VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

NÁVRH TECHNOLOGICKÉHO IT KURZU PRO INTERNÍ VZDĚLÁVÁNÍ

TECHNOLOGICAL IT COURSE FOR INTERNAL EDUCATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **MICHAL ORLÍČEK**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. JAN PLUSKAL**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Orlíček Michal**
Program: Informační technologie
Název: **Návrh technologického IT kurzu pro interní vzdělávání**
Technological IT Course for Internal Education
Kategorie: Softwarové inženýrství

Zadání:

1. Zjistěte, jaké jsou požadavky pro vytvoření technologického kurzu interního vzdělávání. Jaké formy materiálů je vhodné použít a v jakém zastoupení.
2. Dle pokynů vedoucího navrhněte osnovu celotýdenního kurzu (40 vyučovacích hodin) zaměřeného na programování v jazyce C#, stanovte anotaci a požadované prerekvizity.
3. Vytvořte materiály pro Vámi vybraný kurz, respektujíce zjištěné požadavky z bodu 1. Součástí materiálů budou demonstrační aplikace s využitím různých technologií a frameworků.
4. Diskutujte logickou strukturu kurzu, její opodstatnění, výběr technologií a užití vytvořených materiálů účastníky.

Literatura:

- Albahari, J., & Albahari, B. (2017). *C# 7.0 in a Nutshell: The Definitive Reference*. " O'Reilly Media, Inc.".
- Microsoft. (2018, July 24). Course 20483C: Programming in C#. Retrieved from https://www.microsoft.com/en-in/learning/course.aspx?cid=20483
- Gamma, E. (1995). *Design patterns: elements of reusable object-oriented software*. Pearson Education India.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Pluskal Jan, Ing.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 28. října 2018

Abstrakt

Cieľom práce je navrhnúť technologický kurz pre interné vzdelávanie, ktorého témou je programovací jazyk C#. Pred vytvorením samotného kurzu bol prevedený prieskum študijných materiálov k danému jazyku, prieskum prezentačných technológií a prieskum vývojových prostredí. Na základe týchto prieskumov bola zvolená vhodnú dĺžku kurzu, jeho jazyk, materiály, ktoré v ňom boli použité, ich zastúpenie, a technológie, ktoré boli použité na tvorbu týchto materiálov. Ku kurzu sú vytvorené potrebné doplňujúce dokumenty, ako je osnova, časový harmonogram, anotácie, prerekvizity a súhrn získaných znalostí účastníkov. V neposlednom rade sú ku kurzu vytvorené študijné materiály skladajúce sa z trinástich teoretických prezentácií ku prednáškam, a piatich praktických cvičení.

Abstract

The aim of this bachelor's thesis is to create a technological course on C# programming language for internal education. Prior to the creation of the course itself, the research of relevant study materials, presentation technologies, and development environments was done. From that research, it was possible to choose the appropriate length of the course, its language, the structure of suitable materials, and technologies which were used to create these materials. Additional documents, such as the study plan, the timetable, the annotations, the prerequisites, and the list of the acquired knowledge, are created as well. Last but not least, there are course materials themselves, which contains thirteen theoretical presentations suited for lectures and five practical exercises.

Kľúčové slová

technologický IT kurz, GitPitch, Markdown, C#, .NET, objektovo orientované programovanie, študijné materiály

Keywords

technological IT course, GitPitch, Markdown, C#, .NET, object oriented programming, study materials

Citácia

ORLÍČEK, Michal. *Návrh technologického IT kurzu pro interní vzdělávání.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Pluskal

Návrh technologického IT kurzu pro interní vzdělávání

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Jana Pluskala. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Michal Orlíček
15. mája 2019

Poďakovanie

Chcel by som sa poďakovať vedúcemu práce Ing. Janovi Pluskalovi za ochotu a odbornú pomoc. Kamarátom Bc. Miroslavovi Igazovi a Bc. Tomášovi Belobradovi za korektúru textu. A v neposlednom rade mojej priateľke a rodine za psychickú podporu pri písaní tejto práce.

Obsah

Kapitola 1

Úvod

V súčasnosti je k dispozícii veľké množstvo kurzov a aj kníh, ktoré sa zameriavajú na programovací jazyk C#. Existuje mnoho stránok s kurzami, ktoré majú užívateľa naučiť programovať v danom jazyku. Väčšina z nich do jazyka len uvádza a vysvetľuje jeho základné štruktúry. Nevenuje sa tomu, ako v jazyku písať správne a čisto, ako dodržiavať *clean code*, alebo ako správne využívať návrhové vzory a objektovú orientáciu jazyka. Mnohé z nich sú spoplatnené, ich materiály nie je možné voľne šíriť alebo sa nedajú upravovať komunitou a sú zastaralé. Presný prehľad toho, aké možnosti poskytujú jednotlivé kurzy sú spracované v prieskume študijných materiálov tejto práce, viď sekcia 2.1.

Vzhľadom na uvedené problémy absolventi nezískajú dostatočne hlboké znalosti, ale iba akýsi úvod do problematiky. Nenaučia sa správne programátorské praktiky, ako je napríklad *clean code* a refaktorizácia *legacy kódu*, a tiež aj rôzne technológie a *frameworky* ako napríklad *Entity Framework* alebo *Auto Mapper*, ktoré sa pri práci na reálnych projektoch používajú. Znalosti, ktoré účastníci získajú z mnou vytvoreného kurzu viď sekcia 3.5.

Technologický kurz vytvorený spolu s touto prácou je voľne dostupný všetkým pod *MIT* licenciou, ktorá ho umožňuje upravovať a používať na osobné aj komerčné účely. Vďaka tomu, že je voľne dostupný, je jednoduché udržiavať ho aktuálnym. Účastníci kurzu sa naučia vyvíjať software v programovacom jazyku C#, pričom je kladený dôraz na dodržiavanie dobrých programátorských praktík, ktoré sa v jazyku využívajú. Hlavným cieľom je naučiť účastníkov kurzu vytvárať *desktopové aplikácie* napojené na databázu.

Prvá časť práce sa zameriava na požiadavky technologického kurzu. Obsahuje prieskum študijných materiálov (viď sekcia 2.1), ktoré cielia na jazyk C#. Prieskum sa delí na dve časti, v prvej skúma materiály od spoločnosti *Microsoft*, v druhej materiály tretích strán. Druhým prieskumom je prieskum prezentačných technológií (viď sekcia 2.2), ktorý sa snaží nájsť najlepšiu technológiu pre vytváranie a zdieľanie prezentácií. Posledným prieskumom je prieskum vývojových prostredí (viď sekcia 2.3), ktorý hľadá ideálne prostredie na prácu s kódom v jazyku C#. Prieskum vývojových prostredí je rovnako ako prieskum študijných materiálov rozdelený na dve časti podla spoločnosti, ktorá za vývoj prostredia zodpovedá. Záver prvej časti tvorí zmysel technologického kurzu (viď sekcia 2.4). Vysvetľuje, akú úlohu by mal kurz spĺňať a aký by mal byť jeho prínos.

Druhá časť práce sa zaoberá samotnou náplňou kurzu a obsahuje potrebné doplňujúce dokumenty. Deklaruje jeho osnovu (viď sekcia 3.1) a časový harmonogram (viď sekcia 3.2). Venuje sa potrebným prerekvizitám účastníkov (viď sekcia 3.4), ako aj ich získaným znalostiam po absolvovaní kurzu (viď sekcia 3.5). Obsahuje aj anotácie ku každej prednáške (viď sekcia 3.3.1) a cvičeniu (viď sekcia 3.3.2) v kurze.

Tretia časť rozoberá zvolenú formu technologického kurzu. V prvej sekcii sa venuje jeho dĺžke a rozloženiu (viď sekcia 4.1), a v druhej sekcii jazyku (viď sekcia 4.2). Následne analyzuje, aké formy materiálov sa v kurze použili, v akom rozsahu a z akého dôvodu, viď sekcia 4.3. V poslednej sekcii zdôvodňuje zvolenie technológií a služieb vybraných na distribúciu a tvorbu materiálov (viď sekcia 4.4), pričom vychádza z prieskumu prezentačných technológií a vývojových prostredí.

Poslednou časťou sú samotné technologické materiály, ktoré tvorí trinásť teoretických prezentácií k prednáškam a päť adresárov s kódom k praktickým cvičeniam (pre presnú formu materiálov viď sekcie 4.3.2 a 4.3.3). Materiály ku kurzu sa nenachádzajú priamo v práci, ale sú dostupné online[1] pomocou služby *GitHub*. Prezentácie, ako aj cvičenia som vypracoval počas tvorby tejto práce.

[1] https://github.com/orlicekm/CsharpCourse

Kapitola 2

Požiadavky na vytvorenie technologického kurzu

Kapitola sa zaoberá požiadavkami technologického kurzu. Pred začatím tvorby kurzu je vhodné realizovať prieskum už vytvorených kurzov, kníh, návodov a iných študijných materiálov na dané témy a zistiť, ako rôzni tvorcovia k danej problematike pristupovali a aké technológie použili. Rovnako je vhodné zrealizovať aj prieskum technológií, ktoré je možné použiť na vytvorenie, prezentovanie a zdieľanie materiálov účastníkom. Vďaka daným prieskumom je možné určiť, aké technológie budú pre nový technologický kurz vyhovujúce.

V prvej časti sa nachádza prieskum kurzov a kníh z oficiálnych aj neoficiálnych zdrojov (viď sekcia 2.1), druhá časť obsahuje prieskum prezentačných technológií (viď sekcia 2.2). Treťou časťou kapitoly je prieskum vývojových prostredí (viď sekcia 2.3). Dané prieskumy pomáhajú určiť správnu formu a obsah materiálov technologického kurzu v tretej kapitole práce (viď kapitola 4). Posledná časť kapitoly vysvetľuje zmysel technologického kurzu, úlohu, ktorú by mal kurz spĺňať a prínos, ktorý by mal kurz mať (viď sekcia 2.4).

2.1 Prieskum študijných materiálov

Prvým krokom pri vytváraní tohto kurzu bolo uskutočnenie čo najväčšieho prieskumu technologických kurzov za účelom zistiť, aká forma materiálov a v akom rozsahu je vhodná. Pri vyhľadávaní rôznych kurzov som sa zameral hlavne na tie, ktoré sú podobné vytváranému kurzu. Predovšetkým na kurzy programovania, ktoré majú za úlohu užívateľa naučiť prácu v konkrétnom programovacom jazyku a prostredí. Tam, kde to bolo možné, som sa zameral priamo na kurzy súvisiace s programovacím jazykom C#, keďže sa jedná o tému, ktorú spracovávam.

Pred vytvorením kurzu som uskutočnil aj prieskum prezentačných technológií (viď sekcia 2.2), prieskum vývojových prostredí (viď sekcia 2.3), a vyhľadal a naštudoval som si aj literatúru z pedagogického odboru, ktorá sa zameriava na výber vhodných materiálov a ich zastúpenie pri podobných kurzoch.

2.1.1 Microsoft študijné materiály

Najprv som vyhľadal oficiálne materiály k jazyku C#. Vzhľadom na to, že C# patrí pod *.NET* [40], bol vytvorený spoločnosťou *Microsoft* a až v roku 2018 bol jeho prekladač prepísaný a stal sa *open source*. Oficiálne materiály od tejto firmy považujem za jeden z najvhodnejších zdrojov.

Microsoft Virtual Academy

Microsoft Virtual Academy [39] je oficiálna stránka od *Microsoftu*, ktorá poskytuje online kurzy zadarmo. Stránka neposkytuje kurzy len k jazyku C#, ale aj k ostatným produktom od *Microsoftu* a aj k iným zaujímavým informačno–technologickým témam. Na stránke je možná registrácia, ktorá užívateľovi umožní sledovať jeho pokrok v jednotlivých kurzoch. Kurzy obsahujú videá kombinované s prezentáciami, ktoré sú vhodné predovšetkým na samoštúdium. Kurzy sú zamerané najmä na špecifickejšie témy, a preto majú kratší rozsah, zvyčajne do desiatich hodín. Chýba v nich ale priama spätná väzba od prednášajúcich, a preto účastník nemusí vždy všetko správne pochopiť a pravdepodobne bude nútený hľadať doplňujúce informácie na iných stránkach, prípadne na fóre, ktoré je vytvorené ku každému kurzu. Pozitívom je odbornosť prednášajúcich, ktorá je na vysokej úrovni. Ide hlavne o odborníkov ocenených certifikátom *MVP* a vývojárov pracujúcich pre spoločnosť *Microsoft*.

Suverénne najpopulárnejším je kurz *C# fundamentals for absolute beginners* [65] od Boba Tabora. Ide o kurz, ktorý je na viac ako osem hodín a má naučiť úplne základy jazyka C#. Jeho spôsob výučby je zameraný hlavne na praktickú ukážku kódu a je určený aj pre ľudí, ktorí nemajú s programovaním takmer žiadne skúsenosti. Obsahuje dvadsaťštyri epizód a ku každej je k dispozícii kód na stiahnutie, ktorý si užívateľ má možnosť vyskúšať. Jednotlivé epizódy sa skladajú z videí, pričom ku každému z nich je vytvorená transkripcia.

Prešiel som aj kurzy, ktoré cielia na konkrétne časti jazyka C#, ako je napríklad Windows *10: Data Binding* [26] alebo *Introduction to* JSON *with C#* [12]. Tieto kurzy sú krátke a trvajú len okolo jednej hodiny, preto sú ideálne na prebratie jednej konkrétnej témy.

Tematicky najviac podobným kurzom k vytváranému je *Programming in C# Jump Start* [49]. Kurz je možné využiť ako prípravu na skúšku *Exam Programming in C# 70–483* viď odsek 2.1.1. Dĺžka kurzu je stanovená na viac ako sedem hodín a obsahuje osem kapitol od základov, ako je *OOP*, *podmienka*, *cyklus*, až po konkrétnejšie témy, ako napríklad *REST*, *marshalling* alebo *asynchrónne programovanie*. Každá kapitola obsahuje videá, v ktorých sa daná látka preberá a prezentáciu s teóriou, ktorá je použitá vo videách.

Microsoft Technical Certifications

Microsoft Technical Certifications [38], ako vyplýva z názvu, sú *Microsoft* certifikáty, ktorých získanie je podmienené absolvovaním viacerých skúšok. Pri niektorých z nich je nutné, aby užívateľ mal získané predošlé certifikáty, ktoré mu dajú vstupné znalosti na nadväzujúci certifikát. Nevenoval som sa celým certifikátom, ale len skúške, ktorá má spoločnú tému s mojím kurzom.

Na stránke každej skúšky sú detailne vypísané témy, ktoré daná skúška testuje a tie, ktoré by mal kandidát po jej absolvovaní ovládať. Stránka informuje o viacerých možnostiach, ako sa na skúšku pripraviť, napríklad kurzom vedeným inštruktorom, online videom, knihami alebo blogom od komunity. V neposlednom rade zahŕňa aj zoznam znalostí, ktoré by mal užívateľ ovládať predtým, ako začne s prípravou na skúšku.

Exam Programming in C# 70–483 [30] je skúška, určená hlavne pre developerov, ktorí majú minimálne jeden rok skúseností s programovacím jazykom C#. Absolvovanie skúšky je kľúčovým predpokladom na získanie *MCSD* certifikátu. Kandidát by mal po absolvovaní skúšky ovládať *asynchrónne programovanie*, validáciu dát, *LINQ*, prácu s *polom*, *kolekciami*, *premennými*, *operátormi*, *triedami* a *metódami*. Takisto by mal vedieť ako funguje *event*, *správa výnimiek* a mnoho ďalších. Čo sa týka rozsahu skúšky, obsahuje všetko, čo by mal naučiť môj kurz.

Na stránke skúšky sa nachádza viac možností, ako sa na skúšku pripraviť. Jednou z nich je päťdňový kurz *20483B: Programming in C#* (viď odsek 2.1.1) vedený trénerom. Ak klient preferuje samoštúdium, možnosťou prípravy je kniha (viď odsek 2.1.1) určená priamo k tejto skúške. Ďalšou možnosťou je online tréningový kurz z *Microsoft Virtual Academy* s názvom *Programming in C# Jump Start* (viď odsek 2.1.1). Ku skúške je vytvorené video *Microsoft Certification PREP Talk: Exam 483* [58], v ktorom certifikační experti diskutujú o relevantných zručnostiach potrebných na absolvovanie skúšky. V neposlednom rade existuje skúšobný test[1], ktorý dokáže štúdium na skúšku výrazne zjednodušiť.

Microsoft IT training courses

Microsoft IT training courses sú oficiálne tréningové kurzy od *Microsoftu*, primárne určené na získanie znalostí potrebných na absolvovanie skúšok, a tým na získanie samotného certifikátu [33]. Existujú dve možnosti, ako sa kurzu zúčastniť, ale nie všetky kurzy podporujú obidve.

Prvou možnosťou je *on–demand course* [36]. Ide o online kurz, v ktorom klient získa niekoľkomesačný prístup k študijným materiálom a je len na ňom, aký študijný plán si vytvorí. Nepotrebuje cestovať v určitú hodinu na určité miesto. Jediné, čo potrebuje, je mať vlastný počítač a prístup k internetu. Tieto skúšky vytvárajú odborníci ocenení certifikátom *MCT*, v spolupráci s expertmi v danej oblasti. Druhou možnosťou je *classroom course* [35], ide o kurz vedený naživo miestnym trénerom, ku ktorému je možné sa pripojiť aj vzdialene. Kurz trvá pár dní a má presne danú štruktúru. Výhodou je priama interakcia s trénerom, vďaka čomu klienti môžu klásť otázky v reálnom čase. Nevýhodou je, že tréningy nie sú dostupné vo všetkých štátoch, vzhľadom na to, že spoločnosť *Microsoft* nemá všade vzdelaných a vyškolených trénerov na všetky druhy kurzov.

Course Programming in C# 20483C [29] je oficiálny kurz od Microsoftu, ktorého tematika je veľmi podobná tej mojej. Je prístupný v obidvoch variantoch, v *on–demand course* a aj *classroom course*. V prvom prípade ide o trojmesačný prístup a v druhom o päťdňový tréning.

Vývojári sa v kurze naučia programovacie schopnosti, ktoré sú potrebné na vytváranie *Windows* aplikácií pomocou programovacieho jazyka C#. Počas kurzu sa účastníci zoznámia so základnou štruktúrou jazyka, syntaxou, implementačnými detailmi a následne svoje znalosti uplatňujú pri vytváraní aplikácie, ktorá zahŕňa mnoho funkcií, ktoré umožňuje *.NET Framework*.

Cieľovou skupinou kurzu sú vývojári, ktorí už majú skúsenosti s programovacími jazykmi *C*, *C++*, *JavaScript*, *Objective–C*, *Microsoft Visual Basic* alebo *Java* a chápu koncept *OOP*. Kurz nie je dizajnovaný pre študentov, pre ktorých je programovanie nové. Je cielený skúseným vývojárom s minimálne jedným mesiacom skúseností s programovaním v *OOP* prostredí.

Vývojári by mali mať pred začatím kurzu isté prvotné skúsenosti, ktoré získali z iných programovacích jazykov. Medzi tieto skúsenosti patrí práca s *premennými*, *operátormi*, *podmienkami*, *cyklami*. Taktiež je potrebné vedieť sa pripojiť k *SQL* serveru a získavať z neho dáta. Tieto skúsenosti sú potrebné na dokončenie základných programovacích úloh.

[1] https://www.mindhub.com/70-483-Programming-in-C-p/mu-70-483_p.htm

Microsoft Docs

Microsoft Docs [32] je oficiálna technická dokumentácia pre konečných užívateľov, vývojárov a IT špecialistov, ktorí pracujú s produktami spoločnosti *Microsoft*. Dokumentácia obsahuje technické špecifikácie, články, návody, príručky, referencie *API*, ukažky kódov a ďalšie informácie, týkajúce sa softvéru a webových služieb spoločnosti *Microsoft*. Dokumentácia vznikla v roku 2016 ako náhrada za *MSDN* a *TechNet* knižnice, ktoré niektoré z týchto materiálov dovtedy ponúkali.

Obsah dokumentácie je organizovaný do skupín podľa produktu alebo technológie. Skupiny obsahujú aj úkony s technológiou/produktom spojené, ako je hodnotenie, návod ako začať s používaním produktu/technológie, plánovanie, nasadzovanie, správa a odstraňovanie problémov daného produktu alebo technológie. Navigačný panel u produktov a technológií zobrazuje členenie materiálov podla týchto zásad. Každý článok zobrazuje odhadovaný čas na prečítanie a aj možnosť stiahnutia konkrétnej časti ako *PDF* dokument. Webové stránky dokumentácie sú dostupné aj na mobilných zariadeniach.

Väčšina obsahu dokumentácie je *open source* a prijíma *pull request*. Článok dokumentácie je reprezentovaný ako súbor *Markdown* v *GitHub* repozitári [31]. *Microsoft* vydal rozšírenie pre *Visual Studio Code* nazvané *Docs Authoring Pack*[2], ktoré pomáha pri úpravách obsahu dokumentácie. Rozšírenie obsahuje podporu pre rozšírené funkcie jazyka *Markdown*, ktoré sú špecifické pre *Microsoft Docs*.

Programovací jazyk C# sa nachádza v skupine *.NET*. Okrem jazyka C# a softvérového *frameworku .NET* obsahuje skupina aj súvisiace technológie, akými sú napríklad platforma *.NET Core*, *Entity Framework* alebo *Visual Studio IDE*, pre ktoré existujú v skupine separátne zložky s článkami. K jazyku C# dokumentácia obsahuje kategórie článkov k začiatku práce s jazykom, prehliadke funkcií jazyka, novinkám v jednotlivým verziách jazyka, konceptom jazyka C# a iné [28]. Pod skupinou *.NET* je možné nájsť aj prehľad referencií *API*[3].

Microsoft Learn [34] je platforma, ktorá je súčasťou *Microsoft Docs* a slúži na učenie sa nových technológií. Bola spustená v septembri 2018 a obsahuje zatiaľ viac cez 470 tém.

Celý obsah je organizovaný do učebných ciest, ktoré poskytujú celkový prehľad v určitej technologickej oblasti a zabezpečia, že si účastník vytvorí komplexný súbor poznatkov a zručností v danej oblasti. Učebné cesty sa skladajú z návodov, ktoré prechádzajú danú problematiku postupne po krokoch. Popritom poskytujú bezplatný prístup k prostriedkom *Microsoft Azure* na určitú fixnú dobu, kedy si môže užívateľ technológiu, vysvetľovanú v návode vyskúšať. Pri používaní služby má účastník možnosť sledovania svojho pokroku, kontrolovať svoje vedomosti a overovať praktické úlohy pre získavanie bodov, úrovní, úspechov a trofejí.

Platforma k jazyku C# zatiaľ neposkytuje žiadne učebné cesty. K *.NET* poskytuje dve učebné cesty. Prvá učí ako vytvoriť web *API* pomocou *ASP.NET Core*, druhá učí ako pracovať s databázou pomocou *Entity Framework Core*.

Microsoft Press

Oficiálne *Microsoft* knihy sú zdrojom kvalitných informácií k vytvoreniu kurzu. *Microsoft Press* [44] je vydavateľskou spoluprácou medzi spoločnosťami *Microsoft* a *Pearson plc*. K dispozícii sú knihy v tlačenej, elektronickej a audiovizuálnej forme. *Microsoft Press* bol

[2] https://marketplace.visualstudio.com/items?itemName=docsmsft.docs-authoring-pack
[3] https://docs.microsoft.com/en-us/dotnet/api/

založený v roku 1984 a od svojho začiatku vydáva knihy, ktoré pomáhajú vývojárom, ale aj bežným užívateľom, rozširovať svoje znalosti a zručnosti. Ja som sa zameral predovšetkým na tie, ktoré majú podobnú tematiku s vytváraným kurzom.

Begin to Code with C# [45] je kniha s rozsahom päťstodvanástich strán a jej autorom je Rob Miles. Cieľovou skupinou knihy sú užívatelia so žiadnymi alebo minimálnymi skúsenosťami s programovaním. Dostupná je v obidvoch formách, v tlačenej aj elektronickej. Úlohou je naučiť užívateľa programovať v jazyku C# od úplných základov. Kniha sa skladá z troch častí. Prvá časť má užívateľov naučiť základy programovania, druhá sa zameriava na pokročilejšie časti jazyka a tretia sa venuje vývoju hier. Obsahuje stopäťdesiat vzorových programov, ktoré ilustrujú dôležité koncepty jazyka.

Microsoft Visual C# Step by Step [60] je kniha, ktorej cieľom je naučiť užívateľov základy programovania v programovacom jazyku C# v prostredí *Visual Studio*. Má osemstodvanásť strán a je určená pre vývojárov, ktorí sa chcú naučiť programovanie v jazyku C# od základov. Kniha je dostupná v tlačenej aj elektronickej forme.

Kniha sa skladá zo štyroch častí, pričom každá časť je rozdelená na kapitoly. Prvá časť uvádza vývojára do jazyka C# a prostredia *Visual Studio*. Učí prácu s *premennými*, *operátormi*, *podmienkami*, *cyklami*, vytváranie *metód* a *správu výnimiek*. Druhá časť je cielená na objektový model jazyka, vysvetľuje, ako fungujú *hodnoty*, *referencie*, *triedy*, *objekty*, *rozhrania* a *dedičnosť*, ukazuje, ako použiť *pole* a *garbage collector*. V tretej časti sa autor zameriava na pokročilejšie funkcie jazyka, ako *generické programovanie*, *kolekcie*, *preťažovanie operátorov* a *LINQ*. Posledná, štvrtá časť knihy sa zaoberá programovaním *UWP* aplikácií a *REST API*.

Exam Ref 70–483 Programming in C# [46] slúži ako príprava na najnovšiu verziu skúšky *Exam Programming in C# 70–483* (viď odsek 2.1.1). Dostupná je v tlačenej aj elektronickej forme. Je navrhnutá pre vývojárov, ktorí majú už skúsenosti s jazykom C# a môžu vďaka nej profesijne rásť. Ciele skúšky sú v knihe usporiadané tak, aby štúdium bolo čo najefektívnejšie. Vo svojich štyristotridsiatichdvoch stranách zahŕňa:

- Riadenie toku programu
- Prácu s typmi
- Ladenie aplikácií
- Implementáciu zabezpečenia
- Implementáciu prístupu k dátam

2.1.2 Študijné materiály tretích strán

V tejto časti som sa zameral na materiály k jazyku C# od tretích strán, ktoré nevytvorila spoločnosť *Microsoft*. Vzhľadom k veľkému množstvu nájdených materiálov sa bolo možné zamerať a vyselektovať len tie, ktoré sú pre mňa najužitočnejšie.

Kurzy tretích strán

Výber kurzov tretích strán som zúžil na tie, ktoré považujem za relevantné a mne prístupné. Zvolil som tie, ktoré sú dostupné v online verzii, alebo sa nachádzajú na území Slovenskej a Českej republiky.

Lynda.com [21] je platforma, vytvorená spoločnosťou *LinkedIn*, ktorá poskytuje online kurzy a je spoplatnená mesačne. Platforma neobsahuje kurzy zamerané len na informačné technológie, ale aj na rôzne iné technológie, business, tvorivé schopnosti a podobne.

Každý kurz je tvorený jedným alebo viacerými videami, ku ktorým existuje transkripcia a k väčšine z nich aj súbory na stiahnutie. Kurzy je možné vyhľadať podľa rôznych kritérií a ku každému z nich je možnosť písania poznámok a zaznamenávanie si progresu.

Podľa môjho názoru je táto platforma značne podobná s *Microsoft Virtual Academy (MVA)* (viď sekcia 2.1.1). Výhodou *MVA* je, že nie je spoplatnená, ale na druhej strane *Lynda* poskytuje omnoho väčší výber kurzov aj z rôznych iných odborov. Nachádza sa na nej naozaj veľké množstvo kurzov, ktoré sú zamerané na jazyk C#. Z tisíctristo nájdených kurzov som sa zameral hlavne na tie, ktoré sú najnovšie, aktuálne a najviac relevantné.

Prvým kurzom, na ktorý som sa zameral je *Object Oriented Programming* with C# [4] od Antona Delsinka, určený pre pokročilých vývojárov, ktorí sa chcú naučiť *OOP*. Kurz sa skladá z úvodu, piatich kapitol a záverečného slova, ktoré odporúča ďalšie kroky po absolvovaní kurzu. Samotný kurz obsahuje viac ako päťdesiat videí, ktorých dĺžka spolu je viac ako štyri hodiny. Celý kurz nie je k dispozícii zdarma, a preto som sledoval len bezplatnú časť. V nej autor strieda teoretickú časť, na ktorú používa prezentáciu, s praktickou ukážkou kódu v prostredí *Visual Studio*. Ku kurzu sú prístupné súbory s ukážkami kódu, ale ani k nim som nemal prístup.

Ďalším kurzom je *Using Generics in C#* od Robbyho Millsapa [47]. Kurz obsahuje okrem úvodu a záveru ďalšie dve kapitoly a trvá približne hodinu. Je zameraný na *generické programovanie* a od predchádzajúceho kurzu sa líši tým, že je viac prakticky zameraný. Vo voľne dostupnej časti sa nenachádza žiadna prezentácia, ale len programovanie s komentárom.

Udemy [73] je jedna z najväčších svetových služieb poskytujúcich online kurzy. Okrem bežných online kurzov táto služba poskytuje aj tréningy určené pre firmy. Väčšina z nich je spoplatnená a kurz na jazyk C#, ktorý by bol zadarmo, nebolo možné nájsť. Cieľovou skupinou kurzov nie sú len vývojári vzhľadom na to, že platforma obsahuje aj kurzy zamerané na osobný rozvoj, dizajn, marketing a mnoho ďalších. Služba disponuje okrem webových stránok aj mobilnou aplikáciou.

Užívateľ si môže aj bez zakúpenia prehliadnuť osnovu kurzu, požadované prerekvizity, krátky popis a aj výpis znalostí, ktoré absolvovaním získa. Pod kurzom sa nachádzajú krátke informácie o jeho autorovi a hodnotenie kurzu s komentármi, ktoré pomáha pri rozhodovaní o kúpe. Kurz obsahuje aj informácie o tom, čo všetko užívateľ po zakúpení získa. Konkrétne ide o počet článkov, počet stiahnuteľných súborov a koľko hodín videí bude mať k dispozícii. V prípade zakúpenia tohto kurzu účastníci získavajú celoživotný prístup a po jeho dokončení certifikát.

Najpopulárnejším kurzom týkajúcim sa jazyka C# je kurz *Complete C# Unity Developer 2D: Learn to Code Making Games* [63], ktorý sa zaoberá herným vývojom pomocou platformy *Unity*. Je určený pre úplných začiatočníkov v hernom vývoji a v programovaní. Obsahuje okolo šesťdesiat hodín videí, osemnásť článkov a šesťdesiatsedem stiahnuteľných súborov, rozdelených do takmer štyristo prednášok. Keďže sa kurz zameriava primárne na herný vývoj, neobsahuje veľa užitočných informácií k vytváranému kurzu. Taktiež z dôvodu spoplatnenia nebol možný voľný prístup k obsahom videí, ani k ostatným materiálom.

Druhým najpopulárnejším a zároveň podobným k vytváranému kurzu je kurz *C# Basics for Beginners: Learn C# Fundamentals by Coding* [7]. Je to šesťhodinový kurz zameraný na základy jazyka C#. Obsahuje osemdesiatsedem krátkych prednášok, ktoré majú účastníka previesť úplnými základmi jazyka C#. Ide o prvý diel z trojdielnej série kur-

zov a k jeho videám, jedenástim článkom a desiatim stiahnuteľným súborom sa taktiež nebolo možné dostať.

Kurzy-it.sk [67] je slovenský internetový obchod spoločnosti *Inštitút vzdelávania informačných technológií* [66] poskytujúci počítačové kurzy a školenia pre firmy aj širokú verejnosť. Kurzy sú vedené offline, školiteľom, verejné z nich sú realizované v Nitre, Bratislave, Žiline a Košiciach. Za účasť na kurze je nutné zaplatiť, avšak účastníkovi je garantovaná cena a kvalita. Účastníkom každého počítačového školenia bude vystavený oficiálny doklad o absolvovaní. Obchod poskytuje na programovací jazyk C# dve školenia.

Prvé školenie, *Počítačový kurz C# — základy* [69] si kladie za úlohu naučiť účastníkov základy programovacieho jazyka C#. Rozsahovo je kurz vytvorený na osemnásť hodín v priebehu troch dní od 9 do do 15 hodiny. Jeho cieľovou skupinou sú programátori, skladá sa zo štyroch hlavných tém rozdelených na menšie podtémy. V prvej preberá úvod do programovania, v druhej úvod do samotného programovacieho jazyka C#, v tretej typy, *premenné*, *operátory a výrazy*, v štvrtej vytváranie *metód* a *obory platnosti*. Posledná téma rieši možnosti *vetvenia* a *cykly*.

Nadväzujúce školenie, *Počítačový kurz C# — pokročilý* [68] má rovnaký časový harmonogram ako predchádzajúce. Je vytvorené na osemnásť vyučovacích hodín od 9 do 15 hodiny v priebehu troch dní a je cielené primárne na absolventov predchádzajúceho kurzu. Tvorí ho taktiež 5 tém, pričom prvá rieši *výnimky* a ich zachytávanie, druhá podrobnejšie preberá *OOP*, tretia *generické programovanie*, štvrtá dátové prúdy a posledná *multivláknové programovanie*.

ITnetwork.cz [79] je česká IT sociálna sieť so širokou komunitou aktívnych členov. Jej web je postavený na technológii Simplex Social, ktorá poskytuje webu funkcie sociálnej siete ako napríklad nástenky, chat, správy a fotoalbumy. Na webe je možne nájsť rôzne kurzy, v ktorých sa nevytvárajú iba jednoduché aplikácie ale aj reálne použiteľné komerčné projekty. Na webe je taktiež možné vyplňovať rôzne druhy testov, diskutovať na fórach a nájsť návody a manuály na rôzne, často sa vyskytujúce problémy zo sveta informačných technológií. Okrem online štúdia poskytuje sieť aj prezenčné štúdium, v skupinách okolo desať ľudí.

Web je rozdelený na päť hlavných častí, prvou je programovanie, v ktorej sa zameriava hlavne na jazyky C#, *Java*, *PHP*, *HTML*, *CSS*, *JavaScript*, *Python*, Swift, Kotlin, *SQL*, *C*, *C++* a ďalšie. Druhá časť sa venuje designu, konkrétne databáz, softvéru, webu a algoritmov. Tretia časť sa venuje obecne práci a podnikaniu v IT, štvrtá softvéru a hardvéru a v poslednej je možne nájsť záznamy z prednášok a ostatné, do iných časti nezapadajúce témy z IT sveta.

Sekcia k jazyku C# sa rozdeľuje na ďalšie podsekcie, akými sú napríklad základy, práca so súbormi, práca s databázami, tvorba hier alebo testovanie. Podsekcie obsahujú lekcie a cvičenia, ktoré rozoberajú jednu konkrétnu tému z danej problematiky. Pri každej lekcii a cvičení je uvedený autor s krátkym popisom jeho skúseností a kontaktnými údajmi, čo dodáva stránke na dôveryhodnosti. Cvičenia a články je možné aj komentovať, hodnotiť ale aj zdieľať na sociálnych sieťach.

Sekcia obsahuje okrem podsekcií na témy jazyka aj podsekciu diskusného fóra a podsekciu prezenčných školení. Konkrétne k jazyku C# poskytuje stránka päť typov školení, z čoho ku štyrom má aj vypísané najbližšie termíny a dá sa na ne zaregistrovať. Prvý kurz má učiť základy jazyka, druhý sa zameriava na *OOP*, tretí na *UML* a návrhové vzory, štvrtý na *Git* a testovanie a posledný piaty na *MVC* a *ASP.NET*. Školenia sú spoplatnené

a každé je situované na jeden deň na osem až deväť hodín. Všetky školenia majú ako lokalitu uvedené mesto Praha a ku každému školeniu je napísaná jeho osnova. Po absolvovaní kurzu získa účastník certifikát a prístup k ďalším, rozširujúcim materiálom.

GOPAS [10] je slovenské školiace stredisko poskytujúce kurzy v oblasti informačných technológií. Na ich stránkach sa dá nájsť veľké množstvo zaujímavých tém ako je napríklad *IoT*, IT bezpečnosť, rôzne technológie a programovacie jazyky, medzi ktorými je aj jazyk C#. Okrem poskytovania klasických kurzov, ktorých je cez 1400 je stredisko aj dodávateľom e-learningových technológií. Ponúka systém vzdelávania formou samo-študijných kurzov. Všetky kurzy sú spoplatnené.

K jazyku C# sa v stredisku nachádza 25 kurzov, ktoré prechádzajú jazyk C# od základov programovania až po pokročilé technológie a *frameworky*. Ku kurzom existuje aj graf ich nadväznosti. Po otvorení každého kurzu je možné nájsť jeho krátky popis a osnovu, ako aj najbližšie termíny s ich lokalitou a cenou.

Learn2Code [20] je webová stránka poskytujúca vzdelávanie digitálnych zručností. Web poskytuje kurzy online, prezenčné kurzy a aj REPAS kurzy pre nezamestnaných. Prezenčné kurzy sú všetky platené, no medzi online kurzami sa nachádza pár, ktoré sú zdarma.

K jazyku C# sa na stránkach nachádza jeden kurz, Úvod do programovania v jazyku C# [75], ktorý je bohužiaľ platený. Kurz obsahuje viac ako 5 hodín videotutoriálov rozdelených do 40 kapitol, viacero zadaní na precvičenie a účastník získa po jeho absolvovaní certifikát. Kurz je určený pre začiatočníkov.

Na stránkach kurzu je možnosť nahliadnuť na jeho kapitoly ešte pred jeho zakúpením. Tak isto je možné si prečítať krátky popis toho, čo kurz obsahuje a čím budúceho účastníka obohatí, pozrieť sa na jeho verejné hodnotenie alebo krátke úvodné video.

Windows User Group [76] skrátene WUG je české občianske združenie, ktorého cieľom je fyzicky združovať vývojárov, IT profesionálov a počítačových nadšencov so záujmom o platformu a produkty služby *Microsoft*. Združenie pravidelne usporadúva prednášky, odborné semináre, konferencie a ďalšie akcie pre svojich členov aj širokú verejnosť v dvanástich mestách v Českej republike. Podmienkou pre účasť teda nie je členstvo v združení.

Združenie usporadúva prevažne dva typy akcií. Prvou skupinou sú klasické prednášky zamerané na konkrétnu tému v podaní prednášajúceho, ktorý je odborníkom v danej problematike. Prednášky trvajú zvyčajne okolo troch hodín a je na nich dostupné aj občerstvenie. Druhým typom akcií sú laboratórne cvičenia, ktoré sa konajú v počítačových učebniach, kde si účastníci môžu konkrétnu technológiu, či problematiku vyskúšať a osvojiť.

Organizátori *Windows User Group* sú dobrovoľníci, ktorí usporiadavajú akcie vo svojom voľnom čase. Celý koncept stojí na ochote poskytovateľov priestorov prepožičať ich k tomuto nápadu a na ochote organizátorov nájsť si čas a usporiadať meeting. Rovnaký model dobrovoľnosti platí pre prednášajúc ich, ktorí na *WUG* prednášajú zdarma. Na *WUG* akciách sa vďaka tomu neplatí žiadne vstupné, občestvenie hradí firma *Microsoft*.

Na webových stránkach *Windows User Group* je možná registrácia, vďaka ktorej sa dajú nastaviť pravidelné e-mailové upozornenia o nasledujúcich akciách, na ktoré sa dá pomocou stránky následne prihlásiť. Taktiež sa na stránkach nachádzajú video záznamy a fotky z predchádzajúcich akcií.

Pokiaľ ide o jazyk C#, medzi záznamami sa veľké množstvo prednášok venuje tejto téme. Zväčša ide o špecifické *frameworky*, technológie alebo konkrétne časti a funkcie jazyka, keďže sa prirodzene očakáva, že základnú problematiku jazyka majú už účastníci zvládnutú.

Pluralsight [54] je medzinárodná platforma pre technologické vzdelávanie, ktorá ponúka rôzne vzdelávacie kurzy pre vývojárov softvéru, IT administrátorov a kreatívcov prostredníctvom svojich webových stránok. Platforma spolupracuje s viac ako tisícštyristo expertmi z rôznych oblastí, ktorí sú autormi viac ako šesťtisícpäťsto kurzov, ktoré tvoria katalóg platformy.

Platforma poskytuje online profesionálne technické školenia ako pre jednotlivcov, tak pre firmy a je založená na predplatnom obchodnom modeli. Platforma poskytuje bezplatný účet, ktorý neposkytuje veľké množstvo funkcií, dajú sa pomocou neho iba vyhľadávať kurzy a pozerať ich prehľad. Bezplatný účet teda nemá prístup k prechádzaniu všetkých kurzov, ale iba k pár bezplatným, ktoré sa každý týždeň menia. Pre prístup ku všetkým kurzom je nutné zaplatiť predplatné, ktoré sa dá kúpiť na mesiac alebo rok. Platforma taktiež poskytuje prémiové účty a účty pre firmy s prídavnými funkciami.

Pokiaľ ide o programovací jazyk C#, platforma obsahuje viacero ciest, čo sú skupiny kurzov, ktoré do jazyka C# zasahujú. Medzi cestami sa nachádza jazyk C#, ako písať *unit testy* v jazyku C# pomocou *NUnit*, *DDD*, kurzy vývoja hier v *Unity* a ďalšie. Nie každý kurz na stránke musí byť členom cesty, na stránke sa dajú vyhľadávať kurzy aj individuálne, bez ich cesty. K jazyku C# som na platforme našiel tristotridsaťštyri kurzov.

Cesta nazvaná C# [55] obsahuje 15 kurzov rozdelených do kategórií začiatočník, pokročilý a expert. Trvá 49 hodín a na jej stránke sa dá spraviť rýchly test, ktorý odporučí, ktorou úrovňou je vhodné začať. Stránka obsahuje aj krátky popis jazyka, potrebné prerekvizity, znalosti, ktoré účastník získa a zoznam autorov jednotlivých kurzov. Samotný kurz obsahuje hodnotenie, dĺžku, level zložitosti, informácie o autorovi, krátky popis, obsah a súbory na stiahnutie, ktoré sa v kurze využívajú. Keďže som mal k dispozícii iba bezplatný účet, žiadny z kurzov som si spustiť nemohol, kvôli čomu nemôžem hodnotiť ich kvalitu.

CBT Nuggets [3] je webová stránka poskytujúca *on–demand kurzy* pre IT špecialistov z celého sveta. Stránka ma dlhú históriu, bola založená v roku 1999 ako odpoveď na potrebu inovácie kurzov v IT priemysle. Každý, kto dovtedy chcel tréning, musel zaplatiť veľkú sumu peňazí každý týždeň a sedieť fyzicky na školeniach. Video tréning bol síce dostupný, ale bolo to v podstate iba rovnako finančne nákladné nahrávanie fyzických školení. Z počiatku stránka predávala školenia za jeden dolár v aukciách na stránkach *eBay*.

Stránka sa časom vzdala predaja školení v aukciách a dnes funguje na princípe predplatného, neposkytuje bezplatný účet, preto sa mi k jednotlivým kurzom nepodarilo dostať. Okrem plateného účtu stránka poskytuje aj firemné účty.

CBT Nuggets je dostupná aj ako mobilná aplikácia pre operačné systémy *Android* a *iOS* a ako *desktopová aplikácia* pre *Windows*. Kurzy okrem videí obsahujú aj kvízy a praktické cvičenia vo virtuálnom prostredí. Stránka obsahuje aj kurzy na prípravu na konkrétne certifikáty.

dotNETportal.cz [15] je český webový magazín zameraný na vývoj aplikacií, *.NET Framework* a *Microsoft* technológie. Na stránkach magazínu sa nachádzajú fóra a články rozdelené do blogov a seriálov. Články nemôže písať každý užívateľ, ale iba určitá skupina blogerov. Medzi blogermi sa nachádzajú IT špecialisti, ktorí sú odborníci v oblasti o ktorej píšu.

Medzi kategóriami sa nechachádza iba samotný jazyk C#, ku ktorému je na webovej stránke mnoho článkov, ale aj *.NET*, *Entity Framework*, *LINQ* a ďalšie s jazykom súvisiace témy. Pri každom článku je hodnotenie a komentárová sekcia.

dotNETcollege [14] je český portál ponúkajúci vzdelávanie v oblasti vývoja aplikácii a služieb na platforme *.NET*, webe a databázach. Kladie dôraz na praktické skúsenosti lektorov, ktorí trávia väčšinu času vývojom aplikácií a používaním technológií v danej problematike, a individuálny prístup. Lektori sú uznávaní odborníci, ktorí držia ocenenia *MVP*. Portál sa primárne zameriava na kurzy a konferencie.

Portál poskytuje kurzy ako pre firmy, tak pre jednotlivcov, všetky prebiehajú v učebniach, portál neposkytuje kurzy online. Každý mesiac portál vypisuje niekoľko verejných kurzov v Prahe a Brne. Dĺžka týchto kurzov je zväčša dva až tri dni a obsahujú aj praktickú časť. Druhou alternatívou sú večerné kurzy, prebiehajúce vo večerných hodinách a typicky trvajú štyri až päť týždňov. Portál poskytuje firmám taktiež možnosť vytvorenia kurzu na mieru.

Knihy tretích strán zamerané na jazyk C#

Z kníh zameriavajúcich sa na jazyk C# som vybral tie, z ktorých som čerpal materiály pri vytváraní technologického kurzu. Detailnejší prehľad, v akej časti a z ktorej knihy som čerpal informácie, sa nachádza v materiáloch ku kurzu.

C# 7.0 in a nutshell [1] je bestseller, ktorého rozsah sa neustále zvyšuje. C# sa od svojho vydania v roku 2000 stal jazykom neobvyklej šírky a flexibility, ktorý sa stále rozširuje. Z tohto dôvodu sa aj kniha rozširuje spolu s ním a je možné sa z nej stále niečo nové naučiť.

Kniha má viac ako tisíc strán a obsahuje dvadsaťsedem kapitol. Tie prechádzajú všetky zákutia jazyka od úplných základov až po pokročilejšie témy, akými sú napríklad *ukazovatele*, *preťažovanie operátorov* alebo *data binding*. Je všeobecnejšia ako *C# in depth* (viď odsek 2.1.2), pretože sa nevenuje iba samotnému jazyku, ale celému C# ekosystému a aj platforme *.NET*. Vďaka jej veľkému rozsahu, je ideálnym podkladom pre vytvorenie technologického kurzu.

C# in depth [62] sa venuje jazyku C# do väčšej hĺbky ako *C# in a nutshell* (viď odsek 2.1.2). Je určená pre vývojárov, ktorí už majú isté skúsenosti s jazykom a chcú sa v ňom ešte viac zdokonaliť. Vďaka jej hĺbke je ideálna na rozšírenie informácií o jazyku, ak informácie v knihe *C# in a nutshell* nie sú postačujúce.

Tretia edícia má šestošestnásť strán, ktoré sú rozdelené do piatich častí a šestnástich kapitol. Prvá časť obsahuje úvod do knihy, a preto ju skúsení vývojári môžu úplne vynechať, ale zároveň obsahuje aj užitočné rady, ako z knihy vyťažiť maximum. Druhá časť pokrýva celý programovací jazyk C# do verzie 2. Tretia časť pokrýva C# do verzie 3, štvrtá do verzie 4 a piata do verzie 5. Ďalšie novinky je možné nájsť až v novšom vydaní knihy.

Knihy tretích strán zamerané na špecifickejšie oblasti

Knihy *C# in depth* (viď odsek 2.1.2) a *C# 7.0 in a nutshell* (viď odsek 2.1.2) sa zameriavajú na jazyk C#, prípadne na platformu *.NET*. Vo vytváranom kurze sa zaoberám aj dobrými programátorskými praktikami, a z toho dôvodu som zvolil aj literatúru, ktorá je v tomto smere nápomocná.

Agile principles, patterns, and practices in C# [25] preberá základy agilného vývoja a agilného dizajnu. Nevenuje sa príliš dlho *UML* modelom a rýchlo prechádza na kód

napísaný v jazyku C#. Úvodné kapitoly prechádzajú základy agilného vývoja, zatiaľ čo neskoršie predstavujú osvedčené techniky priamo v akcii. Kniha obsahuje veľa zdrojových kódov, ktoré je možné aj stiahnuť priamo v *GitHub* repozitári autora[4].

Clean code [24] sa snaží naučiť čitateľa, ako písať čistý a udržiavateľný kód. Je rozdelená do troch častí, pričom prvá opisuje princípy a praktiky písania čistého kódu. Druhá časť sa skladá z viacerých príkladov so zvyšujúcou sa zložitosťou, v ktorých má užívateľ upraviť kód a spraviť ho čistým a efektívnym. Tretia časť obsahuje databázu správnych a nesprávnych spôsobov, ktoré boli použité pri vytváraní príkladov.

Design patterns [8] je kniha zahŕňajúca katalóg, ktorý obsahuje dvadsaťtri jednoduchých a všeobecných riešení pre bežne vyskytujúce sa problémy v návrhu. Tieto riešenia, návrhové vzory, umožňujú programátorom vytvárať pružnejšie, elegantnejšie a znovu použiteľné návrhy bez toho, aby bolo nutné ich znovu objavovať vo vlastnej réžii. Kniha opisuje, čo návrhové vzory sú a ako pomáhajú pri dizajne v *OOP*, ale aj to, ako tieto vzory využiť na riešenie vlastných návrhových problémov.

Ku každému vzoru sú v knihe vypísané okolnosti, pri ktorých je použiteľný, jeho obmedzenia a aj rôzne variácie. Všetky vzory pochádzajú z reálne používaných systémov alebo aplikácií, a sú založené na príkladoch z reality. Každý vzor obsahuje aj zdrojový kód, ktorý demonštruje, ako má byť návrhový vzor použitý. Zdrojové kódy však nie sú napísané v jazyku C#, ale v programovacích jazykoch *C++* a *Smalltalk*.

The art of unit testing with examples in C# [52] je kniha, ktorá učí čitateľa testovať aplikácie. Začína sa tým, ako písať prvé jednoduché testy a dostáva sa až k vývoju robustných testovacích setov, ktoré sú udržiavateľné, čitateľné a dôveryhodné. Po vysvetlení základov sa presúva na vysoko–cenené objekty ako *mock* alebo *stub*. Ku každému z nich tiež odporúča *framework*, ktorého použitie zároveň vysvetľuje. Popritom kniha učí, ako testovať *legacy kód*, „netestovateľný“ kód, databázy a iné technológie.

Refactoring [6] je kniha zameraná na refaktorizáciu kódu, prepísanie kódu a na zlepšenie jeho čitateľnosti alebo štruktúry, bez ovplyvnenia jeho významu a správania. Kniha vysvetľuje, čo to refaktorizácia je, prečo by sa kód mal refaktorizovať, a ako to úspešne robiť nezávisle na jazyku, v ktorom sa programuje. Kniha obsahuje aj ukážky kódu v programovacom jazyku *JavaScript*.

2.2 Prieskum prezentačných technológií

Ďalším krokom po prieskume študijných materiálov (viď sekcia 2.1) bolo uskutočnenie prieskumu prezentačných technológií. Prieskum má za úlohu zistiť rozdiely medzi technológiami slúžiacimi na prezentovanie, zdieľanie a vytváranie prezentácií a tým zistiť, aký spôsob prezentácie je pre kurz najvhodnejší.

Pred vytvorením kurzu som uskutočnil aj prieskum vývojových prostredí (viď sekcia 2.3). Vyhľadal a naštudoval som si aj vhodné materiály z pedagogického odboru, ktoré súvisia s danou problematikou.

[4] https://github.com/unclebob/PPP

2.2.1 Prezentačné technológie určené pre širokú verejnosť

Medzi *prezentačné technológie určené pre širokú verejnosť* som zaradil technológie, ktoré sú určené pre bežného užívateľa a sú využívane masovo. Pre ich používanie nie je potrebné ovládať žiaden programovací jazyk alebo pokročilú prácu s počítačom. Sú intuitívne a majú prívetivé užívateľské rozhranie. Väčšina z nich je súčasťou rôznych kancelárskych balíčkov.

Google Slides

Google Slides [9] je prezentačný program od firmy *Google*, ktorý je súčasťou bezplatného balíčku kancelárskeho softvéru fungujúceho nad službou *Google Drive*. Program je k dispozícii ako webová aplikácia, mobilná aplikácia pre *Android*, *iOS*, *Windows*, *BlackBerry* a ako *desktopová aplikácia* v operačnom systéme *ChromeOS*. Aplikácia je kompatibilná s formátmi súborov *PowerPoint*.

Program umožňuje užívateľom vytvárať a upravovať prezentácie online a spolupracovať na nich s ostatnými užívateľmi v reálnom čase. Úpravy sú sledované históriou verzií, ktorá sleduje zmeny v prezentácii. Pozícia každého editora je zvýraznená farebným kurzorom špecifickým pre konkrétneho užívateľa. Systém obsahuje rôzne stupne oprávnení slúžiace na reguláciu dostupných možností užívateľov. Zmeny sú automaticky ukladané na serveri spoločnosti *Google*.

Aplikácia obsahuje bežné funkcie na úpravu prezentácií, ktoré je možné poznať z aplikácie *PowerPoint*. Medzi ďalšie funkcie patrí strojové učenie, ktoré vytvára automatické návrhy na základe obsahu každého slajdu a akčné položky, ktorými je možné priraďovať editorom úlohy.

PowerPoint

PowerPoint [37] je aplikácia na vytváranie a prezentovanie prezentácií vlastnená a vyvíjaná firmou *Microsoft*. Aplikácia je súčasťou balíku *Microsoft Office* a je dostupná pre operačné systémy *Windows*, *Mac OS* a mobilné zariadenia.

Aplikácia je spoplatnená a určená pre širokú verejnosť na jednoduché a intuitívne vytváranie prezentácii. *PowerPoint* obsahuje veľké množstvo možností, medzi základné patria témy, formátovania, zdieľanie multimediálneho obsahu, animácie, prechody a ďalšie. Aplikáciu je možné prepojiť so službou *One Drive*, čo umožňuje spolupracovať na zdieľanej prezentácii.

Keynote

Keynote [2] je bezplatná aplikácia na vytváranie prezentácií a ich prezentovanie od firmy *Apple*. Je určená primárne pre užívateľov produktov firmy *Apple* na operačných systémoch *Mac OS X* a *iOS*.

Možnosti, ktoré poskytuje *Keynote* sú porovnateľné s možnosťami technológie *PowerPoint*. Aj táto technológia je určená pre širokú verejnosť. Základnými vlastnosťami sú témy, podpora dvoch monitorov, export do veľkého množstva formátov, medzi ktoré patria aj formáty súborov *PowerPoint* a *PDF* alebo vzdialené ovládanie z *Apple* mobilných zariadení pomocou technológie *Keynote Remote*.

OpenDocument Format

OpenDocument Format skrátene ODF [51] je kompresovaný formát súborov založený na *XML*. Cieľom jeho vývoja bolo poskytnúť *open source* formát pre kancelárske aplikácie a ponúknuť tak alternatívu k uzavretým formátom, ako napríklad používa kancelársky balíček *Microsoft Office*. Formát zahrňuje textové dokumenty ako poznámky, knihy, tabuľky, grafy, databázy a pre tento prieskum najpodstatnejšie prezentácie.

ODF je široko podporovaný v celom odvetví významnými komerčnými dodávateľmi, ako sú *Microsoft*, *IBM*, *Google*, ako aj open source projektami ako *OpenOffice.org*, *LibreOffice*, *Calligra* a *Gnome Office*.

LibreOffice [71] je balík kancelárskych programov šírený ako *open source* pod *LGPL* licenciou. Tvorcom je firma *The Document Foundation*, ktorej vývojári v roku 2010 opustili projekt *OpenOffice.org* z obáv, že jeho vtedajší vlastník *Oracle Corporation* projekt zruší rovnako, ako sa to stalo s projektom *OpenSolaris*.

Balík sa skladá z textového a tabuľkového procesoru, vektorového grafického editoru, prezentačného nástroja, programu na spravovanie databáz a nástroja pre vytváranie matematických vzorcov. Dostupný je vo viac ako stovke jazykov, na väčšine v dnešnej dobe používaných operačných systémov, ako aj online. Hlavné verzie balíka sa vydávajú približne raz za pol roka.

Apache OpenOffice [70] predtým nazývaný *OpenOffice.org*. je podobne ako *LibreOffice*, balík kancelárskych programov šírený ako *open source* pod licenciou *Apache Licence*. V roku 2011 firma *Oracle* ukončila vývoj *OpenOffice.org* a odovzdala ho Apache Software Foundation, kde je naďalej vyvíjaný ako *Apache OpenOffice*.

Balík obsahuje rovnakú sadu programov ako *LibreOffice*, je dostupný na väčšine operačných systémov a jeho hlavným cieľom je konkurovať komerčným dodávateľom kancelárskych balíkov ako je *Microsoft Office*.

2.2.2 Markdown prezentačné technológie

Pod *Markdown prezentačné technológie* som zaradil aplikácie a služby, ktoré vytvarujú prezentácie pomocou značkovacieho jazyka *Markdown*. *Markdown* je veľmi jednoduchý značkovací jazyk zvyčajne využívaný na vytváranie *README* súborov. Tieto súbory sa bežne využívajú pri vývoji softvéru, a vďaka tomu, že kurz cieli na vývojárov, tak s nimi väčšina účastníkov kurzu vie pracovať.

Výhodou technológií založených na jazyku *Markdown* je aj to, že sú viac späté s vývojármi softvéru. Vývojári jazyk *Markdown* používajú častejšie, a vďaka tomu tieto technológie obsahujú výhody, ktoré sa v technológiách určených pre bežných užívateľov nenachádzajú. Príkladom jednej z častých výhod je zobrazovanie zdrojového kódu priamo v prezentácii.

Marp

Marp [13] je skrátený názov pre aplikáciu *Markdown Presentation Writer*. Ide o jednoduchú *desktopovú aplikáciu*, ktorá vytvára prezentácie z kódu napísaného v jazyku *Markdown*. Aplikácia je dostupná na operačných systémoch *Windows*, *Mac OS X* aj *Linux*.

V aplikácii sa dá nájsť niekoľko funkcií, ako je živý náhľad na výslednú prezentáciu s tromi druhmi módov, podpora tém a prezentácia podporuje okrem základného *Mar-*

kdown formátovania aj renderovanie emotikonov, matematické výrazy, pozadia či veľkosti a číslovania strán. Prezentácie sa z aplikácie dajú exportovať vo formáte *PDF*.

Podľa informácií na *GitHub* stránke aplikácie[5] bol *Marp* iba vo verzii pred vydaním a už bol jeho vývoj zastavený a presunutý na nový *GitHub* repozitár[6], ktorý je avšak ešte stále v rannej fáze vývoja. V novom depozitári sa nevyvíja iba *desktopová aplikácia*, ale celá rodina *Marp* produktov ako sú *frameworky*, aplikácie a integrácie do iných softvérov.

Pandoc

Pandoc [23] je univerzálny konvertor dokumentov distribuovaný ako konzolová aplikácia dostupná pre väčšinu moderných operačných systémov. Je k dispozícii aj ako *Haskell* knižnica na napísanie vlastných konvertorov. Zdrojové súbory sú dostupné v *GitHub* repozitári[7] pod *GPL* licenciou.

Pandoc bol vytvorený na konvertovanie jedného *Markup* jazyka na iný. Časom sa rozrástol a v dnešnej dobe vie konvertovať veľké množstvo formátov medzi sebou. Vďaka rozsiahlej prehľadne spracovanej dokumentácii sa dá veľmi jednoducho dohľadať, aké formáty sa dajú konvertovať na aké, a s akými možnosťami.

Prezentácie môžu byť v aplikácii vytvorené z *Markdown* textových súborov. A to konvertovaním do prezentácií uložených ako *HTML* a *JavaScript* súbory, ktoré môžu byť zobrazené pomocou webového prehliadača. Aplikácia obsahuje 5 rôznych možností, ako konverziu previesť. Druhou možnosťou konvertovania je formát *PDF*, pri ktorom existuje taktiež viacero možností konverzie, avšak pri tejto konverzii je nutné mať nainštalovaný *pdflatex*. Treťou a poslednou možnosťou je konvertovanie do formátov používaných v *PowerPoint* prezentáciách.

revealJS

revealJS [11] je *HTML* prezentačný *framework* umožňujúci vytváranie interaktívnych prezentácií. *Framework* obsahuje rozsiahle spektrum funkcií. Medzi hlavné patria vnorené slajdy, obsah v jazyku *Markdown*, export do formátu *PDF*, poznámky pre prednášajúceho a *JavaScript API*.

Framework je určený k voľnej distribúcii pod *MIT* licenciou a na svojich stránkach uvádza online editor *Slides* ako svoj oficiálny online editor. *Framework* je využívaný veľkým množstvom iných služieb, medzi ktoré patria aj *Gitpitch* alebo *Hacker slides*.

Slides [64] je stránka na prezentovanie, vytváranie a zdieľanie prezentácií založená na technológii *revealJS*. Obsahuje rozsiahlu škálu rôznych funkcií ako je platený účet, ktorý pridáva ďalšiu funkcionalitu. Medzi hlavné z nich patrí možnosť vytvorenia súkromnej prezentácie, zobrazenie analytiky prezentácie a všetky možnosti, ktoré ponúka *framework*.

GitPitch

GitPitch [50] je *Markdown* prezentačná služba pre užívateľov *Gitu*, ktorá automaticky pretvára obsah *Markdown* súborov na interaktívnu prezentáciu. Služba poskytuje veľké množstvo funkcionalít, ako spoplatnených tak aj bezplatných, ku ktorým je vytvorená online dokumentácia[8].

[5] https://github.com/yhatt/marp
[6] https://github.com/marp-team/marp
[7] https://github.com/jgm/pandoc

Služba *GitPitch* prechádza repozitár verzného systému *Git* a zo súborov *Markdown* spraví prezentácie, ktoré sú dostupné online na *GitPitch* webových stránkach. Technológia podporuje repozitáre uložené lokálne alebo na hostingoch *GitHub*, *GitLab*[9], *Bitbucket*[10].

K používaniu služby existuje 5 druhov účtov v závislosti na výške platby. Základný účet je zdarma a poskytuje základné služby verejným online repozitárom. Najlacnejší platený účet poskytuje možnosť vytvárania prezentácií lokálne, k čomu využíva technológiu *Docker*. Ďalším druhom plateného účtu je osobný účet, ktorý obsahuje všetky možnosti najlacnejšieho plateného účtu, ale aj možnosť vytvárania prezentácií v súkromných repozitároch. Posledné dva druhy účtov sú určené pre tím a organizáciu, obsahujú výhody všetkých ostatných druhov účtov s výhodnejšími platovými podmienkami.

Platené účty poskytujú aj ďalšie rozšírené možnosti, medzi ktoré patrí vloženie prieskumu priamo do prezentácie, ten môže byť využitý napríklad k jej ohodnoteniu. Platené účty umožňujú pridávanie *UML* diagramov, matematických funkcií, smajlíkov, príspevkov zo sociálnej siete *Twitter*, záznamov z terminálu, offline videí a pridávajú aj rozšírené možnosti pre zobrazovanie kódov a obrázkov.

Pokiaľ ide o bezplatné možnosti, *GitPitch* ich obsahuje veľké množstvo. Medzi základné patrí určovanie pozície obsahu ako pre celú prezentáciu, tak pre jednotlivé slajdy, určovanie veľkosti a formu písma, prezentovanie kódu a *Git commit* rozdielov priamo zo zdrojových súborov v repozitári s možnosťou zvýrazňovania jeho častí. Pokiaľ ide o obrázky, služba poskytuje možnosti vkladania obrázkov do prezentácií, pozadia, menenie ich veľkostí, zmenu farby pozadia, priehľadnosti a pozície. Pozadie je možné vytvoriť aj pomocou farby alebo prechodov farieb. Do prezentácie je možne vkladať aj grafy, videá, poznámky prednášajúceho a k prezentácii sa automaticky generuje aj obsah.

Nastavenia prezentácie sú umiestnené v samostatnom súbore. Služba umožňuje veľké množstvo rôznych nastavení. Medzi základné patrí téma, štýl zvýrazňovania kódu, logo, pozadie, titulok, päta prezentácie, cesta k *CSS* súboru, ktorý predefinuje design a ďalšie nastavenia, ktoré menia jednotlivé možnosti služby.

Pre vytváranie prezentácií *GitPitch* používa prezentačný *framework RevealJS* a poskytuje pre platené účty možnosť stiahnutia vo formáte *PDF* a vo formátoch služieb *PowerPoint*, *Google Slides* alebo *KeyNote*. Prípadne umožňuje stiahnutie vo verzii vhodnej pre tlač a aj zdieľanie pomocou sociálnych sietí.

Hacker slides

Hacker slides [61] je jednoduchá aplikácia umožňujúca písanie kódu v jazyku *Markdown*, ktorý je v reálnom čase konvertovaný do prezentácie. Hlavnými komponentami aplikácie sú *Ace Editor* a *RevealJS*. Aplikácia je vyvíjaná pod *MIT* licenciou a beží na platforme *Sandstorm* [56]. *Sandstorm* je *open source* platforma pre webové aplikácie hostované na privátnych serveroch.

Hacker Slides je veľmi jednoduchá, obsahuje prakticky iba dve okná, pričom prvé slúži na písanie formátovaného textu v jazyku *Markdown* a druhé slúži ako náhľad vytvorenej prezentácie. Prezentácia sa dá otvoriť v novom samostatnom okne vhodnom na prezentovanie. Žiadne možnosti importu alebo exportu prezentácie aplikácia neobsahuje. Ukladanie práce zaisťuje platforma *Sandstorm*.

[9] https://gitlab.com/
[10] https://bitbucket.org/

Landslide

Landslide [78] je *Python* balíček, ktorý generuje prezentáciu z jazykov *Markdown*, *ReST* alebo *Textile*. Prezentácia vychádza z Google *html5slides*[11] šablóny. Balíček vyžaduje okrem nainštalovaného jazyka *Python* aj ďalšie podporné balíčky, a pre export do formátu *PDF* aj aplikáciu *Prince*[12], ktorá slúži na konvertovanie *HTML* dokumentov do formátu *PDF*.

Pre vytváranie prezentácií existuje v balíčku rozsiahle množstvo možností, ako sú rôzne štýly formátovania, konfigurácie pomocou konfiguračných súborov, makrá, poznámky prednášajúceho, témy ale aj používanie *CSS* a *JavaScript* súborov, či publikovanie prezentácie online na vlastný *HTTP* webserver.

Deckset

Deckset [74] je nástroj na vytváranie prezentácií exkluzívne pre operačný systém *Mac OS X* pomocou *Markdown* súborov. Nástroj je spoplatnený s týždenným skúšobným režimom a možnosťou zľavy pre študentov a organizácie. Nástroj okrem štandardných možností podporovaných priamo v jazyku *Markdown* obsahuje aj rozšírené možnosti.

Medzi ne patria vytváranie a zdieľanie tém, automatické zvýrazňovanie kódu vo viac ako 100 programovacích jazykoch, konfigurácia príkazov, číslá stránok, hlavičky, päty a podpora smajlíkov. Ďalšou možnosťou je pridávanie fotiek, videí, matematických vzorcov, obrázkových filtrov a poznámok prednášajúceho. Prezentácie sa dajú exportovať do formátu *PDF* a do kolekcie obrázkov, pričom v oboch prípadoch je možné zvoliť rôzne rozlíšenia.

2.2.3 Služby sprostredkovávajúce zdieľanie prezentácií

Pri *službách sprostredkovávajúcich zdieľanie prezentácií* som sa zameral na služby, ktorých primárnym zameraním je zdieľanie prezentácií. Urobil som tak preto, aby som sa zacielil na nadštandardné funkcie, ktoré tieto služby poskytujú oproti bežným dátovým úložiskám. Medzi nadštandardné funkcie patrí napríklad vyhľadávanie, kategorizovanie a zobrazovanie prezentácií priamo na stránkach služieb.

Speaker Deck

Speaker Deck [5] je webová služba sprostredkovávajúca zdieľanie prezentácií online. Umožňuje jednoduché nahratie prezentácie vo formáte *PDF* a jej následne zdieľanie prostredníctvom internetového odkazu alebo priamo na sociálnych sieťach *Facebook* a *Twitter*. Speaker Deck umožňuje aj vloženie prezentácie priamo do kódu webovej stránky pomocou formátu *oEmbed*. Služba nie je spoplatnená.

SlideShare

SlideShare [22] je hostovacia služba pre prezentácie, infografiku, dokumenty a videá. Jej užívatelia môžu nahrávať súbory, privátne alebo verejne, vo formáte *PDF* alebo formátoch aplikácií *PowerPoint*, *Word* a *OpenDocument*. Nahraté súbory sú následne k dispozícii pre zobrazenie na stránkach služby.

SlideShare je v dnešnej dobe primárne úložisko pre prezentácie a od roku 2012 patrí pod sociálnu sieť *LinkedIn*. Jej prezentácie môžu byť hodnotené, komentované a zdieľané. Služba taktiež poskytuje funkciu *Zipcast*. *Zipcast* je sociálny webový konferenčný systém, ktorý

[11] https://code.google.com/archive/p/html5slides/
[12] https://www.princexml.com/

umožňuje prednášajúcim online vysielanie audio alebo video zdroja pri riadení prezentácie a pritom umožňuje účastníkom komunikáciu prostredníctvom zabudovaného chatu. Všetky možnosti služby sú k dispozícii zdarma.

SharePresentation

SharePresentation [59] je webová platforma, ktorá umožňuje užívateľom vyhľadávať, nahrávať a zdieľať prezentácie vo formátoch *PPT*, *PPTX*, *PPSX* a *PDF*. Služby platformy sú bezplatné a pre ich použitie je nutná registrácia.

Po nahratí prezentácie je možné upraviť jej názov, popis, zvoliť jej kategóriu a nastaviť ju buď verejnou alebo privátnou. Následne sa prezentácia spracuje a užívateľ získa odkaz na stránku prezentácie. Na stránke sa nachádzajú okrem samotnej prezentácie aj informácie o nej, jej autorovi a možnosť ju komentovať, zdieľať alebo stiahnuť. Po kliknutí na autora sa zobrazí jeho profil, kde sú vidieť všetky ním zverejnené prezentácie.

2.3 Prieskum vývojových prostredí

Pred vytvorením kurzu som uskutočnil okrem prieskumu vývojových prostredí aj prieskum študijných materiálov (viď sekcia 2.1) a prieskum prezentačných technológií (viď sekcia 2.2). Prieskum vývojových prostredí si kladie za úlohu zmapovať, aké prostredia je možné použiť pri vytváraní materiálov v programovacom jazyku C#.

Prieskum zisťuje, aké funkcionality jednotlivé prostredia majú, na akých operačných systémoch je možné ich použiť, aké technológie využívajú a čo všetko užívateľom ponúkajú. Základnou podmienkou pre každé prostredie je, aby vedelo spúšťať alebo kompilovať kód napísaný v jazyku C#.

2.3.1 Microsoft vývojové prostredia

Sekcia obsahuje prieskum vývojových prostredí vytvorených spoločnosťou *Microsoft*. No obsahuje aj prostredia, ktoré boli predchodcami *Microsoft* prostredí, kvôli čomu nie sú všetky prostredia v tejto časti vytvorené firmou Microsoft. Niektoré sú *open source* ale ich správcom je spoločnosť *Microsoft*.

Visual Studio

Visual Studio [42] je plnohodnotné integrované vývojové prostredie (*IDE*) pre operačný systém *Windows*, vyvinuté firmou *Microsoft*. Je používané na vytváranie počítačových programov pre *Windows*, web, cloud, mobilné zariadenia, ako aj k vývoju hier, *multiplatformných aplikácií*, prácu s databázou a ďalších. *Visual Studio* používa *Microsoft* softvérové vývojárske platformy, ako je *Windows API*, *Windows Form*, *WPF* alebo *Windows Store*. Vie produkovať ako strojový kód, tak aj kód bežiaci pod *CLR*.

Visual Studio obsahuje editor kódu obsahujúci nástroj IntelliSense, ktorý inteligentne dopisuje rozpísanú časť kódu, ako aj možnosti refraktorizácie. Vstavaný *debugger* funguje ako so zdrojovým kódom, tak aj so strojovým kódom. Prostredie okrem toho obsahuje veľké množstvo iných nástrojov, ako sú dizajnéri pre triedy, *UI*, weby a databázy, nástroj vyhľadávajúci optimalizácie kódu, nástroje na testovanie, analyzovanie kódu alebo napríklad nástroj na správu *Git* repozitára. Do prostredia je možné doinštalovať rozsiahle množstvo doplnkov, ktoré vedia rozšíriť jeho funkcionalitu takmer na každej úrovni.

Prostredie podporuje viac ako tridsať programovacích jazykov, vrátane jazyka C#. Zabudovaný editor kódu a *debugger* môže podporovať takmer každý programovací jazyk pomocou doinštalovania rozšírení. *Visual Studio* má taktiež vstavané nástroje na prácu s *Microsoft Azure*, vďaka ktorým je možné priamo z *Visual Studia* nasadzovať aplikácie priamo do *Azure*, robiť vzdialený *debugging* alebo spravovať zdroje.

Pokiaľ ide o vývoj, v posledných rokoch prichádza nová verzia prostredia každé dva roky. Najnovšia verzia, *Visual Studio 2019*, vyšla v apríli roku 2019. *Visual Studio* ponúka tri druhy platobných programov, nazývaných edície. Základná *Community* edícia je k dispozícii zadarmo. Edícia *Professional* je platená a určená pre menšie firmy a jednotlivcov. Edícia *Enterprise* je taktiež platená a určená pre veľké organizácie. Edícia *Professional* obsahuje všetky funkcie *Community* edície plus ďalšie prídavné, ako napríklad plná funkcionalita pre *CodeLens*, ktorá analyzuje kód. Edícia *Enterprise* obsahuje všetky funkcie edície *Professional*, plus ďalšie prídavné, ako napríklad *IntelliTrace* alebo *IntelliTest*. Presné rozdiely medzi verziami je možné nájsť na stránkach[13] platformy.

Visual Studio for Mac

Visual Studio for Mac vzniklo ako zmena názvu *Xamarin Studio* po verzii 6.3. *Xamarin Studio* vzniklo z prostredia *MonoDevelop* firmou *Xamarin*, pri jeho verzii 4.0. Prostredie *MonoDevelop* vzniklo oddelením sa od *SharpDevelop* a naďalej sa vyvíja osobitne.

Visual Studio for Mac [43] je integrované vývojové prostredie (*IDE*) pre operačný systém *macOS* od firmy *Xamarin* patriacej pod *Microsoft* a je k dispozícii zdarma. Prostredie vie vytvárať webové stránky s použitím *ASP.NET Core*, hry za pomoci *Unity* a mobilné aplikácie pre *iOS* a *Android* s použitím platformy *Xamarin*. Prostredie má podporu pre programovacie jazyky C#, F#, Razor, *HTML5*, *CSS*, *JavaScript*, *TypeScript*, *XAML* a *XML*.

Prostredie obsahuje mnoho rovnakých nástrojov ako aj jeho obdoba pre operačný systém *Windows*. Napríklad .NET kompilátor pomenovaný *Roslyn*, ktorý sa používa na *IntelliSense* a refaktorizáciu. Projekt systém používa *MSBuild*. Prostredie obsahuje aj *debugger*, integrovaný spúšťač testov a manažér pre verzný systém podobný tomu pre systém *Windows*.

Xamarin Studio [77] bolo samostatné IDE pre vytváranie mobilných aplikácií na operačných systémoch *Windows* a *macOS*. Prvý raz bolo vydané v roku 2013 z *open source* projektu *MonoDevelop* s pridaním nástroja *debugger*. Na *OS Windows* sa vývoj prostredia zastavil a nahradil ho *Xamarin* pre *Visual Studio*, zatiaľ čo na *OS macOS* je *Xamarin Studio* stále vo vývoji, ale od roku 2016 zmenilo názov na *Visual Studio for Mac*.

MonoDevelop [48] je *open source IDE* pre *Linux*, *macOS* a *Windows*. Obsahuje funkcie podobné tým v prostredí *Visual Studio*, ako je automatické dopĺňanie kódu, manažér verzného systému alebo web dizajnér. V *MonoDevelop* je integorvaný *UI* dizajnér s názvom *Stetic*. Prostredie vzniklo koncom roku 2003 oddelením od prostredia *SharpDevelop*.

SharpDevelop [16] je *open source IDE* šírené pod *MIT* licenciou. Funguje na OS *Windows* a obsahuje podobné funkcie ako staršie verzie prostredia *Visual Studio*. Slúži k vývoju aplikácií v programovacích jazykoch C#, Visual basic a Boo. Ku kompletizácii kódu používa prostredie vlastný parser. Posledná stabilná verzia vyšla v roku 2016.

[13] https://visualstudio.microsoft.com/vs/compare/

Visual Studio Code

Visual Studio Code [41] je editor zdrojového kódu od firmy *Microsoft* pre operačné systémy *Windows*, *Linux* a *macOS*. Zdrojový kód editoru je zdarma, *open source* a dostupný ako repozitár[14] v službe *GitHub*. Skompilovaná aplikácia je voľne použiteľná pre súkromné aj komerčné účely. Editor je postavený na *frameworku Elektron*, ktorý používa *Monaco*, rovnako ako *Azure DevOps*.

Pokiaľ ide o podporu jazykov, editor podporuje takmer všetky dnes používané programovacie jazyky. Nejaké z nich v základe, ako je napríklad C#, *JavaScript*, *CSS* alebo *HTML*. Pre podporu ďalších programovacích jazykov je možné stiahnuť rozšírenia. Rozšírenia umožňujú okrem podpory ďalších jazykov pridávať aj nové témy, *debuggery* a pripájať aplikáciu k ďalším službám. Rozšírenia bežia na separátnych procesoch, takže nespomaľujú samotnú aplikáciu.

Namiesto systému projektov, *Visual Studio Code* užívateľom povoľuje otvoriť jednu alebo viac zložiek, ktoré môžu byť uložené v pracovnom prostredí pre budúce použitie. Vďaka tomu môže aplikácia pracovať ako editor kódu pre akýkoľvek jazyk, narozdiel od prostredia *Visual Studio*, ktoré používa súbory riešení. Čo sa týka funkcionalít, *Visual Studio Code* podporuje debug aplikácií, zvýrazňovanie syntaxe a automatické dopĺňanie kódu za pomoci *IntelliSense*, a aj vstavané *Git* príkazy na prácu so vzdialeným repozitárom.

2.3.2 Vývojové prostredia tretích strán

V tejto sekcii som sa zameral na vývojové prostredia pre programovací jazyk C#, ktoré boli vytvorené tretími stranami a teda nevytvorila ich alebo ich nespravuje spoločnosť *Microsoft*. Vzhľadom na veľké množstvo rôznych vývojových prostredí, ako aj jednoduchých editorov a prekladačov, bolo možné zamerať sa iba na tie najužitočnejšie.

JetBrains Rider

JetBrains Rider [17] je platené multiplatformové integrované vývojové prostredie (*IDE*) pre operačné systémy *Windows*, *Linux* a *macOS* od spoločnosti *JetBrains*. Prostredie sa používa na vývoj programov založených na *.NET Framework*, *ASP.NET*, *.NET Core*, *Xamarin* a *Unity*. Poskytuje bohatú podporu pre editáciu a písanie kódu. Podporuje programovacie jazyky C#, VB.NET, F#, *JavaScript*, *TypeScript*, *XAML*, *XML*, *HTML*, *CSS*, *JSON*, *SQL* a ďalšie. *Rider* dokáže otvárať, upravovať, kompilovať, spúšťať a debugovať väčšinu .NET aplikácií s výnimkou *UWP*.

Prostredie je veľmi výkonné, s rýchlou odozvou na požiadavky vďaka tomu, že nie je zakliesnené do jedného procesu. Obsahuje inteligentný editor kódu s mnohými možnosťami doplňovania, automatického importu častí a používania šablón. Prostredie taktiež používa cez dvetisícdvesto inšpekcií kódu, ktoré pomáhajú nájsť chyby a možné problémy, väčšina z nich obsahuje aj možnosť automatickej rýchlej opravy. Ďalšími funkciami prostredia sú rýchla navigácia a hľadanie v kóde, ktoré umožňuje aj hľadanie v celom riešení medzi viacerými použitými programovacími jazykmi, dekompilátor kódu, *debugger*, správa *unit testov*, rozsiahle možnosti refaktorizácie, správca verzií kódu a *SQL* editor. Väčšinu funkcií získalo prostredie z rozšírenia *Resharper*, ktoré je ňom priamo vstavané. Ďalšie funkcionality je možné doinštalovať pomocou rozšírení.

[14] https://github.com/Microsoft/vscode

Resharper [18] je spoplatnené rozšírenie do prostredia *Visual Studio* pre *.NET* vývojárov od spoločnosti *JetBrains*. Zároveň je použité v prostredí *Rider*, ktorému dodáva veľké množstvo hlavných funkcionalít.

Medzi funkcionality patrí analýza kvality kódu a navrhovateľ automatických opráv a vylepšení kódu. Okrem toho rozšírenie vylepšuje *IntelliSense*, pridáva možnosti automatickej transformácie kódu a rôzne možnosti refaktorizácie, automatické importovanie *namespace* a vytváranie časti kódu za pomoci šablón, vylepšené hľadanie a rozšírenú navigáciu. Rozšírenie pridáva možnosť vytvoriť si štandard, akým má byť kód napísaný a čistiť kód pomocou tohto štandardu. Medzi ďalšie funkcionality patrí správca *unit testov*, asistent pri debugovaní, zobrazovač vzájomností projektov, špeciálne funkcie pre jednotlivé jazyky a iné.

scriptcs

scriptcs [57] je softvérový balíček distribuovaný pomocou manažéra balíčkov *Chocolatey*. Balíček je *open source*, pod *Apache* licenciou a jeho zdrojové súbory sa nachádzajú v *GitHub* repozitári[15]. Stránka *Katacoda* umožňuje vyskúšanie balíčka[16] priamo vo webovom prehliadači.

Balíček umožňuje spúšťanie kódu napísaného v jazyku C# bez použitia vývojového prostredia. Jednoducho sa mu ako argument predá cesta k súboru s kódom, alebo sa kód napíše priamo doň. Balíček následne kód vykoná. Vďaka tomu je možné písať C# kód v ľubovolnom editore. Balíček taktiež spravuje *NuGet* závislosti.

2.4 Zmysel technologického kurzu

Technologický kurz si kladie za úlohu naučiť základy programovacieho jazyka C# vývojárov, ktorí už majú skúsenosti s objektovo orientovaným programovaním (*OOP*), avšak s programovacím jazykom C# majú iba minimálne alebo žiadne skúsenosti. Popritom zdôrazňuje dobré programátorské praktiky, ktoré sa v tomto jazyku používajú. Účastník by sa mal okrem písania čistého a funkčného kódu naučiť aj vytvárať návrh aplikácií, tímovú spoluprácu a správne používanie návrhových vzorov. Po absolvovaní by mal byť schopný samostatne vytvoriť *desktopovú aplikáciu* s pripojením na databázu. Presné znalosti, ktoré absolventi kurzu získajú, je možné nájsť v sekcii 3.5.

Každá z teoretických prednášok zoznamuje účastníka kurzu s jednou témou. Pomocou praktických cvičení účastník získané vedomosti z prednášok spája a využíva ich na vyriešenie jedného, komplexného problému, ktorým je vytvorenie *desktopovej aplikácie*. Výsledná aplikácia je rozdelená do viacerých vrstiev a používa niekoľko rôznych technológií. Vrstvy a technológie sú spolu prepojené a tým vytvárajú súhrnne riešenie problému.

Zmyslom kurzu je priniesť tieto znalosti účastníkom bezplatne. Zdrojové súbory kurzu sú voľne dostupné na stiahnutie v *GitHub* repozitorári ako *open source* projekt pod *MIT* licenciou. Licencia umožňuje použiť všetky zdrojové súbory kurzu ku komerčným aj súkromným účelom. Možná je aj modifikácia a ďalšia distribúcia zdrojových súborov. Súbory sú v anglickom jazyku, aby bolo možné ich použitie na svetovej úrovni, viac o jazyku viď sekcia 4.2.

Pri výbere vhodných technológií na tvorbu a distribúciu materiálov sa kládol dôraz na použitie moderných technológii. Použité technológie umožňujú jednoduchú úpravu zdrojových súborov kurzu, ako aj použitie súborov z praktických cvičení priamo v teoretických prezentáciách. Zdrojové súbory kurzu môžu byť, vďaka technológii *pull request* menené aj ďalšími osobami. Viac o použitých technológiách viď sekcia 4.4.

[15] https://github.com/scriptcs/scriptcs
[16] https://www.katacoda.com/courses/dotnet-in-docker/scriptcs-playground

Kapitola 3

Náplň technologického kurzu

Kapitola sa zaoberá náplňou technologického kurzu, jeho osnovou (viď sekcia 3.1), anotáciami jednotlivých častí (viď sekcia 3.3) a časovým harmonogramom (viď sekcia 3.2). Táto kapitola zároveň uvádza, aké znalosti a iné prerekvizity musí uchádzač splniť pred zápisom kurzu (viď sekcia 3.4) a takisto poznatky a vedomosti, ktoré uchádzač získava jeho úspešným absolvovaním (viď sekcia 3.5).

3.1 Osnova technologického kurzu

Osnova technologického kurzu sa skladá z dvoch celkov: teoretických prednášok a praktických cvičení. Teoretická časť sa skladá z trinástich prednášok (viac k prednáškam viď sekcia 4.3.2) a praktická časť sa skladá z piatich cvičení (viac k cvičeniam viď sekcia 4.3.3). Pre zdôvodnenie, prečo je kurz rozdelený týmto spôsobom viď sekcia 4.3.1.

3.1.1 Osnova prednášok

Prednášky sú rozdelené do trinástich častí podľa preberaných tématických celkov. Ku každej prednáške je vytvorená teoretická prezentácia, pre viac informácií k forme teoretických materiálov viď sekcia 4.3.2. Samotná osnova prednášok vyzerá následovne:

1. Úvod do jazyka C#, prostredia *Visual Studio* a *.NET*
2. Objektovo orientované programovanie (*OOP*) a pokročilé konštrukty v jazyku C#
3. *.NET Standard* a *LINQ*
4. *Entity Framework*
5. Testovanie aplikácií a Continous integration (*CI*) v jazyku C#
6. Návrhové vzory v jazyku C#
7. Čistý, udržiavateľný kód
8. Návrhové vzory *repozitár*, *UnitOfWork*, *mapper* a *fasáda*
9. Návrhový vzor *Model–View–ViewModel* (*MVVM*)
10. *Windows presentation foundation* (*WPF*)
11. *Paralelné programovanie* a *asynchrónne programovanie* v jazyku C#
12. Profilácia výkonu a správa pamäte v jazyku C#
13. *Multiplatformné* programovanie, *.NET Core*, *kontajnerizácia* aplikácií

3.1.2 Osnova cvičení

Cvičenia sú rozdelené do piatich častí podľa počtu dní kurzu. Na každý deň pripadá jedno praktické cvičenie, ktoré zahŕňa prax na všetku prebratú teóriu od prechádzajúceho cvičenia. Pre inšpiráciu a odôvodnenie rozdelenia viď sekcia 4.3.1. Každý praktický materiál obsahuje dve *C# solution*, pre viac informácií k forme praktických materiálov viď sekcia 4.3.3. Samotná osnova cvičení vyzerá následovne:

1. Práca v prostredí *Visual Studio*
2. Správa dát pomocou technológie *Entity Framework*
3. Čistý, udržiavateľný kód
4. Vytváranie viewmodel, základy *WPF*
5. Viazanie *WPF* view s viewmodel

3.2 Časový harmonogram kurzu

Dĺžka kurzu je stanovená na jeden pracovný týždeň, pričom každý deň začína o 8:00 hod. a končí o 17:00 hod. V tomto časovom intervale kurz obsahuje prednášky, cvičenia a prestávky. Pre viac informácií o rozdelení materiálov viď sekcia 4.3.

Pondelok	Utorok	Streda	Štvrtok	Piatok
8:00-10:00 Prednáška č. 1 — Úvod do jazyka C#, prostredia *Visual Studio* a *.NET* **10:00-10:20** Prestávka **10:20-12:00** Cvičenie č. 1 — Práca v prostredí *Visual Studio* **12:00-13:00** Prestávka na obed **13:00-14:40** Prednáška č. 2 — Objektovo orientované programovanie (*OOP*) a pokročilé konštrukty v jazyku C# **14:40-15:00** Prestávka **15:00-17:00** Prednáška č. 3 — *.NET Standard* a *LINQ*	**8:00-9:40** Prednáška č. 4 — *Entity Framework* **9:40-10:00** Prestávka **10:00-12:00** Cvičenie č. 2 — Správa dát pomocou technológie *Entity Framework* **12:00-13:00** Prestávka na obed **13:00-14:40** Prednáška č. 5 — Testovanie aplikácií a Continous integration (*CI*) v jazyku C# **14:40-15:00** Prestávka **15:00-17:00** Prednáška č. 6 — Návrhové vzory v jazyku C#	**8:00-10:00** Prednáška č. 7 — Čistý, udržiavateľný kód **10:00-10:30** Prestávka **10:30-12:30** Cvičenie č. 3 — Čistý, udržiavateľný kód **12:30-14:00** Prestávka na obed **14:00-17:00** Prednáška č. 8 — Návrhové vzory *repozitár*, *UnitOfWork*, *mapper* a *fasáda*	**8:00-10:00** Prednáška č. 9 — Návrhový vzor *Model–View–ViewModel* (*MVVM*) **10:00-10:20** Prestávka **10:20-12:30** Cvičenie č. 4 — Vytváranie viewmodel, základy *WPF* **12:30-14:00** Prestávka na obed **14:00-17:00** Prednáška č. 10 — *Windows presentation foundation* (*WPF*)	**8:00-9:30** Prednáška č. 11 — *Paralelné programovanie* a *asynchrónne programovanie* v jazyku C# **9:30-10:00** Prestávka **10:00-12:20** Cvičenie č. 5 — Viazanie *WPF* view s viewmodel **12:20-13:20** Prestávka na obed **13:20-15:00** Prednáška č. 12 — Profilácia výkonu a správa pamäte v jazyku C# **15:00-15:20** Prestávka **15:20-17:00** Prednáška č. 13 — *Multiplatformne* programovanie, *.NET Core*, kontajnerizácia aplikácií

3.3 Anotácie technologického kurzu

Táto časť práce obsahuje stručný opis každej prednášky a cvičenia podľa stanovej osnovy. Osnova viď sekcia 3.1. Pre odôvodnenie rozdelenia opisovaných materiálov do prednášok a cvičení viď sekcia 4.3. Pre súhrn znalostí, ktoré účastníci z prednášok a cvičení získajú viď sekcia 3.5.

3.3.1 Anotácie prednášok

Ku každej prednáške je vytvorená prezentácia k zadanej téme. Pre viac informácií k teoretickým materiálom viď sekcia 4.3.2. Každá prednáška preberá jednu komplexnú tému, ale vzájomne na seba nenadväzujú. Z tohto dôvodu je možné v niektorých prípadoch zmeniť poradie prednášok, ale nie vždy sa tento krok odporúča.

Prednášky obsahujú teoretické materiály k zadanej téme, ako aj referencie na jej zdroje. Nachádzajú sa v nich aj odkazy na stránky s rozširujúcimi materiálmi pre prípad, že by účastník potreboval získať viac informácií, pretože tie v prezentáciách považuje za nepostačujúce. Prednášky naviac obsahujú veľké množstvo ukážok kódu, z ktorých väčšina zahrňuje aj *unit* testy, aby si mal účastník možnosť vyskúšať, ako naozaj fungujú.

Úvod do jazyka C#, Visual Studia a .NET

Prvá prednáška uvádza do kurzu, zoznamuje účastníkov s priebehom kurzu a jeho rozložením podľa časového harmonogramu (viď sekcia 3.2). Prednáška obsahuje aj odporúčanú študijnú literatúru ku kurzu.

Táto prednáška v rýchlosti uvádza aj do vývojového prostredia *Visual Studio* a možností, aké vývojárom poskytuje. Odporúča tiež *Visual Studio* rozšírenia a samostatné programy, ktoré sú pre kurz vhodné.

Účastníkom kurzu je prednášané aj o platforme *.NET*. Najprv sa užívateľ dozvie o jej výhodách, prínosoch a možnostiach použitia. Z výhod sa postupne prechádza na jej architektúru a fungovanie jej jednotlivých vrstiev.

Najväčšia časť prednášky sa zaoberá základmi jazyka C#. Postupne prechádza jeho kľúčové slová, *identifikátory*, *operátory* a dátové typy. Vysvetľuje fungovanie *referenčných* dátových typov a *polí*, pričom zľahka uvádza do *OOP*. Venuje sa rôznym typom *parametrov*, ukazuje, ako napísať *podmienku*, *switch*, všetky typy *cyklov* a ostatné riadiace štruktúry.

Objektovo orientované programovanie a pokročilé konštrukty v jazyku C#

Druhá prednáška sa venuje najmä objektovo orientovanému programovaniu (*OOP*). V prvej časti vysvetľuje princípy *OOP*, následne plynule prechádza objektovú orientáciu jazyka C#, kde preberá *referenčné* typy, *identifikátory*, *modifikátory prístupu* a *operátory* pretypovania.

Prednáška sa zaoberá *referenčným* typom *trieda*, kde podrobne rozoberá komponenty, akými sú *field*, *property*, *metóda*, *konštruktor*, *dekonštruktor* a *finalizer*, ale aj jej *abstraktný*, *sealed* a *partial* druh, čo tvorí tú najväčšiu časť prednášky. Zaoberá sa aj typmi interface (*rozhranie*), *enum*, *štruktúra*, pričom vysvetľuje, kedy je lepšie použiť *rozhranie* a kedy *abstraktnú triedu*.

V ďalšej časti sa prednáška venuje pokročilým konštruktom. Začína typovou bezpečnosťou, kde vysvetľuje výhody *generík*, na ktoré postupne prechádza a venuje im podstatnú časť prednášky. Následne sa venuje *výnimkám* a tomu, ako ich správne vyhadzovať a odchytávať.

Po *výnimkách* sa v prednáške rozoberajú *delegáty*, ich fungovanie a vhodné použitie, ale aj výhody a nevýhody oproti typu *rozhranie*. Prednáška sa ďalej venuje konštruktu *event* a jeho štandardnému zápisu. V závere prednášky sa objasňujú *lambda výrazy* a konštrukt *tuple*.

.NET Standard a Language Integrated Query

Prednáška sa skladá z dvoch hlavných tém, prvou z nich je *.NET Standard*. Po rýchlom úvode, v ktorom prednáška vysvetľuje, čo to *.NET Standard* je, význam jeho použitia a stručnú históriu jeho verzií, sa prednáška presúva na jeho *API*, ktoré predstavuje väčšiu časť prednášky. Najdôležitejší *namespace*, System.Namespace, sa podrobne analyzuje a po ňom sa postupne prechádza každý ďalší dôležitejší *namespace*. Vďaka tomu sa užívateľ zoznámi s tými najpoužívanejšími knižnicami jazyka.

V druhej časti sa prednáška zaoberá Language Integrated Query (*LINQ*), pričom začína vysvetlením obidvoch druhov syntaxe a rozdielov medzi nimi, výhod a nevýhod a kedy je vhodné jednu z nich použiť. Následné sa venuje vnútornému fungovaniu *LINQ* a všetkým *operátorom*, pričom ku každému z nich obsahuje aj príklad, pre jeho jednoduchšie pochopenie. Záver prednášky obsahuje ukážky dopytovania nad *SQL*, *XML*, *JSON* a zoznam ďalších možností, na ktoré sa dá *LINQ* použiť.

Entity Framework

Podstatná časť kurzu sa venuje vytvoreniu aplikácie, na ktorej pozadí budú dáta uložené v databáze. Prvým potrebným krokom k dosiahnutiu tohto cieľa je dostať do aplikácie dáta z databázy. Touto problematikou sa zaoberá celá prednáška.

V prvej časti prednáška vysvetľuje základne pojmy ako je *framework* a databáza, ktorej sa v ďalšej časti venuje podrobnejšie a rozdeľuje jej komponenty. Následne vysvetľuje čo je to perzistentnosť, *ACID*, *CAP*, *CRUD*, *DAL*. Po vysvetlení základných pojmov sa venuje rozdielnym typom databáz, podrobnejšie Microsoft *SQL* databázam a vysvetľuje, ako sa pomocou *Visual Studia* pripojiť k *SQL* databáze.

V nasledujúcej časti vysvetľuje, čo je *ORM*, na čo je vhodné a jeho výhody. V plynulej nadväznosti na *ORM* vysvetľuje rôzne druhy technológií umožňujúce prepojenie databázy k aplikácii vytvorenej pomocou programovacieho jazyka C#.

Konkrétne sa venuje technológiám *ADO.NET*, *Entity Framework*, *Dapper* a *NHibernate* a to poslednú, najväčšiu časť prednášky. Ku každej technológii sú vysvetlené jej výhody a nevýhody oproti ostatným, spolu s krátkym príkladom ich použitia. Prednáška v poslednej časti obsahuje výkonnostné porovnanie jednotlivých technológií.

Technológii *Entity Framework* sa prednáška venuje najpodrobnejšie, keďže sa v nasledujúcich častiach kurzu používa. Prednáška vymenúva a vysvetľuje možnosti a verzie daného *frameworku*, jeho inštaláciu a akým spôsobom sa pomocou neho pripojiť k databáze a zachovávať ju perzistentnou. Prechádza základne objekty a štruktúry a možnosti *frameworku*, celkovo prechádza *d*o veľkej hĺbky a informuje účastníka kurzu o takmer všetkých jeho možnostiach.

Testovanie aplikácií a Continous integration v jazyku C#

Prednáška sa skladá z dvoch hlavných tém. Prvá sa venuje testovaniu aplikácií a rôznym druhom testov a druhá Continous integration (*CI*). Prednáška začína krátkym úvodom, v ktorom sa venuje rôznym typom testovacích *frameworkov* podporovaných vo *Visual Stu-*

diu, ku každému ukáže ako *framework* použiť a následne sa podrobnejšie venuje *frameworku xUnit*. Vysvetľuje spôsoby ako spúšťať testy a vhodný spôsob písania testov ako ďalšie možnosti, ktoré *xUnit* umožňuje, medzi ktoré patrí napríklad paralelné a sériové spúšťanie testov. V poslednej časti sa zameriava na pokrytie kódu testami a rôzne druhy testov, prioritne na *smoke*, *unit*, *UI*, *integračným* a *akceptačným* testom.

V druhej časti prednáška vysvetľuje, čo je to Continous integration a aké poskytuje výhody. Rozoberá rôzne *CI* technológie, pričom sa zameriava na službu Azure DevOps[1], ktorú podporuje aj prostredie *Visual Studio*. Objasňuje možnosti služby, ako *Git* repozitáre, *pipelines*, plánovanie vývoja, rozdeľovanie práce v tíme, wiki, testovanie aplikácií a ďalšie. Pri technológii *Git* taktiež vysvetľuje rôzne modely vetvenia a ich výhody, dopodrobna vysvetľuje model *GitFlow*.

Návrhové vzory v jazyku C#

Cieľom prednášky je zoznámiť účastníkov s návrhovými vzormi. Po krátkom úvode, v ktorom vysvetľuje, čo to vlastne návrhové vzory sú, ich začína prechádzať jednotlivo. Prednáška ku každému z nich obsahuje definíciu, frekvenciu použitia, účastníkov, *UML* diagram a príklad aj s výstupom.

Návrhové vzory sú rozdelené do troch skupín, pričom sa informácie čerpajú z Gang of Four (*GoF*). Prvú skupinu tvoria vytváracie vzory, v prednáške ju zastupuje *Abstract Factory*, *Builder*, *Factory Method*, *Prototype* a *Singleton*. Druhú skúpinu tvoria vzory na štrukturovanie a v prednáške ich zastupuje *Adapter*, *Bridge*, *Composite*, *Decorator*, *Facade*, *Flyweight*, *Proxy*. A poslednú skupinu tvoria vzory, ktoré sa vysporiadavajú so správaním, konkrétne ide o *Chain of Responsibility*, *Command*, *Interpreter*, *Iterator*, *Mediator*, *Memento*, *Observer*, *State*, *Strategy*, *Template Method*, *Visitor*.

Čistý, udržiavateľný kód

Hľavným cieľom prednášky je naučiť účastníkov písanie čistého a udržiavateľného kódu. Táto prednáška vysvetľuje *Clean Code* a dôvod, prečo je taký dôležitý. Detailne preberá zásady písania *Clean Code*, akými sú vhodné názvy, vhodné štruktúrovanie, formátovanie kódu a kedy používať komentáre a akého druhu. Popisuje nástroje a doplňujúcu literatúru, ktoré sú schopné, pri dodržiavaní istých zásad, písanie kódu *Clean Code* zjednodušiť. Prednáška obsahuje aj ukážky správnej refaktorizácie kódu a vhodnú prácu s *legacy kódom*, pričom sa zameriava aj na nástroj Reshaper (viď odsek 2.3.2), ktorý vie refaktorizáciu kódu výrazne uľahčiť.

Najväčšiu časť prednášky tvorí *SOLID*, päť pravidiel pre písanie čistého a testovateľného kódu. Všetky pravidlá sú v prednáške podrobne vysvetlené, pričom sa kladie dôraz na ich správne pochopenie s názornými príkladmi. Okrem *SOLID* sa prednáška venuje aj princípom *DRY* a *KISS*, *GRASP* a ďalším.

Návrhové vzory repozitár, UnitOfWork, mapper a fasáda

Prednáška si kladie za cieľ vysvetliť účastníkovi návrhové vzory, ktoré je vhodné použiť na prepojenie *Entity Frameworku* s aplikáciou.

Ako prvému sa prednáška venuje *repozitáru*. Vysvetľuje, čo to *repozitár* je, jeho výhody a zodpovednosti s jednoduchým príkladom použitia. Následne ukazuje rozdiely medzi ním a návrhovým vzorom *UnitOfWork*, pričom vysvetľuje, čo to *UnitOfWork* je a na čo slúži.

[1] https://azure.microsoft.com/en-us/services/devops/

Taktiež sa venuje tomu, prečo by rozhrania DbSet a DbContext z *Entity Frameworku* nemali byt považované za repozitár a *UnitOfWork*.

V nasledujúcej časti sa prednáška venuje návrhovému vzoru *mapper*, vysvetľuje, čo to *mapper* je a ako si naprogramovať. Následne ukazuje *framework* AutoMapper[2], jeho výhody, možnosti a použitie na rovnakom príklade ako prvý jednoduchý *mapper*.

V poslednej časti prednáška vysvetľuje návrhový vzor *fasáda*, jeho účastníkov a definíciu. Ku všetkým návrhovým vzorom v prednáške je vytvorené konkrétne použitie priamo napojené na projekt obsahujúci *Entity Framework*. Ide o rozšírenie príkladu z prednášky o *Entity Frameworku*.

Návrhový vzor Model–View–ViewModel

Prvú časť prednášky tvorí architektúra aplikácií, rôzne návrhové vzory používané pri rôznych druhoch aplikácií. Najväčší dôraz sa kladie na vysvetľovanie desktopových aplikácií, ich výhodám a nevýhodám a architektonickým vzorom, ktoré predchádzali vzor Model–View–ViewModel (*MVVM*). Konkrétne vzorom Model–View–Controller (*MVC*), Model–View–Presenter (*MVP*) a prezentačnému modelu.

Najväčšiu časť prednášky tvorí samotný architektonický návrhový vzor Model–View–ViewModel (*MVVM*), jeho výhody, nevýhody, použitie a podpora v rôznych jazykoch. Následne sa prechádzajú jeho základné komponenty a vysvetľuje sa ich implementácia na rozšírení príkladu z predchádzajúcej prednášky.

V ďalšej časti prednášky sa názornými príkladmi vysvetľujú návrhové vzory *Messenger*, *ViewModelLocator*, a *IoC* kontajner, ktoré použité *MVVM* značne spriehľadňujú.

V poslednej časti sa prednáška venuje rôznym *MVVM frameworkom*, ku každému obsahuje stručný výčet jeho možností a odkaz na ďalšie informácie. Podrobne sa venuje *frameworku* MVVM light[3], ktorého funkcionality jednotlivo preberá a taktiež implementuje do príkladu z predchádzajúcej prednášky, vďaka čomu je možné porovnať kód, ktorý bol vytvorený s a bez použitia *frameworku*.

Windows presentation foundation

Prednáška sa zaoberá užívateľským rozhraním, jeho vytvorením pomocou technológie Windows presentation foundation (*WPF*) a jeho následne prepojenie so zvyškom aplikácie pomocou návrhového vzoru Model–View–ViewModel (*MVVM*).

V prvej časti prednáška vysvetľuje, čo to Windows presentation foundation (*WPF*) je, jeho fungovanie, výhody a čo všetko umožňuje vytvárať. Následne ukazuje základy písania *WPF* kódu pomocou jazyka *XAML* na príkladoch, pričom objasňuje význam jednotlivých základných elementov, taktiež vysvetľuje, čo to *XAML* je a aké ma možnosti.

Druhá časť prednášky zobrazuje rozdiel medzi deklaratívnou a imperatívnou notáciou. Prechádza základné komponenty a hierarchiu *tried*, ktorým venuje podstatnú časť prednášky. Na záver v rýchlosti uvádza do prostredia *Blend for Visual Studio*.

Nasledujúcu časť prednášky tvorí *data binding*. Táto prednáška učí jeho rôzne druhy a smery a zároveň vysvetľuje, ako fungujú na názorných príkladoch. Venuje sa aj aplikovaniu *data binding* na *kolekcie*, konverterom, otváraniu nových view a návrhovému vzoru *Command* a jeho použitiu pri prepojení view a viewmodelu.

[2] https://automapper.org/
[3] http://www.mvvmlight.net/

V ďalšej časti sa prednáška zaoberá rozšírenejšími funkcionalitami *WPF*. Ukazuje, ako vytvárať vlastné komponenty, a ako používať už vopred vytvorené štýly a sady na jednoduchých príkladoch pričom na to taktiež ukazuje *framework* Material Design[4], slúžiaci na preštýlovanie aplikácií.

V úplne poslednej časti sa prednáška v krátkosti zaoberá technológiami, ktoré používajú *WPF*. Účastníkov kurzu uvádza do technológií *Silverlight*, Universal Windows Platform (*UWP*) a *Xamarin*. Pri každej z nich vysvetľuje ich výhody, nevýhody a kedy je vhodné ktorú z nich použiť.

Paralelne a asynchrónne programovanie v jazyku C#

Prednáška postupne prechádza všetky možnosti, ktoré poskytuje *paralelné programovanie* a *asynchrónne programovanie* v programovacom jazyku C#. Najprv uvádza do danej problematiky, vysvetľuje jej obmedzenia a vhodné možnosti použitia.

V prvej časti prednášky sa vysvetľuje rozdiel medzi sériovými a paralelnými výpočtami, výhodami paralelných a možnosťami ich použitia. Následne sa venuje rozdielu medzi synchrónnymi a asynchrónnymi výpočtami a taktiež výhodami a problémami, ktoré prichádzajú s používaním asynchronity.

Najväčšiu časť prednášky predstavujú jednotlivé možnosti, konkrétne *proces*, *vlákno*, *task*, s príkladmi samotného použitia. Prednáška rozoberá ich rozdiely, výhody a možnosti, ktoré jednotlivé úrovne abstrakcie poskytujú.

V závere sa venuje kľúčovým slovám async a await, ich výhodám, štruktúre a správe výnimiek pri ich použití. K tomu sú uvedené príklady, v ktorých samotné objekty v *.NET Frameworku* kľúčové slova používajú, ako aj objekty umožňujúce paralelný prístup.

Profilácia výkonu a správa pamäte v jazyku C#

Prednáška detailne preberá procesy, ktoré slúžia na profiláciu výkonu a správu pamäte v programovacom jazyku C#. V prvej časti sa zameriava na to, ako jednotlivé procesy v jazyku C# prebiehajú, podrobne preberá funkciu *garbage collectora*, ktorý manažuje správu pamäte v jazyku.

Druhá časť prednášky sa zameriava na externé programy a rozšírenia do prostredia *Visual Studio*, ktoré vedia manažment výrazne zjednodušiť. Podrobne sa venuje *NuGet* balíčku BenchmarkDotNet [5].

Multiplatformové programovanie, .NET Core, kontajnerizácia aplikácií

Prvá časť prednášky sa zameriava na *multiplatformové* programovanie, ktoré jazyk C# a platforma *.NET* umožňujú. Prednáška sa venuje všetkým známejším technológiám, na rôznych platformách, pričom aj štandardu *.NET Standard*, ktorý väčšina z nich podporuje. Najväčšiu čas venuje technológii *.NET Core*, ktorá je jednou z aktuálne najpoužívanejších.

V druhej časti prednáška preberá *kontajnerizáciu* aplikácií. Vysvetľuje základné pojmy z oblasti kontajnerizácie a následné rozoberá technológiu *Docker*, ktorá kontajnerizáciu umožňuje. Účastník sa naučí, ako zkontajnerizovať jednoduchú *.NET Core* aplikáciu, ako s ňou v technológii *Docker* pracovať a akú podporu poskytuje prostredie *Visual Studio* pre *Docker* a *orchestracia aplikácií*orchestráciu aplikácií.

[4] http://materialdesigninxaml.net/
[5] http://materialdesigninxaml.net/

3.3.2 Anotácie cvičení

Ku každému cvičeniu sú vytvorené *C# solution* na zadanú tému. Pre viac informácií k materiálom viď sekcia 4.3.3. Cvičenia na seba vzájomne nadväzujú a z toho dôvodu nie je možné meniť ich poradie.

Cvičenia obsahujú praktické materiály, kód v jazyku C# ku zadanej téme. Účastník kurzu si v nich postupne skúsi vytvoriť desktopovú aplikáciu, napojenú na databázu. Každé cvičenie obsahuje aj referenčný *C# solution*, ktorým si môže účastník následne skontrolovať svoju prácu.

Práca v prostredí Visual Studio

Účastník sa na cvičení zoznámi s prácou v prostredí *Visual Studio*. Cvičenie na začiatku obsahuje demonštráciu inštalácie tohto prostredia, aj jeho potrebných komponentov a rozšírení. Po dokončení inštalácie prechádza jednotlivé pohľady, ktoré Visual Studio obsahuje, ako aj prispôsobenie na mieru samotnému účastníkovi.

V druhej časti cvičenia sa vytvárajú rôzne typy projektov. Účastníci sa učia pripojiť ku repozitáru pomocou technológie *Git* a oboznamujú sa s vetviacim modelom *GitFlow*. Následne sa prechádza refaktorizácia pomocou nástroja *Resharper* (viď odsek 2.3.2), a ako efektívne robiť *debugging* a odchytávať výnimky. Na konci sa demoštrujú najčastejšie používané skratky.

V poslednej časti cvičenia si účastníci sami skúsia vytvoriť vlastný projekt, vytvoriť jednoduchú aplikáciu a upraviť už predom vytvorenú aplikáciu kalkulačka, ktorá demonštruje základné časti jazyka.

Správa dát pomocou technológie Entity Framework

Cvičenie začína demonštráciou inštalácie *NuGet* balíčka a práce s konzolou, slúžiacou na správu balíčkov, pričom ako príklad využíva technológiou *Entity Framework* code first.

Spočiatku cvičenia účastníci pracujú s dvomi, už predom vytvorenými projektami. Do prvého je za úlohu vytvoriť *databázovú vrstvu aplikácie* pomocou *Entity Frameworku* a do druhého vytvoriť testy na ňu.

Pri vytváraní *databázovej vrstvy* je nutné využiť skúsenosti z demonštrácie zo začiatku cvičenia. Je nutné nainštalovať *NuGet* balíček, pracovať s konzolou, vytvoriť objekt reprezentujúci prístup k databáze a entity reprezentujúce jednotlivé tabuľky z databázy, v neposlednom rade je potrebné povoliť a vytvoriť migráciu a overiť funkčnosť vytvorenej *databázovej vrstvy* pomocou druhého projektu, v ktorom budú vytvorené testy demonštrujúce jej fungovanie a spôsob použitia.

Čistý, udržiavateľný kód

Účastníkovi sa na cvičení predstaví funkčný kód aplikácie, ktorý je ale vo veľmi zlom stave z hľadiska čistoty a udržateľnosti. V priebehu cvičenia sa kód za spolupráce účastníkov upravuje, aby dodržiaval zásady *Clean Code*, ktoré účastníci poznajú z prednášky.

Od účastníkov sa taktiež očakáva použitie nástroja Resharper (viď odsek 2.3.2), ktorý vie čistenie a refaktorizáciu kódu značne zjednodušiť. Taktiež sa od učastníkov očakáva používanie skratiek, ktoré keď si osvoja, tak budú vedieť pracovať s kódom značne rýchlejšie.

Cvičenie ako jediné nenadväzuje na ostatné, a preto je možné ho zameniť aj za iné. Je nutné ale dbať na to, aby účastníci už mali odprezentovanú prednášku o čistote kódu (viď sekcia 3.3.1).

Vytváranie viewmodelu, základy WPF

Cvičenie nadväzuje na vytvorenú *databázovú vrstvu* aplikácie z druhého cvičenia. Účastníci pokračujú vo vytváraní *desktopovej aplikácie*, založenej na návrhovom vzore Model–View–ViewModel (*MVVM*).

V prvom kroku účastníci vytvoria model a pomocou návrhového vzoru *mapper* ho prepoja s entitami z *databázovej vrstvy*, ku ktorým sa následne vytvorí *repozitár*. Následne sa vytvori jednoduchý viewmodel, pre každé z budúcich view, ktorý bude obsahovať základne *metódy*. Samotné view sa v cvičení nevytvárajú.

Taktiež sa prechádzajú základy práce a vytvorenie prvých, jednoduchých užívateľských rozhraní pomocou Windows Presentation Foundation (*WPF*).

Viazanie WPF view s viewmodel

Posledné cvičenie nadväzuje na to prechádzajúce, v ktorom sa vytvárala *desktopová aplikácia* po viewmodel. V tomto cvičení sa aplikácia dokončí a vykonajú sa všetky finálne úpravy a náležitosti, aby bola plne funkčná.

Prvú časť cvičenia predstavuje dokončenie viewmodelu z predchádzajúceho cvičenia, následne vytvorenie view a ich následné viazanie. Účastník si vyskúša rôzne formy viazania, ktoré umožňuje *data binding* a návrhový vzor *Command*, ako aj prepojenie a komunikáciu viacerých viewmodel medzi sebou.

Posledná časť cvičenia rieši rôzne pokročilé možnosti jazyka *XAML*, ako sú prevádzače, vlastné komponenty a štýlovanie aplikácie pomocou Material Design *NuGet* balíčku.

3.4 Prerekvizity účastníkov kurzu

Kurz je určený pre skúsených programátorov, ktorí už majú vedomosti a skúsenosti s programovaním a rozumejú konceptom *procedurálneho*, *funkcionálneho* a objektovo orientovaného (*OOP*) programovania.

Kurz nie je vhodný pre úplných začiatočníkov v oblasti programovania, pretože je mierený priamo na profesionálov s minimálne polročnými skúsenosťami v programovacích jazykoch *C*, *C++* alebo *Java*.

Základné znalosti programovacieho jazyka C# nie sú vyžadované, ale sú nápomocné pri rýchlejšom porozumení zložitejších konceptov jazyka. Vyžadované znalosti v ľubovoľnom jazyku z *rodiny C* sú:

- Vedieť pomenovať, deklarovať, inicializovať a priradiť *hodnotu premennej*
- Vedieť používať primárne dátové typy a ovládať ich vzájomné pretypovania
- Vedieť používať aritmetické *operátory* v kontexte s *premennými*
- Vedieť používať relačné *operátory* na testovanie vzťahov *premenných* a výrazov
- Vedieť používať logické *operátory* na kombinovanie výrazov, ktoré obsahujú relačné *operátory*
- Vedieť vytvoriť jednoduché vetvenie pomocou *podmienky*
- Vedieť vytvoriť jednoduchý *cyklus* prechádzajúci prvky *poľa*
- Vedieť vytvoriť *funkciu*, ktorá akceptuje *argumenty* a vracia *hodnotu* špecifického typu
- Vedieť zoradiť dáta v *poli*

- Vedieť vytvoriť a vyextrahovať jednoduchú konzolovú aplikáciu
- Vedieť sa pripojiť na ľubovoľnú *SQL* databázu na získavanie a ukladanie dát

Kurz od účastníkov očakáva aj základnú znalosť *OS Windows* a prácu s počítačom na úrovni bežného užívateľa.

3.5 Získané znalosti účastníkov kurzu

Sekcia zhŕňa základné znalosti, ktoré získajú účastníci technologického kurzu po absolvovaní všetkých prednášok a cvičení. Pre zistenie, aké znalosti sa preberajú v akých prednáškach viď sekcia 3.3, ktorá obsahuje anotácie k jednotlivým cvičeniam a prednáškam. Hlavnou znalosťou, ktorú účastníci získali, je tvorba *desktopových aplikácii* napojených na databázu pomocou programovacieho jazyka C#.

Pokiaľ ide o znalosti programovacieho jazyka C#, absolvent kurzu by mal vedieť popísať základnú syntax a možnosti jazyka, mal by vedieť používať:

- Kľúčové slová
- *Identifikátory*
- *Operátory*
- *P*arametre
- Riadiace štruktúry

Zo znalostí *objektovo orientovaného programovania* v jazyku C# by mal ovládať rozdiely medzi rôznymi dátovými typmi, *modifikátormi prístupu*. Mal by vedieť taktiež vytvárať *triedy*, definovať a implementovať *rozhrania*, vytvárať a používať *generické kolekcie*. Mal by vedieť rozdiely medzi jednotlivými členmi *tried* a kedy je vhodné použiť ktorý. Taktiež by mal vedieť použiť:

- *Generiká*
- *Výnimky*
- *Delegáty*
- *Lambda výrazy*
- *Tuple*

Absolvent by mal rozumieť vnútornému fungovaniu architektúry *.NET*, jednotlivým vrstvám a *CLR*. Bol oboznámený s rozdielmi medzi *.NET Standard*, *.NET Framework* a *.NET Core*, preto by mal vedieť, aké funkcionality sa kde nachádzajú. V neposlednom rade by mal vedieť *kontajnerizovať* aplikácie.

Z kurzu by sa mal absolvent naučiť základnú prácu s prostredím *Visual Studio*, aké obsahuje komponenty, skratky a ako ich používať, ale aj ako *debuggovať* aplikácie, *odchytávať výnimky* v prostredí a pracovať na tímovom projekte pomocou *Git* repozitára. Absolvent sa naučil aj rôzne modely vetvenia a prácu s technológiu *DevOps*.

Okrem základnej práce s prostredím *Visual Studio* sa absolvent učil aj o rôznych rozšíreniach prostredia. Mal by po absolvovaní vedieť pracovať s rozšírením *Resharper* a s jeho pomocou refaktorizovať kód, ale aj vedieť písať čistý, udržiavateľný a testovateľný kód, ako aj testovacie sady k nemu. Okrem iného by mal vedieť písať *asynchrónny kód* a vedieť, kedy ho použiť, ako aj návrhové vzory z *GoF*.

Absolvent kurzu by nemal mať problém s nainštalovaním a prácou s *NuGet* balíčkami a rôznymi *frameworkami*. Mal by sa vedieť prostredníctvom jazyka C# napojiť na *SQL* databázu a vedieť z nej získavať dáta, rovnako tak aj dáta pridávať, upravovať a mazať. Absolvent by nemal mať problém s vytvorením *databázovej vrstvy* aplikácie s využitím návrhových vzorov *repozitár* a *UnitOfWork*. Na prácu s databázou by mal vedieť používať okrem iného aj *Entity Framework*.

Účastník by mal po absolvovaní kurzu ovládať základne softvérové architektúry a vybrať, ktorá je pre jeho aplikáciu vhodná. Mal by vedieť vytvoriť aplikáciu postavenú na návrhovom vzore *MVVM*, vytvoriť všetky potrebné komponenty k tomu, aby daná aplikácia správne fungovala. Taktiež by mal vedieť obdobnú aplikáciu vedieť vytvoriť s použitím *frameworku MVVM Light*.

Z užívateľských rozhraní by mal absolvent vedieť pracovať s technológiou *WPF*. Mal by vedieť prepájať časti užívateľského rozhrania napísané v jazyku *XAML* s kódom napísaným v jazyku C# pomocou *data binding* a návrhového vzoru *command*, ako aj upravovať design aplikácie pomocou *frameworku Material Design*.

V neposlednom rade by mal absolvent vedieť používať návrhový vzor *mapper* a používať jeho implementáciu v podobe *NuGet* balíčku *AutoMapper*. Mal by vedieť pracovať s knižnicou *BenchmarkDotNet* na testovanie výkonu aplikácie a aj vedieť, kde hľadať rozširujúce informácie k hlavným preberaným témam technologického kurzu.

Kapitola 4

Zvolená forma technologického kurzu

Táto časť práce sa zameriava na formu technologického kurzu, analyzuje, aké typy materiálov sa v ňom použili, v akom rozsahu a z akého dôvodu (viď sekcia 4.3). Takisto sa venuje technológiám využitým na ich tvorbu (viď sekcia 4.4), použitému jazyku (viď sekcia 4.2) a dĺžke kurzu (viď sekcia 4.1).

4.1 Dĺžka a rozloženie technologického kurzu

Zo zadania je dĺžka kurzu stanovená na jeden pracovný týždeň. Každý deň začína o 8:00 hod. a končí 17:00 hod. Presné rozdelenie materiálov na určené hodiny v určitých dňoch sú dostupné v časovom harmonograme kurzu (viď sekcia 3.2).

V priebehu každého dňa sa odohráva viacero prednášok a jedno praktické cvičenie. Cvičenie je vždy v predpoludňajších hodinách tesne pred prestávkou na obed, ktorá je dlhšia a každý deň trvá minimálne hodinu. Medzi jednotlivými prednáškami a cvičeniami sa vždy nachádza kratšia, aspoň dvadsaťminútova prestávka z fyziologických dôvodov účastníkov.

Rozdelenie dňa do menších sekcií, medzi ktorými sa nachádzajú kratšie prestávky a jedna dlhšia prestávka, má aj kurz *Programming in C# 20483C* (viď odsek 2.1.1). Toto rozdelenie je vhodné aj podľa knihy *Všeobecná didaktika* [53], v ktorej autor píše o tom, že prestávka na rozdelenie medzi dopoludňajším a popoludňajším vyučovaním by mala byť väčšia, a mala by trvať aspoň štyridsať minút. Píše aj o tom, že vyučovanie by malo byť celodenné a zodpovedať prirodzenému rytmu dňa, týždňa a roku. Malo by byť rozdelené do sekcií, podľa tém, medzi ktorými by mali byť prestávky.

Prestávky medzi jednotlivými cvičeniami a prednáškami sú veľmi dôležité aj kvôli udržaniu pozornosti účastníkov. Podľa knihy *Praktické využití aktivizačních metod ve výuce* [19] je to veľký problém, hlavne pri vysvetlovaní nezáživnej teórie. Kniha ďalej hovorí o tom, že podľa niektorých výskumov vysokoškolskí študenti a študenti posledného ročníka strednej školy dokážu udržať sústredenú pozornosť asi pätnásť až dvadsať minút. Preto je dôležité študentov zapojiť do procesu výuky a aktivizovať ich.

Zvýšenie pozornosti účastníkov sa dosahuje pomocou použitia obrazových pomôcok ako sú obrázky, grafy a animácie. Školiteľ dokáže zvýšiť pozornosť účastníkov taktiež kladením otázok a krátkou diskusiou a rôznymi aktivizačnými metódami, ako je napríklad vedomostný kvíz na záver.

4.2 Jazyk technologického kurzu

Ako jazyk technologického kurzu som zvolil angličtinu. Učinil som tak preto, lebo téma kurzu, programovací jazyk C#, je viac medzinárodná ako lokálna. Aj technológie *GitHub* a *GitPitch* slúžiace na distribúciu a prezentovanie (viď sekcia 4.4) sú v anglickom jazyku. V neposlednom rade sú aj dokumentácie k použitým technológiám a väčšina zdrojov, z ktorých som čerpal, v angličtine.

Aj podľa knihy *Didaktika* [72] je v súčastnosti väčšina informácií prezentovaná v anglickom jazyku, dôkazom čoho je napríklad internet, či prehľad najznámejších a najuznávanejších svetových časopisov. Anglický jazyk sa stáva univerzálnym dorozumievacím prostriedkom medzi ľuďmi. Dôsledkom toho je aj skutočnosť, že do jednotlivých jazykov prenikajú anglické výrazy a stávajú sa akýmisi medzinárodnými pojmami.

4.3 Rozdelenie materiálov v technologickom kurze

Po preštudovaní dostupných materiálov som sa rozhodol materiály kurzu rozdeliť na dve časti. *Teoretickú* časť, ktorá sa venuje vysvetleniu určitej problematiky a *praktickú*, kde účastník má možnosť vyskúšať si danú problematiku priamo v kóde. Pri rozdeľovaní *teoretickej* časti na menšie časti som sa zameral na logické celky, zatiaľ čo *praktickú* som rozdelil na časti podľa počtu dní kurzu.

4.3.1 Inšpirácia pri rozdelení materiálov

Pri rozdeľovaní materiálov do dvoch častí som sa inšpiroval hlavne kurzom *Programming in C# 20483C* (viď odsek 2.1.1). Takmer všetky technologické kurzy mali v sebe síce teoretickú a praktickú časť, ale väčšina z nich mala obidve časti zlúčené. Konkrétne to znamenalo, že školiteľ plynule prechádzal od prezentácie ku praktickej ukážke kódu a späť. Ja som sa rozhodol tieto časti od seba rozdeliť.

Inšpirácia pri rozdelení teoretických materiálov

Inšpiráciu pri rozdelení teoretických som získal z kurzov v *Microsoft Virtual Academy* (viď sekcia 2.1.1), a takisto z kurzov tretích strán (viď sekcia 2.1.2). Správny počet dobre zvolených kurzov, ktoré sa zameriavajú na konkrétnu časť programovacieho jazyka, môže tvoriť celú teoretickú časť, pričom jeden kurz sa bral ako prezentácia určitej témy. Inšpiroval som sa tiež štandardným rozdelením kníh z *Microsoft press* (viď sekcia 2.1.1), alebo z kníh tretích strán (viď sekcia 2.1.2) na kapitoly, ktoré v tomto technologickom kurze predstavujú prezentáciu určitých tém. Práve preto je teoretická časť rozdelená podľa tematických celkov.

Moje rozhodnutie potvrdzuje aj Erich Petlák, ktorý v knihe *Všeobecná didaktika* [53] píše o tom, že vyučovanie by malo by byť rozdelené do sekcií podľa tém.

Inšpirácia pri rozdelení praktických materiálov

Inšpiráciu pri rozdelení praktických materiálov som získal z kurzu *Programming in C# 20483C* (viď odsek 2.1.1), ktorý mi dodal inšpiráciu túto časť vôbec vytvoriť. Pri rozdeľovaní materiálov mi pomohol aj časový harmonogram kurzu (viď sekcia 3.2). Praktická časť obsahuje päť cvičení, pričom každý deň je jedno v doobedňajších hodinách. Témy do jednotlivých cvičení som rozdelil tak, aby vždy zahŕňali prax z teórie preberanej na prednáškach od predchádzajúceho cvičenia.

Cvičenie každý deň som navrhol aj z dôvodu, že je to podľa knihy *Praktické využití aktivizačních metod ve výuce* [19] spôsob, ako opäť získať pozornosť účastníkov kurzu, ktorých pozornosť počas priebehu dňa klesá.

4.3.2 Forma teoretických materiálov

Teoretické materiály sú rozdelené podľa tematických celkov do prezentácií, pomocou služieb *Gitpitch* a *GitHub* (viac viď sekcia 4.4). Teoretickej časti v kurze je časovo venovaný približne dvakrát taký čas ako praktickým cvičeniam, viď sekcia časový harmonogram 3.2. Osnova prednášok sa nachádza v sekcii 3.1.1.

Prezentáciu, ako formu teoretických materiálov, som zvolil z dôvodu, že bola použitá takmer vo všetkých kurzoch z vykonaného prieskumu. Prezentovanie je efektívna metóda, ako užívateľovi zobraziť podstatné body z preberanej látky a podrobnosti vysvetliť ústnou formou. Vybraná technológia slúžiaca na distribúciu a vytváranie prezentácií sprístupňuje prezentácie užívateľom v online aj offline podobe, a to na väčšine dnes používaných operačných systémov. Materiály prezentácií sú k dispozícii aj vo forme *Markdown* súborov.

V kurze očakávam, že prezentácie budú prednesené účastníkom od lektora vo forme prednášok. Podľa knihy *Výukové metody* [27] sa s prednáškami stretávame najmä u vysokoškolských študentov a dospelých. Prednáška sa na rozdiel od vysvetľovania vyznačuje dlhším, uceleným prejavom sprostredkovávajúcim závažnú tému skupine zainteresovaných poslucháčov. Prednáška sprostredkováva informácie rýchlo a aktuálne a aby zaujala, musí prednášajúci zvládnuť techniku rečového prejavu.

Osoba prednášajúca materiály by mala byť dobrým rečníkom, toho dôležitým predpokladom je nesedieť, ale primerane sa pohybovať a zrakom účastníkov sledovať. Je taktiež nutné zvoliť primeranú silu hlasu, meniť tempo a melódiu reči a využívať všetky rečové prostriedky a rečnícke postupy vrátane reči tela.

4.3.3 Forma praktických materiálov

Rozdelenie praktických materiálov je vytvorené podľa harmonogramu kurzu (viď sekcia 3.2), pričom na každý deň pripadá jedno praktické cvičenie, zahrňujúce prax na všetku prebratú teóriu od prechádzajúceho cvičenia. Praktických materiálov v kurze je približne dvakrát menej ako teoretických. Každý praktický materiál obsahuje dve *C# solution*. Prvé obsahuje čiastočný kód, ktorý majú študenti v priebehu cvičenia vyplniť a druhé obsahuje konečné, referenčné riešenie, ktoré môže slúžiť na kontrolu správnosti. Materiály boli vytvorené pomocou nástroja *Visual Studio*.

C# solution považujem za najlepší dostupný spôsob, ako vytvoriť praktický materiál ku kurzu. Čiastočne som sa inšpiroval kurzom Programming in C# 20483C (viď odsek 2.1.1), v ktorom sa tiež používajú praktické materiály. Naviac sa v ňom využívajú zvirtualizované *OS Windows*, s pripraveným a nakonfigurovaným prostredím *Visual Studio* a inými prerekvizitami. Túto časť som ale do vytváraného kurzu nezahrnul, napriek tomu, že je podľa môjho názoru vhodná, z dôvodu vysokej technickej náročnosti. Nebolo by ani také náročné vytvoriť zvirtualizované *OS Windows* s prostredím *Visual Studio*, zložitejšia by bola konfigurácia, správa materiálov a prerekvizít, ako aj distribúcia medzi účastníkov. Každopádne je to možnosť, ako v budúcnosti kurz dostať na vyššiu technickú úroveň.

Cvičenia sú vedené školiteľom. Na začiatku každého cvičenia školiteľ vysvetlí zadanie a postupne účastníkom pomáha s jeho riešením, avšak v kontraste s prednáškami, školiteľova úloha časom upadá. Podla knihy *Výukové metody* [27] sa učiteľova pomoc žiakovi postupne zoslabuje, až dosiahne úplnej samostatnosti žiaka, ktorá je konečným cieľom všetkej edu-

kácie. Žiakové funkcie vo výuke sa tiež menia smerom k väčšej otvorenosti k technickým inováciám, ktoré žiak prijíma so samozrejmosťou a ústretovosťou.

4.4 Technológie využité na tvorbu a distribúciu materiálov

Materiály boli vytvárané na *OS Windows* pomocou technológie *Git* a služieb *GitHub*, *Gitpitch*. Pri vytváraní bol využitý aj webový prehliadač *Google Chrome* a *Visual Studio IDE*. Na vytvorenie materiálov bol použitý prevažne programovací jazyk C# a značkovací jazyk *Markdown*.

Všetky teoretické, ako aj praktické materiály[1], potrebné k tomuto kurzu sú k dispozícii v službe *GitHub*. Materiály sú voľne dostupné pod *MIT* licenciou, vďaka ktorej si ich môže stiahnuť a upravovať každý. Účastníci môžu meniť aj samotný obsah repozitáru, vďaka technológii *pull request*, ktorú *Git* poskytuje.

GitPitch

GitPitch je služba, ktorú som sa rozhodol použiť na distribúciu a vytváranie prezentácií. S daným rozhodnutím mi pomohol prieskum prezentačných technológií viď sekcia 2.2. Pomocou technológie *Markdown* sú v *GitHub* repozitári uložené zdrojové súbory s informáciami, ktoré následne služba *GitPitch* prevádza na samotné prezentácie. Vďaka tomu sú informácie uložené ako vo forme formátovaného textu v jazyku *Markdown*, tak aj vo forme prezentácií v službe *GitPitch*.

Základom služby *GitPitch* je prezentačný *framework revealJS*, ktorý dáva službe veľké množstvo výhod. Oproti službám pre širokú verejnosť, ako sú *Google Slides*, *PowerPoint* alebo *Keynote*, je určený pre vývojárov. Pridávať ukážky kódov do týchto služieb je obecne zdĺhavé a nepríjemné. Prezentačné nástroje založené na *revealJS* prichádzajú s tradíciou *WYSIWYG*. To je veľmi užitočné, pretože to uľahčuje vloženie čohokoľvek na slajd. Ale môže to byť aj veľmi bolestivé, pretože môže trvať dlhý čas dostať pixely na správne miesto, namiesto sústredenia sa na obsah.

Ďalšou nepríjemnosťou služieb určených pre širokú verejnosť je formát ich súborov. Binárne súbory nie sú vhodné na použitie vo verzných systémoch ako je *Git*, ktorý je v kurze používaný. Väčšina z týchto služieb síce umožňuje spoluprácu na úpravách prezentácií, no táto funkcia nie je kompatibilná s verznými systémami. Využitie verzného systému v kurze umožňuje uložiť zdrojové súbory prezentácií aj cvičení na jedno miesto a to do *Git* repozitára.

Vďaka uloženiu *Markdown* súborov vo verznom systéme nie je potrebné využívať ďalšiu službu na ukladanie a zdieľanie prezentácií, ako napríklad *SlideShare*, *Speaker Deck* alebo nejakú formu cloud úložiska. Zdrojové súbory prezentácií sú uložené vo verznom systéme a samotná prezentácia je generovaná z nich a uložená na serveroch služby *GitPitch*.

Služba *GitPitch* oproti ostatným službám založeným na *revealJS* alebo inej technológii, ktorá používa ako zdroj *Markdown* obsahuje omnoho viac funkcionalít. Najdôležitejšou z nich je možnosť vkladania zdrojových súborov z *Git* repozitára priamo do prezentácie pomocou vloženia cesty k súboru. Vďaka tomu je v prezentácii vždy najnovšia verzia súboru bez potreby úprav zdrojového kódu prezentácie pri každej zmene v danom súbore.

[1] https://github.com/orlicekm/CsharpCourse

Visual Studio

Ako vývojové prostredie pre tvorbu materiálov v jazyku C# som vybral *Visual Studio*. Boli ním vytvárané *C# solution* pre cvičenia, ako aj ukážky kódov k prezentáciám a samotné súbory prezentácií. Spočiatku som na vytváranie materiálov používal *Visual Studio 2017* a následne po vydaní *Visual Studio 2019*.

S rozhodnutím používať *Visual Studio* mi pomohol prieskum vývojových prostredí viď sekcia 2.3. Kedže kurz obsahuje časti, ktoré sú relevantné iba pre operačný systém *Windows*, ako je napríklad *.NET Framework*, možnosť pracovať na inej platforme mi neprišla pri výbere prostredia dôležitá. Preto som sa primárne sústredil na prostredia pre tento *OS*. Pokiaľ ide o množstvo funkcionalít, *Visual Studio* ich poskytuje najväčšie množstvo oproti ostatným prostrediam. *scriptcs* nie je ani prostredie, ale iba balíček na spúšťanie kódu a aj *Visual Studio Code* je editor zdrojového kódu, nie plnohodnotné *IDE*.

Prostredie *JetBrains Rider* obsahuje funkcionality, ktoré by boli pri vytváranií materiálov užitočné a samotné *Visual Studio* ich neobsahuje. Každopádne, *JetBrains Rider* tieto funkcionality získalo z rozšírenia *Resharper*, ktoré v ňom je v ňom priamo vstavané. *Resharper* je rozšírenie pre prostredie *Visual Studio*, ktoré som doň doinštaloval a používal. Vďaka tomu som získal funkcionality *JetBrains Rider* v prostredí *Visual Studio*, pre ktoré som sa aj vďaka tomu rozhodol.

Na prístup ku *GitHub* repozitáru s kurzom a na jeho následnú úpravu bol použitý *Visual Studio Team Explorer* klient, umožňujúci komunikáciu s *Git* repozitárom. Z dostupných doplnkov bol použitý spomínaný *Resharper* a *Markdown Editor*[2]. Vďaka *Markdown Editoru* som mohol vytvárať prezentácie a aj *README* súbory priamo v prostredí *Visual Studio*, ktoré mi zobrazovalo aj výsledné formátovanie zo zdrojového kódu.

[2] https://marketplace.visualstudio.com/items?itemName=MadsKristensen.MarkdownEditor

Kapitola 5

Záver

Cieľom práce bolo vytvoriť technologický kurz v programovacom jazyku C#. Kurz sa zameriava na využívanie dobrých programátorských praktík a na moderné technológie, umožňujúce jeho jednoduchú úpravu a šírenie. Hlavným zámerom kurzu bolo naučiť účastníkov vytvárať aplikácie s užívateľským rozhraním napojené na databázu pomocou programovacieho jazyka C#. Kurz sa ďalej cielil predovšetkým na tvorbu *desktopových aplikácii* pomocou návrhového vzoru *Model–View–ViewModel* a užívateľského rozhrania *Windows Presentation Foundation*.

Celkovo som pri vytváraní kurzu uskutočnil tri prieskumy. Prvý z nich bol prieskum kurzov a kníh (viď sekcia 2.1), ktoré sa venujú programovaciemu jazyku C#. Prieskum som rozdelil na oficiálnu a neoficiálnu časť, pričom som sa snažil nájsť čo najvhodnejšie a najrelevantnejšie zdroje a druhy materiálov k vytváranému kurzu. Okrem toho som uskutočnil prieskum prezentačných technológií (viď sekcia 2.2), z ktorého som vybral ako najvhodnejší variant pre teoretické prezentácie technológiu *GitPitch*. Pre túto technológiu som sa rozhodol pre veľké množstvo funkcionalít, z ktorých najužitočnejšou je možnosť vkladania zdrojových súborov z *Git* repozitára priamo do prezentácie pomocou vloženia cesty k súboru. Posledným bol prieskum vývojových prostredí (viď sekcia 2.3), z ktorého som sa rozhodol pre prostredie *Visual Studio* s použitím doplnkov *Resharper* a *Markdown* editor. *Visual Studio* poskytuje z preskúmaných prostredí najväčšie množstvo funkcionalít a vďaka doplnku *Resharper* získa aj funkcionality z prostredia *JetBrains Rider*.

Za ten najlepší nájdený kurz považujem oficiálny kurz od firmy *Microsoft Programming in C# 20483C* (viď odsek 2.1.1). Jeho mínusom je však spoplatnený prístup. V porovnaní s ostatnými kurzami je na omnoho vyššej technickej úrovni, a to vďaka používaniu virtuálnych operačných systémov na praktických cvičeniach. Väčšina ostatných kurzov praktickú časť ani nemá, alebo je spojená s teoretickou časťou. Najužitočnejšiou knihou z prieskumu je podľa môjho názoru *C# 7.0 in a nutshell* (viď odsek 2.1.2), napriek tomu, že nie je od spoločnosti *Microsoft*. A práve túto knihu som použil ako hlavný zdroj pri tvorbe teoretických materiálov, pretože poskytuje široký rozsah znalostí.

Kurz celkovo obsahuje osemnásť študijných materiálov, trinásť teoretických prednášok a päť praktických cvičení, na ktoré boli použité moderné technológie *GitHub*, *GitPitch*, *Markdown*, *Visual Studio* a ďalšie viď sekcia 4.4. Ku kurzu boli okrem základných študijných materiálov vytvorené aj ďalšie, doplňujúce informácie. Konkrétne ide o osnovu kurzu, časový harmonogram kurzu, požadované prerekvizity od budúcich účastníkov kurzu, vedomosti, ktoré kurz absolventov naučil, ale aj anotácie ku všetkým študijným materiálom (viď kapitola 3). Takisto sa určila dĺžka kurzu, jazyk kurzu a ďalšie potrebné náležitosti, aby kurz mohol vôbec vzniknúť.

Pri vytváraní práce som sa stretol s viacerými problémami. Najväčším z nich bolo získavanie informácií z platených zdrojov ako aj kurzov, ktoré boli dostupné iba osobne. Namiesto nich som si vyhľadal obdobné varianty s voľným prístupom a použil tie. Menšie problémy mi spôsobila aj technológia *GitPitch*, ktorá je stále vo vývoji. Stalo sa mi, že jej funkcionalita na zvýrazňovanie určitých riadkov kódu v prezentácii prestala správne fungovať. Všetok kód bol zobrazený ako nezvýraznený a kvôli tomu nebol skoro vôbec čitateľný. Po ubezpečení, že problém nie je na mojej strane, som kontaktoval autora technológie, ktorý daný problém do pár hodín vyriešil.

Na záver by som doplnil, že nadväzovať na prácu by sa dalo vytvorením ďalších kurzov, ktoré by sa venovali technológiám, ktoré sa v tomto kurze nestihli prebrať. Napríklad tvorbe webových aplikácií alebo rôznych služieb pomocou jazyka C#, poprípade vytvorením podobných technologických kurzov pre iné programovacie jazyky.

Literatúra

[1] Albahari, J.: *C# 7.0 in a nutshell*. O'Reilly, siedme vydanie, 2018, ISBN 978-1-4919-8765-0.

[2] Apple Incorporation: *Keynote*. [Online; navštívené 27.4.2019].
URL https://www.apple.com/lae/keynote/

[3] CBT Nuggets: *CBT Nuggets*. [Online; navštívené 1.5.2019].
URL https://www.cbtnuggets.com/

[4] Delsink, A.: *Object Oriented Programming with C#*. [Online; navštívené 5.1.2019].
URL https://www.lynda.com/C-tutorials/Object-Oriented-Programming-C/765320-2.html

[5] Fewer and Faster LLC: *Speaker Deck*. [Online; navštívené 28.4.2019].
URL https://speakerdeck.com/

[6] Fowler, M.: *Refactoring*. Addison-Wesley, druhé vydanie, 2018, ISBN 978-1-5093-0698-5.

[7] Gabriel, A.: *C# Basics for Beginners: Learn C# Fundamentals by Coding*. [Online; navštívené 20.1.2019].
URL https://www.udemy.com/csharp-tutorial-for-beginners/

[8] Gamma, E.: *Design patterns*. Addison-Wesley, 1995, ISBN 978-0-2016-3361-0.

[9] Google: *Google Slides*. [Online; navštívené 28.4.2019].
URL https://www.google.com/slides/about/

[10] GOPAS SR, a.s: GOPAS SR. [Online; navštívené 16.4.2019].
URL https://www.gopas.sk/Novinky-sk.aspx

[11] Hakim El Hattab and contributors: *Reveal.JS*. [Online; navštívené 25.4.2019].
URL https://revealjs.com/#/

[12] Harrison, C.: *Introduction to JSON with C#*. [Online; navštívené 2.1.2019].
URL https://mva.microsoft.com/en-US/training-courses/introduction-to-json-with-c-12742

[13] Hattori, Y.: *Marp*. [Online; navštívené 24.4.2019].
URL https://yhatt.github.io/marp/

[14] Herceg, T.; Jecha, T.: *dotNETcollege*. [Online; navštívené 1.5.2019].
URL https://www.dotnetcollege.cz/

[15] Herceg, T.; Jecha, T.: *dotNETportal.cz*. [Online; navštívené 1.5.2019].
URL https://www.dotnetportal.cz/

[16] IC#Code: SharpDevelop.
URL http://www.icsharpcode.net/OpenSource/SD/Default.aspx

[17] JetBrains s.r.o.: JetBrains Rider. [Online; navštívené 5.5.2019].
URL https://www.jetbrains.com/rider/

[18] JetBrains s.r.o.: *Resharper*. [Online; navštívené 13.1.2019].
URL https://www.jetbrains.com/resharper/

[19] Kotrba, T.; Lacina, L.: *Praktické využití aktivizačních metod ve výuce*. Barrister & Principal, 2007, ISBN 978-80-87029-12-1.

[20] Learn2Code: Learn2Code. [Online; navštívené 17.4.2019].
URL https://www.learn2code.sk

[21] LinkedIn Corporation: *Lynda.com*. [Online; navštívené 4.1.2019].
URL https://www.lynda.com/

[22] LinkedIn Corporation: *SlideShare*. [Online; navštívené 28.4.2019].
URL https://www.slideshare.net/

[23] MacFarlane, J.: *Pandoc*. [Online; navštívené 24.4.2019].
URL https://pandoc.org/

[24] Martin, R. C.: *Clean code*. Prentice Hall, 2009, ISBN 978-0-1323-5088-4.

[25] Martin, R. C.; Martin, M.: *Agile principles, patterns, and practices in C#*. Prentice Hall, 2007, ISBN 978-0-1318-5725-4.

[26] May, D.; Morgan, G.: *Windows 10: Data Binding*. [Online; navštívené 2.1.2019].
URL https://mva.microsoft.com/en-US/training-courses/windows-10-data-binding-14579

[27] Maňák, J.; Švec, V.: *Výukové metody*. Paido, 2003, ISBN 80-7315-039-5.

[28] Microsoft Corporation: *C# Guide — Microsoft Docs*. [Online; navštívené 12.1.2019].
URL https://docs.microsoft.com/en-us/dotnet/csharp/

[29] Microsoft Corporation: *Course Programming in C# 20483C*. [Online; navštívené 29.12.2018].
URL https://www.microsoft.com/en-us/learning/course.aspx?cid=20483

[30] Microsoft Corporation: *Exam Programming in C# 70–483*. [Online; navštívené 28.12.2018].
URL https://www.microsoft.com/en-us/learning/exam-70-483.aspx

[31] Microsoft Corporation: *GitHub — Microsoft Docs*. [Online; navštívené 1.5.2019].
URL https://github.com/MicrosoftDocs

[32] Microsoft Corporation: *Microsoft Docs*. [Online; navštívené 31.12.2018].
URL https://docs.microsoft.com/en-us/

[33] Microsoft Corporation: *Microsoft IT Training Courses*. [Online; navštívené 28.12.2018].
URL https://www.microsoft.com/en-us/learning/course-list.aspx

[34] Microsoft Corporation: *Microsoft Learn*. [Online; navštívené 31.12.2018].
URL https://docs.microsoft.com/en-us/learn/

[35] Microsoft Corporation: *Microsoft Learning Partners*. [Online; navštívené 5.1.2019].
URL https://www.microsoft.com/en-us/learning/partners.aspx

[36] Microsoft Corporation: *Microsoft Official Courses On-Demand*. [Online; navštívené 5.1.2019].
URL
https://www.microsoft.com/en-us/learning/on-demand-online-courses.aspx

[37] Microsoft Corporation: *Microsoft PowerPoint*. [Online; navštívené 27.4.2019].
URL https://products.office.com/en/powerpoint

[38] Microsoft Corporation: *Microsoft Technical Certifications*. [Online; navštívené 28.12.2018].
URL
https://www.microsoft.com/en-us/learning/certification-overview.aspx

[39] Microsoft Corporation: *Microsoft Virtual Academy*. [Online; navštívené 28.12.2018].
URL https://mva.microsoft.com/

[40] Microsoft Corporation: *.NET Programming Languages — Microsoft Docs*. [Online; navštívené 4.22.2019].
URL https://dotnet.microsoft.com/languages

[41] Microsoft Corporation: Visual Studio Code. [Online; navštívené 5.5.2019].
URL https://code.visualstudio.com/

[42] Microsoft Corporation: *Visual Studio*. [Online; navštívené 6.1.2019].
URL https://visualstudio.microsoft.com/cs/vs/

[43] Microsoft Corporation: *Visual Studio for Mac*. [Online; navštívené 4.5.2019].
URL https://visualstudio.microsoft.com/vs/mac/

[44] Microsoft Corporation and Pearson plc: *Microsoft Press*. [Online; navštívené 1.1.2019].
URL https://www.microsoftpressstore.com/

[45] Miles, R. S.: *Begin to Code with C#*. Microsoft Press, 2016, ISBN 978-1-5093-0115-7.

[46] Miles, R. S.: *Exam ref 70–483, Programming in C#*. Microsoft Press, druhé vydanie, 2018, ISBN 978-1-5093-0698-5.

[47] Millsap, R.: *Using Generics in C#*. [Online; navštívené 5.1.2019].
URL https://www.lynda.com/C-tutorials/Using-Generics-C/731737-2.html

[48] Mono community: *MonoDevelop*. [Online; navštívené 4.5.2019].
URL https://www.monodevelop.com/

[49] Nixon, J.; May, D.: *Programming in C# Jump Start*. [Online; navštívené 2.1.2019].
URL https://mva.microsoft.com/en-us/training-courses/programming-in-c-jump-start-1425

[50] One Tap Limited: *GitPitch*. [Online; navštívené 11.1.2019].
URL https://gitpitch.com/

[51] OpenDoc Society: *OpenDocument Format*. [Online; navštívené 28.4.2019].
URL http://opendocumentformat.org

[52] Osherove, R.: *The art of unit testing with examples in C#*. Manning, druhé vydanie, 1995, ISBN 978-1-6172-9089-3.

[53] Petlák, E.: *Všeobecná didaktika*. Iris, 1997, ISBN 80-88778-49-2.

[54] Pluralsight LLC: *Pluralsight*. [Online; navštívené 30.4.2019].
URL https://www.pluralsight.com

[55] Pluralsight LLC: *Pluralsight — C#*. [Online; navštívené 1.5.2019].
URL https://app.pluralsight.com/paths/skills/csharp

[56] Sandstorm contributors: *Sandstorm*. [Online; navštívené 25.4.2019].
URL https://sandstorm.io/

[57] scriptcs community: scriptcs. [Online; navštívené 5.5.2019].
URL http://scriptcs.net/

[58] Seymour, J.; Nixon, J.: *Microsoft Certification PREP Talk: Exam 483*. [Online; navštívené 6.1.2019].
URL https://www.microsoft.com/en-us/videoplayer/embed/382796de-e68a-4f14-9b0b-fa9d2a626ed8?pid=56c39b0b-8cd3-47c3-889d-4c1f10758b72-innerdiv-oneplayer

[59] SharePresentation: *SharePresentation*. [Online; navštívené 30.4.2019].
URL https://www.sharepresentation.com/

[60] Sharp, J.: *Microsoft Visual C# Step by Step*. Microsoft Press, deviate vydanie, 2018, ISBN 978-1-5093-0776-0.

[61] Singleton, J.: *Hacker Slides*. [Online; navštívené 25.4.2019].
URL https://github.com/jacksingleton/hacker-slides

[62] Skeet, J.: *C# in depth*. Manning, tretie vydanie, 2014, ISBN 978-1-6172-9134-0.

[63] Slater, F.: *Complete C# Unity Developer 2D: Learn to Code Making Games*. [Online; navštívené 20.1.2019].
URL https://www.udemy.com/unitycourse/

[64] Slides Inc.: *Slides*. [Online; navštívené 28.4.2019].
URL https://slides.com/

[65] Tabor, B.: *C# Fundamentals for Absolute Beginners*. [Online; navštívené 5.1.2019].
URL https://mva.microsoft.com/en-US/training-courses/c-fundamentals-for-absolute-beginners-16169

[66] Inštitút vzdelávania informačných technológií, s.: Inštitút vzdelávania informačných technológií. [Online; navštívené 1.2.2019].
URL http://www.ivit.sk/

[67] Inštitút vzdelávania informačných technológií, s.: Kurzy-it.sk. [Online; navštívené 1.2.2019].
URL https://www.kurzy-it.sk/

[68] Inštitút vzdelávania informačných technológií, s.: Počítačový kurz C# — pokročilý. [Online; navštívené 1.2.2019].
URL https://www.kurzy-it.sk/c-sharp-pokrocily

[69] Inštitút vzdelávania informačných technológií, s.: Počítačový kurz C# — základy. [Online; navštívené 1.2.2019].
URL https://www.kurzy-it.sk/c-sharp-zaklady

[70] The Apache Software Foundation: *OpenOffice.org*. [Online; navštívené 30.4.2019].
URL https://www.openoffice.org/

[71] The Document Foundation: *LibreOffice*. [Online; navštívené 30.4.2019].
URL https://www.libreoffice.org/

[72] Turek, I.: *Didaktika*. Wolters Kluwer, 2014, ISBN 978-80-8168-004-5.

[73] Udemy Incorporation: *Udemy*. [Online; navštívené 20.1.2019].
URL https://www.udemy.com/

[74] Unsigned Integer UG: *Deckset*. [Online; navštívené 26.4.2019].
URL https://www.deckset.com/

[75] Vzorek, M.: Úvod do programovania v jazyku C#. [Online; navštívené 17.4.2019].
URL
https://www.learn2code.sk/kurzy/uvod-do-programovania-v-jazyku-csharp

[76] WUG: *Windows User Group*. [Online; navštívené 30.4.2019].
URL https://www.wug.cz/

[77] Xamarin: *Xamarin Studio 6.3*. [Online; navštívené 4.5.2019].
URL https://developer.xamarin.com/releases/studio/xamarin.studio_6.3/xamarin.studio_6.3/

[78] Zapletal, A.: *Landslide*. [Online; navštívené 25.4.2019].
URL https://github.com/adamzap/landslide

[79] Čápka, D.: ITnetwork.cz. [Online; navštívené 30.3.2019].
URL https://www.itnetwork.cz

Slovník

.NET je zastrešujúci názov pre súbor technológií v softvérových produktoch, ktoré tvoria celú platformu. Základným komponentom je *Microsoft .NET Framework* [81]. Str. 5, 8, 13, 14, 23, 25, 26, 27, 31, 34, 48, 50

.NET Core je slobodný softvérový *framework*, s voľne prístupným kódom [79]. Str. 8, 23, 25, 26, 31, 34

.NET Framework je softvérový *framework*, vyvinutý spoločnosťou *Microsoft*, ktorý beží primárne na *OS Windows*, obsahuje veľkú knižnicu tried, pomenovanú Framework Class Library [80]. Str. 7, 13, 23, 31, 34, 40, 48, 52, 55

.NET Standard je formálna špecifikácia *.NET API*, ktorá je určená na použitie vo všetkých implementáciách *.NET* [18]. Str. 25, 26, 28, 31, 34

Abstract Factory je návrhový vzor, ktorý zaisťuje rozhranie pre vytváranie rodín súvisiacich alebo závislých objektov, bez ich konkrétnej špecifikácie [4]. Str. 29

abstraktná trieda je *trieda*, ktorá je určená na to, aby bola rodičovskou triedou iných tried [10]. Str. 27

Ace Editor je samostatný editor kódu napísaný v jazyku *JavaScript* [1]. Str. 19

Adapter je návrhový vzor, ktorý prevádza rozhranie *triedy* na iné, akceptované klientom [4]. Str. 29

ADO.NET je technológia prístupu k dátam z *.NET Frameworku*, ktorá poskytuje komunikáciu medzi relačnými a nerelačnými systémami [29]. Str. 28

akceptačný test je testovanie, ktoré sa uskutočňuje, aby sa zistilo, či sú splnené požiadavky špecifikácie alebo zmluvy [27]. Str. 28

Apache Licence je názov slobodnej softvérovej licencie. Licencia vyžaduje od užívateľa zachovanie autorstva a tzv. Disclaimer, teda vzdať sa zodpovednosti. Licencia umožňuje užívateľovi slobodné používanie softvéru na rôzne účely; distribúciu, úpravy, následnú redistribúciu upravenej verzie [30]. Str. 17, 24

ASP.NET je *open source* webový *framework*, slúžiaci na vytváranie moderných webových aplikácií a služieb pomocou *.NET* [11]. Str. 8, 11, 22, 23

asynchrónne programovanie je programovanie s využívaním udalostí nezávislých od hlavného toku programu [32]. Str. 6, 25, 26, 31, 34

Blend for Visual Studio je prostredie na návrh a vytváranie UI, vyvinutý a predávaný spoločnosťou *Microsoft* [72]. Str. 30

Bridge je návrhový vzor, ktorý oddeľuje abstrakciu a implementáciu tak, aby sa obidve mohli od seba nezávisle líšiť [4]. Str. 29

Builder je návrhový vzor, ktorý oddeľuje vytváranie komplexného objektu od jeho reprezentácie [4]. Str. 29

business logic je časť programu, ktorá zakóduje pravidlá reálneho sveta, ktoré určujú, ako môžu byť údaje vytvorené, uložené a zmenené [33]. Str. 49

C je jeden z najpopulárnejších programovacích jazykov, ktorý stojí za založením *rodiny jazykov C* [40]. Str. 7, 11, 33, 53

C# solution je kontajner pre jeden alebo viac podobných C# projektov [21]. Str. 25, 31, 32, 38, 39

C++ je *objektovo orientovaný* programovací jazyk z *rodiny jazykov C*, ktorý poskytuje možnosti nízkoúrovňovej práce s pamäťou [41]. Str. 7, 11, 15, 33

Chain of Responsibility je návrhový vzor, ktorý bráni zahlteniu prijímateľa vďaka viac ako jednému objektu, ktorý môže spracovať žiadosť. Posiela požiadavku naprieč reťazcom objektov, pokiaľ ho objekt neobslúži [4]. Str. 29

Chocolatey je správca balíčkov pre systém *Windows* [2]. Str. 24

classroom course je kurz, pri ktorom sa študenti učia od inštruktora v triede určenej na vzdelávanie [35, 60]. Str. 7

Clean Code je kód, ktorý je formátovaný tak, aby ho iný programátor mohol jednoducho pochopiť a upraviť [112]. Str. 3, 29, 32

Command je návrhový vzor, ktorý zapúzdruje požiadavku do objektu [4]. Str. 29, 30, 33, 35

Composite je návrhový vzor, ktorý skladá objekty do stromových štruktúr [4]. Str. 29

cyklus je riadiaca štruktúra programu, kde sa opakovane vykonáva postupnosť príkazov, opakovanie a ukončenie cyklu je riadené *podmienkou* [39]. Str. 6, 7, 9, 11, 27, 33

Dapper knižnica *NuGet*, ktorá rozširuje IDbConnection rozhranie [113]. Str. 28

data binding je proces, ktorý vytvára spojenie medzi používateľským rozhraním aplikácie a *business logic* [42]. Str. 14, 30, 33, 35

debugging je metodický postup pre nachádzanie a znižovanie množstva chýb v počítačových programoch [65]. Str. 21, 22, 23, 32, 34

Decorator je návrhový vzor, ktorý umožnuje pridávanie ďalších povinností k objektu dynamicky [4]. Str. 29

dedičnosť je mechanizmus založenia objektu alebo triedy na inom objekte alebo triede [59]. Str. 9

dekonštruktor je spôsob, ako rozložiť *objekt* späť do jeho častí [17]. Str. 27

delegát odkazuje na hodnotenie člena (*property* alebo *metóda*) jedného objektu (prijímateľa) v kontexte iného pôvodného objektu (odosielateľa) [44]. Str. 27, 34

desktopová aplikácia je aplikácia v desktopovom prostredí [45]. Str. 3, 13, 16, 17, 18, 24, 32, 33, 34

Docker je počítačový program, ktorý vykonáva *kontajnerizáciu* [46]. Str. 19, 31

Entity Framework je objektovo relačný *framework*, ktorý umožňuje vývojárom *.NET* pracovať s databázou pomocou *.NET objekt*ov [14]. Str. 3, 8, 13, 25, 26, 28, 29, 30, 32, 34

enum je dátový typ pozostávajúci z množín pomenovaných hodnôt [47]. Str. 27

event je akcia alebo udalosť, rozpoznávaná softvérom, ktorá často pochádza asynchrónne z externého prostredia, ktoré môže byť ovládané softvérom [48]. Str. 6, 27

Facade je návrhový vzor, ktorý vytvára jednotné rozhranie pre súbor rozhraní v subsystéme. Definuje rozhranie vyššej úrovne, ktoré uľahčuje používanie subsystému [4]. Str. 25, 26, 29, 30

Factory Method je návrhový vzor, ktorý definuje rozhranie pre vytváranie objektu, pričom necháva podtriedy rozhodnúť, ktorá trieda sa má konkretizovať [4]. Str. 29

field v *OOP* sú dáta, zapúzdrené v rámci triedy alebo objektu [49]. Str. 27, 53

finalizer je špeciálny typ podprogramu, volaný na vykonanie akéhokoľvek potrebného konečného vyčistenia, keď inštanciu triedy ruší *garbage collector* [15]. Str. 27

Flyweight je návrhový vzor, ktorý umožňuje podporu veľkého množstva podobných objektov [4]. Str. 29

framework je abstrakcia, v ktorej je softvér poskytujúci všeobecnú funkčnosť selektívne zmenený dodatočným užívateľom napísaným kódom, vďaka čomu poskytuje softvér špecifický pre aplikáciu [99]. Str. 3, 8, 12, 15, 18, 19, 22, 28, 30, 34, 35, 39, 48, 50, 54

funkcia je časť kódu vnútri väčšieho programu, ktorá vykonáva špecifickú úlohu [90]. Str. 33, 52

funkcionálne programovanie je programovacia paradigma, založená na zápise programu v tvare výrazu [50]. Str. 33

garbage collector je spôsob automatickej správy pamäte, vyhľadáva a uvoľňuje úseky, ktoré už program nepoužíva [51]. Str. 9, 31, 50

generické programovanie je štýl programovania, v ktorom sú algoritmy napísané z hľadiska typov, ktoré majú byť špecifikované neskôr a sú pri použití inštancované pre špecifické typy poskytnuté ako parametre [52]. Str. 9, 10, 11, 27, 34

Git je systém na kontrolu verzií, ktoré umožňujú sledovanie zmien v súboroch a koordináciu prác na týchto súboroch medzi viacerými ľuďmi [54]. Str. 11, 18, 21, 23, 29, 32, 34, 39, 40, 41, 50, 51, 53

Git commit je príkaz, používaný na uloženie zmien v systéme *Git* [6]. Str. 19

GitFlow je model vetvenia pre systém *Git* [5]. Str. 29, 32

GitHub je webová hostingová služba[1] pre verzný systém *Git*, ktorá bola v nedávno odkúpená spoločnosťou *Microsoft*. Služba sa používa predovšetkým na zdieľanie počítačového kódu a umožňuje vytvorenie verejného repozitára zadarmo, vďaka čomu je vhodná na väčšinu *open source* projektov [53]. Str. 4, 8, 14, 18, 22, 24, 36, 38, 39, 40

Haskell je štandardizovaný funkcionálny programovací jazyk s voľnou sémantikou (lenivým vyhodnocovaním) pomenovaný po logikovi Haskellovi Currym [58]. Str. 18

herný engine je sada znovu použiteľných softvérových komponentov, ktoré sústreďujú všeobecné funkcie používané v počítačových hrách [55]. Str. 55

hodnota je reprezentácia nejakej entity programu, s ktorou môže program pracovať [56]. Str. 9, 33, 53

identifikátor je lexikálny token, ktorý pomenováva entity [57]. Str. 27, 34

integračný test je test, v ktorom sú jednotlivé softvérové moduly skombinované a testované ako skupina [61]. Str. 28

Interpreter je návrhový vzor, ktorý v danom jazyku definuje reprezentáciu gramatiky spolu s tlmočníkom [4]. Str. 29

Iterator je návrhový vzor, ktorý poskytuje sekvenčný prístup k elementom agregovaného objektu, bez odhaľovania jeho vnútornej reprezentácie [4]. Str. 29

Java je *objektovo orientovaný* programovací jazyk, založený na *triedach* z *rodiny jazykov C* [63]. Str. 7, 11, 33

JavaScript multiplatformný, *objektovo orientovaný* skriptovací jazyk [64]. Str. 11, 15, 18, 20, 22, 23, 48, 54

kolekcia je zoskupenie premenlivého počtu dátových položiek, ktoré majú spoločný význam pre vyriešenie problému a musia byť riadené spoločne nejakým kontrolovaným spôsobom [36]. Str. 6, 9, 30, 34

konštruktor je špeciálny typ podprogramu, volaný na vytvorenie objektu [37]. Str. 27

kontajnerizácia je virtualizácia na úrovni operačného systému [85]. Str. 25, 31, 34, 50

lambda výraz je anonymná funkcia, ktorú je možné použiť na vytvorenie delegátov alebo výrazových stromových typov [16]. Str. 27, 34

legacy kód je kód, ktorý už nie je naďalej vyvíjaný, nevychádzajú jeho nové verzie [66]. Str. 3, 15, 29

mapper je návrhový vzor, ktorý vykonáva obojsmerný prenos dát medzi trvalým dátovým úložiskom (väčšinou relačnou databázou) [43]. Str. 25, 26, 30, 33, 35

[1] https://github.com/

Markdown je odľahčený značkovací jazyk, ktorý slúži na úpravu prostého textu a jeho následný prevod na formátovaný text publikovateľný na webe [68]. Str. 8, 17, 18, 19, 20, 38, 39

Markup alebo značkový jazyk kombinuje text a informáciu o texte (metainformáciu). Metainformácia, napríklad o logickej štruktúre alebo spôsobe prezentácie textu, sa vyjadruje použitím značiek, ktoré sú zmiešané s primárnym textom [69]. Str. 18, 54

marshalling je proces transformácie pamäťovej reprezentácie objektu do dátového formátu, použiteľného na uchovávanie alebo prenos [97]. Str. 6

Mediator je objekt, ktorý zapúzdruje komunikáciu objektov [4]. Str. 29, 30

Memento je návrhový vzor, ktorý bez porušenia zapúzdrenia zachytí vnútorný stav objektu, aby sa mohol neskôr obnoviť [4]. Str. 29

metóda je *funkcia* v *OOP*, ktorá môže pracovať s dátami *triedy* alebo *objektu* [70]. Str. 6, 9, 11, 27, 33, 49, 53

Microsoft Azure je cloudová platforma spoločnosti *Microsoft*, vytvorená k hosťovaniu a škálovaniu webových aplikácií cez datacentra Microsoftu [71]. Str. 8, 21

mock simuluje správanie zložitého, skutočného objektu a preto je užitočný, keď je reálny objekt nepraktické alebo nemožné začleniť do *unit test*u [76]. Str. 15

modifikátor prístupu sú kľúčové slová v objektovo orientovaných jazykoch, ktoré určujú dostupnosť tried, metód a ďalších členov [28]. Str. 11, 27, 34

multiplatformový softvér je softvér pre viacero typov platforiem [77]. Str. 21, 25, 26, 31

namespace vyjadruje sémantickú kategóriu kódu [78]. Str. 24, 28

NHibernate *open source ORM* pre *.NET Framework* [23]. Str. 28

NuGet je bezplatný a *open source* manažér pre správu balíčkov, určený pre vývojovú platformu spoločnosti *Microsoft* [82]. Str. 24, 31, 32, 33, 34, 35, 49

objekt je konkrétna inštancia triedy, je kombináciou premenných, funkcií a dátových štruktúr [83]. Str. 9, 49, 50, 52, 54, 55

Observer je návrhový vzor, ktorý definuje závislosť 1 ku N medzi objektami, vďaka čomu keď objekt zmení stav, všetky jeho závislosti sú informované [4]. Str. 29

on–demand course je kurz, pri ktorom študent získa prístup k študijným materiálom kdekoľvek a kedykoľvek [24]. Str. 7, 13

open source je akákoľvek informácia, dostupná verejnosti za podmienky, že možnosť jej slobodného šírenia zostane zachovaná. [84]. Str. 5, 8, 16, 17, 19, 21, 22, 24, 48, 51, 52, 53, 54, 55

operátor je konštrukcia, ktorá sa všeobecne správa ako *funkcia*, ale líši sa od nej syntakticky alebo sémanticky [86]. Str. 6, 7, 9, 11, 27, 28, 33, 34

orchestracia aplikacií je ich automatizovaná konfigurácia, koordinácia a správa [87]. Str. 31

paralelné programovanie je koncept, ktorý umožňuje naprogramovať úlohy, ktoré sú schopné paralelného (súčasného) priebehu [88]. Str. 25, 26, 31

parameter je špeciálny druh premennej, používaný v podprograme na referencovanie jedného z údajov, poskytnutých ako vstup do podprogramu [89]. Str. 27, 33, 34

partial trieda je *trieda*, krorej definícia môže byť rozdelená na dva alebo viac zdrojových súborov [19]. Str. 27

pipelines slúžia na automatické zostavovanie a nasadzovanie aplikácií [12]. Str. 29

podmienka je riadiaca štruktúra programu, ktorá umožňuje rozdielne správanie, v závislosti od logickej podmienky, ktorá je vyhodnotená ako pravda alebo nepravda [38]. Str. 6, 7, 9, 11, 27, 33, 49

pole je dátová štruktúra, pozostávajúca zo súboru *hodnôt* alebo *premennaých* [31]. Str. 6, 9, 27, 33

premenná je pamäťové miesto so svojou hodnotou [109]. Str. 6, 7, 9, 11, 33, 53, 54

preťažovanie operátorov je špecifický prípad polymorfizmu, kedy rôzni operátori majú odlišné implementácie v závislosti od svojich argumentov [52]. Str. 9, 14

procedurálne programovanie je programovacia paradigma, postavená na koncepte volania procedúr [92]. Str. 33

proces je inštancia počítačového programu, ktorý sa vykonáva [93]. Str. 31

property je špeciálnym členom *triedy*, ktorého funkčnosť je medzi *field* a *metódou*. Syntax na čítanie a zápis vlastností je rovnaký ako pre *field*, ale sú preložené na volania *metód* [94]. Str. 27, 49

Prototype je špeciálny druh objektu, určený na vytváranie prototypov a nových objektov, pomocou kopírovania spomínaných prototypov [4]. Str. 29

Proxy je návrhový vzor, ktorý vytvára zástupné miesto pre iný objekt na kontrolu prístupu k nemu [4]. Str. 29

pull request je navrhovaná zmena do *Git* repozitára, odoslaná od užívateľa a akceptovaná alebo odmietaná spolupracovníkmi repozitára [7]. Str. 8, 24, 39

Python je vysoko úrovňový skriptovací programovací jazyk, vyvýjaný ako *open source* projekt [95]. Str. 11, 19

README súbor obsahuje informácie o iných súboroch v adresári alebo archíve počítačového softvéru [26]. Str. 17, 40

referencia je *hodnota*, ktorá umožňuje programu nepriamo získať prístup k určitým dátam [96]. Str. 9, 27

repozitár je jeden z najužitočnejších a najrozšírenejších návrhových vzorov, ktoré sa kedy objavili [3]. Str. 25, 26, 29, 33, 34

rodina C jazykov sú jazyky založené na programovacom jazyku *C*, kvôli jeho veľkej popularite [67]. Str. 33, 49, 51

rozhranie je spoločná hranica, cez ktorú si môžu samostatné komponenty počítačového systému vymieňať informácie [62]. Str. 9, 27, 34

sealed trieda je *trieda*, ktorá zabraňuje iným triedam zdediť z nej [20]. Str. 27

Silverlight je *framework* určený na písanie a spúšťanie internetových aplikácií [73]. Str. 31

Singleton je návrhový vzor, ktorý umožňuje mať triede iba jednu inštanciu a zaisťuje jej globálny prístupový bod [4]. Str. 29

smoke test je krátky test, ktorý slúži na rýchle overenie, či je vyvíjaná aplikácia pripravená na ďalšie testovanie [98]. Str. 28

SOLID päť dizajnových princípov, ktoré sú určené na to, aby softvérové návrhy boli zrozumiteľnejšie, flexibilnejšie a udržateľnejšie [100]. Str. 29

správa výnimiek je špeciálny prípad softvérového prerušenia, vyvolaného následkom výnimočnej situácie pri spracovaní v procesore [110]. Str. 6, 9, 11, 27, 34

State je návrhový vzor, ktorý umožňuje objektu zmeniť jeho správanie, keď sa zmení jeho vnútorný stav. Zdá sa, že zmenil svoju triedu [4]. Str. 29

Strategy je návrhový vzor, ktorý definuje skupinu algoritmov, pričom každý z nich zapúzdruje. Klientovi umožňuje voľbu, ktorý algoritmus použiť [4]. Str. 29

štruktúra je zložený dátový typ, pozostávajúci zo skupiny *premenných*, s ktorými umožňuje pracovať ako s jedným celkom [101]. Str. 27

stub je program, ktorý simuluje správanie softvérových komponentov, na ktorých závisí modul podstupujúci testy [104]. Str. 15

switch je riadiaca štruktúra programu, ktorá umožňuje hodnote premennej alebo výrazu zmeniť tok programu pomocou vyhľadávacej mapy [102]. Str. 27

task je jednotka vykonávania programu [103]. Str. 31

TechNet je program spoločnosti *Microsoft*, ktorý zahŕňa rozsiahle zdroje technických informácií, určených pre IT odborníkov [74]. Str. 7

Template Method je návrhový vzor, ktorý definuje kostru algoritmu, pričom niektoré presúva do podtried. Necháva podtriedy redefinovať kroky algoritmu bez zmeny jeho štruktúry [4]. Str. 29

Textile je ľahký *Markup*, ktorý používa textom formátovanú syntax na konverziu obyčajného textu do štruktúrovaného *HTML* kódu [105]. Str. 19

trieda je rozšíriteľná šablóna programového kódu na vytváranie *objekt*ov [34]. Str. 6, 9, 27, 30, 34, 48, 51, 52, 53, 54

tuple je typ, ktorý sa definuje podla ľahkej syntaxe [13]. Str. 27, 34

TypeScript je *open source* programovací jazyk vytvorený a spravovaný firmou Microsoft. Jedná sa o nadstavbu nad jazykom *JavaScript* [107]. Str. 22, 23

ukazovateľ je *objekt* v programovacom jazyku, ktorý v sebe ukladá pamäťovú adresu inej hodnoty, umiestnenej v pamäti počítača [91]. Str. 14

UnitOfWork je návrhový vzor, ktorý udržiava v pamäti aktualizácie a posiela ich do databázy ako jednu transakciu [8]. Str. 25, 26, 29, 34

unit test označuje automatické testovanie a overovanie fungovania a správnosti implementácie systému, testuje iba jeden daný konkrétny prípad [108]. Str. 13, 23, 24, 27, 28, 52

Unity je multiplatformný *herný engine* [25]. Str. 10, 13, 22, 23

ViewModelLocator je návrhový vzor, ktorý umožnuje zvoliť, aký ViewModel ide k view [9]. Str. 30

Visitor je návrhový vzor, ktorý reprezentuje operáciu vykonanú na elementoch objektu. Umožňuje definíciu novej operácie bez zmenenia tried elementov, na ktorých operuje [4]. Str. 29

Visual Studio je plnohodnotné integrované vývojové prostredie (*IDE*) pre *Android*, *iOS*, *Windows*, web aj cloud [42]. Pre viac informácií v projekte viď paragraf 2.3.1. Str. 8, 9, 10, 25, 26, 27, 28, 29, 31, 32, 34, 38, 39, 40

vlákno je najmenšia sekvencia programových inštrukcií, ktoré je možné spracovať nezávisle [106]. Str. 11, 31

Windows je séria niekoľkých rodín operačných systémov od spoločnosti *Microsoft* [75]. Str. 6, 7, 13, 16, 17, 21, 22, 23, 34, 38, 39, 40, 48, 49, 55

Xamarin nástroje určené na písanie aplikácií pre viacero *OS* [111]. Str. 22, 23, 31

xUnit je bezplatný, *open source* komunitný testovací nástroj pre *.NET Framework* [22]. Str. 28

Skratky

ACID Atomicity, Consistency, Isolation, Durability. 28

API Application programming interface. 7, 8, 9, 18, 21, 28, 48

CAP Consistency, Availability, Partition tolerance. 28

CI Continuous integration. 25, 26, 28, 29

CLR Common Language Runtime. 21, 34

CRUD Create, read, update and delete. 28

CSS Cascading Style Sheets. 11, 19, 20, 22, 23

DAL Data access layer. 28, 32, 33, 34

DDD Domain-driven design. 13

DRY Don't repeat yourself. 29

GoF Gang of Four. 29, 34

GPL GNU General Public License. 18

GRASP General Responsibility Assignment Software Patterns. 29

HTML Hypertext Markup Language. 11, 18, 19, 22, 23, 54

HTTP Hypertext Transfer Protocol. 20

IDE Integrated development environment. 8, 21, 22, 23, 39, 40, 55

IoC Inversion of control. 30

IoT Internet of things. 12

JSON JavaScript Object Notation. 6, 23, 28

KISS Keep it simple, stupid. 29

LGPL GNU Lesser General Public License. 17

LINQ Language Integrated Query. 6, 9, 13, 25, 26, 28

MCSD Microsoft Certified Solutions Developer. 6

MCT Microsoft Certified Trainers. 7

MIT Massachusetts Institute of Technology. 3, 18, 19, 22, 24, 39

MSDN Microsoft Developer Network. 7

MVC Model-View-Controller. 11, 30

MVP Most Valuable Professional. 5, 13, 30

MVVM Model-View-ViewModel. 25, 26, 30, 32, 35, 41

OOP Object-oriented programming. 6, 7, 10, 11, 15, 24, 25, 26, 27, 33, 34, 49, 50, 51, 52

ORM Object-relational mapping. 28, 52

OS Operating system. 22, 34, 38, 39, 40, 48, 55

PDF Portable Document Format. 8, 16, 17, 18, 19, 20, 21

PHP Hypertext Preprocessor. 11

ReST reStructuredText. 19

REST Representational state transfer. 6, 9

SQL Structured Query Language. 7, 11, 23, 28, 33, 34

UI User interface. 21, 22, 28

UML Unified Modeling Language. 11, 14, 19, 29

UWP Universal Windows Platform. 9, 23, 31

WPF Windows Presentation Foundation. 21, 25, 26, 30, 31, 33, 35, 41

WYSIWYG What You See Is What You Get. 39

XAML Extensible Application Markup Language. 22, 23, 30, 33, 35

XML Extensible Markup Language. 16, 22, 23, 28

Literatúra k slovníku

[1] Ajax.org: *Ajax.org*. [Online; navštívené 25.4.2019].
URL https://github.com/ajaxorg/ace/

[2] Chocolatey Software, Inc.: Chocolatey. [Online; navštívené 5.5.2019].
URL https://chocolatey.org/

[3] Csaba, P.: *The Repository Design Pattern*. [Online; navštívené 22.1.2019].
URL https://code.tutsplus.com/tutorials/the-repository-design-pattern--net-35804

[4] Data & Object Factory, LLC: *.NET Design Patterns — Dofactory*. [Online; navštívené 25.1.2019].
URL https://www.dofactory.com/net/design-patterns

[5] Driessen, V.: *Introducing GitFlow*. [Online; navštívené 22.1.2019].
URL https://datasift.github.io/gitflow/IntroducingGitFlow.html

[6] Fournova Software GmbH: *Git Commands — Git Tower*. [Online; navštívené 27.4.2019].
URL https://www.git-tower.com/learn/git/commands/git-commit

[7] GitHub Incorporation: *Pull request — GitHub Glossary*. [Online; navštívené 13.1.2019].
URL https://help.github.com/articles/github-glossary/#pull-request

[8] Koirala, S.: *Unit of Work Design Pattern*. [Online; navštívené 24.1.2019].
URL https://www.codeproject.com/Articles/581487/Unit-of-Work-Design-Pattern

[9] Michels, B. L.: *ViewModel Locator in WPF*. [Online; navštívené 20.4.2019].
URL https://www.c-sharpcorner.com/UploadFile/20c06b/viewmodel-locator-in-wpf/

[10] Microsoft Corporation: *Abstract — Microsoft Docs*. [Online; navštívené 23.1.2019].
URL https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/abstract

[11] Microsoft Corporation: *ASP.NET*. [Online; navštívené 31.3.2019].
URL https://asp.net

[12] Microsoft Corporation: *Azure Pipelines*. [Online; navštívené 24.1.2019].
URL https://azure.microsoft.com/en-us/services/devops/pipelines/

[13] Microsoft Corporation: *C# tuple types — Microsoft Docs*. [Online; navštívené 23.1.2019].
URL https://docs.microsoft.com/en-us/dotnet/csharp/tuples

[14] Microsoft Corporation: *EntityFramework — Microsoft Docs*. [Online; navštívené 8.1.2019].
URL https://docs.microsoft.com/en-us/ef/

[15] Microsoft Corporation: *Finalizers — Microsoft Docs*. [Online; navštívené 23.1.2019].
URL https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/destructors

[16] Microsoft Corporation: *Lambda expressions — Microsoft Docs*. [Online; navštívené 23.1.2019].
URL https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/statements-expressions-operators/lambda-expressions

[17] Microsoft Corporation: *.NET Framework — What's New in C# 7.0*. [Online; navštívené 23.1.2019].
URL https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/destructors

[18] Microsoft Corporation: *.NET Standard — Microsoft Docs*. [Online; navštívené 8.1.2019].
URL https://docs.microsoft.com/en-us/dotnet/standard/net-standard

[19] Microsoft Corporation: *Partial Classes and Methods — Microsoft Docs*. [Online; navštívené 23.1.2019].
URL https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/partial-classes-and-methods

[20] Microsoft Corporation: *Sealed — Microsoft Docs*. [Online; navštívené 23.1.2019].
URL https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/sealed

[21] Microsoft Corporation: *Solutions and projects — Microsoft Docs*. [Online; navštívené 11.1.2019].
URL https://docs.microsoft.com/en-us/visualstudio/ide/solutions-and-projects-in-visual-studio?view=vs-2017

[22] .NET Foundation: *xUnit*. [Online; navštívené 19.4.2019].
URL https://xunit.net/

[23] NHibernate Community: *NHibernate*. [Online; navštívené 19.4.2019].
URL https://nhibernate.info/

[24] Training Industry: *On-Demand Learning*. [Online; navštívené 6.1.2019].
URL https://trainingindustry.com/glossary/on-demand-learning/

[25] Unity Technologies: *Unity*. [Online; navštívené 20.1.2019].
URL https://unity3d.com/

[26] Wikipedia: .

[27] Wikipedia: *Acceptance testing — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Acceptance%20testing&oldid=874072331, 2019, [Online; navštívené 25.1.2019].

[28] Wikipedia: *Access modifiers — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Access%20modifiers&oldid=875542460, 2019, [Online; navštívené 23.1.2019].

[29] Wikipedia: *ADO.NET — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=ADO.NET&oldid=858190230, 2019, [Online; navštívené 19.4.2019].

[30] Wikipedia: *Apache License — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Apache%20License&oldid=891703194, 2019, [Online; navštívené 30.4.2019].

[31] Wikipedia: *Array data structure — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Array%20data%20structure&oldid=866978477, 2019, [Online; navštívené 7.1.2019].

[32] Wikipedia: *Asynchrony (computer programming) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Asynchrony%20(computer%20programming)&oldid=835151094, 2019, [Online; navštívené 4.1.2019].

[33] Wikipedia: *Business logic — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Business%20logic&oldid=864157913, 2019, [Online; navštívené 4.1.2019].

[34] Wikipedia: *Class (computer programming) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Class%20(computer%20programming)&oldid=860065842, 2019, [Online; navštívené 7.1.2019].

[35] Wikipedia: *Classroom — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Classroom&oldid=874039952, 2019, [Online; navštívené 6.1.2019].

[36] Wikipedia: *Collection (abstract data type) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Collection%20(abstract%20data%20type)&oldid=799613276, 2019, [Online; navštívené 7.1.2019].

[37] Wikipedia: *Constructor (object-oriented programming) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Constructor%20(object-oriented%20programming)&oldid=862770779, 2019, [Online; navštívené 23.1.2019].

[38] Wikipedia: *Control flow if-then-(else) statements — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Control%20flow&oldid=8748402222#If-then-(else)_statements, 2019, [Online; navštívené 6.1.2019].

[39] Wikipedia: *Control flow loops — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Control%20flow&oldid=8748402222#Loops, 2019, [Online; navštívené 6.1.2019].

[40] Wikipedia: *C (programming language) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=C%20(programming%20language)&oldid=880112456, 2019, [Online; navštívené 27.1.2019].

[41] Wikipedia: *C++ — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=C%2B%2B&oldid=879914581, 2019, [Online; navštívené 27.1.2019].

[42] Wikipedia: *Data binding — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Data%20binding&oldid=874677878, 2019, [Online; navštívené 4.1.2019].

[43] Wikipedia: *Data mapper pattern — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Data%20mapper%20pattern&oldid=875748028, 2019, [Online; navštívené 22.1.2019].

[44] Wikipedia: *Delegation (object-oriented programming) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Delegation%20(object-oriented%20programming)&oldid=873930616, 2019, [Online; navštívené 23.1.2019].

[45] Wikipedia: *Desktopové prostredie — Wikipedia, The Free Encyclopedia.* http://sk.wikipedia.org/w/index.php?title=Desktopov%C3%A9%20prostredie&oldid=6586533, 2019, [Online; navštívené 21.1.2019].

[46] Wikipedia: *Docker (software) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Docker%20(software)&oldid=879852107, 2019, [Online; navštívené 26.1.2019].

[47] Wikipedia: *Enumerated type — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Enumerated%20type&oldid=870007823, 2019, [Online; navštívené 23.1.2019].

[48] Wikipedia: *Event (computing) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Event%20(computing)&oldid=870366273, 2019, [Online; navštívené 13.1.2019].

[49] Wikipedia: *Field (computer science) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Field%20(computer%20science)&oldid=852139888, 2019, [Online; navštívené 23.1.2019].

[50] Wikipedia: *Funkcionálne programovanie — Wikipedia, The Free Encyclopedia.* http://sk.wikipedia.org/w/index.php?title=Funkcion%C3%A1lne%20programovanie&oldid=6400080, 2019, [Online; navštívené 27.1.2019].

[51] Wikipedia: *Garbage collection (computer science) — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Garbage%20collection%20(computer%20science)&oldid=873228320, 2019, [Online; navštívené 4.1.2019].

[52] Wikipedia: *Generic programming — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Generic%20programming&oldid=875496455, 2019, [Online; navštívené 7.1.2019].

[53] Wikipedia: *GitHub — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=GitHub&oldid=877328890, 2019, [Online; navštívené 9.1.2019].

[54] Wikipedia: *Git — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Git&oldid=876004333, 2019, [Online; navštívené 9.1.2019].

[55] Wikipedia: *Herní engine — Wikipedia, The Free Encyclopedia*. http://cs.wikipedia.org/w/index.php?title=Hern%C3%AD%20engine&oldid=16215691, 2019, [Online; navštívené 20.1.2019].

[56] Wikipedia: *Hodnota (informatika) — Wikipedia, The Free Encyclopedia*. http://cs.wikipedia.org/w/index.php?title=Hodnota%20(informatika)&oldid=16019324, 2019, [Online; navštívené 7.1.2019].

[57] Wikipedia: *Identifier — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Identifier&oldid=874281190#In_computer_science, 2019, [Online; navštívené 14.1.2019].

[58] Wikipedia: ıHaskell (programming language) — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Haskell%20(programming%20language)&oldid=893176890, 2019, [Online; navštívené 25.4.2019].

[59] Wikipedia: *Inheritance (object-oriented programming) — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Inheritance%20(object-oriented%20programming)&oldid=876686584, 2019, [Online; navštívené 7.1.2019].

[60] Wikipedia: *Instructor-led training — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Instructor-led%20training&oldid=876627305, 2019, [Online; navštívené 6.1.2019].

[61] Wikipedia: *Integration testing — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Integration%20testing&oldid=865103223, 2019, [Online; navštívené 25.1.2019].

[62] Wikipedia: *Interface — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Interface&oldid=870815776, 2019, [Online; navštívené 7.1.2019].

[63] Wikipedia: *Java (programming language) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Java%20(programming%20language)&oldid=879795563, 2019, [Online; navštívené 27.1.2019].

[64] Wikipedia: *JavaScript — Wikipedia, The Free Encyclopedia.* http://cs.wikipedia.org/w/index.php?title=JavaScript&oldid=16853929, 2019, [Online; navštívené 31.3.2019].

[65] Wikipedia: *Ladění (programování) — Wikipedia, The Free Encyclopedia.* http://cs.wikipedia.org/w/index.php?title=Lad%C4%9Bn%C3%AD%20(programov%C3%A1n%C3%AD)&oldid=13115457, 2019, [Online; navštívené 22.1.2019].

[66] Wikipedia: *Legacy code — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Legacy%20code&oldid=822508543, 2019, [Online; navštívené 10.1.2019].

[67] Wikipedia: *List of C-family programming languages — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=List%20of%20C-family%20programming%20languages&oldid=879524385, 2019, [Online; navštívené 27.1.2019].

[68] Wikipedia: *Markdown — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Markdown&oldid=877391375, 2019, [Online; navštívené 11.1.2019].

[69] Wikipedia: *Markup language — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Markup%20language&oldid=893246462, 2019, [Online; navštívené 25.4.2019].

[70] Wikipedia: *Method (computer programming) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Method%20(computer%20programming)&oldid=875290752, 2019, [Online; navštívené 7.1.2019].

[71] Wikipedia: *Microsoft Azure — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Microsoft%20Azure&oldid=893532628, 2019, [Online; navštívené 1.5.2019].

[72] Wikipedia: *Microsoft Blend — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Microsoft%20Blend&oldid=867687062, 2019, [Online; navštívené 26.1.2019].

[73] Wikipedia: *Microsoft Silverlight — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Microsoft%20Silverlight&oldid=869454912, 2019, [Online; navštívené 26.1.2019].

[74] Wikipedia: *Microsoft TechNet — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Microsoft%20TechNet&oldid=888070148, 2019, [Online; navštívené 1.5.2019].

[75] Wikipedia: *Microsoft Windows — Wikipedia, The Free Encyclopedia*. http://sk.wikipedia.org/w/index.php?title=Microsoft%20Windows&oldid=6762280, 2019, [[Online; navštívené 6.1.2019].

[76] Wikipedia: *Mock object — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Mock%20object&oldid=875743810, 2019, [Online; navštívené 10.1.2019].

[77] Wikipedia: *Multiplatformový softvér — Wikipedia, The Free Encyclopedia*. http://sk.wikipedia.org/w/index.php?title=Multiplatformov%C3%BD%20softv%C3%A9r&oldid=6374565, 2019, [Online; navštívené 26.1.2019].

[78] Wikipedia: *Namespace — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Namespace&oldid=875161689, 2019, [Online; navštívené 27.1.2019].

[79] Wikipedia: *.NET Core — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=.NET%20Core&oldid=876997691, 2019, [Online; navštívené 8.1.2019].

[80] Wikipedia: *.NET Framework — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=.NET%20Framework&oldid=875426353, 2019, [Online; navštívené 4.1.2019].

[81] Wikipedia: *.NET — Wikipedia, The Free Encyclopedia*. http://sk.wikipedia.org/w/index.php?title=.NET&oldid=6465183, 2019, [Online; navštívené 14.1.2019].

[82] Wikipedia: *NuGet — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=NuGet&oldid=858189652, 2019, [Online; navštívené 22.1.2019].

[83] Wikipedia: *Object (computer science) — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Object%20(computer%20science)&oldid=875165207, 2019, [Online; navštívené 7.1.2019].

[84] Wikipedia: *Open source — Wikipedia, The Free Encyclopedia*. http://sk.wikipedia.org/w/index.php?title=Open%20source&oldid=6724020, 2019, [Online; navštívené 12.1.2019].

[85] Wikipedia: *Operating-system-level virtualization — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Operating-system-level%20virtualization&oldid=875683892, 2019, [Online; navštívené 26.1.2019].

[86] Wikipedia: *Operator (computer programming) — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Operator%20(computer%20programming)&oldid=876982881, 2019, [Online; navštívené 7.1.2019].

[87] Wikipedia: *Orchestration (computing) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Orchestration%20(computing)&oldid=874133738, 2019, [Online; navštívené 26.1.2019].

[88] Wikipedia: *Paralelní programování — Wikipedia, The Free Encyclopedia.* http://cs.wikipedia.org/w/index.php?title=Paraleln%C3%AD%20programov%C3%A1n%C3%AD&oldid=15419962, 2019, [Online; navštívené 8.1.2019].

[89] Wikipedia: *Parameter (computer programming) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Parameter%20(computer%20programming)&oldid=867205673, 2019, [Online; navštívené 14.1.2019].

[90] Wikipedia: *Podprogram — Wikipedia, The Free Encyclopedia.* http://sk.wikipedia.org/w/index.php?title=Podprogram&oldid=5313183, 2019, [Online; navštívené 7.1.2019].

[91] Wikipedia: *Pointer (computer programming) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Pointer%20(computer%20programming)&oldid=875349749, 2019, [Online; navštívené 7.1.2019].

[92] Wikipedia: *Procedural programming — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Procedural%20programming&oldid=878649532, 2019, [Online; navštívené 27.1.2019].

[93] Wikipedia: *Process (computing) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Process%20(computing)&oldid=872231543, 2019, [Online; navštívené 26.1.2019].

[94] Wikipedia: *Property (programming) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Property%20(programming)&oldid=873147296, 2019, [Online; navštívené 23.1.2019].

[95] Wikipedia: *Python — Wikipedia, The Free Encyclopedia.* http://cs.wikipedia.org/w/index.php?title=Python&oldid=17051738, 2019, [Online; navštívené 31.3.2019].

[96] Wikipedia: *Reference (computer science) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Reference%20(computer%20science)&oldid=874716317, 2019, [Online; navštívené 7.1.2019].

[97] Wikipedia: *Serialization — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Serialization&oldid=872278702, 2019, [Online; navštívené 4.1.2019].

[98] Wikipedia: *Smoke testing (software) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Smoke%20testing%20(software)&oldid=869215972, 2019, [Online; navštívené 25.1.2019].

[99] Wikipedia: *Software framework — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Software%20framework&oldid=876257901, 2019, [Online; navštívené 8.1.2019].

[100] Wikipedia: *SOLID — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=SOLID&oldid=873665817, 2019, [Online; navštívené 25.1.2019].

[101] Wikipedia: *Struktura (programovací jazyk C) — Wikipedia, The Free Encyclopedia.* http://cs.wikipedia.org/w/index.php?title=Struktura%20(programovac%C3%AD%20jazyk%20C)&oldid=14566769, 2019, [Online; navštívené 23.1.2019].

[102] Wikipedia: *Switch statement — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Switch%20statement&oldid=866307445, 2019, [Online; navštívené 14.1.2019].

[103] Wikipedia: *Task (computing) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Task%20(computing)&oldid=858478262, 2019, [Online; navštívené 26.1.2019].

[104] Wikipedia: *Test stub — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Test%20stub&oldid=835449711, 2019, [Online; navštívené 10.1.2019].

[105] Wikipedia: *Textile (markup language) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Textile%20(markup%20language)&oldid=880816637, 2019, [Online; navštívené 26.4.2019].

[106] Wikipedia: *Thread (computing) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Thread%20(computing)&oldid=875467045, 2019, [Online; navštívené 26.1.2019].

[107] Wikipedia: *TypeScript — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=TypeScript&oldid=895051287, 2019, [Online; navštívené 5.5.2019].

[108] Wikipedia: *Unit testing — Wikipedia, The Free Encyclopedia.* http://cs.wikipedia.org/w/index.php?title=Unit%20testing&oldid=11480059, 2019, [Online; navštívené 10.1.2019].

[109] Wikipedia: *Variable (computer science) — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=Variable%20(computer%20science)&oldid=872285972, 2019, [Online; navštívené 7.1.2019].

[110] Wikipedia: *Výjimka (programování) — Wikipedia, The Free Encyclopedia*. http://cs.wikipedia.org/w/index.php?title=V%C3%BDjimka%20(programov%C3%A1n%C3%AD)&oldid=13326277, 2019, [Online; navštívené 7.1.2019].

[111] Wikipedia: *Xamarin — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Xamarin&oldid=879903826, 2019, [Online; navštívené 26.1.2019].

[112] Wiktionary: *clean code — Wiktionary, The Free Dictionary*. https://en.wiktionary.org/w/index.php?title=clean_code&oldid=50769273, 2019, [Online; navštívené 22.1.2019].

[113] ZZZ Projects: *Dapper*. [Online; navštívené 19.4.2019]. URL https://dapper-tutorial.net/

Príloha A

Syllabus in English

A.1 Syllabus of lectures:

1. Introduction to C#, Visual studio and .NET
2. Object-oriented programming and advanced constructs in C#
3. .NET Standard and Language Integrated Query (LINQ)
4. Entity framework
5. Application testing and continous integration in C#
6. Design patterns in C#
7. Clean code in C#
8. Repository, UnitOfWork, Facade and Mapper design patterns
9. Model–View–ViewModel (MVVM) design pattern
10. Windows Presentation Foundation (WPF)
11. Asynchronous and parallel programming in C#
12. Performance analyzing and memory managment in C#
13. Multiplatform development, .NET Core, application containerization

A.2 Syllabus of laboratories:

1. Work in Visual studio environment
2. Data management using Entity Framework
3. Clean code
4. View Model creation, WPF basics
5. WPF View binding to View Model