Vyšší odborná škola a Střední průmyslová škola elektrotechnická Olomouc,
Božetěchova3

PRAKTICKÁ ZKOUŠKA Z ODBORNÝCH PŘEDMĚTŮ

Tutoriál knihovny WxPython

2020

Filip Plachý

Vyšší odborná škola a
Střední průmyslová škola elektrotechnická
Božetěchova 3, 772 00 Olomouc

ZADÁNÍ PRAKTICKÉ ZKOUŠKY Z ODBORNÝCH PŘEDMĚTŮ

pro: **Filipa PLACHÉHO**

Studijní obor: **78-42-M/01 Technické lyceum**

Třída: **4L**

Ředitelství Vyšší odborné školy a Střední průmyslové školy elektrotechnické Olomouc Vám podle vyhlášky Ministerstva školství, mládeže a tělovýchovy č. 177/2009 Sb., o bližších podmínkách ukončování vzdělávání ve středních školách maturitní zkouškou, ve znění vyhlášky č. 90/2010 Sb., vyhlášky č. 274/2010 Sb., vyhlášky č. 54/2011 Sb. a vyhlášky č. 273/2011 Sb., určuje tuto praktickou zkoušku z odborných předmětů.

Téma: **Tutoriál knihovny wxPython**

Způsob zpracování a pokyny k obsahu:

- Nastudujte použití knihovny wxWidgets v programovacím jazyce Python.
- Vytvořte tutoriál pro knihovnu wxPython.
- Vyberte vhodný způsob zápisu tutoriálu (dokumentace) s ohledem na jeho snadné čtení a pozdější úpravy a aktualizace.
- Použijte objektově orientované programování.
- Vytvořte poster (formát PDF, velikost A2) prezentující maturitní práci.
- Připravte podklady pro aktivní účast v jednotlivých kolech SOČ.
- **Všechny body zadání jsou závazné, pokud toto žák nedodrží, přechází praktická maturitní zkouška z formy dlouhodobé maturitní práce na praktickou maturitní zkoušku jednodenní.**

Rozsah: 25 až 35 stran

Kritéria hodnocení: Hodnocení práce probíhá ve třech fázích.

Průběžné hodnocení zohledňuje postupné plnění zadaných úkolů, dodržování termínů, míru samostatnosti žáka. Hodnocení závěrečné posuzuje míru splnění všech požadavků vyplývajících ze zadání práce a funkčnost produktů. Hodnocena je přehlednost, úplnost, srozumitelnost a formální stránka textové části práce. Hodnocení obhajoby práce zahrnuje způsob a srozumitelnost projevu, vzhled prezentace, odpovědi na dotazy.

Délka obhajoby: 15 minut

Počet vyhotovení: 1 výtisk

Vedoucí práce: Ing. Marek Nožka

Datum zadání: 5. října 2020 Datum odevzdání: 26. března 2021

V Olomouci dne 5. října 2020

.....
ředitel školy

Zadání převzal dne 5. října 2020

.....
podpis žáka

Prohlašuji, že jsem praktickou zkoušku z odborných předmětů vypracoval samostatně a všechny prameny jsem uvedl v seznamu použité literatury.

……
jméno a příjmení žáka

Chtěl bych vyslovit poděkování panu Ing. Markovi Nožkovi za odborné konzultace a poskytnuté informace.

……
jméno a příjmení žáka

Prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé práce nebo její části se souhlasem školy.

……
jméno a příjmení žáka

ABSTRAKT

Cílem diplomové práce je vytvořit sbírku tutoriálů popisující základy knihovny wxWidgets v programovacím jazyce Python. Samotná knihovna Pythonu podporující wxWidgets se nazývá wxPython. Jedná se o alternativu k vytváření aplikací s uživatelským grafickým rozhraním.

Sbírka je určena pro studenty programování začínající s GUI, kteří již mají základní zkušenosti s Pythonem. Součástí tutoriálů je i úvod do objektově orientovaného programování, které se při tvorbě grafického rozhraní využívá. Po dokončení tutoriálů je součástí sbírky i sada názorných příkladů, řešící základní jednoduché aplikace, tutoriál popisující aplikaci WxGlade a seznam základních prvků wxPythonu.

OBSAH

ÚVOD

Práce se rozděluje na 2 části. Teoretická a praktická. Teoretická obsahuje dopodrobna popsané nástroje a funkce, které byly při tvorbě tutoriálu využity. V praktické části si rozebereme samotný tutoriál knihovny wxPython. Tutoriál je určen pro studenty s již základními zkušenosti v programovacím jazyce python. Při práci s wxPython je potřeba mít i znalost Objektově orientovaného programování (OOP), tudíž součástí sady tutoriálů je i lekce do OOP. Cílem práce je vytvořit představu a znalostní základ, se kterým student zvládne vytvářet aplikace s grafickým uživatelským rozhraním.

1. TEORETICKÁ ČÁST

Teoretická část obsahuje základní informace všech programů a nástrojů, které byly při práci využity.

1.1 PYTHON[1]

Cílem této podkapitoly je obecný úvod do programovacího jazyka Python.

1.1.1 ÚVOD DO JAZYKA PYTHON

Python je univerzální programovací jazyk[2].

Jazyk patří mezi tzv. "interpretované jazyky". To znamená, že napsaný zdrojový kód v jazyce Python je převeden (interpretován) pomocí programu (interpreter/tlumočník) do jazyka, se kterým pracuje počítač. Samotný tlumočník pro Python je k dispozici na všech operačních systémech přímo na stránkách www.python.org.

Pro práci s pythonem slouží tzv. „IDE“ (Integrated Development Environment), česky: „editor“, který pomáhá uživateli s formátováním kódu tak, aby byl nejen přehledný pro uživatele, ale také aby se dal převést do formátu proveditelného počítačem. Nejznámějšími editory jsou: Pycharm, Pydev, Visual Studio Code, VIM, Atom/Atom-IDE, IDLE, Spyder...

Python byl navržen tak, aby se jeho veškeré funkce nacházely přímo v jádru programovacího jazyka.

Má jednodušší a méně přeplněnou syntaxi a gramatiku, např. díky využití mezer. Další jeho výhodou je rozšiřitelnost o další knihovny/moduly.

1.1.2 HISTORIE

Programovací jazyk Python byl navrhnut mezi roky 1990-1991 Holanďanem Guidem van Rossumem v národním výzkumném institutu pro matiku a informatiku (CWI) v Amsterdamu. Samotné pojmenování Pythonu nemá nic společného s druhem hada, neboť Van Rossum pojmenoval Python po televizním pořadu anglické BBC Monty Pythonův létající cirkus.

[1] Hlavní zdroj kapitoly - https://en.wikipedia.org/wiki/Python_(programming_language)

[2] Prostředek pro zápis algoritmů, které provádí počítač: https://en.wikipedia.org/wiki/Programming_language

První verzi kódu zveřejnil Van Rossum v únoru roku 1991 (verze 0.9.0). Již v této fázi vývoje bylo možné v Pythonu pracovat s třídami a dědičnostmi (viz podkapitola OOP). K dispozici již také byly i základní datové typy jako string[3], list[4] či dict[5] (slovník). V počátečním vydání Python již obsahoval modulový systém[6].

Python dosáhl verze 1 v lednu 1994. Novými funkcemi byla lambda[7] a práce s mapováním[8], filtrováním[9] a redukování[10] vyšších funkcí[11] (Funkce vyšších řádů berou jednu či více funkcí nebo atributů a jako výsledek vrací jednu funkci). Později Van Rossum opouští CWI a pokračuje na vývoji Pythonu ve CNRI (Korporace pro národní výzkumné iniciativy) ve Virginii. Další důležitou verzí je 1.4, kde se objevily pojmenované parametry[12] nebo podpora komplexních čísel.

Během Van Rossumova pobytu ve CNRI se institut snažil o zpřístupnění možnosti programování pro veřejnost se základním vzděláním. Tohle mělo později za následek menší "šachování" mezi licencemi Pythonu 1.6, které měly za cíl získání licence svobodného softwaru[13] (Free-software licence). Další verze 1.6.1 sice neobsahovala žádné důležité funkce, ale měla upravenou licenci na GPL/GNU[14] (General Public License nebo-li bezplatná softwarová licence).

Verze 2.0 byla vydána v říjnu 2000, která upravila seznamy do podoby jak je nyní známe.

---

[3] Formát textu - https://en.wikipedia.org/wiki/String_(computer_science)

[4] Datový typ, který obsahuje položky různých typů v jedné proměnné - https://en.wikipedia.org/wiki/List_comprehension

[5] Slovníky obsahují uložená data ve formátu *key : value*, kdy daný klíč obsahuje jednu nebo více položek - https://en.wikipedia.org/wiki/Associative_array

[6] Metoda funkčnosti programu na nezávislých zaměnitelných modulech - https://en.wikipedia.org/wiki/Modular_programming

[7] Komplexní (tzv. „anonymní“) funkce, která je se často využívá pro zkracování kódu pomocí zápisu na jeden řádek - https://en.wikipedia.org/wiki/Anonymous_function

[8] Aplikování funkce na element každé položky v seznamu - https://en.wikipedia.org/wiki/Map_%28higher-order_function%29

[9] Funkce zjišťuje datový typ položky a vrací bool hodnotu - https://en.wikipedia.org/wiki/Filter_%28higher-order_function%29

[10] Velmi obsáhlá funkce k uspořádávání a analýze dat - https://en.wikipedia.org/wiki/Fold_%28higher-order_function%29

[11] Jedná se o funkce, které berou při vstupu jednu nebo více funkcí jako argumenty a vrací je jako jednu funkci - https://en.wikipedia.org/wiki/Higher-order_function

[12] Funkce je schopna zkrátit kód tím, že uživatel si „zkrátí“ parametry do jedné proměnné, se kterou dále pracuje - https://en.wikipedia.org/wiki/Named_parameter

[13] Licence svobodného softwaru - https://en.wikipedia.org/wiki/Free-software_license

[14] Obecná veřejná licence GNU - https://en.wikipedia.org/wiki/GNU_General_Public_License

2.2 představila sjednocení typů Pythonu (typů napsaných v jazyku C) a tříd (typů napsaných v Pythonu) do jedné hierarchie.

2.5 představil prohlášení *with*[15] umožňující otevření a zavření souboru a další funkce.

Python 2.6 byl vydán, aby se shodoval se souběžným vývojem Pythonu 3.0 a varoval hlavně o funkcích, které jsou ve verzi 3.0 odstraněny.

Podobně byl vydán i 2.7, který se shodoval s 3.1. Python 2.7 byl posledním vydáním ve druhé sérii. V listopadu 2014 byl oznámen konec podpory 2.7 do roku 2020. Uživatelé byli vyzváni, aby postupně přešli na Python 3.0.

1. ledna 2020 byl "zmražen"[16] kód Pythonu 2.7. Konečné vydání, 2.7.18, došlo 20. dubna 2020 a zahrnovalo opravy kritických chyb a blokace vydání.

Verze 3.0 přišla 3. prosince 2008. 3.0 byla navržena, aby napravila základní konstrukční chyby jazyka. Tyhle změny avšak znemožnily zpětnou kompatibilitu se staršími verzemi (tudíž došlo k samotnému oddělení verzí z 2.x na 3.0). Hlavním mottem změn bylo odstranění nadbytečných, duplicitních konstrukcí a modulů. Vznikl nástroj tzv. 2to3[17], který dokázal přepsat automaticky Python 2 do nové verze, avšak nástroj nefunguje na 100% a některé aspekty nedokáže převést.

Hlavními úpravami byly:

- Změna *print*, aby se jednalo o vestavěnou funkci. V Pythonu 2.6 a 2.7 *print()* je k dispozici jako vestavěná funkce, avšak je maskovaná syntaxí příkazu, který lze deaktivovat zadáním *„from __future import print_functionv“* v hlavičce souboru
- Odebrání inputu ve verzi 2, ze které byl přebrat raw_input, který byl přejmenován na klasický input
- Přidání podpory pro anotace jednotlivých funkcí (Když v kódu byla nepoužívaná funkce se špatnou syntaxí, tak se program ve 2.x nespustil)
- Sjednocení str / unicode typů
- Odebrání funkcí zpětné kompatibility, včetně tříd starého stylu, výjimek řetězců a implicitních importů

[15] Handler with, který se využívá pro práci se čtením a zápisem do souborů (např. txt, json atd.) - https://en.wikipedia.org/wiki/Python_syntax_and_semantics#With_statements

[16] Oficiální ukončení kódu - https://en.wikipedia.org/wiki/Freeze_%28software_engineering%29

[17] Nástroj 2to3 - https://docs.python.org/3/library/2to3.html

- Změna funkce celočíselného dělení. (ve 2.0 *5 / 2 = 2*, nyní *5 / 2 = 2.5*. ve 3.0 vznikla náhradní syntaxe *5 // 2 = 2*)

1.2 wxPython[18]

WxPython je obal pro multiplatformní GUI (grafické uživatelské rozhraní) aplikačního rozhraní wxWidgets (napsán v C++) pro programovací jazyk Python. Jedná se o otevřený (veřejný kód) rozšiřující modul Pythonu. Oficiální stránky wxPythonu: https://www.wxpython.org

1.2.1 Historie wxPythonu

WxPython byl vytvořen Robinem Dunnem, když potřeboval GUI k operačnímu systému HP-UX a Windows verze 3.1 (1992-1995). Při hodnocení komerčních řešení narazil na vazby Pythonu se sadou nástrojů wxWidgets.

První verze byly vytvořeny ručně. Avšak brzy se kódová základna velmi obtížně udržovala synchronizovaná s novými verzemi wxWidgets. Pozdější verze byly vytvořeny pomocí SWIG[19], který výrazně snížil množství práce na aktualizaci.

První "moderní" verze 0.3, která byla oznámena v roce 1998.

Práce na tutoriálu probíhala ve verzi 4.1.0 s podtitulem „Escaping the Quarantine“.

Celý vývoj wxPython nalezneme na oficiálních stránkách:

https://wxpython.org/pages/changes/index.html

1.3 wxWidgets[20]

Samotná sada wxWidgets je knihovna nástrojů pro vytváření graficky uživatelských rozhraní napříč všemi platformami. WxWidgets umožňuje kódu GUI kompilovat a spouštět na několika počítačových platformách s žádnými nebo minimálními změnami kódu.

Jedná se o bezplatný a otevřený software distribuovaný za podmínek licence WxWidgets, která je obdobná GPL/GNU u Pythonu. Stránky wxWidgets: https://www.wxwidgets.org/

---

[18] Hlavní zdroj WxPython - https://en.wikipedia.org/wiki/WxPython

[19] Nástroj pro propojení knihoven jazyku C se skriptovacími jazyky - https://en.wikipedia.org/wiki/SWIG

[20] Hlavní zdroj WxWidgets - https://en.wikipedia.org/wiki/WxWidgets

1.3.1 Historie wxWidgets

WxWidgets (původně wxWindows) zahájil v roce 1992 Julian Smart z Edinburské Univerzity. V roce 2004 došlo k přejmenování wxWindows v důsledku požadavků společnosti Microsoft pro distribuci v UK.

Hlavní verze byla vydána 6. ledna 2004. Samotná verze 3.0 byla vydána 11. listopadu 2013.

1.4 WxGlade

WxGlade je aplikace pro vytváření GUI aplikací ve wxPythonu napsána v klasickém Pythonu. Obsahuje přehledné rozhraní pro práci s widgety a sizery knihovny. Během celé práce dynamicky zobrazuje vzhled celé aplikace a na konci vygeneruje kód (viz praktická část - 23).

Webové stránky - http://wxglade.sourceforge.net/

1.5 Jupyter Notebook

Jedná se o webové interaktivní prostředí pro vytváření stejnojmenného dokumentu, který je součástí Projektu Jupyter. Samotný notebook je kombinace Markdown[21] dokumentu formátovaným do JSON[22]. Zápis do notebooku je prováděn pomocí tzv. "buněk", které mohou obsahovat funkční kód mnoha jazyků, text, matematiku, grafy, obrázky. Dokumenty mají koncovku ".ipynb".

Seznam programovacích jazyků podporované Jupyter Notebookem - https://jupyter4edu.github.io/jupyter-edu-book/jupyter.html

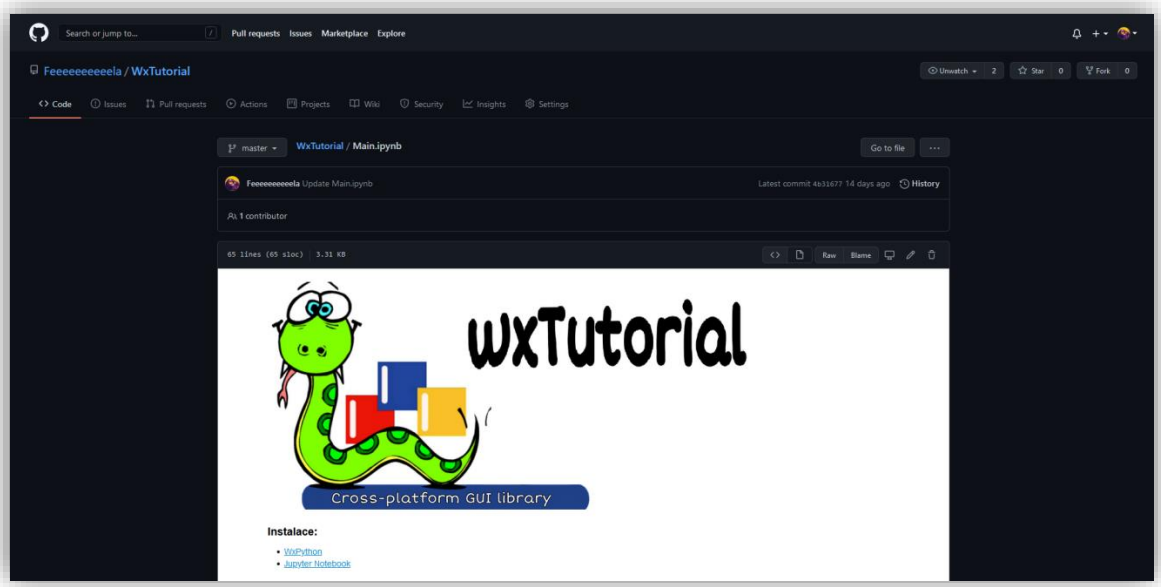
1.5.1 Historie Jupyter Notebooku

V roce 2014 vznikl Projekt Jupyter oddělením od IPythonu. IPython nadále existuje jako prostředí Pythonu, zatímco Notebook a další jazykově nezávislé části IPythonu se přesunuly pod názvem Jupyter.

V roce 2015 GitHub a projekt Jupyter oznámili nativní vykreslování formátu souborů notebooků Jupyter (soubory .ipynb) na platformě GitHub. (viz kapitola 1.9)

[21] Typ souboru podobný poznámkového bloku - https://en.wikipedia.org/wiki/Markdown

[22] Typ souboru slovníku - https://en.wikipedia.org/wiki/JSON



Obrázek č. 1 - Zobrazení souborů .ipynb na GitHubu

Zdroj: Můj osobní GitHub

1.6 Project Jupyter[23]

Projekt Jupyter je nezisková organizace vytvořená s motivy rozvinutí otevřeného softwaru pro interaktivní práci na počítači s desítkami programovacích jazyků. Jupyter vznikl oddělením od IPythonu v roce 2014, za kterým stál Fernando Pérez. Pod Projekt Jupyter spadají interaktivní výpočetní produkty Jupyter Notebook, JupyterHub a JupyterLab.

1.7 IPython[24]

IPython (Interactive Python) je příkazový shell[25] pro interaktivní výpočty ve více programovacích jazycích (původně pouze pro Python). Podporuje práci s multimédii[26], introspekci[27] (schopnost programu zkoumat typ a vlastnosti

[23] Hlavní zdroj Projektu Jupyter - https://en.wikipedia.org/wiki/Project_Jupyter

[24] Hlavní zdroj IPythonu - https://en.wikipedia.org/wiki/IPython

[25] Příkazový řádek pro práci s počítačem - https://en.wikipedia.org/wiki/Shell_%28computing%29

[26] Multimédia zahrnují typy souborů, které jsou určeny pro uživatele (obrázky, videa, animace, audio atd.) - https://en.wikipedia.org/wiki/Interactive_media

[27] Prozkoumávání typu nebo vlastností objektu v reálném čase při běhu programu- https://en.wikipedia.org/wiki/Type_introspection

objektu při běhu), nástroje pro paralelní výpočty[28] a rozhraní poznámkového bloku s podporou kódu, textu, matematických výrazů a dalších médií (podobně jako Markdown).

1.8 NbViewer

Jedná se o nový produkt Jupyter Notebooku, který vytváří webové rozhraní z veřejných URL[29] notebook dokumentu a převede ho do formátu HTML[30], který zobrazí jako webovou stránku.

Obrázek č. 2 - webové rozhraní NbViewer

Zdroj: screenshot z https://nbviewer.jupyter.org/

1.9 GitHub

GitHub je poskytovatel internetového hostingu[31] (webhostingu) určený pro vývoj softwaru pomocí systému pro distribuci Git. Hlavními funkcemi, kromě úložiště, je správa zdrojového kódu, systém aktualizací a verzí.

---

[28] Výpočet komplikovaných příkladů pomocí rozdělení na menší díly, které se vyřeší rychleji a jednodušeji - https://en.wikipedia.org/wiki/Parallel_computing

[29] Jednotná adresa zdroje (Uniform Resource Locator) pro orientaci v síti - https://en.wikipedia.org/wiki/URL

[30] Jazyk pro tvorbu webových stránek - https://en.wikipedia.org/wiki/HTML

[31] Pronájem prostoru na cizím serveru - https://en.wikipedia.org/wiki/Internet_hosting_service

GitHub své služby nabízí zdarma, avšak existují komerční profesionální verze.

Dříve bývaly zdarma pouze projekty s otevřeným zdrojovým kódem[32] , avšak od roku 2019 GitHub začal postupně rušit omezení pro neplacenou verzi. Nyní může uživatel mít neomezeně úložišť (tzv. repositářů), ať už veřejných, tak soukromých.

Aktuálně GitHub má přes 60 milionů uživatelů a více než 210 milionů úložišť, což ho činí největším hostitelem zdrojového kódu na Světě.

1.9.1 GIT[33]

Jedná se o nejpopulárnější systém pro správu verzí pro programátory při vývoji softwaru s GNU licencí.

Nástroj funguje formou kontroly verzí, ve které sleduje změny souborů v daných složkách.

Výhodami samotného Gitu je jeho rychlost, integrita dat[34] a podpora pracovních toků při vývoji (tzv. branche/větve).

1.10 OBJEKTOVĚ ORIENTOVANÉ PROGRAMOVÁNÍ

Tématem kapitoly je programovací paradigma OOP.

1.10.1 PROGRAMOVACÍ PARADIGMATA[35]

Programovací paradigma je způsob, jak rozdělit programovací jazyky. Paradigmata se zabývají samotným prováděním jazyka (styl psaní kódu). Rozdělují se na:

1. Imperativy - uživatel upravuje/instruuje samotné zařízení/stroj, tím mění jeho stav
   - Procedurální programování (Strukturované/imperativní programování): nejzákladnější/nejjednodušší paradigma, které

[32] Veřejně dostupný kód - https://en.wikipedia.org/wiki/Open_source

[33] Hlavní zdroj Gitu - https://en.wikipedia.org/wiki/Git

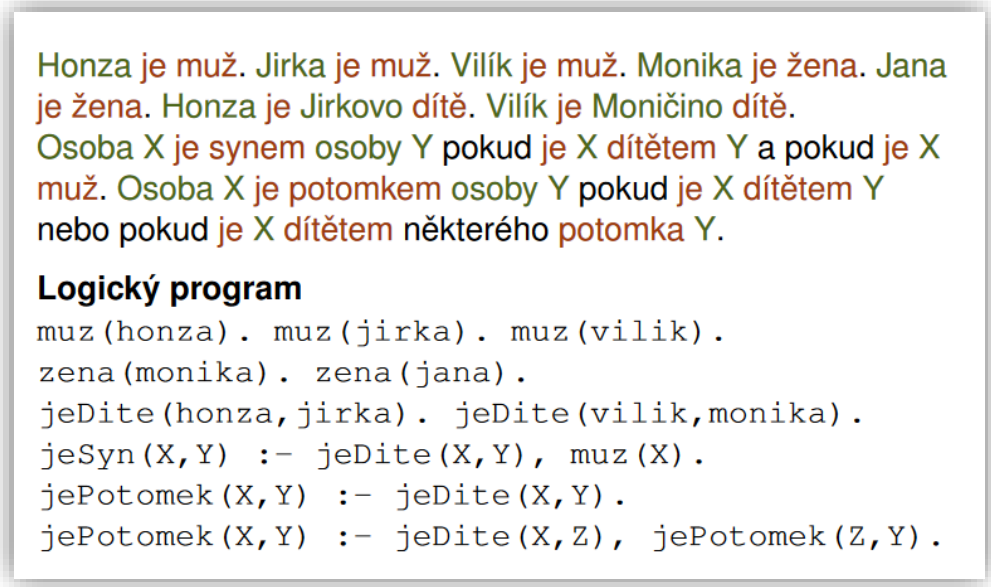
[34] Neboli jednotvárnost dat, která se snaží o aktualitu dat a zamezuje duplikace či zastaralá data - https://en.wikipedia.org/wiki/Data_integrity

[35] Hlavní zdroj Programovacích paradigmat - https://en.wikipedia.org/wiki/Programming_paradigm

seskupuje pokyny do postupů (jednoduše obsahuje řadu výpočtových kroků, které vedou k cílenému výsledku).

- Objektově orientované programování (OOP): paradigma je založené na konceptu tzv. "objektů" (viz 1.10).

2. Deklarativní - uživatel pouze upravuje vlastnosti požadovaného výsledku, ale ne způsob jeho výpočtu
   - Funkcionální programování: programy jsou konstruovány funkcemi, které se skládají do stromů či samotných výrazů, které vracejí hodnotu, podle které se mění stav programu.
   - Logické programování: je psán sadami vět v logické podobě, které řeší programové problémy (viz Obrázek č. 3 - Příklad logického programování).
   - Matematické programování: využívá mnohé způsoby matematických funkcí (lineární, celočíselné, kvadratické, dynamické programování...)



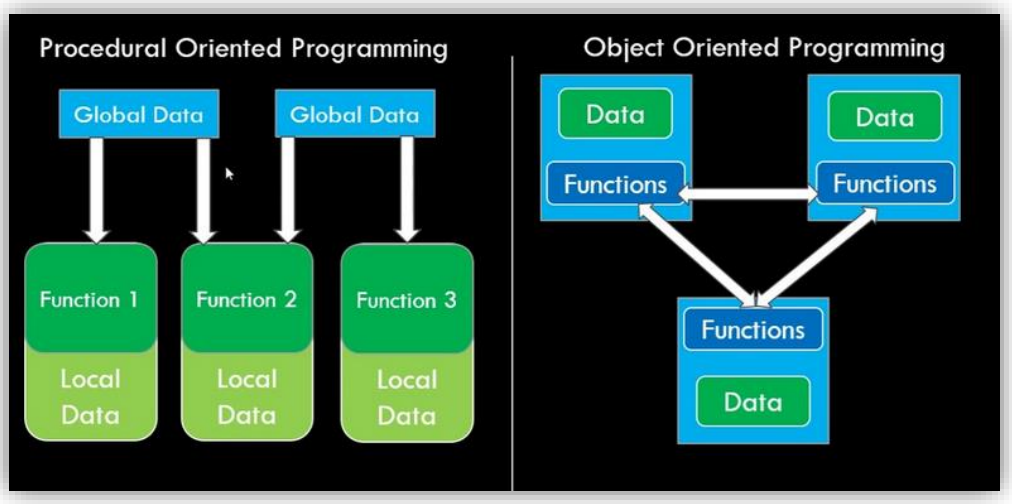
Obrázek č. 3 - Příklad logického programování

Zdroj: prezentace Logické programování - Tomáš Kühr: https://www.inf.upol.cz/downloads/ruzne/LPproSS.pdf

1.10.2 OOP[36]

OOP paradigma tedy funguje na principů objektů - objekty obsahují buď data (atributy, vlastnosti) nebo funkce (metody).

Funkce objektů spočívá v tom, že objekty mohou přistupovat a upravovat datová pole podle sebe sama (podle jednoho atributu se upraví atribut z jiného objektu). Tudíž objekty mezi sebou interagují. (viz - Obrázek č. 4 - Rozdíl mezi



proceduralním a objektově orientovaným programováním.)

Jazyky využívající OOP paradigma je mnoho. Mezi nejoblíbenější však patří ty, které jsou založeny na tzv. „třídách[37]“. Objekty jsou poté vytvářeny do tříd, kde se stávají tzv. jejich instancemi[38], které určují jejich datové typy.

Mnoho nejrozšířenějších programovacích jazyků (např. C++, Python, Java atd…) jsou založeny na více paradigmatech a podporují OOP ve větší či menší míře.

Mezi významné objektově orientované jazyky patří: Java, C++, C#, Python, R, PHP(pro weby), Visual Basic(Microsoft), JavaScript(pro weby), Ruby, Pearl, Object Pascal, Objective-C, Dart, Swift, Scala, Kotlin, Common Lisp, MATLAB atd…

Obrázek č. 4 - Rozdíl mezi procedurálním a objektově orientovaným programováním.

Zdroj: obrázek z videa Simple Snippets o OOP v Javě

[36] Hlavní zdroj OOP - https://en.wikipedia.org/wiki/Object-oriented_programming

[37] Jedná se o konstrukční prvek, který udává „předpis“ objektům, které jsou v ní - https://en.wikipedia.org/wiki/Class_%28computer_programming%29

[38] Jestliže se objekt nachází v dané třídě, stává se její instancí - https://en.wikipedia.org/wiki/Instance_%28computer_science%29

1.10.3 FUNKCIONALITA OOP V PYTHONU

V Pythonu se využívá tzv. dědičnost[39] pro opětovné použití kódu a rozšířenost ve formě tříd. V konceptu:

- Třídy - definice datového formátu a dostupné postupy pro daný typ objektu nebo třídy objektu
- Objekt - instance tříd (objekty dědí vlastnosti dané třídy)

Funkce v OOP jsou známe jako metody.

Proměnné jako pole[40], členy, atributy nebo vlastnosti.

Existují tedy:

- Proměnné třídy - data patří do třídy jako celku
- Proměnné/atributy instance - data patří pouze jednotlivým objektům
- Členské proměnné[41] - označujeme tak proměnné třídy nebo instance, které jsou definovány konkrétní třídou
- Metody třídy - patří do třídy jako celku a mají přístup pouze k proměnným a volaným vstupům třídy
- Metody instance - patří k jednotlivým objektům, mají přístup pouze k proměnným konkrétního objektu, na který jsou volány vstupy a proměnné třídy

[39] Dědění/přejímání funkcí a dat mezi třídami (nebo objekty) - https://en.wikipedia.org/wiki/Inheritance_%28object-oriented_programming%29

[40] Záznamy v tabulce zorganizovány do řádků - https://en.wikipedia.org/wiki/Field_%28computer_science%29

[41] Proměnná spojená čistě s objektem - https://en.wikipedia.org/wiki/Member_variable

2. PRAKTICKÁ ČÁST - WXTUTORIAL

Praktická část popisuje výhody využití a aplikaci zmíněných nástrojů v teoretické části a samotnou tvorbu tutoriálu.

2.1 PROSTŘEDÍ PYTHONU A GIT

Při tvorbě tutoriálu byly využity stabilní verze Pythonu 3.8.6 a 3.7.9.

Nejnovější verze 3.9.0 není zatím kompatibilní s wxPythonem (rok 2020).

Při studování knihovny wx jsem pracoval ve dvou prostředích a to hlavně v Atomu[42] a příležitostně ve Visual Studio Code[43] (VSCode).

Atom je svobodný IDE navržen týmem GitHubu roku 2014, jehož podstatnou součástí jsou stahovatelné balíčky od komunity upravující samotný editor. Atom má v sobě zabudovaný i Git, který jsem, s kombinací aplikací GitHub Desktop[44], aktivně při práci používal.

2.2 JUPYTER NOTEBOOK A NBVIEWER

Jupyter notebook byl jeden z doporučených nástrojů mým vedoucím pro prostředí tutoriálů během mého učení knihovny wx. Pro tutoriál je ideální díky své kombinaci markdownu s funkčním kódem, kde se kód dá velmi jednoduše okomentovat. Původní myšlenkou bylo, že by uživatel přímo využíval rozhraní Notebooku (nebo by se dodatečně vytvořila celá webová stránka pro tutoriál), avšak to se změnilo po listopadu roku 2020, kdy Jupyter Notebook zveřejnil novou službu NBViewer pro zobrazování souborů .ipynb skrze adresu veřejného repositáře na GitHubu.

Využití NBVieweru vedlo k úpravě tutoriálů tak, že bylo nutné přidat zobrazení výstupů, neboť NBViewer buňky s kódem zobrazuje, avšak je nedokáže spustit.

[42] Oficiální webové stránky Atom IDE - https://atom.io/
[43] Oficiální webové stránky VSCodu - https://code.visualstudio.com/
[44] Hlavní webová stránka GitHub Desktopu- https://desktop.github.com/

Obrázek č. 6 - Spustitelné buňky v rozhraní Jupyter Notebooku

Zdroj: Jupyter Notebook - úvod a instalace

Obrázek č. 5 - Zobrazení v NBVieweru

Zdroj: [NBViewer Jupyter Notebook]

Dalším šikovným využitím NBVieweru bylo vytvoření tzv. „menu“ pro navigaci v celém tutoriálu skrze .ipynb soubor obsahující odkazy na všechny soubory. [NBViewer menu]

2.3 Struktura Tutoriálů

Pro přehlednost byl tutoriál rozdělen do pěti sekcí.

2.3.1 Instalační soubory

Část obsahuje instalaci a krátký úvod do wxPythonu a Jupyter Notebooku. Tutoriály obsahují několik alternativ instalace, řešení nejčastějších chyb a praktický úvod.

1. WxPython
2. Jupyter Notebook

2.3.2 HLAVNÍ SADA TUTORIÁLŮ

Sada po sobě jdoucích tutoriálů popisující základní logiku/syntaxi knihovny wx. Do sady jsou zahrnuty i tutoriály OOP.

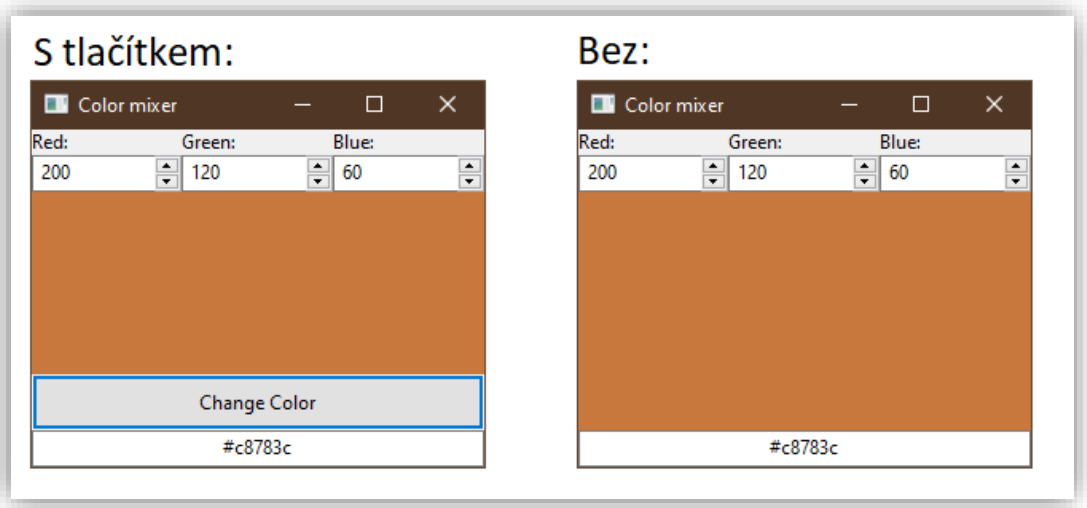
1. OOP Teorie - uvádí uživatele do Objektově Orientovaného programování
2. OOP Praxe - názorná ukázka práce s třídami a dědění mezi nimi
3. První program - popsání nejjednodušší aplikace ve wx v procedurálním programování
4. Přepsání do OOP a StaticText - Přepsání prvního programu do OOP s třídou wx.Frame
5. Velikost a pozice framu - Alternativy pro nastavování velikosti framu, funkce *SetSizeHints()* a pozicování na obrazovce
6. Tlačítko - Příklad vytvoření tlačítka a eventové funkce (konkrétně tzv. „exitbutton“)
7. Vstupy pro uživatele - V lekci se podobněji probere práce s *TextCtrl()*. Proberou se metody *GetValue()*/*SetValue()*. Do větší hloubky se rozeberou bindovací eventy. Ve vstupním dialogu je popsán další widget wx *TextEntryDialog()* a metoda *ShowModal()*
8. Menubar - v lekci je názorná ukázka vytvořené lišty
9. Třída wx.App - Další přepsání základního programu do OOP, kdy vytvoříme třídu instanci wx.App. (Jedná se o strukturu, která se aktivně používá v programátorské komunitě)
10. Sizery - práce s rozvržení prvků a jejich chování při změně velikosti okna

2.3.3 PŘÍKLADY

Názorné příklady do větší hloubky popisují práci se sizery, eventy a slouží uživateli pro inspiraci ať už z hlediska rozvržení, tak i logiky.

1. Mixér barev - Program přijímá 3 inputy (pro RGB) a převádí je do hex[45], která se využívá pro nastavování barvy v Pythonu. Soubor obsahuje 2 alternativy programu.

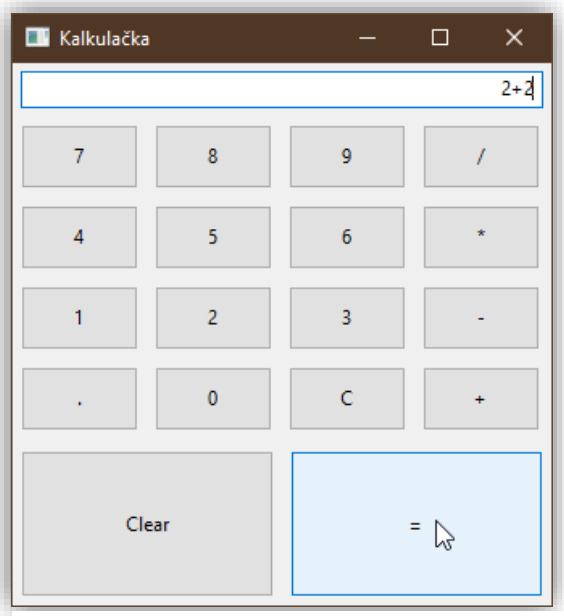
[45] Typ zápisu barvy v hexadecimální soustavě - https://en.wikipedia.org/wiki/Web_colors#Hex_triplet



Obrázek č. 7 - Mixér barev - náhled

Zdroj: Mixér barev

2. Kalkulačka - jednoduchá kalkulačka s logikou přes funkci *eval()*



Obrázek č. 8 - Náhled kalkulačky

Zdroj: Kalkulačka

3. Práce s více panely - propojující program předchozích 2 příkladů skrze lištu

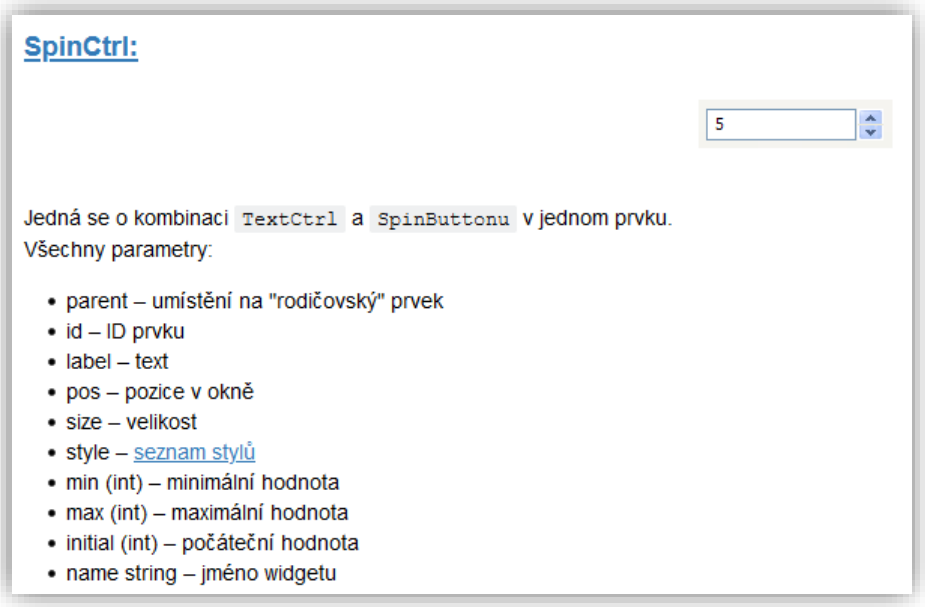
2.3.4 WxGlade

Část je rozdělena na instalaci/úvod a na názorný příklad mixéru barev.

1. Instalace a důvod - návod na instalaci a spuštění
2. Mixér barev - názorná ukázka práce s wxGlade krok po kroku

2.3.5 Základní widgety

Seznam widgetů - seznam všech wx widgetů (bez přídavných knihoven) s náhledy, parametry a s odkazy na oficiální dokumentaci.



Obrázek č. 9 - příklad ze seznamu widgetů

Zdroj: Seznam widgetů

Seznam byl čerpán z galerie wx widgetů z oficiální dokumentace

2.4 Odkazy pro studium wxPythonu

Při studiu knihovny wx jsem vycházel z mnoha tutoriálů a stránek, kterých je tolik, že není možné je všechny vypsat. Proto zmíním jen ty nejdůležitější:

Kromě oficiálních instalačních stránek s malým úvodem jsem se nejvíce v začátcích učil z tutoriálu ze ZetCode, který mi také vytvořil představu o struktuře tutoriálu.

Nejčastěji jsem vycházel z oficiálních stránek daných prvků na wxPython API dokumentace, který obsahuje vše, co je pro práci s knihovnou potřeba.

Pro studium tlačítka a lišty jsem využil tutoriálovou sérii od thenewboston - thenewboston - tlačítko, thenewboston - lišta

Při učení jsem si povšiml, že se často pro přehlednost využívá třída wx.App, na kterou jsem shlédl celý tutoriál - Johnyboycurtis - first app

Kdykoliv jsem byl v koncích ohledně nějaké chyby, tak jsem řešení většinou nacházel na Stackoverflow.

Pro pochopení a práci se sizery mi nejvíce pomohla vizualizace skrze aplikaci WxGlade.

ZÁVĚR

Vzhledem k rychlému, zbrklému vybrání témata z důvodu náhlé změny podmínek pro DPM, kvůli kterému jsem neměl čas kontaktovat firmu, se kterou jsem měl spolupracovat již při seminární práci. Téma mi bylo navrhnuto Ing. Markem Nožkou. Zadání bylo napsáno velmi obecně, tudíž jsem při práci věnoval velmi značné úsilí, abych splnil představy mého vedoucího práce.

Nejtěžší částí bylo vytvoření samotné základní struktury a pojetí tutoriálů, neboť jsem neměl se psaním tutoriálů, vůbec s knihovnou wx, žádné zkušenosti. Práce byla tudíž několikrát celá přepsaná a upravená, měnil se způsob zápisu a nebo i celá struktura. To má za následek přes 250 tzv. „commitů“ na GitHubu, kde se průběžné změny dají dohledat.

Dalším problémem také bylo, že při postupování ve vytváření tutoriálu začalo nabývat množství alternativních řešení pro dané problematiky, které závisí na konkrétních aplikování. Mým řešením byl pokus o co nejvíce obecné popsání a rozšíření o komplexní úlohy.

Velmi těžké také byla samotná interpretace tutoriálu, ať už odborné výrazy, gramatika nebo srozumitelná skladba vět.

Finálním produktem tedy je menší sbírka tutoriálů, která studentům vytvoří praktický základ pro jejich další rozvoj. Tutoriál je v téhle podobě velmi rozšiřitelný o další funkce a praktiky knihovny wx, např. práce s fonty, validatory a nebo další propojení s ostatními knihovnami (knihovny pro zobrazování grafů atd.).

SEZNAM POUŽITÉ LITERATURY

[1] Wikipedia contributors. (2021, February 19). Python (programming language). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:46, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&oldid=1007718549

[2] Wikipedia contributors. (2021, February 18). Programming language. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Programming_language&oldid=1007469992

[3] Wikipedia contributors. (2021, February 18). String (computer science). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:51, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=String_(computer_science)&oldid=1007410250

[4] Wikipedia contributors. (2021, January 13). List comprehension. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:51, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=List_comprehension&oldid=1000138931

[5] Wikipedia contributors. (2021, February 7). Associative array. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:53, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Associative_array&oldid=1005355786

[6] Wikipedia contributors. (2021, February 17). Modular programming. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:53, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Modular_programming&oldid=1007300521

[7] Wikipedia contributors. (2021, February 18). Anonymous function. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:53, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Anonymous_function&oldid=1007587353

[8] Wikipedia contributors. (2021, January 8). Map (higher-order function). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:54, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Map_(higher-order_function)&oldid=999131400

[9] Wikipedia contributors. (2020, November 5). Filter (higher-order function). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:54, February 21, 2021, from

https://en.wikipedia.org/w/index.php?title=Filter_(higher-order_function)&oldid=987173509

[10] Wikipedia contributors. (2021, January 3). Fold (higher-order function). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:54, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Fold_(higher-order_function)&oldid=998078750

[11] Wikipedia contributors. (2021, February 3). Higher-order function. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:54, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Higher-order_function&oldid=1004633268

[12] Wikipedia contributors. (2021, January 18). Named parameter. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:55, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Named_parameter&oldid=1001119037

[13] Wikipedia contributors. (2021, January 29). Free-software license. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:55, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Free-software_license&oldid=1003615687

[14] Wikipedia contributors. (2021, February 20). GNU General Public License. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:55, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=GNU_General_Public_License&oldid=1007872034

[15] Wikipedia contributors. (2021, February 4). Python syntax and semantics. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:56, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Python_syntax_and_semantics&oldid=1004753236

[16] Wikipedia contributors. (2021, February 7). Freeze (software engineering). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:57, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Freeze_(software_engineering)&oldid=1005313942

[17] Python team contributors. (2021, February 21). 2to3 - Automated Python 2 to 3 code translation. In *Python tm*. Retrieved 15:02, February 21, 2021, from https://docs.python.org/3/library/2to3.html

[18] Wikipedia contributors. (2021, February 18). WxPython. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:05, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=WxPython&oldid=1007527885

[19] Wikipedia contributors. (2020, October 29). SWIG. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:06, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=SWIG&oldid=985974151

[20] Wikipedia contributors. (2021, February 11). WxWidgets. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:06, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=WxWidgets&oldid=1006116530

[21] Wikipedia contributors. (2021, February 20). Markdown. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:09, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Markdown&oldid=1007872031

[22] Wikipedia contributors. (2021, February 5). JSON. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:10, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=JSON&oldid=1005074494

[23] Wikipedia contributors. (2021, February 20). Project Jupyter. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:11, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Project_Jupyter&oldid=1007949968

[24] Wikipedia contributors. (2021, February 18). IPython. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:12, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=IPython&oldid=1007521229

[25] Wikipedia contributors. (2021, January 11). Shell (computing). In *Wikipedia, The Free Encyclopedia*. Retrieved 14:12, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Shell_(computing)&oldid=999685044

[26] Wikipedia contributors. (2020, December 22). Interactive media. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:12, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Interactive_media&oldid=995757547

[27] Wikipedia contributors. (2020, December 31). Type introspection. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:12, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Type_introspection&oldid=997445542

[28] Wikipedia contributors. (2021, January 16). Parallel computing. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:12, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Parallel_computing&oldid=1000667608

[29] Wikipedia contributors. (2021, February 17). URL. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:13, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=URL&oldid=1007276500

[30] Wikipedia contributors. (2021, February 11). HTML. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:13, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=HTML&oldid=1006099685

[31] Wikipedia contributors. (2021, February 20). Internet hosting service. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:13, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Internet_hosting_service&oldid=1007898126

[32] Wikipedia contributors. (2021, February 4). Open source. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:15, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Open_source&oldid=1004805511

[33] Wikipedia contributors. (2021, February 19). Git. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:16, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Git&oldid=1007689366

[34] Wikipedia contributors. (2021, February 15). Data integrity. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:16, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Data_integrity&oldid=1006834899

[35] Wikipedia contributors. (2021, February 17). Programming paradigm. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:16, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Programming_paradigm&oldid=1007324887

[36] Wikipedia contributors. (2021, February 20). Object-oriented programming. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:17, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Object-oriented_programming&oldid=1007940260

[37] Wikipedia contributors. (2021, February 17). Class (computer programming). In *Wikipedia, The Free Encyclopedia*. Retrieved 14:17, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Class_(computer_programming)&oldid=1007287664

[38] Wikipedia contributors. (2020, December 31). Instance (computer science). In *Wikipedia, The Free Encyclopedia*. Retrieved 14:17, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Instance_(computer_science)&oldid=997342925

[39] Wikipedia contributors. (2021, February 11). Inheritance (object-oriented programming). In *Wikipedia, The Free Encyclopedia*. Retrieved 14:17, February 21, 2021, from

https://en.wikipedia.org/w/index.php?title=Inheritance_(object-oriented_programming)&oldid=1006197330

[40] Wikipedia contributors. (2021, January 16). Field (computer science). In *Wikipedia, The Free Encyclopedia*. Retrieved 14:17, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Field_(computer_science)&oldid=1000806287

[41] Wikipedia contributors. (2021, January 19). Member variable. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:18, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Member_variable&oldid=1001393302

[45] Wikipedia contributors. (2021, February 5). Web colors. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:18, February 21, 2021, from https://en.wikipedia.org/w/index.php?title=Web_colors&oldid=1005014083

SEZNAM OBRÁZKŮ A TABULEK

PŘÍLOHY

Příloha č. 1: Poster k maturitní práci.