

**Hodnocení:**

**Autor práce:** Václav Mašl, Matěj Černý, Filip Beneš

**Třída:** 4.I

**Vedoucí práce:** Alan Koukol

**Dne:**18.12.2023

**DLOUHODOBÁ MATURITNÍ PRÁCE S OBHAJOBOU**

Téma: Úniková hra

**Vyšší odborná škola   
a Střední průmyslová škola elektrotechnická   
Plzeň, Koterovská 85**

** Vyšší odborná škola**

**a Střední průmyslová škola elektrotechnická**

**Plzeň, Koterovská 85**

Zadání dlouhodobé maturitní práce

*Zde vložte originál zadání podepsaný ředitelem školy*

Anotace

**Účel anotace**

Hlavním účelem anotace je vytvořit vhodný text nebo také shrnutí, které bude sloužit pro snadné vyhledání v odborných katalozích či databázích. Ideální je tedy vložit do anotace vhodná klíčová slova, i když to není nutně podmínkou.

**Čeho se u anotací vyvarovat**

Do anotací rozhodně nepatří obrázky, tabulky, grafy, citace nebo obsáhlejší text složený z více odstavců. Text by měl být jasný a stručný, jednoduše a výstižně popisující obsah celé navazující práce. Nikdy by se neměl byť jen krátký text anotace rozdělovat do několika odstavců.

**Formální úprava anotace**

Anotace společně s titulním listem, zadáním, čestným prohlášením a obsahem mají číslování skryté, jedná se o tzv. neviditelné číslování stránek, ale přitom se tyto stránky započítávají.

**Hlavní účel anotace**

Pamatujte na to, že anotace by primárně měla nalákat potenciálního čtenáře a už na první dojem zaujmout tématem Vaší práce. Jde o stručné uvedení do problematiky, kterou jste si pro svou práci vybrali a které se budete věnovat.

**Jak správně napsat anotaci**

Nejprve začněte tím nejjednodušším, a to nadpisem. Anotaci uveďte nadpisem jako i ostatní kapitoly nebo úvod své práce. Nikdy nepište klíčové slovo do nadpisu, ta se píší pouze do vět. Do nadpisu tedy uveďte pouze jednoduše „Anotace“. Nadpis „Anotace“ není číslován.

Anotaci můžete uvést např. větou „Maturitní práce se zaměřuje na …“ nebo „Maturitní práce s názvem „Plaťte pokutu moderně“ se zabývá …“ a dále pak pokračovat informací o konkrétní problematice, které se budete věnovat. Můžete představit i jednotlivé kapitoly, to vše však jasně a stručně.

Tip: Anotaci nikdy nepište jako první. Nejlépe je anotaci napsat, až dopíšete svou maturitní práci. Ke tvorbě anotace samozřejmě potřebujete mít potřebné podklady, ze kterých budete vycházet.

**Nezapomeňte na klíčová slova**

Klíčová slova se vyplatí nevynechat, i když to není, co se anotace týče, povinný údaj. Vyberte nejlépe 5 až 8 slov, která budou Vaší práci nejlépe vystihovat, a zakomponujte je vhodně do textu.

Klíčová slova

Klíčová slova tvoří výčet slov na jednom řádku a jsou oddělena čárkami a měla by být seřazena podle abecedy.

C#, online kurz, programování, …

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil(a) literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

„Souhlasím s využitím mé práce učiteli VOŠ a SPŠE Plzeň k výuce.“

V Plzni dne: …..................... Podpis: ….....................................

Obsah

[Zadání dlouhodobé maturitní práce 7](#_Toc161987801)

[Anotace 8](#_Toc161987802)

[Klíčová slova 9](#_Toc161987803)

[Obsah 6](#_Toc161987804)

[Úvod 7](#_Toc161987805)

[1 Vývojový deník 8](#_Toc161987806)

[2 Uživatelská příručka 9](#_Toc161987807)

[3 Části projektu 10](#_Toc161987808)

[3.1 Popis funkčností hry 10](#_Toc161987809)

[3.2 Front end aplikace 10](#_Toc161987810)

[3.3 Back end aplikace 10](#_Toc161987811)

[3.4 Inventář 10](#_Toc161987812)

[3.4.1 Sbírání itemů 10](#_Toc161987813)

[3.4.2 Předávání mezi sloty 11](#_Toc161987814)

[3.4.3 Crafting systém 11](#_Toc161987815)

[3.4.4 Vracení předmětů z inventáře do scény 11](#_Toc161987816)

[3.5 UI 12](#_Toc161987817)

[3.6 Grafika a design 12](#_Toc161987818)

[3.7 Minihry 12](#_Toc161987819)

[3.7.1 Minihra1 12](#_Toc161987820)

[3.7.2 Minihra2 12](#_Toc161987821)

[3.7.3 Minihra3 12](#_Toc161987822)

[3.7.4 Minihra4 12](#_Toc161987823)

[3.8 Zvuky 13](#_Toc161987824)

[3.9 Systém Ukládání 14](#_Toc161987825)

[3.10 ScriptableObject jako úložiště 15](#_Toc161987826)

[3.11 Prezentace projektu 16](#_Toc161987827)

[3.12 Dokumentace aplikace 16](#_Toc161987828)

[3.13 Další části odpovídající zadání projektu 16](#_Toc161987829)

[4 Technická dokumentace 17](#_Toc161987830)

[Závěr 18](#_Toc161987831)

[Literatura 19](#_Toc161987832)

[Seznam obrázků a tabulek 21](#_Toc161987833)

[Seznam tabulek 21](#_Toc161987834)

[Seznam obrázků 22](#_Toc161987835)

[Seznam grafů 23](#_Toc161987836)

Úvod

* Motivace pro realizaci projektu
* Popis použitých technologií
* Souhrn existujících řešení a porovnání s vlastním projektem
* Slovníček pojmů
* Popis napojení na třetí strany

# Vývojový deník

Chronologický popis postupu vývoje aplikace během realizace projektu. Možno vložit Sprint protokoly.

# Uživatelská příručka

* Návod, jak použít základní funkce systému
  + z pohledu všech uživatelských rolí
  + z pohledu administrátora
* Řešení nejčastějších problémů
* Odkazy na vyvíjenou aplikaci
  + server, stažení aplikace
* Přístupy do administrátorské části

# Části projektu

Obsah této části bude odpovídat konkrétnímu zadání projektu. Měl by obsahovat rozdělení celé práce na jednotlivé části včetně delegování částí na členy týmu. Musí být jasně specifikováno kdo co dělal.

## Popis funkčností hry

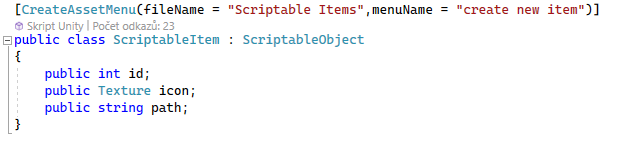
## Front end aplikace

## Back end aplikace

## 

## Inventář

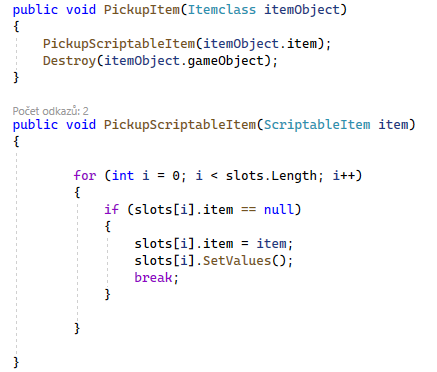
Náš inventář funguje jako pole jednotlivých slotů. Každý slot má atributy item a image kdy item je ScriptableObject který hodnotami reprezentuje daný objekt a image je RawImage která reprezentuje Item vizuálně v inventáři.



Dále je zde důležitá metoda SetValues() která zajistí promítnutí z ScriptaleItem.icon do image daného slotu. Díky tomu tak sloty můžou logicky nabývat jiných itemů než vizuálně a tyto hodnoty se pak synchronizují až touto metodou.

### Sbírání itemů

Sbírání itemů funguje následovně. GameObject item má na sobě script Itemclass kde je přiřazený ScriptableItem odpovídající danému itemu. Při sebrání se spustí metoda PickupItem() která zničí gameObject a zároveň ScriptableObject itemu předá do invetáře. Tam je přidělen nejbližšímu volnému slotu a pomocí metody SetValues() následně zobrazen. Při sebrání je také item odebrán z úložiště scény.

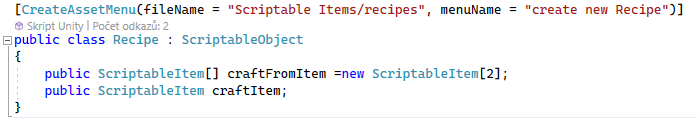


### Předávání mezi sloty

Předávání předmětů mezi sloty funguje následovně. Při započnutí tažení ze slotu se do Atributu DraggedfromSlot dosadí hodnota slotu ze kterého je taženo. SlotClass Implementuje rozhraní IDropHandler a díky tomu můžeme za pomocí metody onDrop() získat data o upuštěném objektu. V tomto momentě tak stačí z Dropped objektu získat atribut DraggedfromSlot a máme odkaz na slot ze kterého bylo taženo i slot do kterého bylo puštěnoPak už stačí těmto dvěma slotům jen vzájemně prohodit jejich itemy. Z tohoto důvodu jsou itemy public atributy. Pak jen opět u obou slotů zavolat metodu SetValues() a prohození už bude viditelné i vizuálně. Kromě toho se při tažení předmětu změní Parent dané Image a lze s ní pohybovat i mimo daný slot. To vytvoří hezký vizuální kdy item následuje při tažení myš a po puštění se navrátí na svou pozici.

### Crafting systém

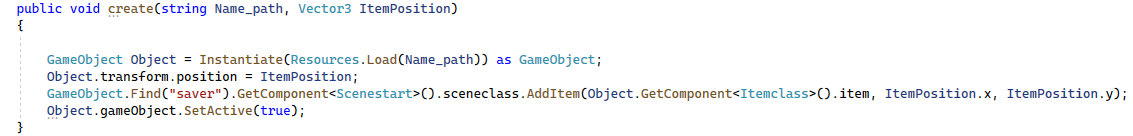
Vytváření nových předmětů funguje na základě spojení dvou předmětů v jeden. To funguje na principu receptů které jsou ScriptableObjecty.



Crafting Systém obsahuje pole těchto receptů. Při každém položení předmětu do jednoho ze vstupních slotů se otestuje jestli kombinace předmětů ve vstupních slotech neodpovídá nějaké kombinaci z receptů, pokud ano vytvoří ve výstupním slotu výstupní item získaný z příslušného receptu. Zároveň je ošetřeno aby při sebrání předmětu z výstupního slotu zmizely předměty ze vstupních slotů a nevznikalo tak nekonečně mnoho předmětů. Podobným stylem jsou ošetřeny i vstupní sloty .

### Vracení předmětů z inventáře do scény

Vracení předmětů funguje tak že v momentě kdy uživatel pustí předmět v prostoru výstupního slotu (pravá horní část inventáře) tak je daný předmět inicializován ve scéně z prefabu za pomocí metody Instantiate() kdy cestu na daný Item získá ze ScriptableItemu který získá přes onDrop() obdobě jako při předávání předmětů mezi sloty. Pak se jen tomuto předmětu nastaví pozice na které byl upuštěn. A je přidán do uložiště scény.



## UI

## Grafika a design

## Minihry

### Minihra1

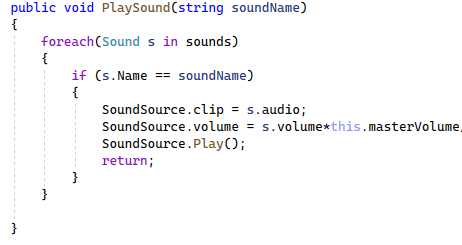
### Minihra2

### Minihra3

### Minihra4

## Zvuky

Zvuky v naší aplikaci jsou řešeny pomocí objektů, které spravují přehrávání zvuků, nazývaných AudioManagery. Máme dva druhy těchto AudioManagerů, univerzální a unikátní. Oba tyto objekty fungují podobně až na pár výjimek. AudioManager má přiřazeno pole objektů zvuků, které mají definované různé atributy, jako například název, odchylku od hlasitosti a danou zvukovou stopu. AudioManager má právě jeden zdroj zvuku (AudioSource), do kterého se po každém zavolání metody play(zvuk) dosadí příslušná zvuková stopa a další argumenty (hlasitost, název atd). Proto se tak nedají v jednu chvíli překrývat dva zvuky ze stejného AudioManageru. Přehrávání hudby funguje na naprosto stejném principu jako přehrávání zvuků, s tím rozdílem, že se zvuková stopa přehrává neustále dokola.

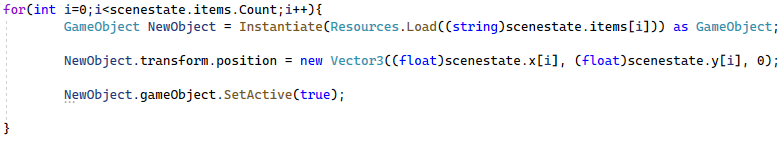


Hlavním rozdílem mezi univerzálním a unikátním AudioManagerem je obsažení zvuků. Univerzální obsahuje zvuky, které se používají v UI vrstvě a které se vyskytují ve více scénách. Univerzální AudioManager se přenáší mezi scénami a je proto jen jeden v celém projektu. Také se používá k přehrávání hudby a zajišťuje tak, aby vždy hrála jen jedna hudba a nepřeskakovala při přepínání mezi jednotlivými scénami. Unikátní AudioManager je v každé scéně právě jeden, a zvuky, které spouští, jsou tak unikátní pro danou scénu. Také nedokáže spouštět hudbu. Univerzální AudioManager je potomkem unikátního, který rozšiřuje. Zajímavá je také synchronizace jednotlivých AudioManagerů a jejich hlasitosti, kterou můžeme měnit. Když totiž změníme hlasitost, změna se může promítnout pouze do AudioManagerů, které existují v momentální scéně. Proto máme metody PullSynchronizeVolume() a PushSynchronizeVolume(), které vždy při vytvoření nového unikátního AudioManageru (při přepínání mezi scénami) synchronizují hlasitost nově vytvořeného objektu s Univerzálním AudioManagerem. Zároveň tyto metody zajišťují při změně hlasitosti i změnu hlasitosti všech momentálně existujících AudioManagerů

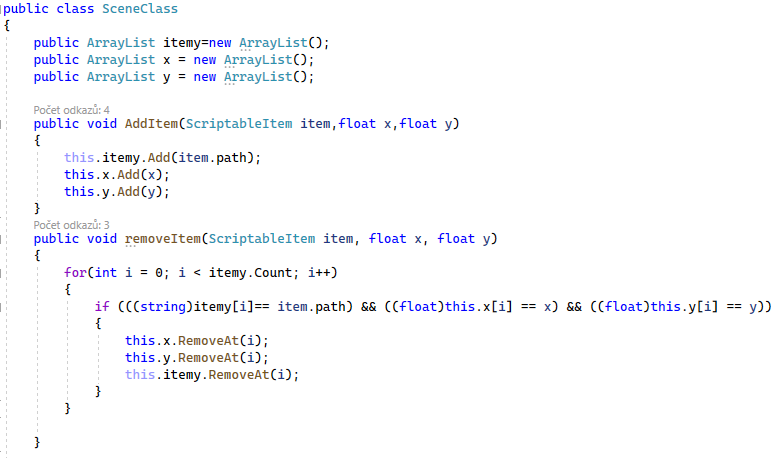
Testování aplikace

## Systém Ukládání

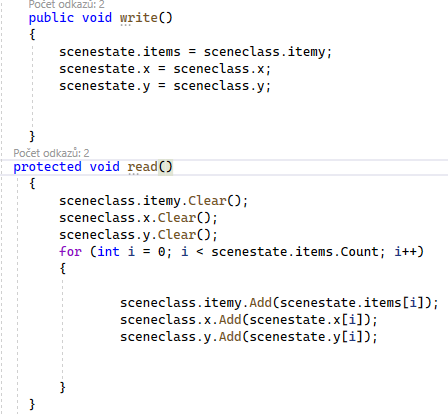
V Naší Aplikaci nemáme pevné ukládání dočasné (při vypnutí aplikace nezůstane nic uloženo). Dočasné ukládání za běhu aplikace využíváme především při přechodu mezi scénami. To zajištuje aby při odchodu a následném návratu do místnosti byla místnost ve stejném stavu jako po odchodu. K vysvětlení toho jak náš systém ukládání funguje je třeba nejdříve říct, že naše ukládání nefunguje na principu ukládání stavů ale spíše jednotlivých atributů a vlastností. Téměř veškeré objekty se kterými uživatel jakkoliv interaguje nejsou přímo staticky ve scéně. Při načtení scény se na objektu saver spustí script sceneStart a ten zajistí že se na scéně vytvoří všechny tyto objekty. Ty se inicializují z prefabů pomocí metody Instantiate(). Objekty k inicializeci získáme ze ScriptableObjektu dané místnosti které zde používáme k dočastnému ukládání.



U vytvořených objektů rovnou nastavíme jejich pozici kterou jsme získali taktéž z našeho ScriptableObjekt úložiště. Díky tomu že tyto objekty nejsou staticky v místnosti ale jejich vytvoření je závislé na datech z úložiště se nám tak ušetří spousta problémů které by jsme měli kdybychom k ukládání přistupovali jako ke stavům místnosti.



Třída SceneClass nám skvěle ukazuje jak jednoduše se teď dá k ukládání přistupovat. K přidání objektu stačí pouze zavolat metodu AddItem() s a jako argumenty dát daný item a jeho pozici. Jeho odebrání je stejně snadné. Díky tomu tak můžeme jednoduše a přehledně ukládat i pozice jednotlivých Objektů. Takže například když necháme v místnosti pod postelí a poté se do ní vrátíme nalezneme ho na stejném místě kde jsme ho nechali a ne na místě kde je jeho výchozí pozice. Tyto metody můžeme volat z jakéhokoliv skriptu ze kterého potřebujeme a můžeme tak dočasně ukládat na základě libovolných událostí. Toto je ovšem jen Třída pro ulehčení práce s ukládáním samotné ukládání se probíhá v metodě write() kde se ArrayListy Třídy SceneClass uloží do Arraylistů ScriptableObjektu dané třídy. Metoda read() funguje naopak(Naplní SceneClass hodnotami ze ScriptableObjektu).Tyto metody jsou volány vždy jen při změně scény aby nedocházelo k neustálému zápisu do ScriptableObjektu ale pouze v případech kdy je zápis relevantní. Každá scéna má vlastní objekt saver který zajištuje inicializaci itemů a jejich ukládání.



## ScriptableObject jako úložiště

ScriptableObject je datový kontejner, který můžete použít k uložení dat nezávisle na instancích třídy. Jedním z hlavních případů použití ScriptableObjects je snížení využití paměti projektu tím, že se vyhnete kopírování hodnot. Data, která uložíte z Editor Tools do ScriptableObjects jako aktivum, jsou zapsána na disk, a proto jsou mezi relacemi trvalá. ScriptableObject tak slouží skvěle pro naše účely vzhledem k tomu , že drží své hodnoty mezi relacemi ale po ukončení aplikace se zárověň vrátí do výchozího stavu. Proto jsme se je rozhodli je upřednostnit před pevným ukládáním např. do .json nebo .csv souborů. Se ScriptableObjekty se také přívětivěji pracuje v unity Inspectoru. Tyto objekty v našem projektu ovšem nepoužíváme jen jako uložiště ale třeba i pro reprezentaci jednotlivých Itemů kdy mají ScriptableObjekty pevná neměnná data. Toho využíváme především v UI vrstvě v inventáři, jelikož ScriptaleObject narozdíl od Monobehavior nevyžaduje připojení na GameObject. Díky tomu s nimi můžeme v inventáři operovat nezávisle na jakémkoliv GameObjectu.

## Prezentace projektu

## Dokumentace aplikace

## Další části odpovídající zadání projektu

# Technická dokumentace

* Identifikovat v aplikaci kritické procesy z pohledu bussinesu
* Popis kódu zajišťující vybrané procesy (třídy, metody, DB, …)
* Zabezpečení aplikace

Závěr

* Vlastní zhodnocení průběhu práce
* Důvody změn, které nastali při realizaci oproti zadání
* Důvody, proč nebyly realizovány některé části
* Možnosti rozšíření produktu
* Info o použití AI

Literatura

Azure Machine Learning. Azure.microsoft.com [online]. [cit. 2022-03-25]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-is-machine-learning-platform/><https://azure.microsoft.com/cs-cz/services/machine-learning/>

Bill Connelly Principal Component Analysis(PCA) For Dummies [online]. [cit. 2022-03-11]. Dostupné z: <http://www.billconnelly.net/?p=697>

Dimenzionalita dat. [online]. [cit. 2022-03-11]. Dostupné z: <https://www.statisticshowto.com/dimensionality/>

Dokumentace Pandas – Metody korelace [online]. [cit. 2022-03-17]. Dostupné z: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html>

FastAPI. *Fastapi* [online]. [cit. 2022-03-24]. Dostupné z: <https://fastapi.tiangolo.com>

FastAPI. *Uvicorn* [online]. [cit. 2022-03-24]. Dostupné z: <https://www.uvicorn.org>

Fyzikální veličina. Wikipedia [online]. [cit. 2022-03-11]. Dostupné z: <https://cs.wikipedia.org/wiki/Fyzik%C3%A1ln%C3%AD_veli%C4%8Dina>

GitLab [online]. [cit. 2022-03-11]. Dostupné z: <https://www.gitlab.com/>

Korelace. ExcelTown.com [online]. Praha: ExcelTown, 2020 [cit. 2022-03-11]. Dostupné z: <https://exceltown.com/navody/pokrocila-analyza-regrese-korelace/korelace-co-to-vlastne-je/>

Lineární regrese. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2022-04-06]. Dostupné z: <https://cs.wikipedia.org/wiki/Line%C3%A1rn%C3%AD_regrese>

Logistická regrese. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2022-04-06]. Dostupné z: <https://cs.wikipedia.org/wiki/Logistick%C3%A1_regrese>

Microsoft Teams. Microsoft [online]. [cit. 2022-03-11]. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-teams/group-chat-software>

Pandas.pydata.org [online]. [cit. 2022-03-11]. Dostupné z: <https://pandas.pydata.org/>

Pearsonův korelační koeficient. Wallstreetmojo [online]. 2019 [cit. 2022-03-11]. Dostupné z: <https://www.wallstreetmojo.com/pearson-correlation-coefficient/>

Prediction. DataRobot [online]. [cit. 2022-04-07]. Dostupné z: <https://www.datarobot.com/wiki/prediction/>

Principal Component Analysis Step-by-step. Builtin [online]. 2021 [cit. 2022-03-11]. Dostupné z: <https://builtin.com/data-science/step-step-explanation-principal-component-analysis>

PyCharm [online].  [cit. 2022-03-11]. Dostupné z: <https://www.jetbrains.com/pycharm/>

Scikit-learn.org [online]. [cit. 2022-03-11]. Dostupné z: <https://scikit-learn.org/stable/getting_started.html>

SOAP. Wikipedia [online]. [cit. 2022-03-24]. Dostupné z: <https://cs.wikipedia.org/wiki/SOAP>

Spotipy.readthedocs.io [online]. [cit. 2022-03-11]. Dostupné z: <https://spotipy.readthedocs.io/en/2.19.0/>

The beginner’s guide to Scrumban. Monday.com [online]. Moira Alexander, 2019 [cit. 2022-03-25]. Dostupné z: <https://monday.com/blog/rnd/the-beginners-guide-to-scrumban/>

Unit testing. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2022-03-11]. Dostupné z: <https://cs.wikipedia.org/wiki/Unit_testing>

What is CRISP DM? *Data Science Process Alliance* [online]. [cit. 2022-03-20]. Dostupné z: <https://www.datascience-pm.com/crisp-dm-2/>

XML-RPC. Wikipedia [online]. [cit. 2022-03-24]. Dostupné z: <https://cs.wikipedia.org/wiki/XML-RPC>

Seznam obrázků a tabulek

Seznam tabulek

[Tabulka 1 - Tabulka pojmů 6](#_Toc101432684)

[Tabulka 2 - Stavové kódy 16](#_Toc101432685)

[Tabulka 3 - Regulární výrazy 18](#_Toc101432686)

[Tabulka 4 - Rozdíl mezi hodnotami 18](#_Toc101432687)

[Tabulka 5 - Informace o skladbě 26](#_Toc101432688)

Seznam obrázků

[Obrázek 1 - Ukázka použité nástěnky 8](#_Toc101376658)

[Obrázek 2 - Ukázka webového repositáře 8](#_Toc101376659)

[Obrázek 3 - Graf vzdáleností prvků od přímky 11](#_Toc101376660)

[Obrázek 4 - Graf hlavních komponent 11](#_Toc101376661)

[Obrázek 5 - Ukázka metody \_get\_data() 16](#_Toc101376662)

[Obrázek 6 - Ukázka medoty \_transform\_data() 17](#_Toc101376663)

[Obrázek 7 - Ukázka metody \_present\_data() 18](#_Toc101376664)

Seznam grafů

Graf 1 – Produktivita práce 8

Graf 2 – Poměr vojáků 8

Graf 3 – Stav lesů ve 20 století 11