

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

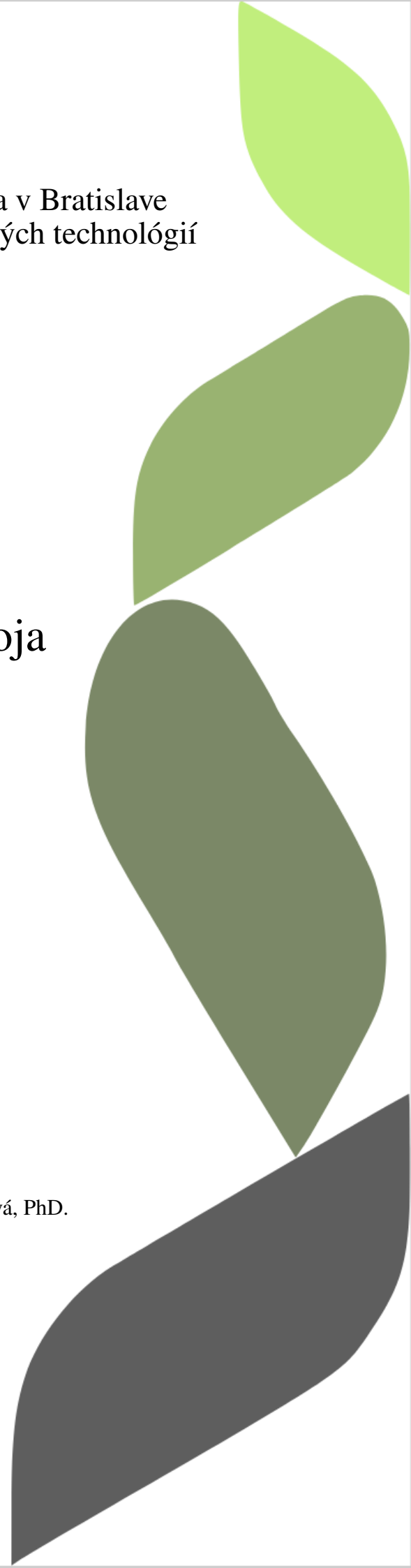
Tím č.5

# Metodika vývoja aplikácie

Vedúci tímu: Mgr. Martin Sabo, PhD., Ing. Marta Prnová, PhD.

Vypracoval: Jakub Kučečka

Kontakt: [mikasa.fiit@gmail.com](mailto:mikasa.fiit@gmail.com)



# 1 Manipulácia s aplikáciou

---

Pre zefektívnenie a zachovanie rovnorodosti práce na projekte sú v tomto dokumente definované pravidlá manipulácie s aplikáciou.

Odporúčaným IDE pre písanie programov v jazyku Python je Pycharm. Zvoleným jazykom pre vytváranie pomenovaní je anglický jazyk. Na integráciu medzi Python-om a JavaScript-om budeme používať Eel

## Názvoslovie pre súbory a adresáre

Všetky priečinky budú pomenované: *v anglickom jazyku* Keďže aplikácia je vyvíjaná pomocou technológií Python, JavaScript, CSS a HTML, jednotlivé súbory sa budú pomenovávať v závislosti od ich obsahu:

- **Python triedy.** V prípade, že obsahom súboru bude definícia triedy, súbory sa pomenujú cez *camelCase*, s výnimkou veľkého začiatočného písmena. Napríklad trieda, ktorá definuje príspevky pre používateľov sa bude volať *Sample*.
- **Python scripty.** Ak obsahom súboru nie sú triedy, súbory budú pomenované cez *lower\_case*, t.j. všetky slová v názve budú začínať malým začiatočným písmenom (okrem štartovacieho scriptu) a budú oddelené cez podčiarkovník. Napríklad, pre funkcie upravujúce reťazce, by bol názov súboru *ASR\_app*.
- **JavaScript scripty.** Rovnako ako pri Pythone, všetky súbory budú pomenované *lower\_case*, t.j. všetky slová v názve budú začínať malým začiatočným písmenom a budú oddelené cez podčiarkovník.
- **CSS.** Súbory budú začínať slovom "stylesheet" nasledované podčiarkovníkom a ľubovoľným *lower\_case* pomenovaním.
- **HTML.** Aplikácia bude obsahovať iba jeden HTML súbor a to *index.html*, ten musí byť umiestnený v *Application/public/index.html*.

## Adresáre v projekte Application

Pre zefektívnenie, sprehľadnenie a zjednodušenie práce na projekte budú v projekte Application existovať nasledujúce adresáre:

- **eel**. Súbory potrebné pre fungovanie knižnice eel.
- **lib**. Scripty pre správy a jazyky.
  - **Classification**. Pôvodné súbory klasifikácie.
- **log**. Adresár pre ukladanie logov za behu programu.
- **public**. Obsahuje súbory rôznych typov, ako napríklad .css, .js, .html a ďalšie.
  - **images**. Adresár pre obrázky potrebné pre aplikáciu, teda ikony a obrázky na pozadí.
  - **models**. Adresár pre natrénované modely, načítajú sa pri štarte aplikácie.
  - **tmp**. Adresár pre dočasné súbory.
    - \* **featureCSV**. Adresár pre CSV súbory určené pre klasifikáciu.
    - \* **imageCSV**. Adresár pre CSV súbory určené pre tvorbu obrázkov.
    - \* **images**. Adresár pre vytvorené PNG obrázky.

## 2 Písanie programov

---

### Hlavička dokumentu - Importy a globálne premenné

Všetky globálne importy sa budú nachádzať na začiatku súboru. Prvý import sa umiestni na riadok č.1. V prípade že sa za importom globálne premenné:

- nachádzajú. Nasledujú za importom 1 prázdny riadok.
- nenachádzajú. Nasledujú za importom 2 prázdne riadky.

Následne budú umiestnené deklarácie globálnych premenných. Názvy globálnych premenných budú napísané veľkými písmenami, oddelených podčiarkovníkom. Napríklad globálna premenná pre maximálnu hodnotu bude označená ako *MAX\_VALUE*. Príklad je uvedený na obrázku č. 1

```
import csv

MAX_VALUE = 1000000

def function:
    # some code to add
```

*Obr. 1: Príklad deklarácie hlavičky súboru*

### Telo dokumentu

#### Funkcie

Medzi funkciami budú 2 voľné riadky.

V prípade, ak sa jedná o spustiteľný súbor, je povinná funkcia main v tvare:

```
if __name__ == "__main__":
```

#### Podmienky

Uprednostnenie *or* a *and* pred `||` a `&&`.

Pre podmienky *if*, *elif*, *else* bude blok kódu začínať vždy na novom riadku.

#### Cykly

Blok kódu začínať vždy na novom riadku.

## Polia s kľúčovou hodnotou

Pri deklarácii polí s kľúčovou hodnotou sa odporúča nepridávať čiarku na koniec posledného riadku. Ukážka je uvedený na obrázku č. 2

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

Obr. 2: Príklad deklarácie polí s kľúčovou hodnotou

## Refazce

Deklarácie refazcov budú primárne začínať a končiť dvojitou úvodzovkou ”.

## JSON

Objekty JSON budú obklopené zloženými zátvorkami {}, zapisujú sa do párov kľúč - hodnota. Kľúče musia byť refazce a hodnoty musia byť platným typom údajov JSON (refazec, číslo, objekt, pole, logická hodnota alebo nulová hodnota). Kľúče a hodnoty sú oddelené dvojbodkou. Každý pár kľúč - hodnota je oddelený čiarkou.

## Tvorba pomenovaní

### Python premenné

Názvy premenných v programe sa budú písať cez lower\_case. Názvy premenných by nemali obsahovať slovesá, preferované sú podstatné mená. Odporúča sa používať jednotné číslo, s malými výnimkami.

### JavaScript premenné

Názvy premenných v programe sa budú písať cez camelCase. Názvy premenných by nemali obsahovať slovesá, preferované sú podstatné mená. Odporúča sa používať jednotné číslo, s malými výnimkami.

## **Python funkcie**

Názvy funkcií v programe sa budú písať cez `lower_case`. Názvy funkcií by mali obsahovať kombinácie slovesa a podstatného mena. Odporúča sa používať jednotné číslo, s malými výnimkami.

## **JavaScript funkcie**

Názvy funkcií v programe sa budú písať cez `camelCase`. Názvy funkcií by mali obsahovať kombinácie slovesa a podstatného mena. Odporúča sa používať jednotné číslo, s malými výnimkami.

## **Integrácia Python vs. JavaScript**

Pri vývoji aplikácie pomocou jazykou Python a JavaScriptom pomocou rozhrania Eel je spojených zopár zaujímavostí. Ak deklarujeme funkcie v jednom či druhom jazyku a chceme ich použiť v inom, pred názvom funkcie použijeme deklaráciu `@eel.expose`. Dôležité je ešte načítanie premenných krížom cez jazyky. Nevieme si výstup z Python scriptu uložiť do premennej v JavaScripte. Odporúča sa teda zavolať Python funkciu v `init` funkcii a poslať ju do konkrétnej JavaScript funkcii ako vidíme na obr. 3.

## Python

```
41 @eel.expose
42 def get_lang(language):
43     set_conf("lang", language)
44
45     if language == 'sk':
46         return lang.svk_lang()
47     elif language == 'en':
48         return lang.eng_lang()
49     else:
50         print_log("Unknown language " + language)
51         return lang.eng_lang()
52
53
```

## JavaScript

```
3     function init(lang) {
4         eel.get_lang(lang) (initScene);
5     }
...
...
30 function initScene(lang) {
31
32     setLang(lang);
33
34     document.body.style.width = window.innerWidth + 'px';
35     document.body.style.height = window.innerHeight + 'px';
36     document.getElementById("disableContent").style.width = window.innerWidth + 'px';
37     document.getElementById("disableContent").style.height = window.innerHeight + 'px';
38
39     eel.get_conf("theme") (setTheme);
40 }
```

Obr. 3: Príklad integrácie funkcie `get_lang()` medzi Pythonom a JavaScriptom

Dict Python objekt bude prekonvertovaný na JSON pomocou funkcie `json.dump()`. V JavaScripte bude párovať daný objekt funkciou `JSON.parse()`, tá ho zmení na objekt. K jednotlivým údajom pristupujeme pomocou `[]` (napr. `obj["key"]`). Naopak JSON objekt netreba preložiť na Python dict (napr. `let data = "models": models, "samples": samples` pošleme priamo ako `data` do Python funkcie).

## CSS

V CSS súboroch sú použité premenné definované v `:root{}`, použité sú pomocou `var(meno premennej)`. Názvy tried a identifikátory sú písane camelCase. Rovnaké atribúty vyberáme do samostatných tried a tie používame pre konkrétne komponenty. U url

sú použité relatívne cesty.

Jednotlivé css komponenty sú oddelené voľným riadkom a súbory nie sú minifikované kvôli prehľadnosti. Jednotlivé atribúty CSS komponentu sú na začiatku oddelené tabulátorom.

## HTML

Triedy a id sú písane camelCase. Súbor musí obsahovať `;!DOCTYPE html;` a samotný `html` tag. V `head` tagu sú definované cesty k `stylesheet` a `js` súborom, tiež je zadaná ikona aplikácie a jej názov.

`Body` tag obsahuje sekciu `nav` nasledovanú sekciou `div` s `id = content`. Daný `div` bude fungovať ako dynamický komponent. Jeho obsah sa zobrazí a skrýva podľa potreby.



## A História

---

<b>Meno a Priezvisko</b>	<b>Dátum</b>	<b>Opis zmeny</b>
Jakub Kučečka	03.03.2021	Vytvorenie metodiky
Alena Valová	06.03.2021	Upravenie listov
...	...	...