

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

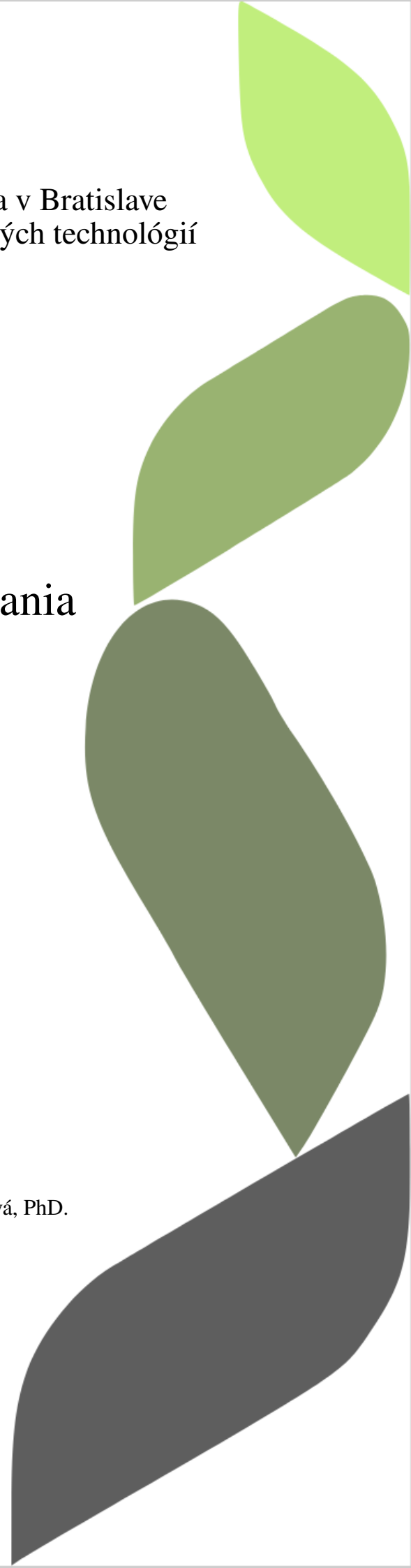
Tím č.5

# Metodika testovania

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

Vypracoval: Maroš Kollár

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



# 1 Testovanie

---

Testovanie je dôležitou súčasťou vývoja, nakoľko pomáha validácií riešenia a identifikácií chýb. Preto by malo zastrešovať čo najväčšiu časť vytvoreného produktu.

Vzhľadom na druh produktu, ktorý vyvíjame a dátovú prácu s ním súvisiacu, sme sa rozhodli využiť najmä jednotkové testy a akceptačné testy.

## Jednotkové testy

Jednotkové testy budú vykonávané pomocou knižnice **pytest**. V rámci využívania tejto knižnice je preto nutné dodržať konvenciu pomenovávania.

- **Súbory:** názov súboru sa musí začínať slovom „test”. Aj napriek metodike vývoja v Pythone sa budú súbory pomenovávať cez *lower\_case* nezávisle na tom, či obsahujú triedy alebo nie. Príklad: *test\_features.py*
- **Triedy:** Podľa metodiky vývoja v Pythone sa triedy budú pomenovávať pomocou *CammelCase*. Prvé slovo v triede musí byť slovo „Test”.
- **Metódy:** Metóda musí začať slovom „test”. Dodržaná bude metodika vývoja v Pythone, čo znamená, že metódy budú písané ako *lower\_case*.

Príklad jednotkového testu:

```
def test_mean_normal(self):  
    value = add_two_numbers(5, 2)  
    assert value == 7
```

## Nutnosť jednotkových testov

Jednotlivé komponenty/metódy budú testované jednotkovými testmi podľa uváženia osoby, ktorá daný komponent/metódu napísala, čím musí zväziť nutnosť jej iného ako fyzického testovania.

Odporúča sa však mať jednotkovými testami pokrytú čo najväčšiu časť kódu, vďaka čomu si môžu vývojári byť neustále istí správnym fungovaním komponentov/metód. Existencia jednotkových testov taktiež napomôže review nových častí kódu, vďaka čomu sa ušetrí čas na manuálnom testovaní.

Taktiež sa odporúča, aby jednotkový test k určitému komponentu/metóde napísala osoba, ktorá daný komponent/metódu vytvorila. Ušetrí sa tak čas potrebný pre naštvovanie komponentu/metódy iným členom tímu.

## **Organizácia testov**

Všetky súbory s testmi budú v špeciálnom adresári „*UnitTests*” z dôvodu ich jednoduchšej organizácie. Testy budú organizované do súborov a tried v závislosti od ich logickej organizácie.

Každá metóda v súboroch s testmi zobrazuje samostatný test.

## **Spustenie jednotkových testov**

Testovanie pomocou jednotkových testov sa bude spúšťať príkazom „*python -m pytest -v*” pomocou konzoly z root adresáru. Prepínač *-v* slúži pre detailnejší výpis úspešnosti jednotlivých testov a je voliteľný.

## **Akceptačné testy**

Na konci fázy tvorby softvéru, ktorý bude možné použiť pre používateľsky jednoduchšiu klasifikáciu vzoriek oleja sa vykoná akceptačný test.

Akceptačný test bude vytvorený na začiatku fázy tvorby softvéru. Test bude spísaný na základe dohody s product ownerom. Zadané budú možné vstupy a predpokladané výsledky k nim. Testované musia byť aj hraničné vstupy, ktorými sa overí schopnosť softvéru reagovať na chyby a zotaviť sa z chybových stavov.

Pre akceptovanie akceptačného testu je nutné bez výnimky splniť všetky dohodnuté body testu.

## A História

---

<b>Meno a Priezvisko</b>	<b>Dátum</b>	<b>Opis zmeny</b>
Maroš Kollár	16.10.2020	Vytvorenie metodiky
Simona Klučková	24.10.2020	Zmena názvov súborov podľa štruktúry v git repozitáry
Maroš Kollár	22.03.2021	Úpravenie popisu spúšťania Jednotkových testov
...	...	...