Напредни Пајтон

Развој тестирањем - TDD

Увод

- TDD је методологија – помаже да креирамо бољи код, али не рјешава проблеме
- Не држати је се као пијан плота
- Не значи ако је нешто имплементирано са TDD да је завршено са тестирањем.
- Оно што нам је поред документације најмрскије, тестирање, стављамо на прво мјесто!

Шта је TDD?

- Скуп ситних цикличних корака:
  - Писање тестова
    - Тестирање (Нормално, прво написани тестови ће сви попадати- црвено)
  - Имплементација,
    - Тестирање док сви тестови не позелене (док не прођу)
  - Рефакторисање
    - Промјена кода која не мијења суштину, уљепшавање кода

Детаљнији кораци

- Написати таман толико кода да дође до правог пада теста
  - ПРВИ ПУТ тест НЕ СМИЈЕ ПРОЋИ. Уколико се то деси, тест највјероватније није потребан
  - **Грешке типа: `ImportError: cannot import` значе да треба нешто додати у коду, али додати само да нема оваквих грешака него да дође до поруке FAILED**
  - **У ову сврху се углавном користи кључна ријеч pass**
- Када се деси прави пад, уписати оно што ће да омогући да тест прође. АЛИ НЕ ИМПЛЕМЕНТАЦИЈУ него сам **ВРИЈЕДНОСТ коју тест очекује**
- Правило само једног теста који пада: Колико год тестова било, само један смије пасти! Ово повећава фокусираност на један проблем
- Када тестови прођу, рефакторисати код. Рефакторисање оставља све тестове зелене – НИКАД НЕ РЕФАКТОРИСАТИ без тестова

Захтјеви су ти који диктирају тестове

- Дизајнирање је нешто што иде паралелно са писањем захтјева
- Навести све захтјеве и уситнити их до нивоа софтверског захтјева
- Сваки софтверски захтјев је сет јединичних тестова

Тестирање ... даље

- Препорука за погледати концепт порука при тестирању