

Тема. Підготовка та використання наборів тестових даних

Очікувані результати заняття

Після цього заняття треба вміти: тестувати створений код

Повторюємо

- які способи сортування масиву ви знаєте?
- як знайти елемент списку, що відповідає певній умові?
- що таке необхідна умова, достатня умова?

Ознайомтеся з інформацією

Для того, щоб перевірити наскільки добре працює написана програма, її тестують. Тестування відбувається різними способами, залежно від того, що це за програма, які її функції, в якому середовищі вона створена. Наприклад, якщо це комп'ютерна гра, то часто випускають на загальну пробну версію цієї гри і всі бажаючі її грають, а потім пишуть в коментарях до неї, що їм не сподобалось, що працювало не так, або вносять якісь свої пропозиції. Якщо це якась спеціалізована програма, то її тестують спеціалісти для яких вона пишеться і теж вказують на певні неточності чи недоопрацювання. Але найпершим кроком тестування є тестування, яке здійснює безпосередньо сам програміст. І тут багато що залежить від середовища, в якому створювалась сама програма. Найпростіший спосіб такого тестування це створення тестового набору даних – якщо таку перевірку програма проходить, то далі її можна давати для тестування користувачам.

Перегляньте презентацію

[Підготовка і використання тестових даних](#)

Виконайте у середовищі програмування

<https://repl.it/languages/python3> або <https://www.techcharge.in/online-python-compiler/>

Задача 1. Види тестових даних(типові дані)

Створено функцію `suma_chusel(a, b)`, яка приймає два числа `a` та `b` і повертає їх суму. Для перевірки правильності роботи функції створено чотири набори тестових даних:

testovi_dani1: додавання двох позитивних чисел (1 і 2).

testovi_dani2: додавання двох негативних чисел (-1 і -2).

testovi_dani3: додавання двох нулів (0 і 0).

testovi_dani4: додавання позитивного числа та негативного числа (10 і -5).

Код

```
def suma_chusel(a, b):  
    return a + b
```

```
# Тестові дані
```

```
testovi_dani1 = (1, 2, 3)    # Типові дані: додавання двох позитивних чисел
```

```

testovi_dani2 = (-1, -2, -3) # Типові дані: додавання двох негативних чисел
testovi_dani3 = (0, 0, 0)   # Типові дані: додавання двох нулів
testovi_dani4 = (10, -5, 5) # Типові дані: додавання позитивного та негативного числа

# Тестування
for dani in [testovi_dani1, testovi_dani2, testovi_dani3, testovi_dani4]:
    a, b, ochikuvanuy_resultat = dani
    resultat = suma_chusel(a, b)
    if resultat == ochikuvanuy_resultat:
        print("Тест пройшов успішно,", "Очікуване значення:", ochikuvanuy_resultat,
"Отримане значення:", resultat)
    else:
        print("Тест не пройшов,", "Очікуване значення:", ochikuvanuy_resultat,
"Отримане значення:", resultat)

```

Задача 2. Види тестових даних(Граничні дані - максимальні і мінімальні значення)

Створено функцію `znaity_maximum(numbers)`, яка приймає список чисел `numbers` і повертає максимальний елемент у цьому списку. У разі, якщо список порожній, функція повертає `None`. Для перевірки правильності роботи функції створено чотири набори тестових даних:

`testovi_dani1`: список з додатніми числами (1, 2, 3).

`testovi_dani2`: список з від'ємними та нульовим числами (-10, -5, 0).

`testovi_dani3`: список з одним елементом (100).

`testovi_dani4`: порожній список.

Код

```

def znaity_maximum(numbers):
    if not numbers:
        return None
    return max(numbers)

# Тестові дані з граничними значеннями
testovi_dani1 = ([1, 2, 3], 3) # Максимальне значення в списку
testovi_dani2 = ([-10, -5, 0], 0) # Мінімальне значення в списку
testovi_dani3 = ([100], 100) # Тільки один елемент у списку
testovi_dani4 = ([], None) # Порожній список

# Тестування
for dani in [testovi_dani1, testovi_dani2, testovi_dani3, testovi_dani4]:
    numbers, ochikuvanuy_resultat = dani
    resultat = znaity_maximum(numbers)
    if resultat == ochikuvanuy_resultat:
        print("Тест пройшов успішно,", "Очікуване значення: ", ochikuvanuy_resultat,
"Отримане значення: ", resultat)
    else:
        print("Тест не пройшов,", "Очікуване значення: ", ochikuvanuy_resultat,
"Отримане значення: ", resultat)

```

Домашнє завдання

Введіть та протестуйте код до задачі Некоректні дані з презентації

Скріншот робочого коду для розв'язування задачі надіслати на HUMAN або на електронну пошту nataliartemiuk.55@gmail.com

Джерело

[На урок](#)