Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 4 з дисципліни

«Основи програмування

2. Модульне програмування»

«Перевантаження операторів»

Виконав студент ІП-14 Нікулін Павло Юрійович
(шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №4

Перевантаження операторів

Мета: вивчити механізми створення класів з використанням перевантажених операторів (операцій).

Хід роботи

Задача:

Розробити клас «Вектор у просторі», який задається координатами його кінця. Реалізувати для нього декілька конструкторів, геттери, метод обчислення довжини вектора. Перевантажити оператори «+» та «*» для знаходження суми і скалярного добутку векторів відповідно. Створити три вектори (M1, M2, M3), використовуючи різні конструктори. Визначити вектор M3 як суму векторів M1 та M2. Знайти довжину вектору M3, а також скалярний добуток векторів M1 та M3.

Розв'язання

1. **Постановка задачі**: результатом роботи мать бути розрахунки довжини вектору M3 та скалярний добуток векторів M1 та M3. Можливість використання різних конструкторів реалізуємо перевантаженням конструктора *__init__*. Для роботи з векторами використаємо список векторів *vectors (v_list)*. Координати у класі будуть захищеного типу, звертання до них буде реалізовано геттерами.

2. Побудова **математичної моделі**. Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Файл з функціями	Файловий	FUNCS.PY	Файл
Список векторів	Класовий	VECTORS/V_LIST	Список
Створення M1 та M2	Універсальний	SET_VECTORS	Функція
Розрахунки	Універсальний	CALCULATIONS	Функція
Клас векторів	Клас	VECTOR	Клас
Конструктор з перевантаженням	Універсальний	__INIT__	Конструктор
Перевантажений «+»	Універсальний	__ADD__	Перевантаження
Перевантажений «*»	Універсальний	__MUL__	Перевантаження

Геттери координат	Дійсний	GET_X/_Y/_Z	Метод
Довжина вектору	Цілий	LENGTH	Метод/ Результат
Аргумент методів/конструктора	Універсальний	SELF	Проміжне дане
Множина аргументів	Універсальний	ARGS	Проміжне дане
Аргумент для перевантаження	Універсальний	OTHER	Проміжне дане
Захищені координати	Цілий	_X/_Y/_Z	Початкове дане
Введені координати	Цілий	A/B/C	Початкове дане
Кортеж координат МЗ	Цілий	МЗ	Результат
Кортеж перемножених координат	Цілий	TEMP	Проміжне дане
Скалярний добуток	Цілий	SCALAR	Результат

Випробування коду

Код

main.py

```python
from funcs import *

vectors = [] # list of vectors
vectors = set_vectors(vectors)

calculations(vectors)

```

funcs.py

```python
class Vector:
    def __init__(self, *args): # overloading constructor = multiple constructors
        if len(args) == 0:
            self._x = 0
            self._y = 0
            self._z = 0
        elif len(args) == 1:
            self._x = args[0]
            self._y = 0
            self._z = 0
        elif len(args) == 2:
            self._x = args[0]
            self._y = args[1]
            self._z = 0
        elif len(args) >= 3:
            self._x = args[0]
            self._y = args[1]
            self._z = args[2]

    def __add__(self, other): # overloading +
        return self._x + other._x, self._y + other._y, self._z + other._z

    def __mul__(self, other): # overloading *
        return self._x * other._x, self._y * other._y, self._z * other._z

    # getters
    def get_x(self):
        return self._x

    def get_y(self):
        return self._y

    def get_z(self):
        return self._z

    def length(self): # method
        return (self._x ** 2 + self._y ** 2 + self._z) ** (1 / 2)

```

```python
def set_vectors(v_list):
    print("Set coordinates for M1 and M2. Enter 'done' to go to next vector:") # skipping allows using multiple constructors
    for i in range(2):
        a = input("\nx coordinate for vector №%d: " % (i + 1))
        if a == "done":
            v_list.append(Vector())
            continue
        b = input("y coordinate for vector №%d: " % (i + 1))
        if b == "done":
            v_list.append(Vector(int(a)))
            continue
        c = input("z coordinate for vector №%d: " % (i + 1))
        if c == "done":
            v_list.append(Vector(int(a), int(b)))
            continue

        v_list.append(Vector(int(a), int(b), int(c)))

    m3 = v_list[0] + v_list[1] # add coordinates using overloaded +
    v_list.append(Vector(m3[0], m3[1], m3[2]))

    print("\nVectors:")
    for i in range(3):
        print("Vector №%d = (%d, %d, %d)" % ((i + 1), v_list[i].get_x(), v_list[i].get_y(), v_list[i].get_z()))

    return v_list

def calculations(v_list):
    print("\nLength of vectors:")
    for i in range(3):
        print("Vector №%d = %f" % ((i + 1), v_list[i].length()))

    scalar = 0
    temp = v_list[0] * v_list[2] # multiply coordinates using overloaded *
    for i in range(len(temp)):
        scalar += temp[i]

    print("\nScalar product of M1 and M3:\nM1 * M3 = %d" % scalar)

```

Результат

```
Set coordinates for M1 and M2. Enter 'done' to go to next vector:

x coordinate for vector №1: 9
y coordinate for vector №1: -1
z coordinate for vector №1: done

x coordinate for vector №2: 4
y coordinate for vector №2: 5
z coordinate for vector №2: -7

Vectors:
Vector №1 = (9, -1, 0)
Vector №2 = (4, 5, -7)
Vector №3 = (13, 4, -7)

Length of vectors:
Vector №1 = 9.055385
Vector №2 = 5.830952
Vector №3 = 13.341664

Scalar product of M1 and M3:
M1 * M3 = 113
```

Висновок

Під час виконання лабораторної роботи було досліджено механізми створення класів з використанням перевантажених операторів. Було створено клас *Vector*, у якому реалізовано конструктор з перевантаженням, перевантаження операторів «+» та «*», геттери для координатів (координати оголошені у захищеному режимі), метод обчислення довжини вектора *length*. Для використання іншого конструктора користувач може ввести «done» замість координати і тоді ця та наступні координати вектора будуть заповнені нулями. Перевантажені оператори було використано під час обчислення вектора М3 та скалярного добутку векторів М1 та М3. Роботу виконано на мові програмування *Python*, задача виконана, програма працює коректно.