Міністерство освіти і науки України  
НТУУ «КПІ ім. І. Сікорського»

Кафедра автоматизації проектування енергетичних процесів і систем

Лабораторна робота №5

«Наслідування, рефлексія»

Виконав: студент II курсу ТЕФ, групи TI-92

Сопронюк Ю. А.

Перевірив: Касьянов А. С.

Дата виконання: 17.12.2020

Київ-2020

**Варіант №11**

1. Створити базовий і похідний класи. Визначити конструктори, методи доступу, віртуальні методи. Показати використання конструкторів цього ж і базового класу. Можна визначити деякі константи або enum.
2. Розробити анотацію, відзначити, їй метод(и) в класі, За допомогою рефлексії обійти методи класу і викликати відмічені анотацією методи за допомогою invoke().
3. За допомогою рефлексії вивести ім'я класу, а також відповідно до варіанта:

|  |  |  |
| --- | --- | --- |
| Номер варіанту | Завдання 1 | Завдання 2 |
| 1,6, ***11***, 16, 21 | Список конструкторів з їх параметрами | Модифікатори класу |

**ВИКОНАНАННЯ РОБОТИ**

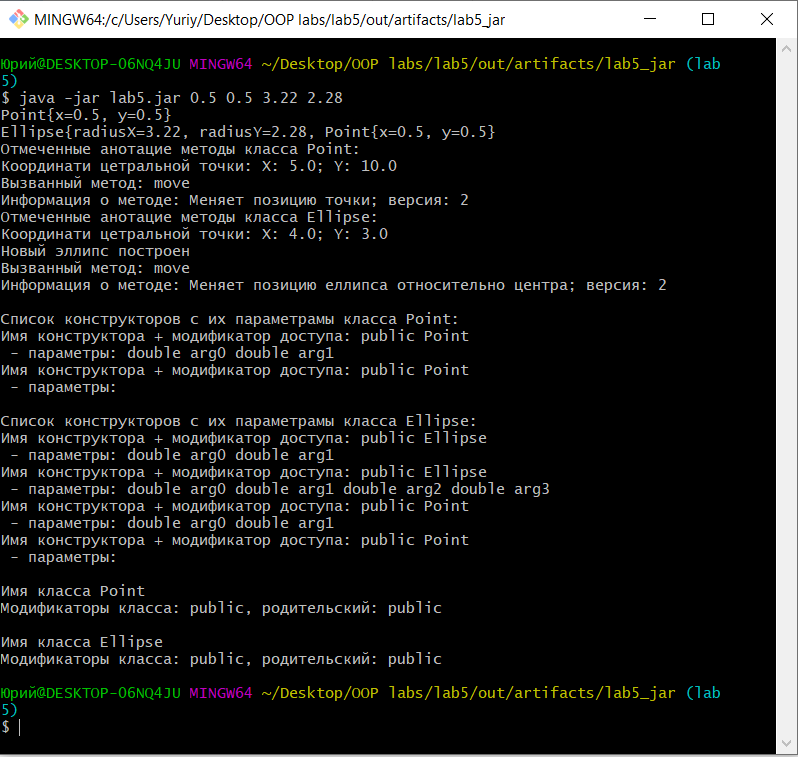
**Код програми**

**Усі необхідні файли проекту можна отримати за посиланням (ветка lab5):**

*https://github.com/steadyGuy/oop\_labs/tree/lab5*

Крім основних файлів з розширенням *.java* та *.jar*, тут також знаходиться звіт у форматі *.docx*.

**Результат роботи програми**

**

*Програму було запущено з консолі, тому аргументи були записані через консоль, а не через* ***Project Properties*** *в IDE*

**ВИСНОВОК**

Під час виконання лабораторної роботи було створено два класи, а саме *Point* – як базовий, а також *Ellipse*, який є наслідником. Були визначені усі необхідні конструктори, геттери та сеттери, а також визначено віртуальний метод (*move*), згідно з завданням варіанту.

Було створено спеціальну анотацію (*InfoAnnotation*), методи якої описували певні метадані програмного коду для функцій (хоча можна описати для класу, пакету і т.д). Також була використана стандартна анотація *@Override*, яка позначала, що ми збираємося перевизначити метод батьківського класу.

За допомогою рефлексії – механізму дослідження даних про програму під час її виконання, було виведено і оброблено інформацію про модифікатори, методи і конструктори класів та їх суперкласів. Також була використана можливість виклику методу об’єкта по імені його класу та відповідної анотації. Було використано велику кількість методів з інтерфейсів, які належать пакету *java.lang.reflect* (в особливості *invoke*, *getClass*, *getAnnotation*, та інші).

Список можливостей *Reflection API* набагато ширший. Використання цього механізму разом з власними анотаціями представляє широкі можливості для роботи з кодом програми, а тому є сенс використання не тільки стандартних анотацій мови, але й для полегшення життя, власних, кастомних анотацій під свої потреби з використанням рефлексії за необхідності.