

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

ЛАБОРАТОРНА РОБОТА № 11

з дисципліни «Основи програмування»

Тема: Посилання на методи (Method references)

Виконали:

Студентки групи ІА-31

Горlach Д., Макассєва М.,

Соколова П.

Перевірів:

Степанов А.

Тема: Посилання на методи (Method references)

Мета: Ознайомитись з наданими нам матеріалами, пригадати інформацію надану на лекціях з даної теми, також навчившись з минулих лабораторних робіт правильно використовувати знання та реалізовувати за допомогою них завдання, виконати поставлену нам задачу. А саме, скористатись новими знаннями та реалізувати за допомогою них подане завдання.

Хід роботи:

Використовуючи класи лабораторної роботи №8 першого семестру демонструємо використання посилань на методи (Method reference) чотирьох типів.

Перший це посилання на статичний метод класу. Звертаємось на пряму до класу FoodProduct, у якого використовуємо посилання на метод printCount, який є статичний, не приймає жодних параметрів і нічого не повертаємо, і отримуємо функціональний інтерфейс Runnable, у якого викликаємо метод run.

Другий це посилання на не статичний метод об'єкту. Створюємо об'єкт FoodProduct, у якого використовуємо посилання на метод toString, який є не статичним, не приймає жодних параметрів і повертає String, і отримуємо функціональний інтерфейс Supplier, типізований під String, у якого викликаємо метод get і отримуємо значення типу String.

Третій це посилання на нестатичний метод класу. Звертаємось на пряму до класу FoodProduct, у якого використовуємо посилання на метод getFoodType, який є не статичний, не приймає жодних параметрів і повертає FoodType, і

отримуємо функціональний інтерфейс Function типізований під два типи це FoodProduct і FoodType, і викликаємо метод apply, параметром якого є об'єкт FoodProduct і отримуємо значення типу FoodType.

Четвертий це посилання на конструктор. Зробили два варіанти: без параметрів і з параметрами. Без параметрів, звертаємось до класу FoodProduct у якого використовуємо посилання на оператор new і отримуємо функціональний інтерфейс Supplier, типізований під FoodProduct, і викликаємо метод get і отримуємо об'єкт типу FoodProduct. З параметрами, для цього створюємо функціональний інтерфейс ProductFoodFactory із методом create, який має параметри такі самі, як і в конструкторі із параметрами в класі FoodProduct. Звертаємо до класу FoodProduct у якого використовуємо посилання на оператор new і отримуємо функціональний інтерфейс ProductFoodFactory із методом create, в який передаємо необхідні параметри і отримуємо об'єкт FoodProduct.

2. На прикладі свого варіанту з л/р №8 першого семестру продемонструвати використання посилань на методи чотирьох типів:

- посилання на статичний метод класу;
- посилання на нестатичний метод об'єкта;
- посилання на нестатичний метод класу;
- посилання на конструктор.

Рис. 1 завдання

Хід роботи:

1. Повторити теоретичні відомості
2. Проаналізувати предметну область завдання свого варіанту (табл.1)
3. Розробити базовий клас (відповідно до завдання можливо абстрактний клас або інтерфейс), клас-нащадок, а також допоміжні класи та/або інтерфейси за необхідністю. Відповідно до предметної області завдання передбачити відповідні методи бізнес-логіки, а також конструктори, сетери та/або гетери, методи equals() та toString().
Продемонструвати використання:
 - this;
 - super;
 - перевантаження (overloading) та заміщення(overriding) методів;
 - перевантаження (overloading) конструкторів.
4. Відповісти на контрольні запитання

Рис. 2 завдання минулого семестру та наш варіант був 9

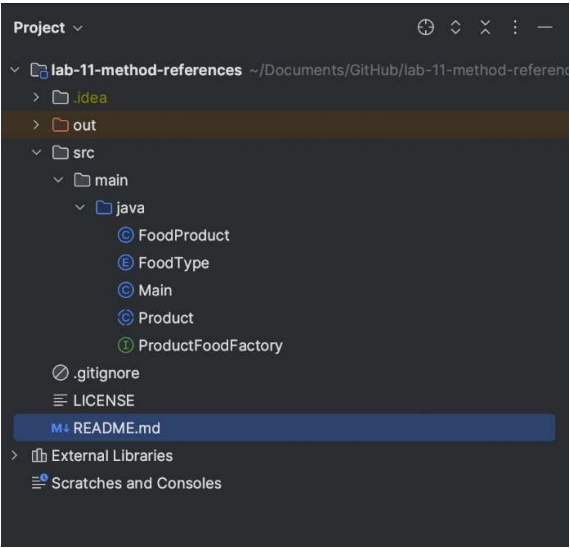


Рис 3. Структура програми

```
1  > import java.util.*;
2
3  < cyber_fire
4  public class Main {
5      < cyber_fire
6      public static void main(String[] args) {
7          classStaticMethodReference();
8          System.out.println();
9          objectNonStaticMethodReference();
10         System.out.println();
11         classNonStaticMethodReference();
12         System.out.println();
13         constructorMethodReference();
14     }
15
16     < usage < cyber_fire
17     private static void classStaticMethodReference() {
18         System.out.println("Class static method reference");
19     }
20
21     < usage < cyber_fire
22     private static void objectNonStaticMethodReference() {
23         System.out.println("Object non-static method reference");
24     }
25
26     < usage < cyber_fire
27     private static void classNonStaticMethodReference() {
28         Runnable printCountRunnable = FoodProduct::printCount;
29         printCountRunnable.run();
30     }
31
32     < usage < cyber_fire
33     private static void constructorMethodReference() {
34         FoodProduct foodProduct = new FoodProduct("apple", 2, 0.5, 10, FoodType.FRUIT);
35         Supplier<String> toStringSupplier = foodProduct::toString;
36         System.out.println(toStringSupplier.get());
37     }
38
39     < usage < cyber_fire
40     private static void classNonStaticMethodReference() {
41         FoodProduct foodProduct = new FoodProduct("apple", 2, 0.5, 10, FoodType.FRUIT);
42         Supplier<String> toStringSupplier = foodProduct::toString;
43         System.out.println(toStringSupplier.get());
44     }
45 }
```

Рис. 4 Main

```
Class static method reference
FoodProduct.Product print count: 0

Object non-static method reference
FoodProduct.Product: name=apple, price=1.0, calories=20 Food Type=FRUIT

Class non-static method reference
VEGETABLE

Constructor method reference
Without parameters: FoodProduct.Product: name=Something, price=0.0, calories=0 Food Type=UNKNOWN
With parameters: FoodProduct.Product: name=pear, price=0.5, calories=10 Food Type=FRUIT

Process finished with exit code 0
```

Рис. 5 output

Висновок:

Після того як ми вивчили лямбда-вирази і функціональні інтерфейси, ми можемо викликати методи за допомогою компактного синтаксису, який називається `method-reference` (посилання на метод). Якщо лямбда вирази зв'язуються з функціональним інтерфейсом, то методи так само можуть бути зв'язані з функціональним інтерфейсом. Зв'язок методу з функціональним інтерфейсом здійснюється за допомогою посилання на метод. Якщо лямбда-вираз може бути переданий в деякий метод як параметр, то посилання на метод також може бути передане як параметр. За допомогою цього посилання можна звертатися до методу, не викликаючи його.