

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

ЛАБОРАТОРНА РОБОТА № 9

з дисципліни «Основи програмування»

Тема: Лямбда-вирази та компаратори

Виконали:

Студентки групи ІА-31

Горlach Д., Макасеєва М.,

Соколова П.

Перевірів:

Степанов А.

Тема: Лямбда-вирази та компаратори

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 сортуємо за допомогою компараторів. Для початку створюємо список з класів FoodProduct, потім отримуємо масив з нього для кожного прикладу із компаратором. Перший приклад це сортування за назвою у вигляді лямбда-виразу. Другий приклад це сортування за назвою у зворотному порядку, методом reversed. Третій приклад це сортування за назвою, або якщо назви однакові, сортування за кількістю, використовуючи method reference, методом thenComparing. Для четвертого прикладу додаємо null в список. Четвертий приклад це сортування за назвою, але при цьому, першим/останнім буде null, використовуючи методи nullFirst|nullLast. І використовуємо метод для відображення результату роботи в консоль printProducts.

Хід роботи:

1. Пригадати як використовувати лямбда-вирази та компаратори.
2. Проаналізувати класи свого варіанту з л/р №8 першого семестру, та виділити що найменше дві властивості, за якими можна сортувати об'єкти цих класів (наприклад, для класу Людина це може бути ім'я та прізвище, для класу Документ – назва та дата створення, ...). Після цього:
 - 2.1) за допомогою лямбда-виразів створити компаратор для сортування за однією з цих двох ознак в порядку зростання;
 - 2.2) за допомогою дефолтного метода `Comparator.reversed()` створити компаратор для сортування за обраною ознакою у зворотному порядку;
 - 2.3) за допомогою дефолтного метода `Comparator.thenComparing()` створити компаратор, який буде порівнювати об'єкти за однією ознакою, а у разі коли вони співпадають порівнювати за іншою ознакою;
 - 2.4) за допомогою статичних методів `Comparator.nullsFirst()` або `Comparator.nullsLast()` створити компаратор, який дозволить порівнювати null-посилання на об'єкти з іншими об'єктами.
3. Продемонструвати використання усіх створених компараторів (відсортувати масив об'єктів та/або зберегти об'єкти у TreeSet).
4. Відповісти на контрольні питання.

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

9	Товар	Продовольчий товар	mutable
---	-------	--------------------	---------

Рис. 2 завдання минулого семестру та наш варіант

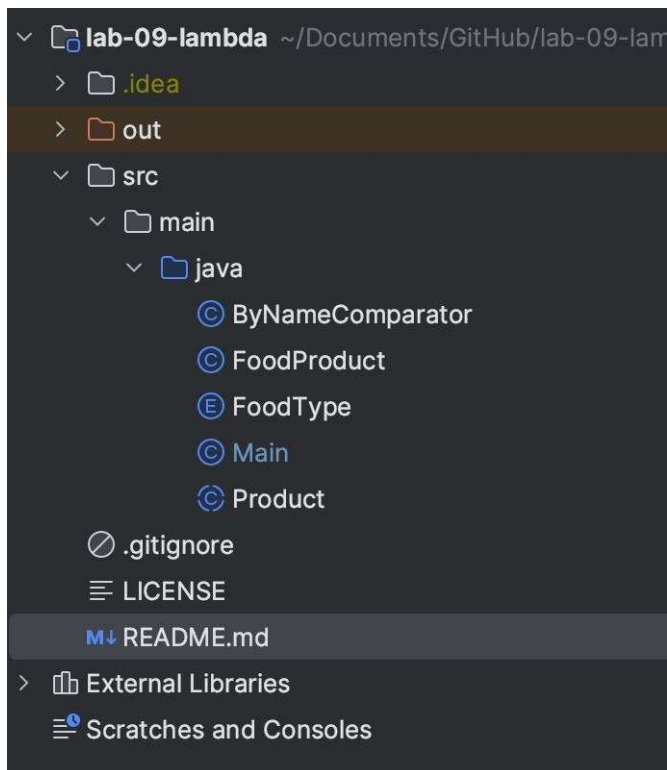


Рис 3. Структура програми

```

System.out.println("Compare by name");
Product[] productArrayCompareByName = products.toArray(new Product[]{});
Arrays.sort(productArrayCompareByName, (p1, p2) -> p1.getName().compareTo(p2.getName()));
printProducts(productArrayCompareByName);

System.out.println("Compare by reverse name");
Product[] productArrayCompareByNameReverse = products.toArray(new Product[]{});
Arrays.sort(productArrayCompareByNameReverse, new ByNameComparator().reversed());
printProducts(productArrayCompareByNameReverse);

System.out.println("Compare by name then by quantity");
Product[] productArrayCompareByNameAndQuantity = products.toArray(new Product[]{});
Arrays.sort(productArrayCompareByNameAndQuantity, new ByNameComparator().thenComparing(Product::getQuantity));
printProducts(productArrayCompareByNameAndQuantity);

System.out.println("Compare by name and null first/last");
products.add(null);
Product[] productArrayCompareByNameAndNulls = products.toArray(new Product[]{});
Arrays.sort(productArrayCompareByNameAndNulls, Comparator.nullsFirst(new ByNameComparator()));
Arrays.sort(productArrayCompareByNameAndNulls, Comparator.nullsLast(new ByNameComparator()));
printProducts(productArrayCompareByNameAndNulls);

```

Рис. 4 Main-частина коду

```

Compare by name
Name: apple Quantity: 2.0
Name: apple Quantity: 1.0
Name: blueberry Quantity: 1.0
Name: pear Quantity: 1.0
Name: pear Quantity: 4.0

Compare by reverse name
Name: pear Quantity: 1.0
Name: pear Quantity: 4.0
Name: blueberry Quantity: 1.0
Name: apple Quantity: 2.0
Name: apple Quantity: 1.0

Compare by name then by quantity
Name: apple Quantity: 1.0
Name: apple Quantity: 2.0
Name: blueberry Quantity: 1.0
Name: pear Quantity: 1.0
Name: pear Quantity: 4.0

Compare by name and null first/last
Null
Name: apple Quantity: 2.0
Name: apple Quantity: 1.0
Name: blueberry Quantity: 1.0
Name: pear Quantity: 1.0
Name: pear Quantity: 4.0

```

Рис. 5 output (Null з nullFirst)

```
Compare by name and null first/last  
Name: apple Quantity: 2.0  
Name: apple Quantity: 1.0  
Name: blueberry Quantity: 1.0  
Name: pear Quantity: 1.0  
Name: pear Quantity: 4.0  
Null
```

Рис. 6 output (Null з nullLast)

Висновок:

Для коректного виконання лабораторної роботи, ми значення Null одразу не додаємо, бо в нас немає обробки та порівняння на Null, якщо додати перед обробленням, то будемо мати NullPointerException. Одразу перед виконанням останнього компаратора, ми додаємо значення Null тому, що в цьому компараторі ми обробляємо Null. Результат сортування відрізняється залежно від методів які ми використовуємо, nullFirst|nullLast, ми будемо мати Null або спочатку або вкінці.