

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Кафедра ЕОМ



Лабораторна робота №7

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Параметризоване програмування»

Варіант № 10

Виконав: ст. гр. КІ-34

Прийняв: Іванов Ю.С

Львів – 2022

Мета: Оволодіти навиками параметризованого програмування мовою Java.

Завдання:

1. Створити параметризований клас, що реалізує предметну область задану варіантом. Клас має містити мінімум 4 методи опрацювання даних включаючи розміщення та виймання елементів. Парні варіанти реалізують пошук мінімального елементу, непарні – максимального. Написати на мові Java та налагодити програму-драйвер для розробленого класу, яка мстить мінімум 2 різні класи екземпляри яких розміщуються у екземплярі розробленого класу-контейнеру. Програма має розміщуватися в пакеті Група.Прізвище.Lab6 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
2. Автоматично згенерувати документацію до розробленого пакету.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
4. Дати відповідь на контрольні запитання.

Код Main.java:

```
package main.kzp.lab7;

import main.kzp.lab7.stuff.Pen;
import main.kzp.lab7.stuff.Scissor;

public class Main {
    public static void main(String[] args) {
        PencilBox pencilBox = new PencilBox();
        pencilBox.putThing(new Pen());
        pencilBox.putThing(new Pen());
        pencilBox.putThing(new Pen());
        pencilBox.putThing(new Scissor(2));
        pencilBox.putThing(new Scissor(5));

        pencilBox.getPens().forEach(Pen::doWork);
        pencilBox.getScissors().forEach(Scissor::doWork);

        System.out.println(pencilBox.getMinimumScissor().getSize());
    }
}
```

Код PencilBox.java:

```
package main.kzp.lab7;

import main.kzp.lab7.stuff.Pen;
import main.kzp.lab7.stuff.Scissor;
import main.kzp.lab7.stuff.Stuff;

import java.util.ArrayList;
import java.util.Comparator;
```

```

import java.util.List;
import java.util.NoSuchElementException;
import java.util.stream.Collectors;

public class PencilBox {
    private List<Stuff> stuffList = new ArrayList<>();

    public <T extends Stuff> void putThing(T thing){
        stuffList.add(thing);
    }

    public Scissor getMinimumScissor(){
        return stuffList.stream().filter(x -> x instanceof Scissor).map(x ->
(Scis-
sor)x).min(Comparator.comparing(Scissor::getSize)).orElseThrow(NoSuchElementException::new);
    }

    public List<Scissor> getScissors(){
        return stuffList.stream().filter(x -> x instanceof Scissor).map(x ->
(Scissor)x).collect(Collectors.toList());
    }

    public List<Pen> getPens(){
        return stuffList.stream().filter(x -> x instanceof Pen).map(x ->
(Pen)x).collect(Collectors.toList());
    }
}

```

Код Stuff.java:

```

package main.kzp.lab7.stuff;

public interface Stuff {
    void doWork();
}

```

Код Pen.java:

```

package main.kzp.lab7.stuff;

public class Pen implements Stuff{
    @Override
    public void doWork() {
        System.out.println("The pen write some text.");
    }
}

```

Код Scissor.java:

```

package main.kzp.lab7.stuff;

public class Scissor implements Stuff{

    private int size;

    public Scissor(int size) {
        this.size = size;
    }

    @Override
    public void doWork() {

```

```
        System.out.println("The scissor cut off smth.");  
    }  
  
    public int getSize() {  
        return size;  
    }  
}
```

Висновок: Я оволодів навиками параметризованого програмування мовою Java.

Контрольні питання:

1. Дайте визначення терміну «параметризоване програмування».

Відповідь: Параметризоване програмування є аналогом шаблонів у C++. Воно полягає у написанні коду, що можна багаторазово застосовувати з об'єктами різних класів.

2. Розкрийте синтаксис визначення простого параметризованого класу.

Відповідь: Параметризований клас – це клас з однією або більше змінними типу.

```
[public] class НазваКласу<параметризованийТип {,параметризованийТип} >  
{...}
```

3. Розкрийте синтаксис створення об'єкту параметризованого класу.

Відповідь: НазваКласу < перелікТипів > = new НазваКласу < перелікТипів > (параметри);

4. Розкрийте синтаксис визначення параметризованого методу.

Відповідь: Параметризовані методи визначаються в середині як звичайних класів так і параметризованих.

```
Модифікатори< параметризованийТип {,параметризованийТип} >  
типПовернення назваМетоду(параметри);
```

5. Розкрийте синтаксис виклику параметризованого методу.

Відповідь: (НазваКласу|НазваОб'єкту).[<перелікТипів>]
НазваМетоду(параметри);

6. Яку роль відіграє встановлення обмежень для змінних типів?

Відповідь: Бувають ситуації, коли клас або метод потребують накладення обмежень на змінні типів. Наприклад, може бути ситуація, коли метод у процесі

роботи викликає з-під об'єкта параметризованого типу метод, що визначається у деякому інтерфейсі.

7. Як встановити обмеження для змінних типів?

Відповідь: використати ключове слово `extends` і вказати один суперклас, або довільну кількість інтерфейсів (через знак `&`), від яких має походити реальний тип, що підставляється замість параметризованого типу.

8. Розкрийте правила спадкування параметризованих типів.

Відповідь:

- Всі класи, що утворені з одного і того ж параметризованого класу з використанням різних значень змінних типів є незалежними навіть якщо між цими типами є залежність спадкування.
- Завжди можна перетворити параметризований клас у «сирий» клас.
- Параметризовані класи можуть розширювати або реалізовувати інші параметризовані класи.

9. Яке призначення підстановочних типів?

Відповідь: Підстановочні типи були введені у мову Java для збільшення гнучкості жорсткої існуючої системи параметризованих типів. На відміну від неї підстановочні типи дозволяють враховувати залежності між типами, що виступають параметрами для параметризованих типів.

10. Застосування підстановочних типів.

Відповідь:

- обмеження підтипу;
- обмеження супертипу;
- необмежені підстановки.