

Отчет по лабораторной работе №3

1 Цель и постановка задачи

В ходе выполнения лабораторной работы №3 необходимо выполнить:

1. Выгрузить все поля и методы класса с помощью рефлексии, вызвать несколько методов класса, вывести всех предков класса.
2. Ознакомиться со всеми коллекциями (list, set, map), сделать программу с примерами использования этих коллекций.
3. Сделать класс с дженериком.
4. Сделать метод с дженериком.

2 Разработка

2.1 Разработка

Сначала были созданы необходимые директории и был создан исполняемый класс Main.

Листинг 1. Листинг класса Main

```
package ru.lab;

import java.lang.reflect.Constructor;
import java.lang.reflect.Method;
import java.lang.reflect.Parameter;
import java.util.*;

public class Main {

    public static void main(String arg[]) throws Exception {
        SomeClass someClass = new SomeClass();
        Class clas = someClass.getClass();
        Class cls = SomeClass.class;
        Class.forName("ru.lab.SomeClass");
        SomeClass someClass1 = (SomeClass)clas.newInstance();

        Constructor[] constructors = clas.getDeclaredConstructors();
        for (Constructor constructor : constructors) {
            System.out.println(constructor.getName());
            Parameter[] parameters = constructor.getParameters();
            for (Parameter parameter : parameters) {
                System.out.println(parameter.getName());
                System.out.println(parameter.getType().getName());
            }
        }
    }
}
```

```

System.out.println(clas.getName());
Method[] methods = clas.getDeclaredMethods();
for(Method method : methods) {
    System.out.println(method);
}

System.out.println(getHierarchy(SomeClass.class));

Account<String> acc1 = new Account<String>("2345", 5000);
String acc1Id = acc1.getId();
System.out.println(acc1Id);

Account<Integer> acc2 = new Account<Integer>(2345, 5000);
Integer acc2Id = acc2.getId();
System.out.println(acc2Id);

Printer printer = new Printer();
String[] people = {"Tom", "Alice", "Sam", "Kate", "Bob", "Helen"};
Integer[] numbers = {23, 4, 5, 2, 13, 456, 4};
printer.<String>print(people);
printer.<Integer>print(numbers);

List listA = new ArrayList();
listA.add("element 1");
listA.add("element 2");
listA.add("element 3");
System.out.println(acc2Id);

List listDest = new ArrayList();
listDest.addAll(listA);

String element0 =(String) listA.get(0);
String element1 =(String) listA.get(1);
String element3 =(String) listA.get(2);

listA.remove("element 1");

System.out.println(listA);

Set setA = new HashSet();
String element = "element 1";
String elementTwo = "element 2";
setA.add(element);
setA.add(elementTwo);

setA.remove(elementTwo);
boolean isEmpty = setA.isEmpty();
System.out.println(isEmpty);

Map<Integer, String> emptyMap = new HashMap<>();

```

```

    emptyMap.put(1, "string 1");
    emptyMap.put(2, "string 2");
    emptyMap.put(3, "string 3");
    for(Map.Entry<Integer, String> entry: emptyMap.entrySet()) {
        Integer key = entry.getKey();
        String value = entry.getValue();
    }

    Iterator<Map.Entry<Integer, String>> itr = emptyMap.entrySet().iterator();
    while(itr.hasNext()) {
        Map.Entry<Integer, String> entry = itr.next();
        Integer key = entry.getKey();
        String value = entry.getValue();
        System.out.println(String.valueOf(key).concat(value));
    }
}

public static String getHierarchy(Class value){
    StringBuilder builder = new StringBuilder(value.getCanonicalName());
    Class klass=value;
    while (true){
        if (klass.getCanonicalName().equals("java.lang.Object")) {
            break;
        }
        klass = klass.getSuperclass();
        builder.append(" => "+klass.getCanonicalName());
    }
    return builder.toString();
}
}

```

Далее был создан типизированный класс Account и класс Printer, в котором находится типизированный метод print

```
class Printer {

    public <T> void print(T[] items){
        for(T item: items){
            System.out.println(item);
        }
    }

}

class Account<T>{

    private T id;
    private int sum;

    Account(T id, int sum){
        this.id = id;
        this.sum = sum;
    }

    public T getId() { return id; }
    public int getSum() { return sum; }
    public void setSum(int sum) { this.sum = sum; }

    public <T> void print(T[] items){
        for(T item: items){
            System.out.println(item);
        }
    }

}
```

2.2 Результат разработки

Далее с помощью команды `javac` данные файлы были скомпилированы. Файл `Main` был запущен командой `java`. При запуске программы в командной строке выводится: `.Результат работы`

```
ru.lab.SomeClass
arg0
java.lang.String
ru.lab.SomeClass
ru.lab.SomeClass
public java.lang.String ru.lab.SomeClass.someMethod(java.lang.String)
ru.lab.SomeClass => ru.lab.Person => ru.lab.NewPerson => java.lang.Object
2345
2345
Tom
Alice
Sam
Kate
Bob
Helen
23
4
5
2
13
456
4
2345
[element 2, element 3]
```