Домашнее задание № 8

Курс:
«Язык программирования Java»

ТЕМА: ИСКЛЮЧЕНИЯ

Задание 1

Создать метод, который выводит в консоль результат целочисленного деления числа, введенного с клавиатуры, на значения элементов одномерного массива целых чисел, заполненный случайным образом – от -10 до 10. Длина массива случайная – от 1 до 10.

Обработать все возможные исключительные ситуации в данном методе.

Задание 2

Создать метод, принимающий на вход число. В случае, если число отрицательное, в методе должно быть брошено проверяемое исключение. Если число больше 100, должно быть брошено непроверяемое исключение. Создать свои исключения для данного примера.

Протестируйте метод с помощью *JUnit*-тестов.

Задание 3

Создать класс, объекты которого будут неизменяемыми. Класс инкапсулирует в себе информацию о треугольнике на плоскости (длины каждой из его ребер). Длины сторон задаются в конструкторе. Если по заданным сторонам нельзя построить треугольник, в конструктор должно бросаться исключение.

Протестируйте класс с помощью *JUnit*-тестов.

Задание 4

Напишите метод бинарного поиска в одномерном массиве. В случае, если массив не отсортирован, метод должен бросать проверяемое исключение.

Задание 5

Дан класс:

```
import java.io.IOException;
import java.util.Random;

public class Runner {
	private static final Random rnd = new Random();

	public void halt() throws IOException {
		if (rnd.nextBoolean()) {
			throw new RuntimeException ();
		}
		else {
			throw new IOException();
		}
	}
}
```

Напишите код, который создает объекты данного класса и вызывает метод *halt*. В случае, если в методе было брошено *RuntimeException*, вывести в консоль *halt*; в противном случае – пробросить исключение наверх.

Задание 6

Дан класс:

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.file.
FileSystemAlreadyExistsException;
import java.util.Random;

public class XmlReader {
      private static final Random rnd = new Random();

      public void read() throws IOException {
            switch (rnd.nextInt(3)) {
                  case 1:
                        throw new
NullPointerException();
                  case 2:
                        throw new Error();
                  case 3:
                        throw new
FileNotFoundException();
                  default:
                        throw new
FileSystemAlreadyExistsException();
            }
      }
}
```

Создайте метод, который принимает массив объектов данного класса, и вызывает у каждого объекта метод ***read***. Если при исполнении будет брошено исключение `FileSystemAlreadyExistsException`, поймать исключение и бросить исключение `FileNotFoundException`.