

Лабораторная работа №2

Задание №1

Тут слишком всё ещё слишком много написано! Упростите код.

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        ClickHandler clickHandler = new ClickHandler() {
            @Override
            public void click() {
                System.out.println("click");
            }
        };
        clickHandler.click();
    }
}
interface ClickHandler{
    void click();
}
```

Задание №2

Создайте объект кнопки и обработайте нажатие

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
    }
}
interface ClickHandler{
    void click();
}
class Button{
    String name;
    ClickHandler clickHandler;
    public Button(String name) {
        this.name = name;
    }
}
```

```

    public void setOnClickListener(ClickHandler c){
        this.clickHandler = c;
        c.click();
    }
}

```

Задание №3

Используя данный код создайте метод, взаимодействующий с массивом с помощью лямбда выражения и воспользуйтесь поочерёдно методами класса Helper.

```

import java.util.*;
public class Main {
    public static void main(String[] args) {
        int[] nums = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
        Expression expr;
    }
}
class Helper{

    static boolean isEven(int n){

        return n%2 == 0;
    }

    static boolean isPositive(int n){

        return n > 0;
    }
}
interface Expression{
    boolean isEqual(int n);
}

```

Задание №4

Просто сделайте свой класс потока и создайте его объект внутри кода с вызовом метода start()

Задание №5

Синхронизируйте данный код, чтобы каждый поток считал от 1 до 4

```

import java.util.*;
public class Program {

    public static void main(String[] args) {

        CommonResource commonResource= new
CommonResource();
        for (int i = 1; i < 6; i++){

            Thread t = new Thread(new
CountThread(commonResource));
            t.setName("Thread "+ i);
            t.start();
        }
    }

    class CommonResource{

        int x=0;
    }

    class CountThread implements Runnable{

        CommonResource res;
        CountThread(CommonResource res){
            this.res=res;
        }
        public void run(){
            res.x=1;
            for (int i = 1; i < 5; i++){
                System.out.printf("%s %d \n",
Thread.currentThread().getName(), res.x);
                res.x++;
                try{
                    Thread.sleep(100);
                }
                catch(InterruptedException e){}
            }
        }
    }
}

```



Задание №6

Пять безмолвных философов сидят вокруг круглого стола, перед каждым философом стоит тарелка спагетти.

1. Вилки лежат на столе между каждой парой ближайших философов.
2. Каждый философ может либо есть, либо размышлять.
3. Философ может есть только тогда, когда держит две вилки — взятую справа и слева.
4. Взятие каждой вилки и возвращение её на стол являются отдельными действиями, которые должны выполняться одно за другим.

За сколько действий поедят все философы, если они начнут есть по порядку? Принимаю только код(индивидуальные решения приветствуются)