

# Лабораторная работа №5

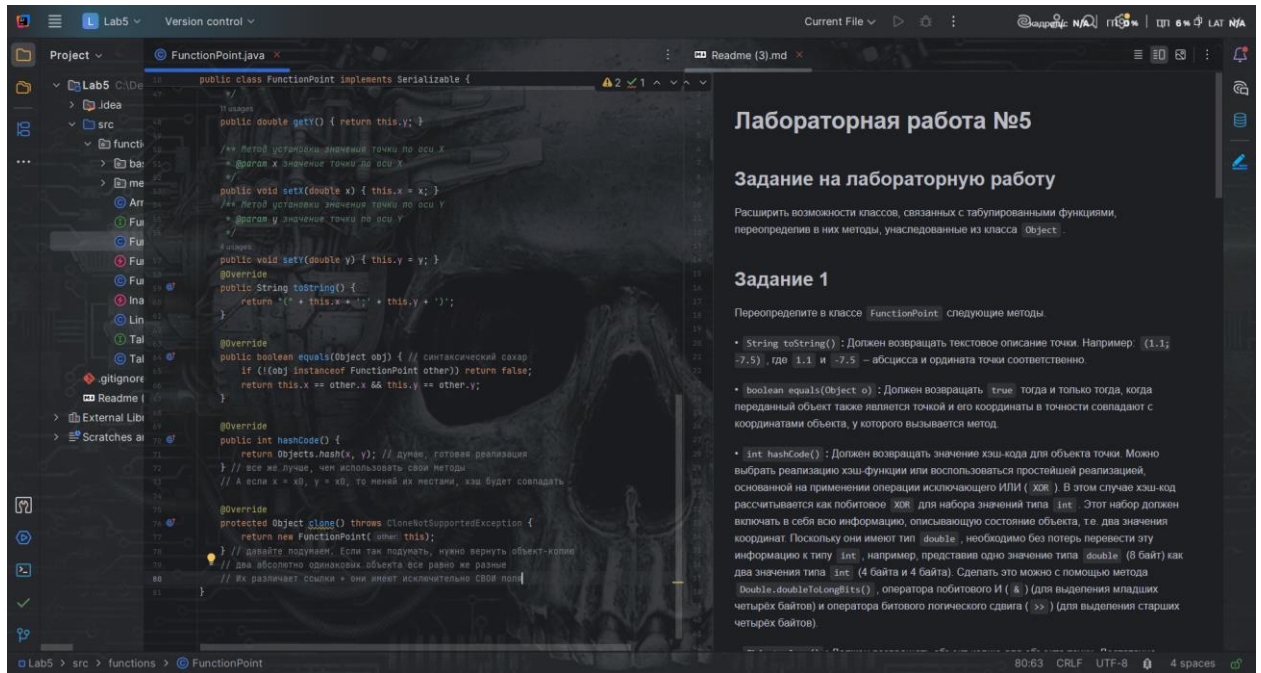
*Выполнил: Беляев Дмитрий Михайлович  
Студент 6203-010302D группы*

# Ход выполнения

Первым делом было прочитано ТЗ. Спасибо за внимание. Можно считать, это самая легкая лабораторная работа...

## Задание 1

Единственный все же имеется вопрос – только к clone(). По сути, тут достаточно вызвать конструктор и вот копия ровно с такими же значениями. В остальном вопросов задание не возникло(см скрин 1)



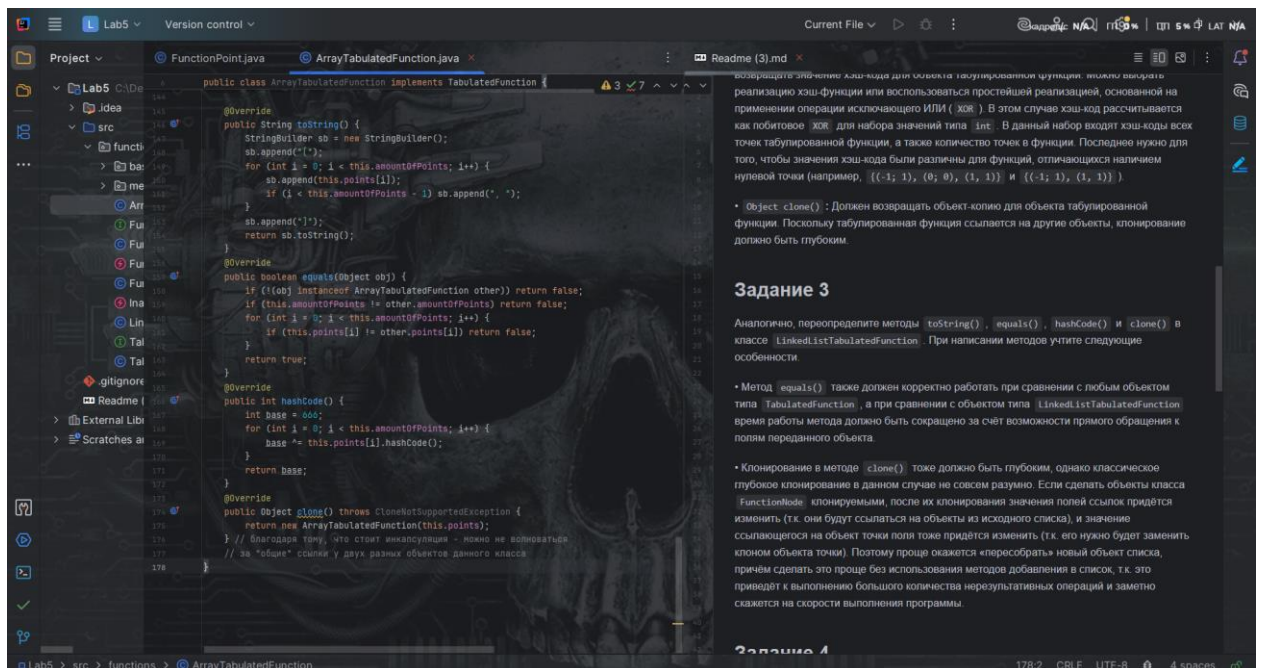
(скрин 1)

Есть небольшая странность в toString. Это я использовать исключительно для того, чтобы кучу, связанную со итернированными строками, не пополнять новыми. Одной строки по идее было бы достаточно. Можно было обойтись String.valueOf(char), но смысла особо нет

*Задание выполнено*

## Задание 2

Здесь нечего комментировать почти. Максимум – для клонирования достаточно конструктора копирования(а там реализована уже инкапсуляция). Результат см скрин 2

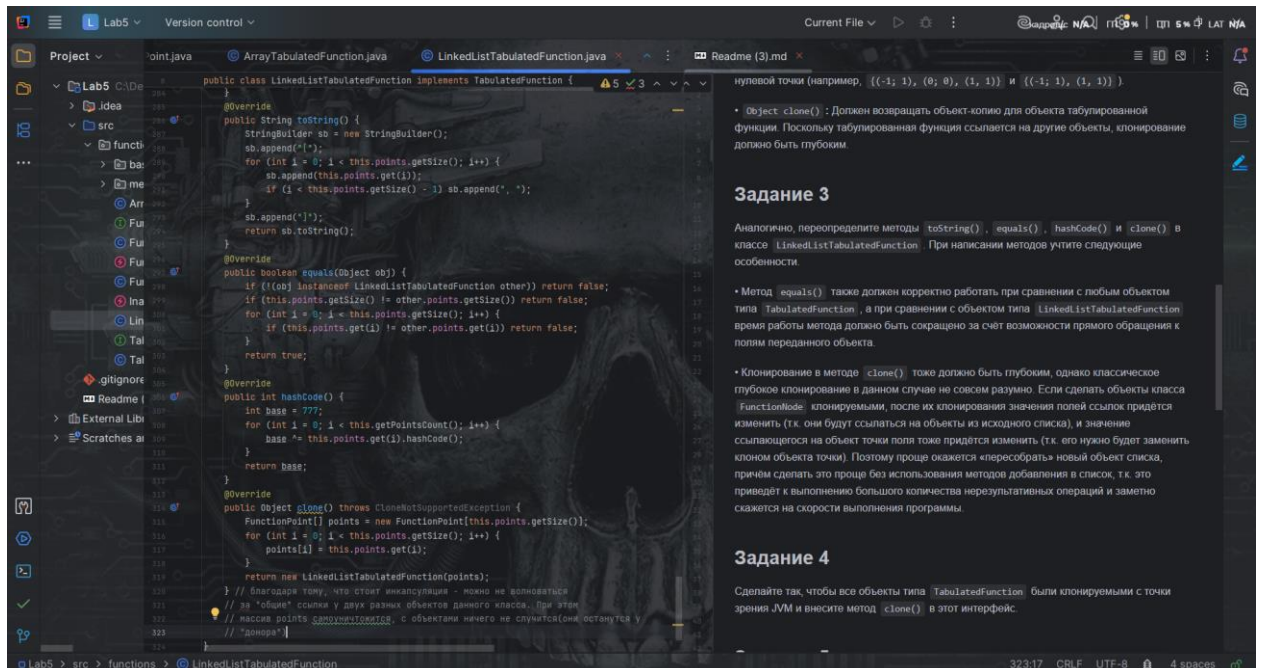


(скрин 2)

Задание выполнено

## Задание 3

Просто копируем реализацию, редактируем(см скрин 3)



(скрин 3)

Задание выполнено

## Задание 4

Результат см скрины 4-5

```

7 usages 2 implementations
public interface TabulatedFunction extends Function, Serializable, Cloneable
/**
 * Метод получения кол-ва точек в данном классе(объекте)
 * @return кол-во точек в данном классе(объекте)

```

(скрин 4)

```

/**
 * Метод клонирования(копирования)
 * @return объект с ровно такими же данными, но уникальными в плане объектов
 * @throws CloneNotSupportedException я фиг знает, но это зависит от реализации
 */
Object clone() throws CloneNotSupportedException;
2 implementations

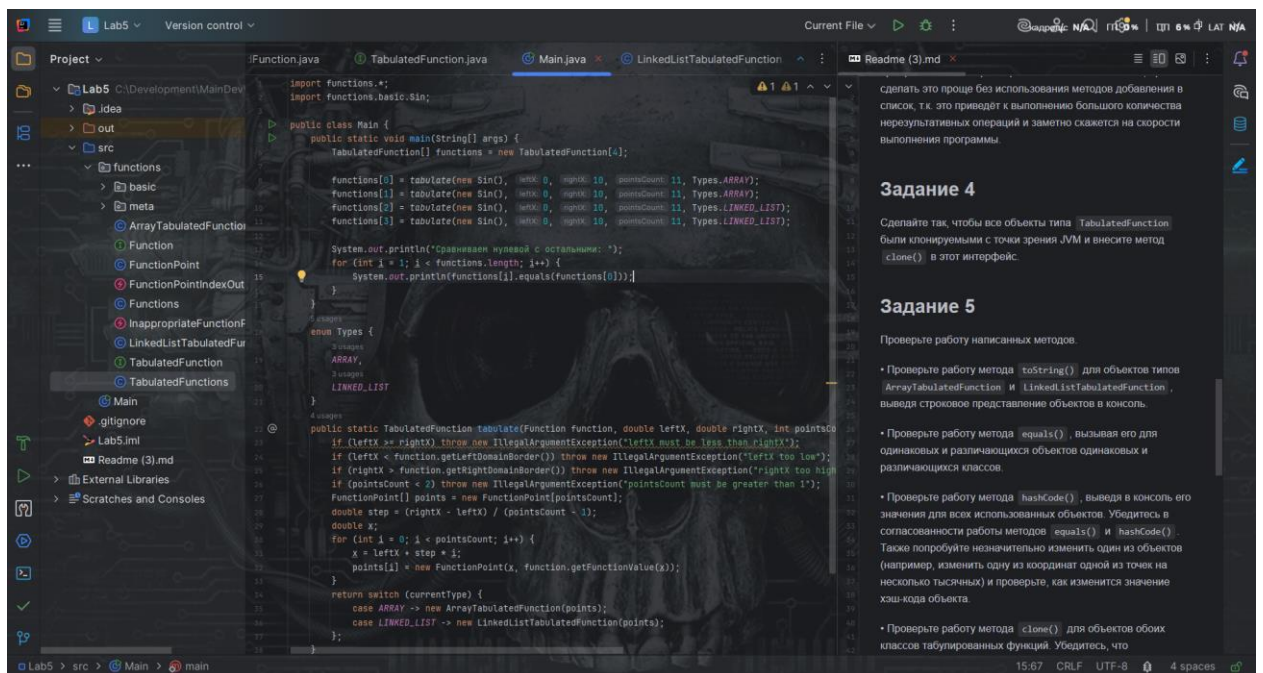
```

(скрин 5)

Задание выполнено

## Задание 5

Применим паттерн Фабрика, напишем пока базовую проверку(см скрин 6)



(скрин 6)

По идее, [0] = [1], [2] = [3]. Запустим(см скрин 7)

```

Сравниваем нулевой с остальными:
false
false
false

Process finished with exit code 0

```

(скрин 7)

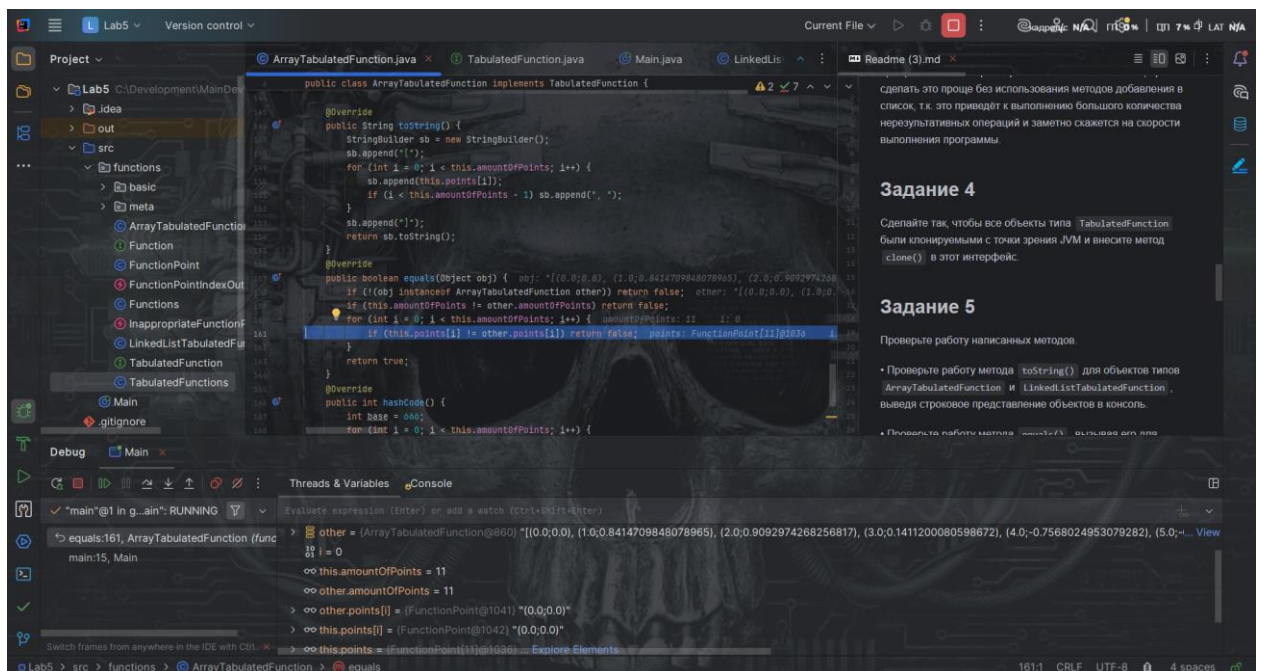


Ага. Видим, что что-то пошло не по плану. Давайте чуть ниже еще выведем их хэшкоды(см скрин 8)

```
false
false
false
Коды:
1483067762
1483067762
1483067617
1483067617
Process finished with exit code 0
```

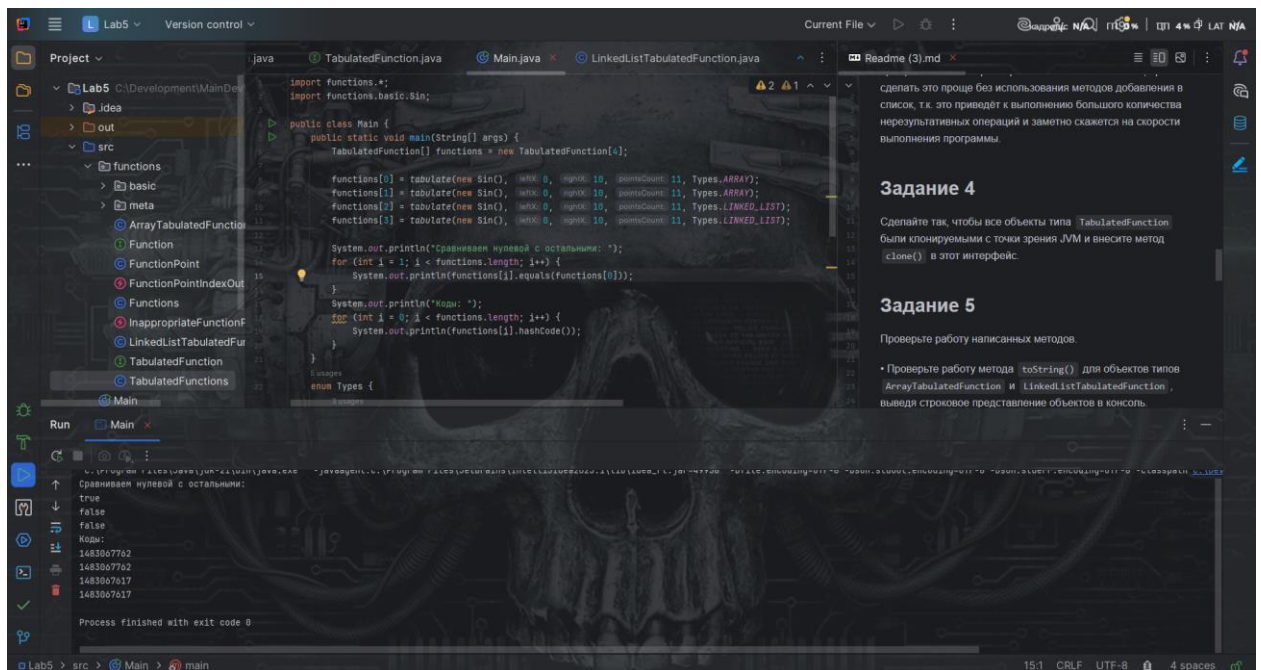
(скрин 8)

Дело не порядок. Придется отлаживать, искать, на каком моменте проваливается проверка. Запустим Debugger, пробежимся до того места, где возвращается false(см скрин 9)



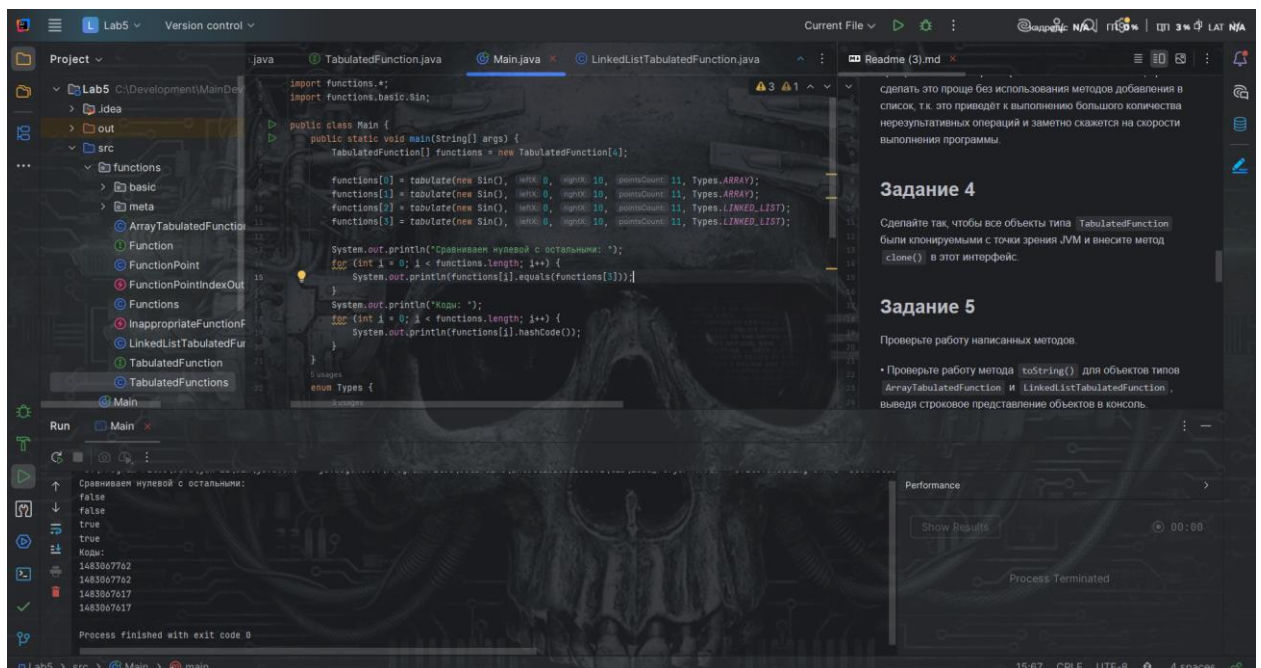
(скрин 9)

Очевидная ошибка найдена. Мы сравниваем ссылки. Исправляем в двух реализациях, заменяем на `equals()`, делаем отрицание. Запускаем заново проверку, получаем результат на скрине 10



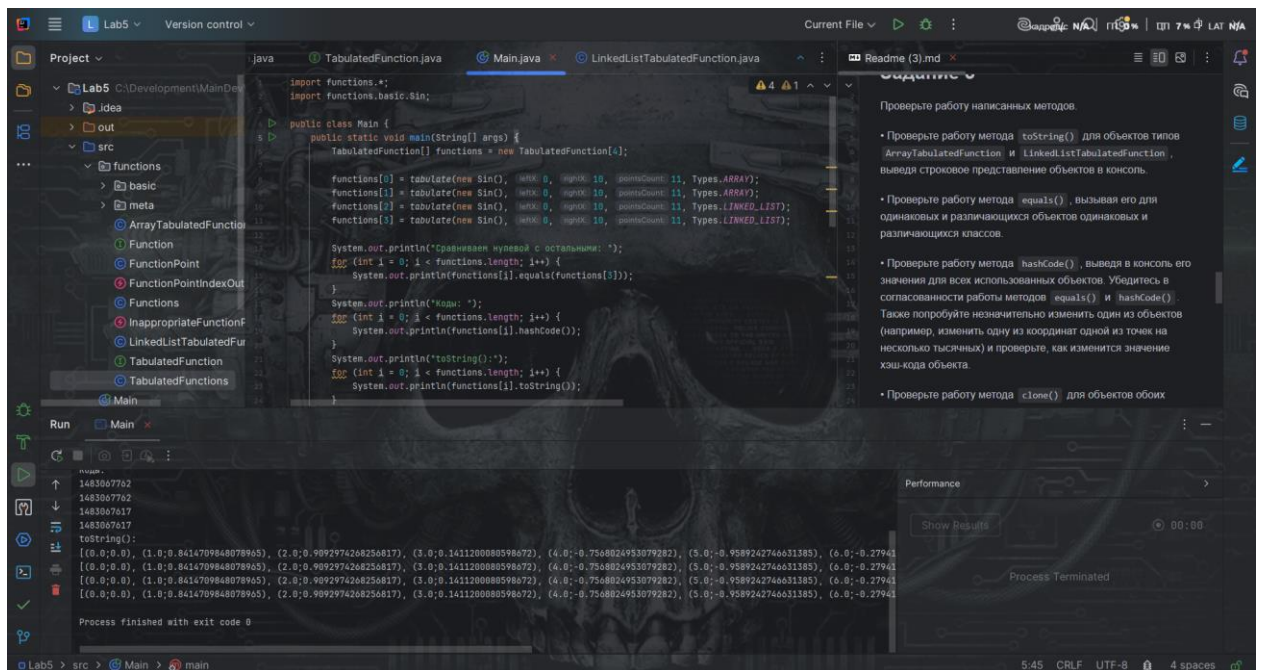
(скрин 10)

Хорошо, проверим для линкеда(см скрин 11)



(скрин 11)

Equals протестировали. Сравниваем строки...(см скрин 12)

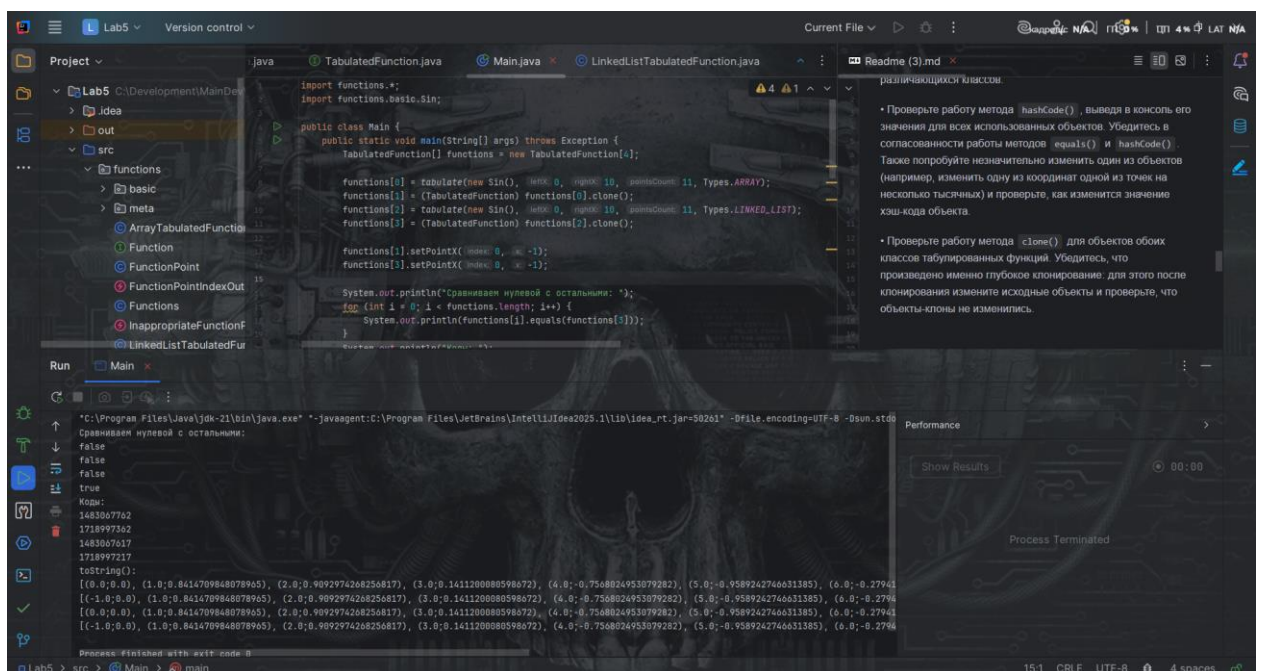


(скрин 12)

Они равны. С этим хорошо.

Давайте попробуем создать 1 и 3 элемента путем клонирования, а затем изменить их какие-либо значения(я изменю левое значение, смещу на -1, значение таким же оставлю).

См скрин 13



(скрин 13)

В принципе, объекты не связаны между собой, с этим все успешно сработано

*Задание выполнено*

**АПДЕЙТ ПОСЛЕ СДАЧИ**



Соответственно, в equals надо добавить еще поддержку сравнения разных классов... реализующих интерфейс TabulatedFunction. См изменения на скринах 14-15

```
public class LinkedListTabulatedFunction implements TabulatedFunction {  
  
    @Override  
    public boolean equals(Object obj) {  
        if (!(obj instanceof TabulatedFunction other)) return false;  
        if (this.getPointsCount() != other.getPointsCount()) return false;  
        for (int i = 0; i < this.points.getSize(); i++) {  
            if (!this.points.get(i).equals(other.getPoint(i))) return false;  
        }  
        return true;  
    }  
}
```

(скрин 14)



```
public class ArrayTabulatedFunction implements TabulatedFunction {  
    }  
  
    @Override  
    public boolean equals(Object obj) {  
        if (!(obj instanceof TabulatedFunction other)) return false;  
        if (this.getPointsCount() != other.getPointsCount()) return false;  
        for (int i = 0; i < this.amountOfPoints; i++) {  
            if (!this.points[i].equals(other.getPoint(i))) return false;  
        }  
        return true;  
    }  
}
```

(скрин 15)

Лабораторная работа была выполнена.