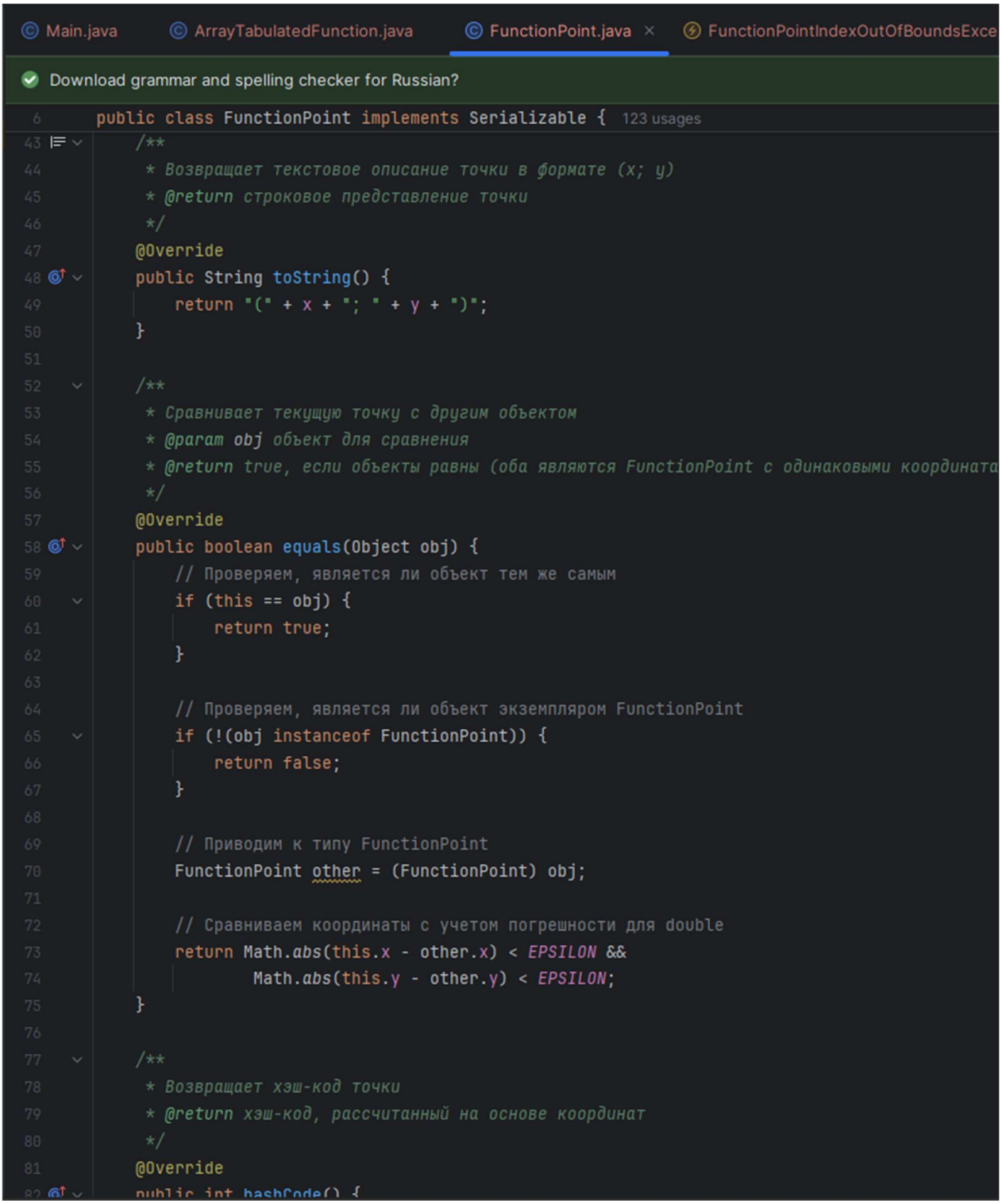
Лабораторная работа №5

Выполнил : Горынин Дмитрий, ст.гр. 6201-120303D.

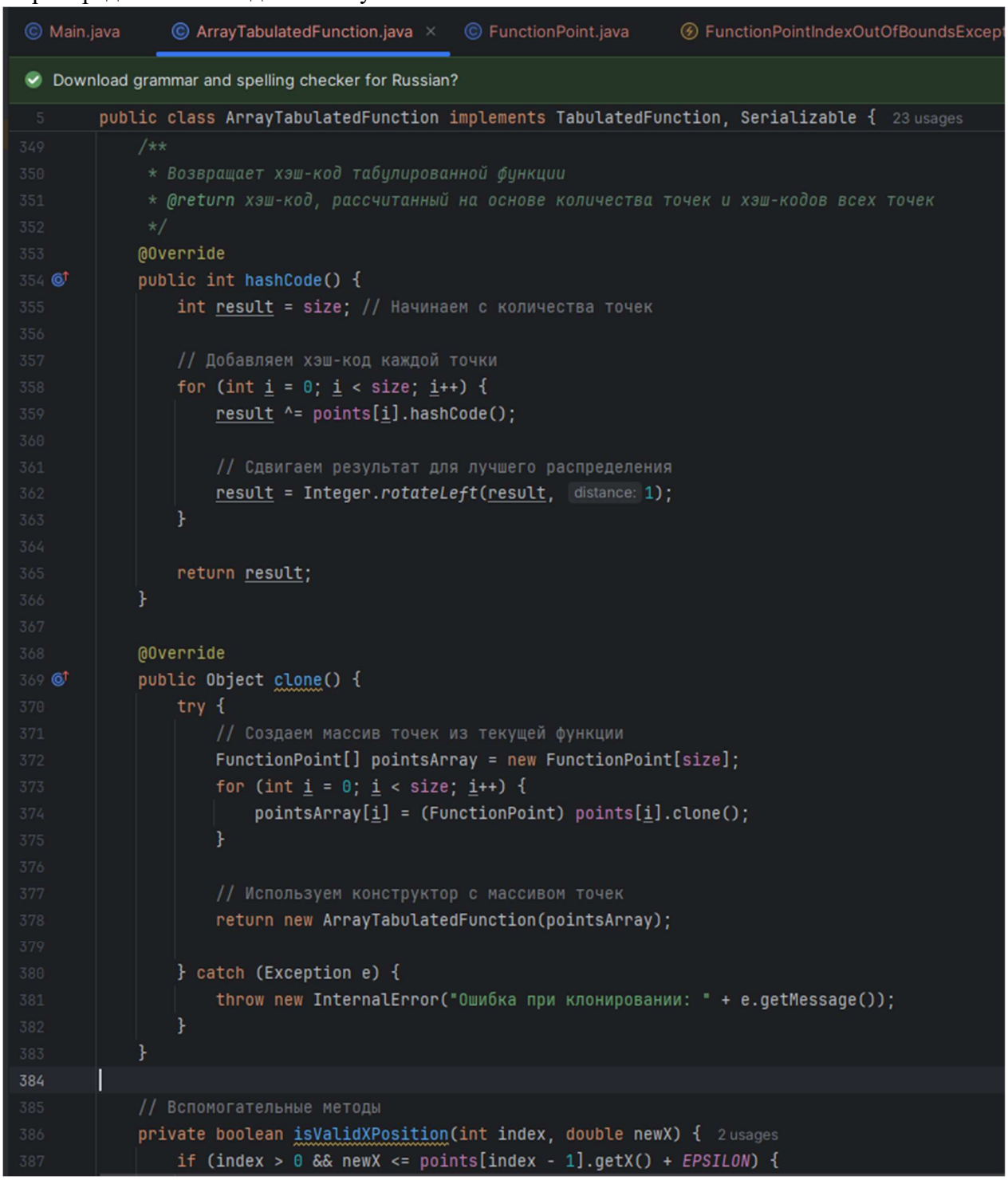
Задание 1.

Переопределение методов в FunctionPoint.



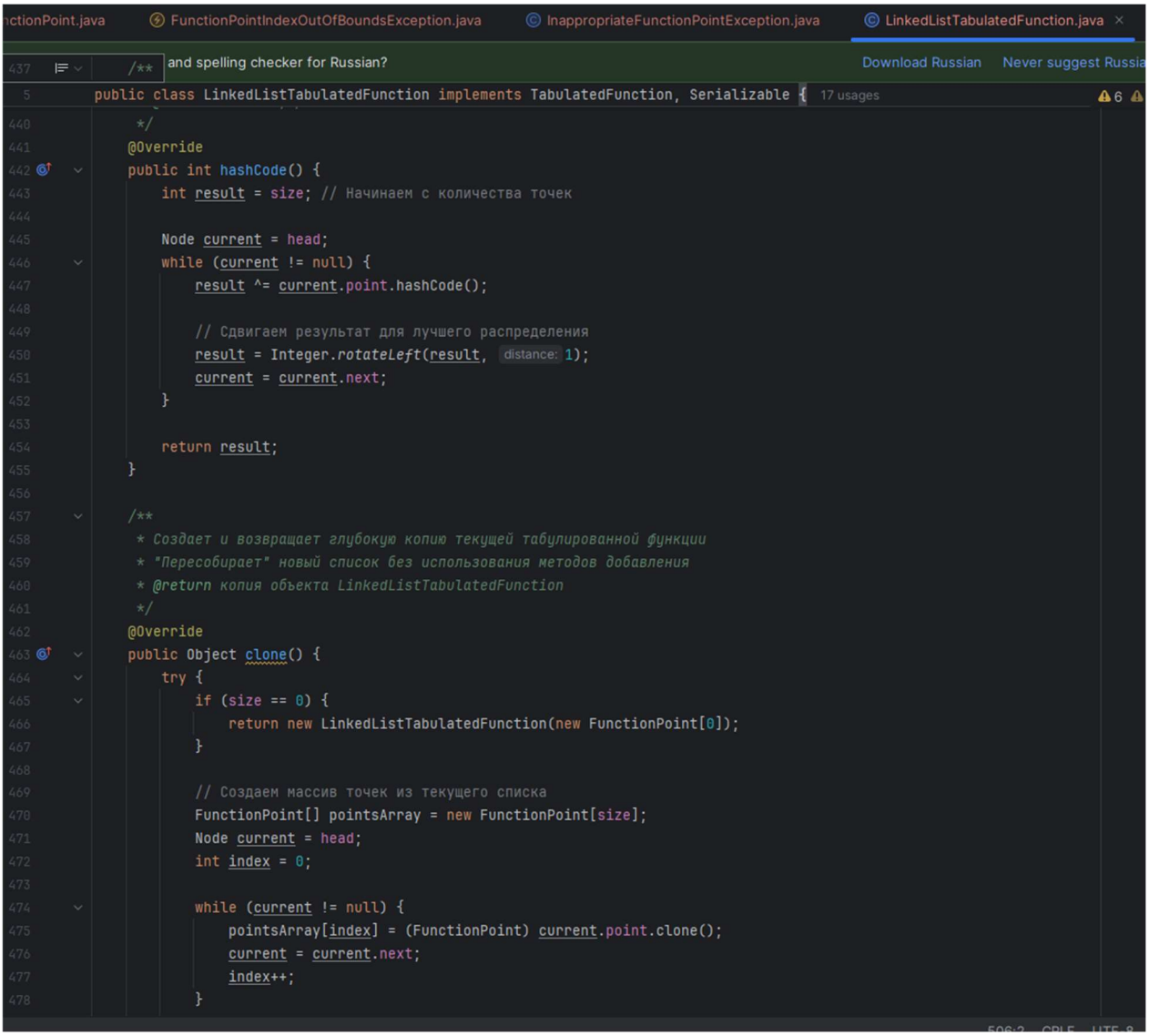
Задание 2.

Переопределение методов в ArrayTabulatedFunction.



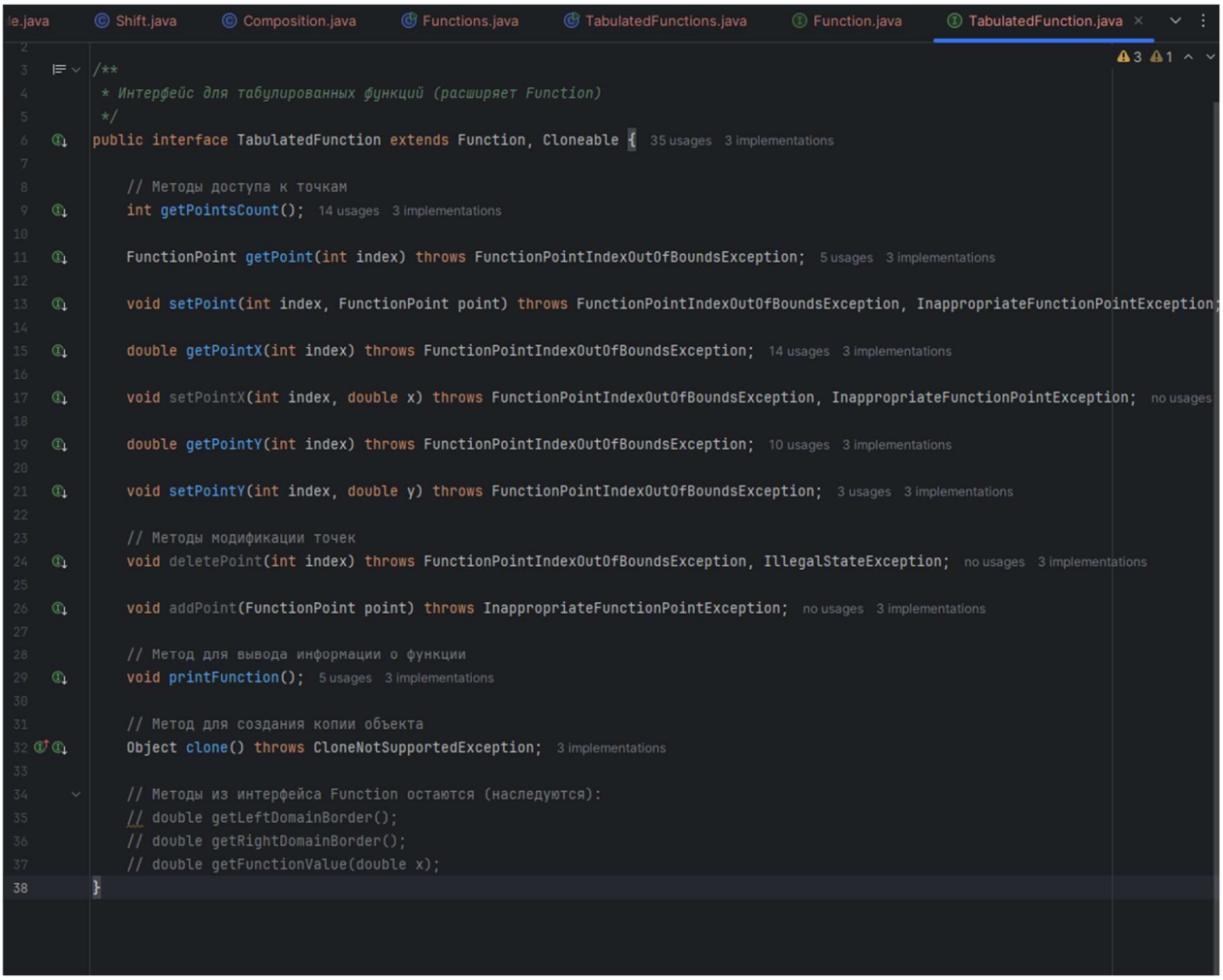
Задание 3.

Аналогично, переопределите методы toString(), equals(), hashCode() и clone() в классе LinkedListTabulatedFunction.



Задание 4.

Сделайте так, чтобы все объекты типа TabulatedFunction были клонируемыми с точки зрения JVM и внесите метод clone() в этот интерфейс.


e.java
Shift.java
Composition.java
Functions.java
TabulatedFunctions.java
Function.java
TabulatedFunction.java
3 1
/**
* Интерфейс для табулированных функций (расширяет Function)
*/
public interface TabulatedFunction extends Function, Cloneable { 35 usages 3 implementations
// Методы доступа к точкам
int getPointsCount(); 14 usages 3 implementations
FunctionPoint getPoint(int index) throws FunctionPointIndexOutOfBoundsException; 5 usages 3 implementations
void setPoint(int index, FunctionPoint point) throws FunctionPointIndexOutOfBoundsException, InappropriateFunctionPointException;
double getPointX(int index) throws FunctionPointIndexOutOfBoundsException; 14 usages 3 implementations
void setPointX(int index, double x) throws FunctionPointIndexOutOfBoundsException, InappropriateFunctionPointException; no usages
double getPointY(int index) throws FunctionPointIndexOutOfBoundsException; 10 usages 3 implementations
void setPointY(int index, double y) throws FunctionPointIndexOutOfBoundsException; 3 usages 3 implementations
// Методы модификации точек
void deletePoint(int index) throws FunctionPointIndexOutOfBoundsException, IllegalStateException; no usages 3 implementations
void addPoint(FunctionPoint point) throws InappropriateFunctionPointException; no usages 3 implementations
// Метод для вывода информации о функции
void printFunction(); 5 usages 3 implementations
// Метод для создания копии объекта
Object clone() throws CloneNotSupportedException; 3 implementations
// Методы из интерфейса Function остаются (наследуются):
// double getLeftDomainBorder();
// double getRightDomainBorder();
// double getFunctionValue(double x);
}

Вывод Main:

```
=== ТЕСТИРОВАНИЕ МЕТОДОВ TABULATED FUNCTION ===

1. СОЗДАНИЕ ТЕСТОВЫХ ОБЪЕКТОВ:
Создано:
  - 3 объекта ArrayTabulatedFunction
  - 3 объекта LinkedListTabulatedFunction

2. ТЕСТ МЕТОДА toString():

ArrayTabulatedFunction arrayFunc1:
  {(1.0; 2.0), (2.0; 4.0), (3.0; 6.0), (4.0; 8.0), (5.0; 10.0)}

ArrayTabulatedFunction arrayFunc2:
  {(0.0; 0.0), (2.5; 0.0), (5.0; 0.0), (7.5; 0.0), (10.0; 0.0)}

LinkedListTabulatedFunction linkedListFunc1:
  {(1.0; 2.0), (2.0; 4.0), (3.0; 6.0), (4.0; 8.0), (5.0; 10.0)}

LinkedListTabulatedFunction linkedListFunc2:
  {(0.0; 0.0), (2.5; 0.0), (5.0; 0.0), (7.5; 0.0), (10.0; 0.0)}

3. ТЕСТ МЕТОДА equals():

Сравнение одинаковых объектов одного класса:
  arrayFunc1.equals(arrayFunc1): true
  arrayFunc1.equals(arrayFunc1Copy): true

Сравнение разных объектов одного класса:
  arrayFunc1.equals(arrayFunc2): false
```

```
Сравнение объектов разных классов с одинаковыми точками:
  arrayFunc1.equals(linkedListFunc1Copy): true

Сравнение объектов разных классов с разными точками:
  arrayFunc2.equals(linkedListFunc2): true

Сравнение с null:
  arrayFunc1.equals(null): false

Сравнение с объектом другого типа:
  arrayFunc1.equals("строка"): false

4. ТЕСТ МЕТОДА hashCode():

Хэш-коды объектов:
  arrayFunc1.hashCode(): -14679889
  arrayFunc1Copy.hashCode(): -14679889
  arrayFunc2.hashCode(): -2133851993
  linkedListFunc1.hashCode(): -14679889
  linkedListFunc1Copy.hashCode(): -14679889
  linkedListFunc2.hashCode(): -2133851993

Проверка согласованности equals() и hashCode():
  arrayFunc1.equals(arrayFunc1Copy): true
  arrayFunc1.hashCode() == arrayFunc1Copy.hashCode(): true
  arrayFunc1.equals(arrayFunc2): false
  arrayFunc1.hashCode() == arrayFunc2.hashCode(): false
```

```
Тест изменения объекта и хэш-кода:
  Исходный хэш-код arrayFunc3: -17301337
  Меняем координату Y у точки с индексом 2 с 3.0 на 3.001...
  Новый хэш-код arrayFunc3: -1289035411
  Хэш-коды различаются

5. ТЕСТ МЕТОДА clone():

Тест клонирования ArrayTabulatedFunction:
  Исходный объект arrayFunc1: {(1.0; 2.0), (2.0; 4.0), (3.0; 6.0), (4.0; 8.0), (5.0; 10.0)}
  Клон arrayFunc1Clone: {(1.0; 2.0), (2.0; 4.0), (3.0; 6.0), (4.0; 8.0), (5.0; 10.0)}
  arrayFunc1 == arrayFunc1Clone: false
  arrayFunc1.equals(arrayFunc1Clone): true

Проверка глубокого клонирования (ArrayTabulatedFunction):
  Изменяем исходный объект arrayFunc1...
  Исходный объект arrayFunc1 после изменения: {(1.0; 2.0), (2.0; 4.0), (3.0; 100.0), (4.0; 8.0), (5.0; 10.0)}
  Клон arrayFunc1Clone после изменения оригинала: {(1.0; 2.0), (2.0; 4.0), (3.0; 6.0), (4.0; 8.0), (5.0; 10.0)}
  Клон изменился? true

Тест клонирования LinkedListTabulatedFunction:
  Исходный объект linkedListFunc1: {(1.0; 2.0), (2.0; 4.0), (3.0; 6.0), (4.0; 8.0), (5.0; 10.0)}
  Клон linkedListFunc1Clone: {(1.0; 2.0), (2.0; 4.0), (3.0; 6.0), (4.0; 8.0), (5.0; 10.0)}
  linkedListFunc1 == linkedListFunc1Clone: false
  linkedListFunc1.equals(linkedListFunc1Clone): true
```

```
Проверка глубокого клонирования (LinkedListTabulatedFunction):
  Изменяем исходный объект linkedListFunc1...
  Исходный объект linkedListFunc1 после изменения: {(1.0; 2.0), (2.0; 4.0), (3.0; 200.0), (4.0; 8.0), (5.0; 10.0)}
  Клон linkedListFunc1Clone после изменения оригинала: {(1.0; 2.0), (2.0; 4.0), (3.0; 6.0), (4.0; 8.0), (5.0; 10.0)}
  Клон изменился? true

6. ДОПОЛНИТЕЛЬНЫЕ ТЕСТЫ:

Тест с функцией из 2 точек:
  smallFunc.toString(): {(0.0; 0.0), (1.0; 0.0)}
  smallFunc.hashCode(): 2145386504

Тест сравнения ArrayTabulatedFunction и LinkedListTabulatedFunction:
  identicalArrayFunc.toString(): {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0)}
  identicalLinkedListFunc.toString(): {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0)}
  identicalArrayFunc.equals(identicalLinkedListFunc): true
  identicalLinkedListFunc.equals(identicalArrayFunc): true

=== ТЕСТИРОВАНИЕ ЗАВЕРШЕНО ===

Process finished with exit code 0
```