ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ имени академика С.П. КОРОЛЁВА»

КАФЕДРА «ТЕХНИЧЕСКАЯ КИБЕРНЕТИКА»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

«Объектно-ориентированное программирование»

ЛАБОРАТОРНАЯ РАБОТА №7

Студент Ветров И.Р.

Группа 6301-030301D

Руководитель Борисов Д. С.

Оценка _____

САМАРА 2025

Задание №1

В интерфейсе TabulatedFunction добавил необходимый родительский тип.

```java
package functions;

public interface TabulatedFunction extends Iterable<FunctionPoint> {
    double getLeftDomainBorder();
    double getRightDomainBorder();
    double getFunctionValue(double x);
    int getPointsCount();
    FunctionPoint getPoint(int index) throws FunctionPointIndexOutOfBoundsException;
    void setPoint(int index, FunctionPoint point) throws FunctionPointIndexOutOfBoundsException, InappropriateFunctionPointException;
    double getPointX(int index) throws FunctionPointIndexOutOfBoundsException;
    void setPointX(int index, double x) throws FunctionPointIndexOutOfBoundsException, InappropriateFunctionPointException;
    double getPointY(int index) throws FunctionPointIndexOutOfBoundsException;
    void setPointY(int index, double y) throws FunctionPointIndexOutOfBoundsException;
    void deletePoint(int index) throws FunctionPointIndexOutOfBoundsException, IllegalStateException;
    void addPoint(FunctionPoint point) throws InappropriateFunctionPointException;

    TabulatedFunction clone();
}
```

ArrayTabulatedFunction

```java
    @Override
    public Iterator<FunctionPoint> iterator() {
        return new Iterator<FunctionPoint>() {
            private int currentIndex = 0;

            @Override
            public boolean hasNext() {
                return currentIndex < pointsCount;
            }

            @Override
            public FunctionPoint next() {
                if (!hasNext()) {
                    throw new NoSuchElementException(s: "Нет следующего элемента");
                }
                // Возвращаем копию точки для защиты инкапсуляции
                return new FunctionPoint(points[currentIndex++]);
            }

            @Override
            public void remove() {
                throw new UnsupportedOperationException(message: "Удаление не поддерживается");
            }
        };
    }
```

LinkedListTabulatedFunction аналогично.

```java
@Override
    public Iterator<FunctionPoint> iterator() {
        return new Iterator<FunctionPoint>() {
            private FunctionNode currentNode = head.getNext();

            @Override
            public boolean hasNext() {
                return currentNode != head;
            }

            @Override
            public FunctionPoint next() {
                if (!hasNext()) {
                    throw new NoSuchElementException(s: "Нет следующего элемента");
                }
                // Возвращаем копию точки для защиты инкапсуляции
                FunctionPoint point = new FunctionPoint(currentNode.getPoint());
                currentNode = currentNode.getNext();
                return point;
            }

            @Override
            public void remove() {
                throw new UnsupportedOperationException(message: "Удаление не поддерживается");
            }
        };
    }
```

Проверка работы методов, реализованная в Main

```
ArrayTabulatedFunction

(1.0; 1.0)
(4.5; 5.0)
(8.0; 3.0)
(11.5; 10.0)
(15.0; 2.0)

LinkedListTabulatedFunction

(1.0; 1.0)
(4.5; 5.0)
(8.0; 3.0)
(11.5; 10.0)
(15.0; 2.0)
```

Задание №2

В пакете functions был описан базовый интерфейс фабрик табулированных функций, также были добавлены три перегруженных метода.

```java
package functions;

public interface TabulatedFunctionFactory {
    TabulatedFunction createTabulatedFunction(FunctionPoint[] points);
    TabulatedFunction createTabulatedFunction(double leftX, double rightX, int pointsCount);
    TabulatedFunction createTabulatedFunction(double leftX, double rightX, double[] values);
}
```

Также были добавлены классы фабрик в ArrayTabulatedFunction и LinkedListTabulatedFunction

```
public static class ArrayTabulatedFunctionFactory implements TabulatedFunctionFactory {
    @Override
    public TabulatedFunction createTabulatedFunction(FunctionPoint[] points) {
        return new ArrayTabulatedFunction(points);
    }

    @Override
    public TabulatedFunction createTabulatedFunction(double leftX, double rightX, int pointsCount) {
        return new ArrayTabulatedFunction(leftX, rightX, pointsCount);
    }

    @Override
    public TabulatedFunction createTabulatedFunction(double leftX, double rightX, double[] values) {
        return new ArrayTabulatedFunction(leftX, rightX, values);
    }
}
```

```
public static class LinkedListTabulatedFunctionFactory implements TabulatedFunctionFactory {
    @Override
    public TabulatedFunction createTabulatedFunction(FunctionPoint[] points) {
        return new LinkedListTabulatedFunction(points);
    }

    @Override
    public TabulatedFunction createTabulatedFunction(double leftX, double rightX, int pointsCount) {
        return new LinkedListTabulatedFunction(leftX, rightX, pointsCount);
    }

    @Override
    public TabulatedFunction createTabulatedFunction(double leftX, double rightX, double[] values) {
        return new LinkedListTabulatedFunction(leftX, rightX, values);
    }
}
```

В TabulatedFunctions создал три перегруженных метода

```
public static TabulatedFunction createTabulatedFunction(FunctionPoint[] points) {
    return factory.createTabulatedFunction(points);
}

public static TabulatedFunction createTabulatedFunction(double leftX, double rightX, int pointsCount) {
    return factory.createTabulatedFunction(leftX, rightX, pointsCount);
}

public static TabulatedFunction createTabulatedFunction(double leftX, double rightX, double[] values) {
    return factory.createTabulatedFunction(leftX, rightX, values);
}
```

Проверка

```
class functions.ArrayTabulatedFunction
class functions.LinkedListTabulatedFunction
class functions.ArrayTabulatedFunction
```

Задание №3

Добавил еще 3 перегруженных метода в TabulatedFunctions

```
public static TabulatedFunction createTabulatedFunction(Class<? extends TabulatedFunction> functionClass,
                                                         FunctionPoint[] points) {
    try {
        Constructor<? extends TabulatedFunction> constructor =
            functionClass.getConstructor(...parameterTypes: FunctionPoint[].class);

        return constructor.newInstance((Object) points);
    } catch (NoSuchMethodException | InstantiationException |
             IllegalAccessException | InvocationTargetException e) {
        throw new IllegalArgumentException(message: "Ошибка при создании объекта через рефлексию", e);
    }
}

public static TabulatedFunction createTabulatedFunction(Class<? extends TabulatedFunction> functionClass,
                                                         double leftX, double rightX, int pointsCount) {
    try {
        Constructor<? extends TabulatedFunction> constructor =
            functionClass.getConstructor(...parameterTypes: double.class, double.class, int.class);

        return constructor.newInstance(leftX, rightX, pointsCount);
    } catch (NoSuchMethodException | InstantiationException |
             IllegalAccessException | InvocationTargetException e) {
        throw new IllegalArgumentException(message: "Ошибка при создании объекта через рефлексию", e);
    }
}

public static TabulatedFunction createTabulatedFunction(Class<? extends TabulatedFunction> functionClass,
                                                         double leftX, double rightX, double[] values) {
    try {
        Constructor<? extends TabulatedFunction> constructor =
            functionClass.getConstructor(...parameterTypes: double.class, double.class, double[].class);

        return constructor.newInstance(leftX, rightX, values);
    } catch (NoSuchMethodException | InstantiationException |
             IllegalAccessException | InvocationTargetException e) {
        throw new IllegalArgumentException(message: "Ошибка при создании объекта через рефлексию", e);
    }
}
```

```
ArrayTabulatedFunction

(1.0; 1.0)
(4.5; 5.0)
(8.0; 3.0)
(11.5; 10.0)
(15.0; 2.0)

LinkedListTabulatedFunction

(1.0; 1.0)
(4.5; 5.0)
(8.0; 3.0)
(11.5; 10.0)
(15.0; 2.0)

-------

class functions.ArrayTabulatedFunction
class functions.LinkedListTabulatedFunction
class functions.ArrayTabulatedFunction

-------

class functions.ArrayTabulatedFunction
{(0,000; 0,000), (5,000; 0,000), (10,000; 0,000)}
class functions.ArrayTabulatedFunction
{(0,000; 0,000), (10,000; 10,000)}
class functions.LinkedListTabulatedFunction
{(0,000; 0,000), (10,000; 10,000)}
class functions.LinkedListTabulatedFunction
{(0,000; 0,000), (0,314; 0,309), (0,628; 0,588), (0,942; 0,809), (1,257; 0,951), (1,571; 1,000), (1,885; 0,951), (2,199; 0,809), (2,513; 0,588), (2,827; 0,309), (3,142; 0,000)}
```