

Лабораторная работа №5

Лебедев Кирилл Дмитриевич 6204-010302D

## Задание 1

В классе FunctionPoint были переопределены следующие методы:

```
@Override
public String toString() {
    return "(" + x + "; " + y + ")";
}

@Override
public boolean equals(Object o) {
    if (this == o) return true; // Проверка на тот же объект
    if (o == null || getClass() != o.getClass()) return false; // Проверка типа

    FunctionPoint that = (FunctionPoint) o;
    return Double.compare(that.x, x) == 0 && Double.compare(that.y, y) == 0; // Сравнение x и y с учётом NaN
}

@Override
public int hashCode() {
    long xBits = Double.doubleToLongBits(x);
    long yBits = Double.doubleToLongBits(y);

    int xHash = (int)(xBits ^ (xBits >>> 32)); // XOR старшей и младшей половин
    int yHash = (int)(yBits ^ (yBits >>> 32)); // XOR старшей и младшей половин

    return xHash ^ yHash; // Комбинирование хэшей координат
}

@Override
public Object clone() {
    return new FunctionPoint(this.x, this.y); // Создание копии точки
}
```

Результат: методы переопределены и работают корректно.

## Задание 2

В классе ArrayTabulatedFunction были переопределены следующие методы:

```

@Override
public String toString() {
    StringBuilder sb = new StringBuilder();
    sb.append("{");
    for (int i = 0; i < pointsCount; i++) {
        sb.append(points[i].toString()); // Используем toString точки
        if (i < pointsCount - 1) sb.append(", ");
    }
    sb.append("}");
    return sb.toString(); // Формат: {(x1; y1), (x2; y2), ...}
}

@Override
public boolean equals(Object o) {
    if (this == o) return true; // Проверка на тот же объект
    if (!(o instanceof TabulatedFunction)) return false; // Проверка интерфейса

    if (o instanceof ArrayTabulatedFunction) { // Проверка класса
        // Оптимизация для одинакового класса
        ArrayTabulatedFunction other = (ArrayTabulatedFunction) o;
        if (this.pointsCount != other.pointsCount) return false;

        for (int i = 0; i < pointsCount; i++) {
            if (!this.points[i].equals(other.points[i])) return false;
        }
        return true;
    } else {
        // Общий случай для любого TabulatedFunction
        TabulatedFunction other = (TabulatedFunction) o;
        if (this.getPointsCount() != other.getPointsCount()) return false;

        for (int i = 0; i < pointsCount; i++) {
            if (!this.getPoint(i).equals(other.getPoint(i))) return false;
        }
        return true;
    }
}
}

```

```

@Override
public int hashCode() {
    int hash = pointsCount; // Начинаем с количества точек
    for (int i = 0; i < pointsCount; i++) {
        hash ^= points[i].hashCode(); // XOR с хэш-кодами точек
    }
    return hash;
}

@Override
public Object clone() {
    FunctionPoint[] clonedPoints = new FunctionPoint[points.length];
    for (int i = 0; i < pointsCount; i++) {
        clonedPoints[i] = (FunctionPoint) points[i].clone(); // Глубокое клонирование точек
    }

    ArrayTabulatedFunction clone = new ArrayTabulatedFunction();
    clone.points = clonedPoints;
    clone.pointsCount = this.pointsCount;
    return clone;
}

```

Результат: методы переопределены и работают корректно.

### Задание 3

В классе `LinkedListTabulatedFunction` были переопределены следующие методы:

```

@Override
public String toString() {
    StringBuilder sb = new StringBuilder();
    sb.append("{");
    FunctionNode current = head.next;
    while (current != head) {
        sb.append(current.point.toString()); // Используем toString точки
        if (current.next != head) sb.append(", ");
        current = current.next;
    }
    sb.append("}");
    return sb.toString(); // Формат: {(x1; y1), (x2; y2), ...}
}

@Override
public boolean equals(Object o) {
    if (this == o) return true; // Проверка на тот же объект
    if (!(o instanceof TabulatedFunction)) return false; // Проверка интерфейса

    TabulatedFunction other = (TabulatedFunction) o;
    if (this.getPointsCount() != other.getPointsCount()) return false;

    // Оптимизация для одинакового класса
    if (o instanceof LinkedListTabulatedFunction) {
        LinkedListTabulatedFunction otherList = (LinkedListTabulatedFunction) o;
        FunctionNode currentThis = this.head.next;
        FunctionNode currentOther = otherList.head.next;

        while (currentThis != head && currentOther != otherList.head) {
            if (!currentThis.point.equals(currentOther.point)) return false;
            currentThis = currentThis.next;
            currentOther = currentOther.next;
        }
        return true;
    } else {
        // Общий случай для любого TabulatedFunction
        for (int i = 0; i < getPointsCount(); i++) {
            if (!this.getPoint(i).equals(other.getPoint(i))) return false;
        }
        return true;
    }
}
}

```

```

@Override
public int hashCode() {
    int hash = getPointsCount(); // Начинаем с количества точек
    FunctionNode current = head.next;
    while (current != head) {
        hash ^= current.point.hashCode(); // XOR с хэш-кодами точек
        current = current.next;
    }
    return hash;
}

@Override
public Object clone() {
    // Создаем массив точек для нового списка
    FunctionPoint[] pointsArray = new FunctionPoint[pointCounts];
    FunctionNode current = head.next;

    for (int i = 0; i < pointCounts; i++) {
        pointsArray[i] = (FunctionPoint) current.point.clone(); // Клонировем каждую точку
        current = current.next;
    }

    // Создаем новый объект через конструктор с массивом точек
    return new LinkedListTabulatedFunction(pointsArray);
}

```

Результат: методы переопределены и работают корректно.

#### Задание 4

Все объекты типа TabulatedFunction клонируемы. Это было сделано с помощью интерфейса Cloneable, который указывает JVM, что объекты этого класса могут быть клонированы.

А также был добавлен метод clone():

Object clone();

Результат: всё работает.

#### Задание 5

Был добавлен метод testObjectMethods() в Main для тестирования методов toString(), equals(), hashCode(), clone().

#### Вывод в main():

Тестирование всего и вся

#### 1. Тест базовых функций

Экспонента:

Область определения:  $[-\text{Infinity}, \text{Infinity}]$

$$\exp(-1,0) = 0,368$$

$$\exp(-0,5) = 0,607$$

$$\exp(0,0) = 1,000$$

$$\exp(0,5) = 1,649$$

$$\exp(1,0) = 2,718$$

Логарифм по основанию 10:

Область определения:  $[0,0, \text{Infinity}]$

$$\log_{10}(0,1) = -1,000$$

$$\log_{10}(1,0) = 0,000$$

$$\log_{10}(10,0) = 1,000$$

Тригонометрические функции на  $[0, \pi]$  с шагом 0.1:

$$x=0,0: \sin=0,000000, \cos=1,000000$$

$$x=0,1: \sin=0,099833, \cos=0,995004$$

$$x=0,2: \sin=0,198669, \cos=0,980067$$

$$x=0,3: \sin=0,295520, \cos=0,955336$$

$$x=0,4: \sin=0,389418, \cos=0,921061$$

$$x=0,5: \sin=0,479426, \cos=0,877583$$

$$x=0,6: \sin=0,564642, \cos=0,825336$$

$$x=0,7: \sin=0,644218, \cos=0,764842$$

$$x=0,8: \sin=0,717356, \cos=0,696707$$

$$x=0,9: \sin=0,783327, \cos=0,621610$$

$$x=1,0: \sin=0,841471, \cos=0,540302$$

$$x=1,1: \sin=0,891207, \cos=0,453596$$

$$x=1,2: \sin=0,932039, \cos=0,362358$$

$$x=1,3: \sin=0,963558, \cos=0,267499$$

x=1,4: sin=0,985450, cos=0,169967  
x=1,5: sin=0,997495, cos=0,070737  
x=1,6: sin=0,999574, cos=-0,029200  
x=1,7: sin=0,991665, cos=-0,128844  
x=1,8: sin=0,973848, cos=-0,227202  
x=1,9: sin=0,946300, cos=-0,323290  
x=2,0: sin=0,909297, cos=-0,416147  
x=2,1: sin=0,863209, cos=-0,504846  
x=2,2: sin=0,808496, cos=-0,588501  
x=2,3: sin=0,745705, cos=-0,666276  
x=2,4: sin=0,675463, cos=-0,737394  
x=2,5: sin=0,598472, cos=-0,801144  
x=2,6: sin=0,515501, cos=-0,856889  
x=2,7: sin=0,427380, cos=-0,904072  
x=2,8: sin=0,334988, cos=-0,942222  
x=2,9: sin=0,239249, cos=-0,970958  
x=3,0: sin=0,141120, cos=-0,989992  
x=3,1: sin=0,041581, cos=-0,999135

## 2. Тест табулированной функции

ArrayTabulatedFunction (2x+1):

- 1 --- (0.0, 1.0)
- 2 --- (1.0, 3.0)
- 3 --- (2.0, 5.0)
- 4 --- (3.0, 7.0)
- 5 --- (4.0, 9.0)
- 6 --- (5.0, 11.0)

Интерполяция ArrayTabulatedFunction:



$$f(0,5) = 2,000$$

$$f(1,5) = 4,000$$

$$f(2,5) = 6,000$$

$$f(3,5) = 8,000$$

$$f(4,5) = 10,000$$

LinkedListTabulatedFunction (2x+1):

$$1 \text{ --- } (0.0, 1.0)$$

$$2 \text{ --- } (1.0, 3.0)$$

$$3 \text{ --- } (2.0, 5.0)$$

$$4 \text{ --- } (3.0, 7.0)$$

$$5 \text{ --- } (4.0, 9.0)$$

$$6 \text{ --- } (5.0, 11.0)$$

Модификация точек:

$$1 \text{ --- } (0.0, 1.0)$$

$$2 \text{ --- } (2.0, 10.0)$$

$$3 \text{ --- } (2.5, 6.25)$$

$$4 \text{ --- } (3.0, 7.0)$$

$$5 \text{ --- } (4.0, 9.0)$$

$$6 \text{ --- } (5.0, 11.0)$$

### 3. Тест мета-функций

Сравнение оригинальных и табулированных функций:

Сравнение  $\sin(x)$  и его табулированного аналога:

$$x=0,0: \text{orig}=0,000000, \text{tab}=0,000000$$

$$x=0,1: \text{orig}=0,099833, \text{tab}=0,097982$$

$$x=0,2: \text{orig}=0,198669, \text{tab}=0,195963$$

x=0,3: orig=0,295520, tab=0,293945  
x=0,4: orig=0,389418, tab=0,385907  
x=0,5: orig=0,479426, tab=0,472070  
x=0,6: orig=0,564642, tab=0,558234  
x=0,7: orig=0,644218, tab=0,643982  
x=0,8: orig=0,717356, tab=0,707935  
x=0,9: orig=0,783327, tab=0,771888  
x=1,0: orig=0,841471, tab=0,835841  
x=1,1: orig=0,891207, tab=0,883993  
x=1,2: orig=0,932039, tab=0,918022  
x=1,3: orig=0,963558, tab=0,952051  
x=1,4: orig=0,985450, tab=0,984808  
x=1,5: orig=0,997495, tab=0,984808  
x=1,6: orig=0,999574, tab=0,984808  
x=1,7: orig=0,991665, tab=0,984808  
x=1,8: orig=0,973848, tab=0,966204  
x=1,9: orig=0,946300, tab=0,932175  
x=2,0: orig=0,909297, tab=0,898147  
x=2,1: orig=0,863209, tab=0,862441  
x=2,2: orig=0,808496, tab=0,798488  
x=2,3: orig=0,745705, tab=0,734535  
x=2,4: orig=0,675463, tab=0,670582  
x=2,5: orig=0,598472, tab=0,594072  
x=2,6: orig=0,515501, tab=0,507908  
x=2,7: orig=0,427380, tab=0,421745  
x=2,8: orig=0,334988, tab=0,334698  
x=2,9: orig=0,239249, tab=0,236716  
x=3,0: orig=0,141120, tab=0,138735  
x=3,1: orig=0,041581, tab=0,040753

Сравнение  $\cos(x)$  и его табулированного аналога:

x=0,0: orig=1,000000, tab=1,000000  
x=0,1: orig=0,995004, tab=0,982723  
x=0,2: orig=0,980067, tab=0,965446  
x=0,3: orig=0,955336, tab=0,948170  
x=0,4: orig=0,921061, tab=0,914355  
x=0,5: orig=0,877583, tab=0,864608  
x=0,6: orig=0,825336, tab=0,814862  
x=0,7: orig=0,764842, tab=0,764620  
x=0,8: orig=0,696707, tab=0,688404  
x=0,9: orig=0,621610, tab=0,612188  
x=1,0: orig=0,540302, tab=0,535972  
x=1,1: orig=0,453596, tab=0,450633  
x=1,2: orig=0,362358, tab=0,357141  
x=1,3: orig=0,267499, tab=0,263648  
x=1,4: orig=0,169967, tab=0,169931  
x=1,5: orig=0,070737, tab=0,070437  
x=1,6: orig=-0,029200, tab=-0,029056  
x=1,7: orig=-0,128844, tab=-0,128549  
x=1,8: orig=-0,227202, tab=-0,224761  
x=1,9: orig=-0,323290, tab=-0,318254  
x=2,0: orig=-0,416147, tab=-0,411747  
x=2,1: orig=-0,504846, tab=-0,504272  
x=2,2: orig=-0,588501, tab=-0,580488  
x=2,3: orig=-0,666276, tab=-0,656704  
x=2,4: orig=-0,737394, tab=-0,732920  
x=2,5: orig=-0,801144, tab=-0,794171  
x=2,6: orig=-0,856889, tab=-0,843917

x=2,7: orig=-0,904072, tab=-0,893664  
x=2,8: orig=-0,942222, tab=-0,940984  
x=2,9: orig=-0,970958, tab=-0,958261  
x=3,0: orig=-0,989992, tab=-0,975537  
x=3,1: orig=-0,999135, tab=-0,992814

Сумма квадратов табулированных син и кос:

$\sin^2(x) + \cos^2(x)$  (табулированные, 10 точек):

x=0,0: result=1,0000000000  
x=0,1: result=0,9753452893  
x=0,2: result=0,9704883234  
x=0,3: result=0,9854291023  
x=0,4: result=0,9849684785  
x=0,5: result=0,9703975808  
x=0,6: result=0,9756244278  
x=0,7: result=0,9993578909  
x=0,8: result=0,9750730614  
x=0,9: result=0,9705859766  
x=1,0: result=0,9858966365  
x=1,1: result=0,9845147652  
x=1,2: result=0,9703137486  
x=1,3: result=0,9759104767  
x=1,4: result=0,9987226923  
x=1,5: result=0,9748077439  
x=1,6: result=0,9706905402  
x=1,7: result=0,9863710813  
x=1,8: result=0,9840679624  
x=1,9: result=0,9702368269  
x=2,0: result=0,9762034362

x=2,1: result=0,9980944042

x=2,2: result=0,9745493369

x=2,3: result=0,9708020143

x=2,4: result=0,9868524365

x=2,5: result=0,9836280701

x=2,6: result=0,9701668157

x=2,7: result=0,9765033061

x=2,8: result=0,9974730266

x=2,9: result=0,9742978404

x=3,0: result=0,9709203989

x=3,1: result=0,9873407022

$3 * \sin(2x)$ :

x=0,000: result=0,000

x=0,785: result=1,148

x=1,571: result=2,121

x=2,356: result=2,772

x=3,142: result=3,000

$\cos(x - \pi/2) + 1$ :

x=0,000: result=1,000

x=0,785: result=1,707

x=1,571: result=2,000

x=2,356: result=1,707

x=3,142: result=1,000

$\sin(x) * \cos(x)$ :

x=0,000: result=0,000

x=0,785: result=0,500

x=1,571: result=0,000

x=2,356: result=-0,500

x=3,142: result=-0,000

#### 4. Тест файловых операций

Экспонента - текстовый формат:

Экспонента (0 до 10, шаг 1):

Оригинал vs Прочитанный (текстовый формат):

x=0,0: orig=1,000000, read=1,000000

x=1,0: orig=2,718282, read=2,718282

x=2,0: orig=7,389056, read=7,389056

x=3,0: orig=20,085537, read=20,085537

x=4,0: orig=54,598150, read=54,598150

x=5,0: orig=148,413159, read=148,413159

x=6,0: orig=403,428793, read=403,428793

x=7,0: orig=1096,633158, read=1096,633158

x=8,0: orig=2980,957987, read=2980,957987

x=9,0: orig=8103,083928, read=8103,083928

x=10,0: orig=22026,465795, read=22026,465795

Логарифм - бинарный формат:

Логарифм (0.1 до 10, шаг 1):

Оригинал vs Прочитанный (бинарный формат):

x=0,1: orig=-2,302585, read=-2,302585

x=1,1: orig=0,092705, read=0,092705

x=2,1: orig=0,740233, read=0,740233

x=3,1: orig=1,130147, read=1,130147

x=4,1: orig=1,409999, read=1,409999

x=5,1: orig=1,628429, read=1,628429

x=6,1: orig=1,807603, read=1,807603

x=7,1: orig=1,959502, read=1,959502

x=8,1: orig=2,091344, read=2,091344

x=9,1: orig=2,207812, read=2,207812

## 5. Тест сериализации (Serializable)

Оригинальная функция  $\ln(\exp(x)) = x$  (0 до 10, 11 точек):

$f(0,0) = 0,000000$

$f(1,0) = 1,000000$

$f(2,0) = 2,000000$

$f(3,0) = 3,000000$

$f(4,0) = 4,000000$

$f(5,0) = 5,000000$

$f(6,0) = 6,000000$

$f(7,0) = 7,000000$

$f(8,0) = 8,000000$

$f(9,0) = 9,000000$

$f(10,0) = 10,000000$

Десериализованная функция:

$f(0,0) = 0,000000$

$f(1,0) = 1,000000$

$f(2,0) = 2,000000$

$f(3,0) = 3,000000$

$f(4,0) = 4,000000$

$f(5,0) = 5,000000$

$f(6,0) = 6,000000$

$f(7,0) = 7,000000$

$f(8,0) = 8,000000$

$f(9,0) = 9,000000$

$$f(10,0) = 10,000000$$

#### 6. Тест сериализации (Externalizable)

Оригинальная функция (Externalizable):

$$f(0,0) = 0,000000$$

$$f(1,0) = 1,000000$$

$$f(2,0) = 2,000000$$

$$f(3,0) = 3,000000$$

$$f(4,0) = 4,000000$$

$$f(5,0) = 5,000000$$

$$f(6,0) = 6,000000$$

$$f(7,0) = 7,000000$$

$$f(8,0) = 8,000000$$

$$f(9,0) = 9,000000$$

$$f(10,0) = 10,000000$$

Десериализованная функция (Externalizable):

$$f(0,0) = 0,000000$$

$$f(1,0) = 1,000000$$

$$f(2,0) = 2,000000$$

$$f(3,0) = 3,000000$$

$$f(4,0) = 4,000000$$

$$f(5,0) = 5,000000$$

$$f(6,0) = 6,000000$$

$$f(7,0) = 7,000000$$

$$f(8,0) = 8,000000$$

$$f(9,0) = 9,000000$$

$$f(10,0) = 10,000000$$

#### 7. Тест методов toString(), equals(), hashCode(), clone()



### Тестирование toString()

ArrayTabulatedFunction 1: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

ArrayTabulatedFunction 2: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

ArrayTabulatedFunction 3: {(0.0; 1.0), (1.0; 2.5), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

LinkedListTabulatedFunction 1: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

LinkedListTabulatedFunction 2: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

LinkedListTabulatedFunction 3: {(0.0; 1.0), (1.0; 2.5), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

### Тестирование equals()

arrayFunc1.equals(arrayFunc2): true (ожидается true)

arrayFunc1.equals(arrayFunc3): false (ожидается false)

listFunc1.equals(listFunc2): true (ожидается true)

listFunc1.equals(listFunc3): false (ожидается false)

arrayFunc1.equals(listFunc1): true (ожидается true)

listFunc1.equals(arrayFunc1): true (ожидается true)

arrayFunc1.equals(arrayFunc4): false (ожидается false)

### Тестирование hashCode()

arrayFunc1.hashCode(): 1075052549

arrayFunc2.hashCode(): 1075052549

arrayFunc3.hashCode(): 1074790405

listFunc1.hashCode(): 1075052549

listFunc2.hashCode(): 1075052549

listFunc3.hashCode(): 1074790405

### Проверка согласованности equals() и hashCode():

arrayFunc1.equals(arrayFunc2) && arrayFunc1.hashCode() ==

arrayFunc2.hashCode(): true (ожидается true)

listFunc1.equals(listFunc2) && listFunc1.hashCode() == listFunc2.hashCode(): true  
(ожидается true)

#### Тест изменения объекта

Исходный arrayFunc1: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

Исходный hashCode: 1075052549

После изменения 'Y[2] = 3.001': {(0.0; 1.0), (1.0; 2.0), (2.0; 3.001), (3.0; 4.0), (4.0; 5.0)}

Новый hashCode: 162421820

Хэш-код изменился: true (ожидается true)

#### Тестирование clone()

Оригинал arrayFunc2: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

Клон arrayClone: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

arrayFunc2.equals(arrayClone): true (ожидается true)

Оригинал listFunc2: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

Клон listClone: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

listFunc2.equals(listClone): true (ожидается true)

#### Проверка глубокого клонирования

После изменения оригиналов:

Оригинал arrayFunc2: {(0.0; 1.0), (1.0; 2.5), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

Клон arrayClone: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

Клон не изменился: 2.0 (ожидается 2.0)

Оригинал listFunc2: {(0.0; 1.0), (1.0; 2.5), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

Клон listClone: {(0.0; 1.0), (1.0; 2.0), (2.0; 3.0), (3.0; 4.0), (4.0; 5.0)}

Клон не изменился: 2.0 (ожидается 2.0)

arrayFunc2.equals(arrayClone): false (ожидается false)

listFunc2.equals(listClone): false (ожидается false)