

Лабораторная №5

Болясников Вадим Андреевич 6204-010302

Задание на лабораторную работу:

Расширить возможности классов, связанных с табулированными функциями, переопределив в них методы, унаследованные из класса Object.

Задание 1

Я переопределил в классе FunctionPoint следующие методы:

- String `toString()`: Метод возвращает текстовое описание точки в формате (x; y), где x и y - абсцисса и ордината точки соответственно. Например: (1.1; -7.5)
- boolean `equals(Object o)`: Метод возвращает true тогда и только тогда, когда переданный объект также является точкой и его координаты в точности совпадают с координатами объекта, у которого вызывается метод. Для корректного сравнения чисел с плавающей точкой используется погрешность `EPSILON = 1e-10`.
- int `hashCode()`: Метод возвращает значение хэш-кода для объекта точки. Реализация основана на применении операции исключающего ИЛИ (XOR) для набора значений типа int. Для преобразования значений типа double используется метод `Double.doubleToLongBits()`, затем каждый long разбивается на два int (старшие и младшие биты) с помощью операторов побитового И (&) и битового сдвига (>>).
- Object `clone()`: Метод возвращает объект-копию для объекта точки. Используется простое клонирование, так как точка не имеет ссылок на другие объекты.

Задание 2

Я переопределил в классе ArrayTabulatedFunction следующие методы:

- String `toString()`: Метод возвращает описание табулированной функции в формате {(x1; y1), (x2; y2), ..., (xn; yn)}, где в круглых скобках указываются координаты точек.
- boolean `equals(Object o)`: Метод возвращает true тогда и только тогда, когда переданный объект также является табулированной функцией и её набор точек в точности совпадает с набором точек функции, у которой вызывается метод. При сравнении с объектом типа - `ArrayTabulatedFunction` время работы сокращено за счёт прямого обращения к элементам состояния переданного объекта.
- int `hashCode()`: Метод возвращает значение хэш-кода для объекта табулированной функции. Хэш-код рассчитывается как побитовое XOR для хэш-кодов всех точек табулированной функции и количества точек в функции.
- Object `clone()`: Метод возвращает объект-копию для объекта табулированной функции. Поскольку табулированная функция ссылается на другие объекты, клонирование является глубоким - клонируется массив точек и каждая точка в нем.

Задание 3

Я переопределил методы `toString()`, `equals()`, `hashCode()` и `clone()` в классе `LinkedListTabulatedFunction`:

- String `toString()`: Метод возвращает описание табулированной функции в том же формате, что и для `ArrayTabulatedFunction`, но с обходом связного списка.
- boolean `equals(Object o)`: Метод корректно работает при сравнении с любым объектом типа `TabulatedFunction`, а при сравнении с объектом типа `LinkedListTabulatedFunction`

время работы сокращено за счёт прямого обращения к полям переданного объекта через обход узлов списка.

- int hashCode(): Метод возвращает значение хэш-кода, рассчитываемое как побитовое XOR для хэш-кодов всех точек и количества точек, с обходом связного списка.

Object clone(): Метод реализует глубокое клонирование через "пересборку" нового объекта списка без использования методов добавления в список для оптимизации производительности.

Задание 4

Я сделал все объекты типа TabulatedFunction клонируемыми с точки зрения JVM и внес метод clone() в этот интерфейс:

Интерфейс TabulatedFunction теперь расширяет интерфейс Cloneable
В интерфейс добавлен метод Object clone()

Задание 5

Я проверил работу написанных методов:

toString(): Проверил для объектов типов ArrayTabulatedFunction и
LinkedListTabulatedFunction, вывел строковое представление объектов в консоль. Формат
корректен: {(0.0; 1.0), (1.0; 3.0), (2.0; 7.0), (3.0; 15.0)}

equals(): Проверил, вызывая его для одинаковых и различающихся объектов одинаковых
и различающихся классов. Метод корректно возвращает true для одинаковых объектов
(даже разных классов) и false для разных объектов.

hashCode(): Вывел в консоль значения хэш-кодов для всех использованных объектов.
Убедился в согласованности работы методов equals() и hashCode() - одинаковые объекты
имеют одинаковые хэш-коды. При незначительном изменении объекта хэш-код
изменяется соответствующим образом.

clone(): Проверил для объектов обоих классов табулированных функций. Убедился, что
произведено именно глубокое клонирование - после изменения исходных объектов
объекты-克隆ы не изменяются, подтверждая их полную независимость.

Все методы работают корректно и согласованно, обеспечивая правильное поведение
объектов в соответствии с контрактами Java.

Вывод:

```
==== ТЕСТИРОВАНИЕ ПЕРЕОПРЕДЕЛЕННЫХ МЕТОДОВ (ЛАБОРАТОРНАЯ 5) ===

1. ТЕСТИРОВАНИЕ МЕТОДА toString()
=====
ArrayTabulatedFunction: {(0.0; 1.0), (1.0; 3.0), (2.0; 7.0), (3.0; 15.0)}
LinkedListTabulatedFunction: {(0.0; 1.0), (1.0; 3.0), (2.0; 7.0), (3.0; 15.0)}

2. ТЕСТИРОВАНИЕ МЕТОДА equals()
=====
array1.equals(array2): true (ожидается: true)
list1.equals(list2): true (ожидается: true)
array1.equals(list1): true (ожидается: true)
array1.equals(array3): false (ожидается: false)
array1.equals(null): false (ожидается: false)

3. ТЕСТИРОВАНИЕ МЕТОДА hashCode()
=====
array1.hashCode(): 1077018628
array2.hashCode(): 1077018628
list1.hashCode(): 1077018628
list2.hashCode(): 1077018628
array1.equals(array2) && array1.hashCode() == array2.hashCode(): true
array1.equals(list1) && array1.hashCode() == list1.hashCode(): true
HashCode до изменения: 1077018628
HashCode после изменения Y[1] на 0.001: 159931453
Хэш-код изменился: true

4. ТЕСТИРОВАНИЕ МЕТОДА clone()
=====
Array – исходная: {(0.0; 1.0), (1.0; 3.0), (2.0; 7.0), (3.0; 15.0)}
Array – клон: {(0.0; 1.0), (1.0; 3.0), (2.0; 7.0), (3.0; 15.0)}
Разные ссылки: true
Равное содержимое: true
После изменения исходного Array:
Исходная: (0.0; 100.0)
Клон: (0.0; 1.0)
Содержимое разное: true
List – исходная: {(0.0; 1.0), (1.0; 3.0), (2.0; 7.0), (3.0; 15.0)}
List – клон: {(0.0; 1.0), (1.0; 3.0), (2.0; 7.0), (3.0; 15.0)}
Разные ссылки: true
Равное содержимое: true
После изменения исходного List:
Исходная: (2.0; 200.0)
Клон: (2.0; 7.0)
Содержимое разное: true
Интерфейс – тип клона: ArrayTabulatedFunction
Интерфейс – клонирование корректно: true

==== ТЕСТИРОВАНИЕ ПЕРЕОПРЕДЕЛЕННЫХ МЕТОДОВ ЗАВЕРШЕНО ===
```

