

Лабораторная работа №5

Автор: Ежова Екатерина

Группа: 6203-010302D

Задание 1

В этом задании я переопределила методы `toString()`, `equals()`, `hashCode()`, `clone()`.

Метод `toString()` возвращает текстовое описание точки. Метод `equals()` проверяет переданный объект с помощью оператора `instanceof` на то, принадлежит ли он классу `FunctionPoint`, и сравнивает значения координат с исходной точкой, если они равны возвращает `true`, иначе – `false`. Метод `hashCode()` возвращает хэш-код объекта. С помощью метода `Double.doubleToLongBits()` преобразуем `x` и `y` в битовое представление, далее сдвигаем на 32 бита вправо получая старшие байты и применяем оператор побитового И, чтобы получить младшие 4 байта, применяем операцию XOR. Метод `clone()` возвращает объект-копию для объекта точки.

Задание 2

Метод `toString()` возвращает текстовое описание табулированной функции. С помощью цикла и метода `toString()` для точки создаем строку, содержащую все точки табулированной функции. В методе `clone()` создаем массив для хранения точек и копируем каждую точку исходной функции в этот массив, возвращаем новый объект, передав в конструктор массив точек. Метод `hashCode()` возвращает значение хэш-кода, которое рассчитывается как побитовое XOR для всех точек табулированной функции и их количества. Метод `equals()` сравнивает функции, если все точки и их количество совпадают, то возвращает `true`, иначе – `false`. Если объект принадлежит такому же классу, то обращаемся напрямую к точке из массива по индексу. С помощью метода `equals()` для точки сравниваем каждую точку и, если хотя бы одна не совпадает, возвращаем `false`. Если объект принадлежит другому классу, то обращаемся к каждой точке через метод `getPoint()`.

Задание 3-4

Все методы реализуются аналогично. Чтобы получить доступ к значению точки, используем метод `getNodeByIndex(i).point` и обратимся к приватному полю. В методе `equals()` используем прямое обращение, если объект принадлежит такому же классу. В интерфейс `TabulatedFunction` вносим метод `clone()`.

Задание 5

```
fun1: y = 3x
{(1.0; 3.0), (12.0; 36.0), (23.0; 69.0), (34.0; 102.0), (45.0; 135.0), (56.0; 168.0), (67.0; 201.0), (78.0; 234.0), (89.0; 267.0), (100.0; 300.0)}

fun2: Функция с значениями из массива
{(1.0; 1.0), (2.0; 2.0), (3.0; 3.0), (4.0; 5.0), (5.0; 9.0), (6.0; 6.0), (7.0; 32.0)}

fun3: log(exp)
{(1.0; 1.0), (2.0; 2.0), (3.0; 3.0), (4.0; 4.0), (5.0; 5.0), (6.0; 6.0), (7.0; 7.0), (8.0; 8.0), (9.0; 9.0), (10.0; 10.0), (11.0; 11.0)}

fun4: fun1 + fun3
{(1.0; 4.0), (2.0; 8.0), (3.0; 12.0), (4.0; 16.0), (5.0; 20.0), (6.0; 24.0), (7.0; 28.0), (8.0; 32.0), (9.0; 36.0), (10.0; 40.0), (11.0; 44.0)}

fun5: Функция с значениями из массива
{(1.0; 3.0), (12.0; 36.0), (23.0; 69.0), (34.0; 102.0), (45.0; 135.0), (56.0; 168.0), (67.0; 201.0), (78.0; 234.0), (89.0; 267.0), (100.0; 300.0)}

fun6: Копия fun4
{(1.0; 4.0), (2.0; 8.0), (3.0; 12.0), (4.0; 16.0), (5.0; 20.0), (6.0; 24.0), (7.0; 28.0), (8.0; 32.0), (9.0; 36.0), (10.0; 40.0), (11.0; 44.0)}

Равенство функций fun1 и fun2 (объекты разных классов)
Функции не равны

Равенство функций fun1 и fun3 (объекты одного класса)
Функции не равны

Равенство функций fun1 и fun5 (объекты разных классов)
Функции равны
```

```
Равенство функций fun4 и fun6 (объекты одного класса)
Функции равны

Хэш-код для fun1
2146021386

Хэш-код для fun2
7208967

Хэш-код для fun3
11

Хэш-код для fun4
2145386507

Хэш-код для fun5
2146021386

Хэш-код для fun6
2145386507

Изменили точку с индексом 1 на 0.001
{(1.0; 4.0), (2.0; 8.001), (3.0; 12.0), (4.0; 16.0), (5.0; 20.0), (6.0; 24.0), (7.0; 28.0), (8.0; 32.0), (9.0; 36.0), (10.0; 40.0), (11.0; 44.0)}

Измененный хэш-код для fun4
1838061317
```

Изменили исходную fun4, при этом копия fun6 не изменилась

```
{(1.0; 4.0), (2.0; 8.0), (3.0; 12.0), (4.0; 16.0), (5.0; 20.0), (6.0; 24.0), (7.0; 28.0), (8.0; 32.0), (9.0; 36.0), (10.0; 40.0), (11.0; 44.0)}
```

fun7: Копия fun2

```
{(1.0; 1.0), (2.0; 2.0), (3.0; 3.0), (4.0; 5.0), (5.0; 9.0), (6.0; 6.0), (7.0; 32.0)}
```

Изменили исходную fun2

```
{(1.0; 1.0), (2.0; 2.0), (3.0; 3.0), (4.0; 5.0), (5.0; 9.0), (6.0; 6.0), (7.0; 32.0), (16.0; 12.0)}
```

Копия fun7 не изменилась

```
{(1.0; 1.0), (2.0; 2.0), (3.0; 3.0), (4.0; 5.0), (5.0; 9.0), (6.0; 6.0), (7.0; 32.0)}
```