

Лабораторная работа № 2

«Разработка простейшего класса на языке Java»

14 февраля 2022 г.

Цель работы

Целью данной работы является изучение базовых возможностей языка Java.

Исходные данные

Каждый публичный класс в языке Java должен размещаться в отдельном файле, базовая часть имени которого совпадает с именем класса. В данной лабораторной работе потребуется разработать два класса: основной класс, реализующий функциональность в соответствии с вариантом задания, и вспомогательный класс Test, демонстрирующий работоспособность основного класса.

Соответственно, создаваемый в рамках лабораторной работы проект будет состоять из двух файлов: файла с основным классом и файла Test.java. Эти файлы нужно разместить в одном каталоге. Компиляция нашего проекта с командной строки может быть выполнена с помощью команды.

```
javac Test.java
```

Файл, содержащий основной класс, в командной строке компилятора Java можно не указывать: компилятор увидит, что основной класс используется в классе Test, и автоматически найдёт и откомпилирует содержащий его файл.

Отметим, что при программировании на языке Java следует соблюдать следующие соглашения об именовании сущностей программы:

1. Имена классов должны быть существительными и должны начинаться с заглавной буквы. Бывают сложные имена классов, состоящие из нескольких слов. В этом случае каждое слово в составе имени класса начинается с заглавной буквы. Например, Color, HashSet, DoubleLinkedList.
2. Имена методов должны быть глаголами и должны начинаться с прописной (т.е. маленькой) буквы. В сложных именах, состоящих из нескольких слов, первое слово начинается с прописной буквы, а следующие слова – с заглавных букв. Например, insert, extractMax, convertToDouble.
3. Имена переменных, параметров и полей должны быть существительными, начинающимися с прописной буквы. В сложных именах, состоящих из нескольких слов, первое слово начинается с прописной буквы, а следующие слова – с заглавных букв.

Для вывода объектов в стандартный поток вывода удобно определить человекочитаемое текстовое представление объектов. Для этого в языке Java предусмотрен метод toString:

```
public String toString()
```

Любой класс по умолчанию наследует реализацию этого метода от класса Object. Однако, эта реализация не очень информативна, и поэтому имеет смысл переопределять метод toString в

каждом классе, объекты которого может потребоваться переводить в текстовую форму. Как это сделать, демонстрируется в следующем примере:

```
public class Point {  
  
    private double x , y ;  
    public Point ( double x , double y ) {  
        this.x = x;  
        this.y = y;  
    }  
  
    public double getX() { return x; }  
    public double getY() { return y; }  
  
    public String toString() {  
        return "("+x+", "+y+")";  
    }  
  
}
```

Метод toString автоматически вызывается при печати объекта с помощью System.out.println и при конкатенации строки и объекта. Например, следующая программа напечатает (10.0, 20.0):

```
public class Test {  
  
    void main (String [] args) {  
        public static Point p = new Point(10,20);  
        System.out.println (p) ;  
    }  
  
}
```

Задание

Выполнение лабораторной работы заключается в составлении на языке Java одного из классов, приведённых в таблице. В классе обязательно должны присутствовать конструктор и метод toString.

Отладку разработанного класса необходимо осуществить в методе main вспомогательного класса Test. Использование контейнерных классов из стандартной библиотеки языка Java не разрешается.

№	Условие задачи
1	Класс симметричных квадратных целочисленных матриц с операциями чтения и записи указанного элемента (часть матрицы, расположенную выше главной диагонали, хранить не надо).
2	Класс квадратных целочисленных матриц с операцией вычисления определителя.
3	Класс n-мерных вещественных векторов с операциями сложения, вычитания и нормализации. Любая операция должна возвращать новый вектор, оставляя исходные вектора нетронутыми.
4	Класс вещественных квадратных матриц с операцией быстрого возведения матрицы в указанную степень.. Операция должна возвращать новую матрицу, оставляя исходную матрицу нетронутой.

- 5 Класс простых дробей с операциями сложения и умножения (числитель и знаменатель должны быть взаимно просты).
- 6 Класс n-мерных векторов, компоненты которых представлены дробями, со следующими операциями: сложение; определение, ортогональны ли два вектора.
- 7 Класс прямых на плоскости с операцией вычисления перпендикуляра, проходящего через точку.
- 8 Класс костяшек домино с операцией присоединения одной костяшки к другой (в итоге костяшки должны образовывать двусвязный список).
- 9 Класс вещественных матриц с операцией транспонирования. Операция должна возвращать новую матрицу, оставляя исходную матрицу нетронутой.
- 10 Класс ломаных линий в двумерном пространстве с операцией вычисления длины ломаной.
- 11 Класс конечных цепных дробей вида $[a_0; a_1; a_2; \dots; a_n] = 1 / (a_1 + 1 / (a_2 + 1 / (\dots)))$ с двумя операциями: изменение заданного коэффициента дроби; вычисление значения дроби в виде числа с плавающей точкой.
- 12 Множество строк с операциями: добавление строки; проверка, можно ли конкатенацией двух строк множества получить палиндром.
- 13 Класс, представляющий доску для игры в крестики-нолики размером $n \times n$ с операциями установки крестика или нолика и определения окончания игры.
- 14 Класс последовательностей целых чисел с операцией удаления элементов, являющихся суммой двух других элементов последовательности. Операция должна возвращать новую последовательность, оставляя исходную последовательность нетронутой.
- 15 Класс, представляющий полином с целыми коэффициентами с операцией деления полинома с остатком.
- 16 Класс, представляющий элемент однонаправленного списка целых чисел, с операцией поиска целого числа в списке.
- 17 Класс квадратных булевских матриц с операцией домножения матрицы на другую матрицу. Аналогом сложения для булевских значений считать операцию ИЛИ, аналогом умножения – операцию И.
- 18 Класс, представляющий матрицу расстояний между всеми парами из n городов с операцией вычисления длины пути, заданного последовательностью посещаемых городов.
- 19 Класс, представляющий кольцевой буфер фиксированного размера, состоящий из целых чисел, с операциями `empty`, `enqueue` и `dequeue`.
- 20 Класс треугольников в трёхмерном пространстве с операцией вычисления площади.
- 21 Класс, представляющий конечное множество целых чисел с операцией симметрической разности.
- 22 Класс последовательностей целых чисел с операцией удаления из последовательности элементов, кратных заданному числу. Операция должна возвращать новую последовательность, оставляя исходную последовательность нетронутой.
- 23 Класс комплексных чисел с четырьмя арифметическими операциями.
- 24 Класс, представляющий последовательность простых чисел с операцией получения n -го простого числа (вычисление чисел должно быть ленивым, т.е. ни одно число не должно вычисляться до вызова операции, и ни одно число не должно вычисляться дважды).
- 25 Множество целых чисел с операциями: добавление числа; добавление минимального количества чисел для того, чтобы числа множества располагались на числовой прямой с равными интервалами.
- 26 Класс стреловидных матриц размера $n \times n$ с операцией вычисления определителя. Все элементы стреловидной матрицы, кроме принадлежащих первой строке, первому столбцу или главной диагонали, равны нулю. Матрица должна быть представлена в виде, исключающем хранение заведомо нулевых элементов.
- 27 Класс n-мерных вещественных векторов с операцией скалярного произведения.
- 28 Класс бинарных отношений на множестве целых чисел от 0 до n с двумя операциями: проверка принадлежности пары чисел отношению; вычисление транзитивного замыкания отношения.
- 29 Класс целочисленных матриц с операциями сложения и умножения. Операции должны возвращать новую матрицу, оставляя исходные матрицы нетронутыми.
- 30 Класс записей шахматных партий с операцией, вычисляющей положение фигуры, которая в начале партии находилась в заданной позиции, на n -ом ходе.
- 31 Класс бинарных отношений на множестве целых чисел от 0 до n с тремя операциями: проверка принадлежности пары чисел отношению; добавление пары чисел в отношение; вычисление композиции

данного отношения с другим отношением.

- 32 Класс идеальных «разменов» рублёвых монет и купюр с операцией получения количества монет или купюр заданного номинала в «размене». Конструктор класса должен принимать в качестве параметра сумму денег (в рублях и копейках). Идеальный «размен» содержит минимально возможное количество монет и купюр.