

Java 02

Николай Обедин

2015-09-16

План на сегодня

- ▶ ООП
- ▶ IO
- ▶ Задачи

- ▶ Когда стоит использовать интерфейс, а когда абстрактный класс?

ООП: пример #1

```
public class Rectangle {  
    public Rectangle(int height, int width);  
    public int getArea() { return height * width; }  
    public int getHeight();  
    public int getWidth();  
    public void setHeight(int height);  
    public void setWidth(int width);  
  
    private int height;  
    private int width;  
}
```

ООП: пример #2

```
public class Rectangle {
    public Rectangle(int height, int width);
    public int getArea() { return height * width; }
    public int getHeight();
    public int getWidth();
    public void setHeight(int height);
    public void setWidth(int width);

    protected int height;
    protected int width;
}

public class Square extends Rectangle {
    public Square(int size);
}
```

ООП: пример #2 (еще раз)

```
public abstract class AbstractRectangle {  
    public int getHeight();  
    public int getWidth();  
    public int getArea() { return getHeight() * getWidth(); }  
}
```

```
public class Rectangle extends AbstractRectangle {  
    public Rectangle(int height, int width);  
    public void setHeight(int height);  
    public void setWidth(int width);  
    private int height;  
    private int width;  
}
```

```
public class Square extends AbstractRectangle {  
    public Square(int size);  
    public void setSize(int size);  
    private int size;  
}
```

ООП: пример #3

```
public interface HasArea {
    int getArea();
}

public abstract class AbstractRectangle implements HasArea {
    public int getHeight();
    public int getWidth();
    public int getArea() { return getHeight() * getWidth(); }
}

public class Rectangle extends AbstractRectangle { ... }

public class Square extends AbstractRectangle { ... }

public class Round implements HasArea {
    public Round(int radius);
    public int getArea() { return Math.PI * radius * radius; }
    private int radius;
}
```

ООП: конструкторы

```
public class Point {  
    public static void main(String[] args) {  
        Point p2 = new Point(2.0, 3.0);  
        Point p1 = new Point(1, 2); // Polar or cartesian?  
    }  
  
    public Point(double x, double y) {  
        System.out.println("CARTESIAN");  
    }  
  
    public Point(double radius, int angle) {  
        System.out.println("POLAR");  
    }  
}
```


ООП: static factory method

```
public class Point {  
    public static void main(String[] args) {  
        Point p2 = Point.fromCartesian(2.0, 3.0);  
        Point p1 = Point.fromCartesian(1, 2);  
    }  
  
    public static Point fromCartesian(double x, double y) {  
        return new Point(x, y);  
    }  
  
    public static Point fromPolar(double radius, int angle) {  
        return new Point(radius * Math.cos(angle), radius * Math.sin(angle));  
    }  
  
    private Point(double x, double y) { ... }  
}
```

IO и исключения

- ▶ Базовая единица: потоки (Streams), работают с байтами (тип byte)
- ▶ `InputStream` – для получения данных, `OutputStream` – для отправки данных
- ▶ `FileInputStream` – для чтения из файла, `FileOutputStream` – для записи в файл
- ▶ `IOException` – общий класс исключений ввода/вывода
- ▶ Специфичные классы: `FileNotFoundException`, `EOFException` и др.

Пример на IO

```
import java.io.*;

public class ConsoleToFile {
    public static void main(String[] args) throws IOException {
        if (args.length < 1) {
            System.err.println("Filename expected");
        } else {
            writeConsoleInputToFile(args[0]);
        }
    }

    private static void writeConsoleInputToFile(String filename)
        throws IOException {
        try(OutputStream out = new FileOutputStream(filename)) {
            while (true) {
                String str = System.console().readLine(
                    "Enter string to write or 'exit' to stop: ");
                if ("exit".equals(str)) {
                    break;
                }
                out.write(str.getBytes());
                out.write(System.lineSeparator().getBytes());
            }
        }
    }
}
```

Задача

Написать программу которая по заданному имени файла и паролю (де)шифрует содержимое файла. Алгоритм шифрования следующий: файл считывается кусками по k байт, где k – длина пароля в байтах. Каждый кусок побайтово XORится с паролем и записывается в выходной файл.

Если имя файла заканчивается на “.xor”, то его надо расшифровать. В противном случае зашифровать в файл с расширением “.xor”.