



Пример 3: Программа "Square"

- Цель примера

Продемонстрировать работу класса с динамическими результатами работы методов в зависимости от свойств объекта.

- Поэтапное создание программы

В рамках данного примера рассмотрим класс – квадрат `Square`, который обладает следующими характеристиками:

- Свойства:
 - Длина стороны: `int sideLength`
- Методы:
 - Расчет собственного периметра

- Шаг 1.

Объявим класс `Square` со свойством:

```
public class Square {  
    public int sideLength;  
}
```

- Шаг 2.

Объявим класс `SquareDemo` для демонстрации работы:

```
public class SquareDemo {  
    public static void main(String[] args) {  
    }  
}
```

- Шаг 3.

В класс `Square` добавим метод, который будет возвращать рассчитанный периметр в зависимости от того, какая длина стороны у квадрата:

```
public class Square {  
  
    public int sideLength;  
  
    public int calculatePerimeter() {  
        return 4 * this.sideLength;  
    }  
  
}
```

Метод возвращает целое число `int`, называется `calculatePerimeter` и не принимает входных параметров.

- Шаг 4.

Создадим экземпляр класса и присвоим размер стороны и выведем в консоль периметр:

```
public class SquareDemo {  
  
    public static void main(String[] args) {  
  
        Square square = new Square();  
  
        square.sideLength = 2;  
        int p1 = square.calculatePerimeter();  
        System.out.println("P when side length is " + square.sideLength + " = " + p1);  
  
        square.sideLength = 5;  
        int p2 = square.calculatePerimeter();  
        System.out.println("P when side length is " + square.sideLength + " = " + p2);  
  
    }  
  
}
```

Результат работы программы:

```
■ P when side length is 2 = 8 P when side length is 5 = 20
```

Обратите внимание, как изменяется периметр у объекта, если изменяется длина его стороны.

• Рекомендации:

- Запустить программу и сравнить результаты;
- Изменить длину стороны и проверить периметр фигуры;
- Добавить метод для расчета площади фигуры;