

# Программирование на языке Java

Методы класса

# Повторение

- ▶ Что такое класс?
- ▶ Что такое объект?
- ▶ Чем отличается объект от класса?
- ▶ Что такое свойство объекта?
- ▶ Как задать свойство?
- ▶ Как получить свойство?
- ▶ Как создать два и больше объекта? Как их различать?

# Что такое объект?

- ▶ Это сущность созданная по шаблону, имеющая определенные свойства
  - ▶ Например:
    - ▶ Шаблон мяча, имеет в себе информацию о том какими должны быть его свойства: Цвет, радиус, форма, материал, страна где был произведен и др.
    - ▶ Объект мяча, имеет в себе информацию о каком то конкретном мяче: Красного цвета, радиус 15 см, круглый, сделан из кожи, сделан в Китае.
    - ▶ Объект мяча №2, имеет в себе информацию о каком то конкретном (но уже другом) мяче: Желтого цвета, радиус 10 см, овальный, сделан из резины, сделан в Америке.

# Что еще может содержать в себе класс (Шаблон)?

- ▶ Помимо свойств, класс может содержать в себе описание поведения(Действий) данного шаблона.
- ▶ Например:
  - ▶ Собака может лаять,
  - ▶ Собака может кушать,
  - ▶ Мяч может взрываться,
  - ▶ Человек может расти
  - ▶ И др.

# Основная деятельность объекта

- ▶ 1. Хранение состояния объекта (цвет, радиус, материал и т.п.)
- ▶ 2. Действия по изменению состояния объекта (изменить цвет, изменить материал, изменить стоимость и т.д.)

# Основная деятельность объекта

- 1. Хранение состояния объекта (цвет, радиус, материал и т.п.) – за это отвечают **Переменные класса(поля, атрибуты) - variables, fields, attributes**
- 2. Действия по изменению состояния объекта (изменить цвет, изменить материал, изменить стоимость и т.д.) – **за это отвечают методы**

# Задача

- ▶ Измерить площадь прямоугольника
  1. Создать класс (шаблон, эскиз прямоугольника)
  2. Создать свойства(переменные) объекта
  3. Создать объект класса
  4. Установить ширину и высоту объекта
  5. Измерить площадь

# Установка свойств - Сеттер (Setter)

1. Создать класс (шаблон, эскиз прямоугольника)
2. Создать свойства(переменные) объекта
3. Создать объект класса
4. Установить ширину и высоту объекта - Setter
5. Измерить площадь



# Установка свойств в Java, через сеттер

```
class Rectangle {  
    int height;  
    int width;  
    int area;  
  
    public void setHeight(int h) {  
        height = h;  
    }  
    public void setWidth(int w) {  
        width = w;  
    }  
}
```

# Правила объявления сеттеров:

- ▶ Методы ничего не возвращают (void)
- ▶ Имена сеттеров очень похожи на имена свойств(переменных), которые они устанавливают. Пример свойство - age, название сеттер - setAge
- ▶ Принимают один параметр
- ▶ Имеют префикс set
- ▶ Пример:
  - ▶ Переменная height
  - ▶ Setter - `void setHeight(int height) { }`

# Получение свойств - геттер (Getter)

1. Создать класс (шаблон, эскиз прямоугольника)
2. Создать свойства(переменные) объекта
3. Создать объект класса
4. Установить ширину и высоту объекта - Setter
5. Измерить площадь - лучше использовать геттер

# Пример геттера - Getter

```
class Rectangle {  
    int height;  
    int width;  
    int area;  
  
    public int getArea() {  
        area = height * width;  
        return area;  
    }  
}
```

# Non-static метод действует только на конкретный объект:

- ▶ Non-static - это такой метод при объявлении которого не было использовано слова static.
- ▶ Если у вас есть много прямоугольников - обычный метод вернёт площадь только по нужному объекту (а не по всем)

```
public int getArea() {  
    area = height * width;  
    return area;  
}
```

# Правила объявления геттеров:

- ▶ Геттер-методы имеют тот же тип что и возвращаемая переменная (В нашем случае - int)
- ▶ Имена геттер-методов похожи на имена переменных, которые они устанавливают.
- ▶ Не принимают параметров.
- ▶ Имеют префикс get
  - ▶ переменная area
  - ▶ геттер - `int getArea(i){}`

# Зачем нужны геттеры и сеттеры?

- ▶ Иногда перед установкой значения нужно дополнительно проверять:
  1. корректность вносимых данных. (А что если пользователь укажет отрицательное значение в качестве высоты?)
  2. Проводить дополнительные вычисления перед присвоением значения или при извлечении (посчитать площадь перед отражением данных)
- ▶ Это первый шаг к инкапсуляции (Указываем действие, а что и как именно изменяется в классе не интересуемся)

# Задача

- ▶ Используя предыдущий класс,
- ▶ Нужно дать возможность задать длину и ширину прямоугольника
- ▶ Создать квадрат указанной ширины (длина и ширина равны).
- ▶ Создать атрибут «цвет» и окрасить квадрат в нужный цвет.
- ▶ При установке высоты или ширины проверять - не отрицательное ли значение, не равно ли оно нулю.



# Как отличить метод от свойства в классе

- ▶ Метод после имени имеет скобки с параметрами (Пример: `rnd.NextInt()`).
- ▶ Очень желательно давать говорящие названия. Чтобы на уровне названия понять - переменная это или метод
  - ▶ `getRate`, `rate`, `bark`, `active`, `isBlocked`, `run`.
    - ▶ Метод - действие, переменная - на что направленно это действие.

*Спасибо за внимание!*