

# Программирование на языке Java

Конструкторы класса

# Повторение

## ► Как создать объект?

Название класса  
(название шаблона  
по которому  
создаем)

Название  
переменной  
(имя объекта)

Название класса и  
скобки

```
Rectangle rect = new Rectangle();
```

Ключевое слово для  
создания объекта по  
шаблону

# Что означают скобки?

```
Rectangle rect = new Rectangle();
```

- ▶ Это конструктор.
- ▶ Скобки помогают задать логику для создания объекта
- ▶ Java по умолчанию создает конструктор без параметров (пустые скобки).
- ▶ При использовании стандартного конструктора, созданный объект «Пустой»

# Конструктор по умолчанию

- ▶ Создает пустой объект, который мы заполняем самостоятельно

```
Rectangle rect = new Rectangle();  
rect.setWidth(15);  
rect.setHeight(10);
```

Заполнение  
«пустого» объекта

Создание «пустого»  
объекта

# Что такое конструктор

- ▶ Конструктор своего рода процедура, которая запускается автоматически при создании объекта. (на самом деле нет, потому что есть статические конструкторы, которые выполняются раньше статических)
- ▶ Конструктор - не метод. Его нельзя вызвать без использования **new**
- ▶ У конструктора нет возвращаемого типа. То есть он ничего не возвращает, даже `void`

# Как определить конструктор?

```
class Rectangle {  
    int height;  
    int width;  
  
    public Rectangle() {  
  
    }  
    public Rectangle(int w, int h) {  
        height = h;  
        width = w;  
    }  
}
```

Конструктор по умолчанию

Конструктор созданный нами

Создание объекта

```
Rectangle rect = new Rectangle(10, 20);
```

# Правила конструкторов

- ▶ Конструкторов может быть сколько угодно, главное чтобы отличались параметры.
- ▶ Если вы объявили конструктор самостоятельно - у вас не будет возможности использовать конструктор по умолчанию. Придется явным образом объявить конструктор по умолчанию (без параметров)
- ▶ Конструктор в 99,9% случаев должен быть `public`.
- ▶ Название конструктора должно быть полностью идентичным названию класса.

# Задание

- ▶ Создайте класс прямоугольника.
- ▶ Создайте свойства, объявите геттеры и сеттеры для ЭТИХ свойств
- ▶ Создайте конструктор для вашего класса
- ▶ Создайте экземпляры класса
- ▶ Заполните экземпляры данными
- ▶ Выведите данные в консоль



Как создать конструктор для квадратов используя тот же класс?



# Ключевое слово `this`

```
public Rectangle(int width, int height) {  
    this.width = width;  
    this.height = height;  
}
```

- ▶ Ключевое слово `this` - дает понять системе, что вы обращаетесь к переменной класса (свойству), а не к параметру или локальной переменной
- ▶ Критично в случаях когда название переменной и параметра в методе или конструкторе идентичны(одинаковые)

# Классная Работа

- ▶ Создать класс Круг
- ▶ Создать переменные (свойства): радиус, цвет и прозрачен ли?
- ▶ Создать геттеры и сеттеры
- ▶ Создать метод получения длины круга (периметр) по формуле  $2 \cdot 3.14 \cdot R$  ( $2\pi r$ )
- ▶ Создать метод получения площади круга по формуле  $3.14 \cdot R \cdot R$  ( $\pi r^2$ )
- ▶ При создании объекта хочу указать
  - ▶ Только радиус
  - ▶ Только радиус и цвет
  - ▶ Только радиус, цвет и прозрачен ли
- ▶ По умолчанию Круг не прозрачен

# Продолжение

- ▶ Вывести информацию о создаваемых объектах (радиус, цвет, прозрачен ли)
- ▶ Увеличить радиус круга в  $N$  раз и отразить в консоли длину и площадь круга после изменения
- ▶ Создать 100 кругов произвольного радиуса.
- ▶ Если радиус круга кратен 3 сохранить их в отдельном массиве

# Вопросы, ответьте своими словами

- ▶ Что такое конструктор?
- ▶ Что такое дефолтный конструктор (по умолчанию)?
- ▶ Как работает конструктор?
- ▶ Для чего нужен конструктор?
- ▶ Какой конструктор вызывается когда конструкторов несколько?
- ▶ Что такое `this`

*Спасибо за внимание!*