



Java 1

Семестр

[ligarobotov.ru](http://ligarobotov.ru)

# Занятие 9. Конструкторы классов. Таймер.

Федеральная сеть  
секций робототехники  
«Лига Роботов»  
[krsk@ligarobotov.ru](mailto:krsk@ligarobotov.ru)  
8 (391) 272-10-40

© «Лига Роботов», Красноярск

# Что нас ждет сегодня?

1. Познакомимся с понятием конструктор класса
2. Узнаем, что такое таймер
3. Выясним, где хранятся файлы нашей программы на жестком диске



# Повторение

- 1) Какие базовые классы вы знаете?
- 2) Важен ли регистр букв при написании программы?
- 3) Что такое «метод»?



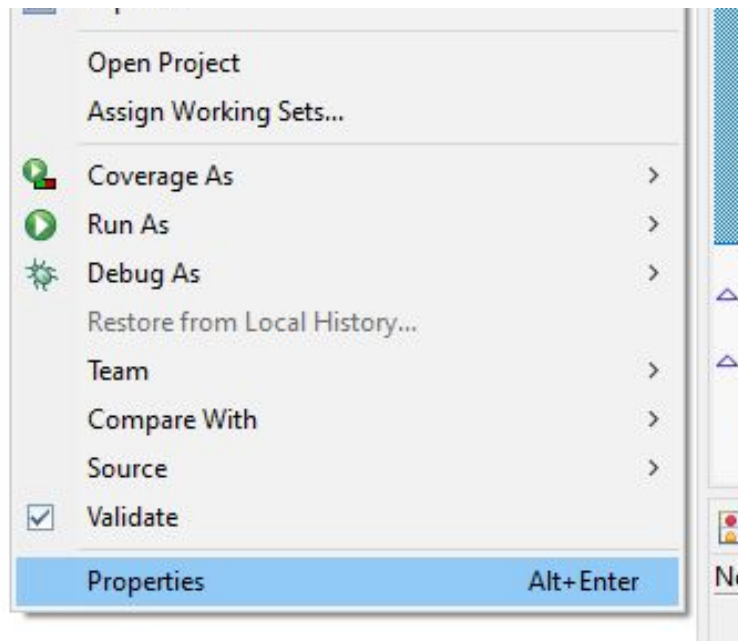
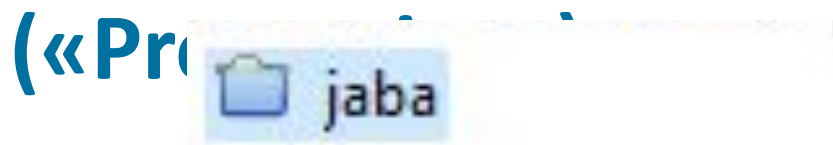
# Подготовка

Откройте программный код с предыдущего занятия.

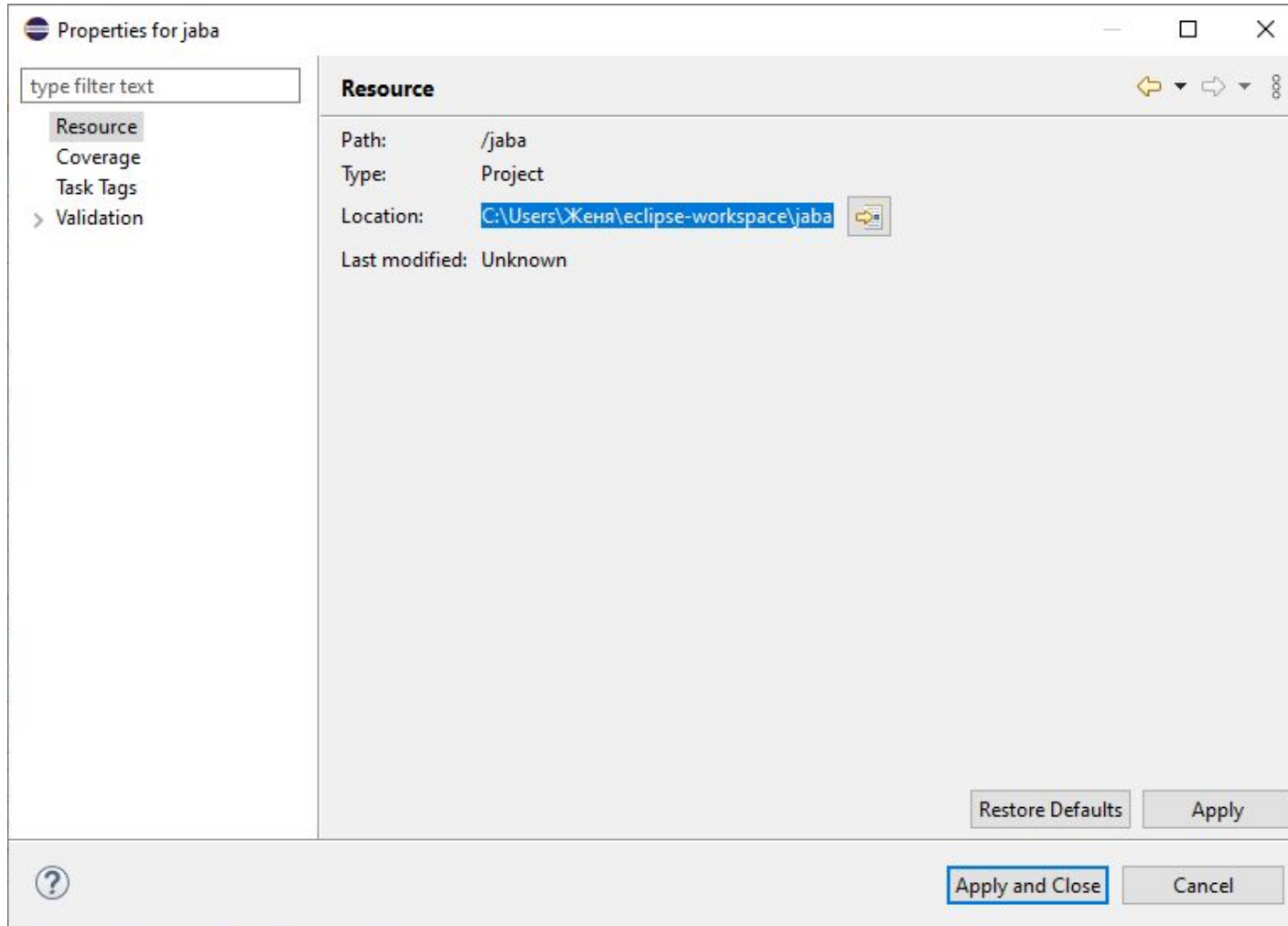


# Где хранятся наши файлы?

Нажимаем правой кнопкой мыши на проект и выбираем вкладку «Свойства»



# Где хранятся наши файлы?



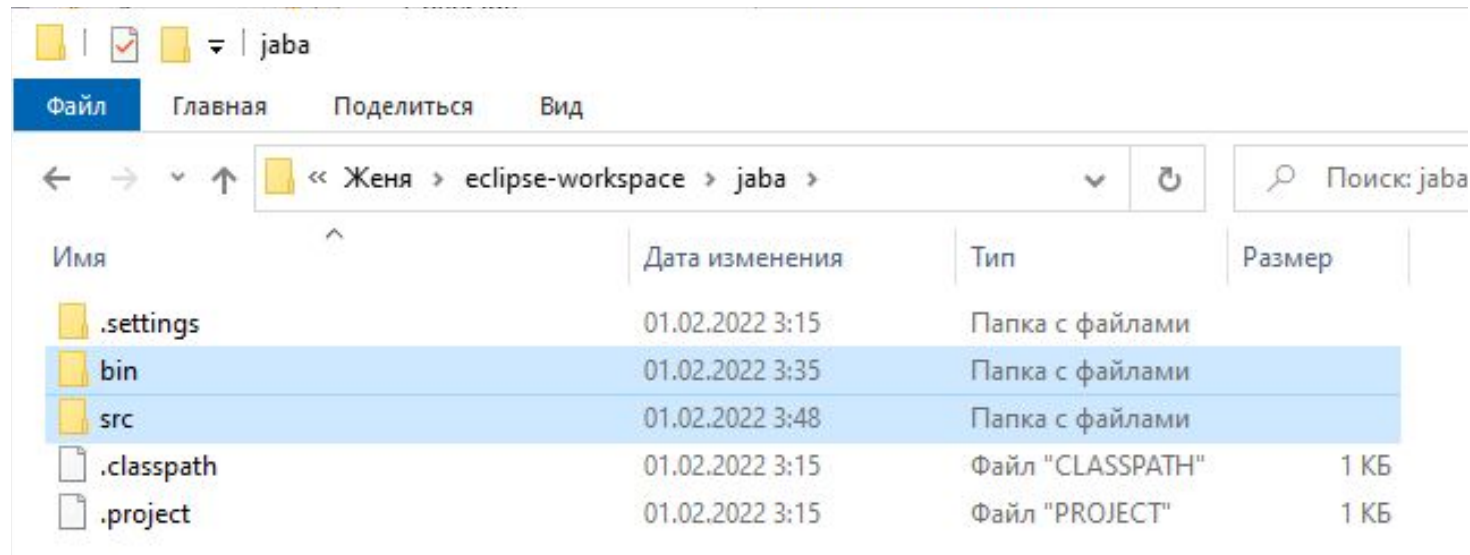
В строке под названием «Расположение» («Location») указан путь к папке, содержащей проект



# Где хранятся наши файлы?

В хранятся `src` файлы, содержащие код создаваемых программ.

А в `bin` – файлы с промежуточным `byte-code` (его виртуальная машина Java преобразует для конкретной операционной системы и запускает).



# Анимация

**Анимация движений – перемещение графического объекта по определенной области или его изменение.**

**Для создания анимации при перемещении изображения программа будет отрисовывать его заново в новых координатах, как бы по кадрам, а область за ним очищать.**





# Анимация

Что нам понадобится?

1. Конструктор класса
2. Таймер
3. Обработчик событий



# Конструктор класса

Конструктор класса – это специальная функция, которая автоматически вызывается после создания объекта этого класса. Это тоже метод, но особенный. Он определяет начальные свойства объекта.

В отличие от обычного метода класса имя конструктора класса всегда совпадает с именем самого класса, и конструктор является открытым (public).



# Конструктор класса

```
class myFrame extends JFrame
{
    public myFrame()
    {
        myPanel np = new myPanel();
        Container cont = getContentPane();
        cont.add(np);
        setBounds(10, 10, 800, 600);
        setVisible(true);
        setBackground(Color.WHITE);
    }
}
```



# Задача 1

**Создайте второй класс проекта и в нем создайте конструктор `myclass`. Пусть при его вызове на консоль выводится надпись «Это работает конструктор класса!».**



# Решение

```
public class test
{
    public static void main(String[] args) {
        myclass ob = new myclass();
    }
}
class myclass {
    public myclass () {
        System.out.println("Это работает конструктор класса!");
    }
}
```



# Перерыв



# Задача 2

Дополним предыдущую задачу. Добавьте метод `show()` и обратитесь к нему через переменную типа `myclass`. При вызове этого метода на консоли появится надпись «Это работает метод класса!».



# Решение

```
public class test
{
    public static void main(String[] args) {
        myclass ob = new myclass();
        ob.show();
    }
}
class myclass {
    public myclass () {
        System.out.println("Это работает конструктор класса!");
    }
    public void show()
    {   System.out.println("Это работает метод класса!");
    }
}
```





# Конструктор класса

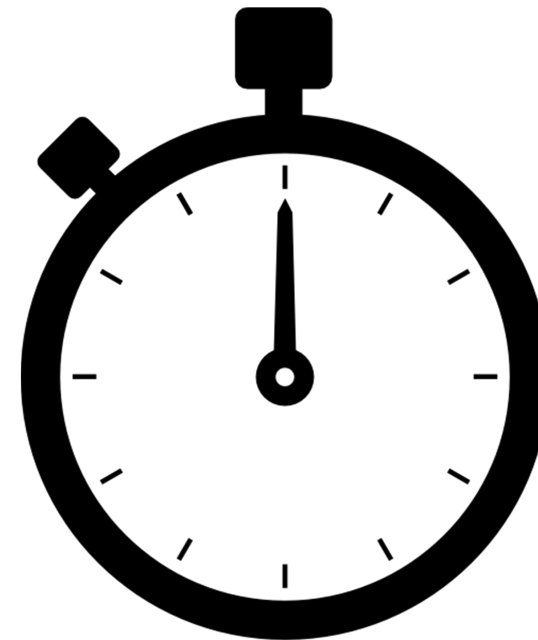
Удалим строку вызова метода `show()` – `ob.show()`;  
и запустим программный код.

Какую из двух надписей мы увидим в  
консольном окне? Попробуйте объяснить,  
почему.



# Таймер

**Таймер – это инструмент, позволяющий создавать события через заданный интервал времени.**



# Таймер

При помощи него можно определять временную задержку с большой точностью (в миллисекундах). Если её не предусмотреть, то перемещение объекта будет происходить мгновенно.

1000 мсек = 1 сек

`import java.awt.event.*` – библиотека для работы с таймером.



# Таймер

Добавим в конструктор класса панели следующий код

В теле метода указывают событие, которое необходимо выполнить после того, как пройдет указанный временной промежуток.

```
Timer timer = new Timer(25, new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        repaint();  
    }  
});  
timer.start();
```



# Подготовка

Подготовимся к следующему шагу. Добавьте закрытые переменные `x` и `y` для хранения координат картинки. Замените координаты на эти переменные в графическом методе. Также создайте закрытую переменную для хранения направления. Значения переменным можно не присваивать.



# Конец занятия

