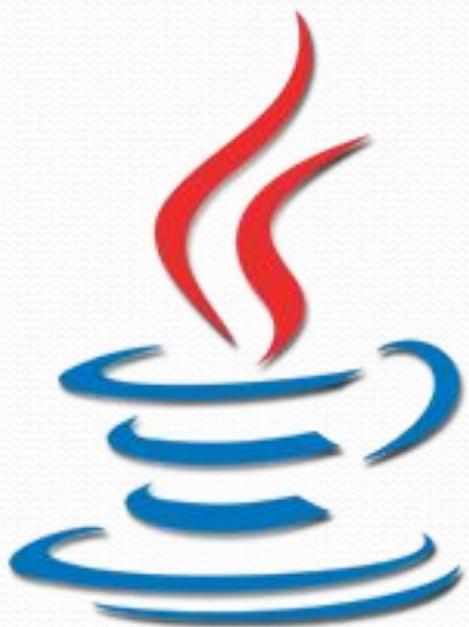
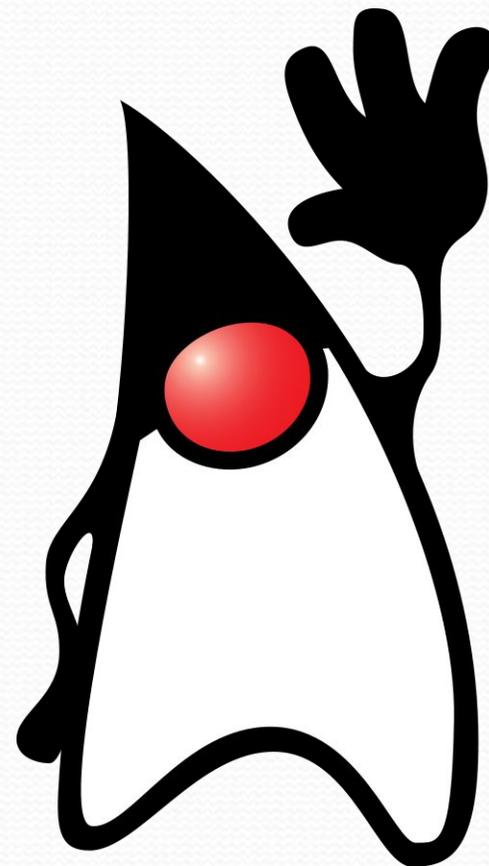


Курс: «Программирование на Java»



Java™



Понятие слова «Класс» в Java

- Класс есть ключевое понятие в объектно-ориентированном программировании, под которое и заточена Java. Класс описывает содержание и поведение некой совокупности данных и действий над этими данными.

Что такое класс?

- **Класс** – логическое описание чего-либо, **шаблон**, с помощью которого можно создавать реальные экземпляры этого самого чего-либо. Другими словами, это просто **описание** того, какими должны быть созданные сущности: какими **свойствами** и **методами** сущности должны обладать.

КЛАСС ПРЕДСТАВЛЯЕТ СОБОЙ ПОЛЬЗОВАТЕЛЬСКИЙ ТИП ДАННЫХ, СОЗДАННЫЙ ПРОГРАММИСТОМ.

Как создать класс в Eclipse?

- Для того, чтобы создать класс необходимо кликнуть правой кнопкой мыши по папке проекта в Eclipse, в появившемся окне выбрать поле «new», затем в новом окне кликнуть по полю «class»
- Появится мастер создания классов. В поле «name» ввести имя класса И кликнуть по кнопке «Finish».
- ДЛЯ ПОЛЬЗОВАТЕЛЬСКИХ КЛАССОВ-ОПИСАНИЙ НЕ ТРЕБУЕТСЯ МЕТОД **main**, ПОЭТОМУ СТАВИТЬ ГАЛОЧКУ В ПОЛЕ - **public static void main** – НЕ НУЖНО!!!



Модуль 1. Занятие 7

The screenshot shows the Eclipse IDE interface. The 'New' menu is open, and the 'Class' option is selected. The Package Explorer on the left shows a project named 'MyProj' with a source folder 'src'. The main editor window displays the code for 'MyProg.java', which includes a 'main' method. The Outline view on the right shows the class structure, including the 'main' method. The Console view at the bottom is empty, displaying the message 'No consoles to display at this time.' The Windows taskbar at the bottom shows the system tray with the date and time '21:01 16.11.2018'.

JavaProj - MyProj/src/MyProg.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer MyProj
src
MyProg.java

New
Go Into
Open in New Window
Open Type Hierarchy
Show In
Copy
Copy Qualified Name
Paste
Delete
Remove from Context
Build Path
Source
Refactor
Import...
Export...
Refresh
Close Project
Assign Working Sets...
Coverage As
Run As
Debug As
Validate
Restore from Local History...
Team
Compare With
Configure
Properties

Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
Task
JUnit Test Case
Example...
Other...

Task List
Find
All
Activate...

Outline
MyProj
main(String[]): void

Problems @ Javadoc Declaration Console
No consoles to display at this time.

MyProj
21:01
16.11.2018

New Java Class

Java Class
Create a new Java class.

Source folder:

Package:

Enclosing type:

Name:

Modifiers: public package private protected
 abstract final static

Superclass:

Interfaces:

Which method stubs would you like to create?

public static void main(String[] args)
 Constructors from superclass
 Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
 Generate comments

Как создать класс в Eclipse?

- При создании нового файла, класс будет объявлен, но в случае, если объявление стерлось, то класс объявляется следующим образом:
- Объявление класса производится с помощью ключевого слова **public** и ключевого слова **class**.

Пример: `public class < имя_класса > {`

`// содержимое класса`

`}` Пример объявления класса с именем **MyClass**

```
public class Animal{
```

```
// содержимое класса
```

```
}
```

Свойства класса

- *Свойства класса* – это характеристики сущности, описание которой мы создаем. Таким образом свойства – некоторые постоянные вещи, которые всегда есть у нашей сущности.
- СВОЙСТВА КЛАССА В ПРОГРАММИРОВАНИИ ПРЕДСТАВЛЕНЫ ПЕРЕМЕННЫМИ.

Например у сущности «**Animal**» (Животное) есть неизменяемая характеристика – «**legs**» -(лапы), которая представляет собой некоторое количество лап.

Пример класса «**Animal**» со свойством лапы «**legs**»

```
public class Animal{  
    int legs; // Характеристика, которая сообщает о том, что у  
}  
    // Сущности «Animal» есть лапы
```

Свойства класса(Итоги)

- 1) **свойства класса** - это всего лишь обычные внутренние **переменные** данного класса, которые описывают характеристики различных сущностей.
- 2) Свойств в классе может быть любое количество.

Методы класса

- **Методы класса** – действия, которые может выполнять сущность, описание которой мы создаем. Таким образом методы – некий алгоритм, который мы описываем.
- **ВСЕ ДЕЙСТВИЯ КЛАССА В ПРОГРАММИРОВАНИИ ПРЕДСТАВЛЕНЫ МЕТОДАМИ.**
- Например сущность «**Animal**» (Животное) может выполнять действие «**eat()**» (Кушать, есть), которое представляет собой процесс поглощения пищи. Пример:

```
public class Animal{  
    public void eat(){ //метод, который выполняет действие  
        System.out.println("Я кушаю");  
    }  
}
```

Методы класса(Итоги)

- 1) ВАЖНО!!

Методы-действия описываемого класса не содержат в себе ключевого слова « **static** ».

- 2) Все действия класса представлены методами

- 3) Методы характеризую одно действие в класса

- 4) Методов в классе может быть любое количество

Понятие слова «Экземпляр класса» (Объект)

- Мы выяснили, что класс – это описание некой сущности, которую мы можем создать.
- Сущности, создаваемые с помощью классов, в программировании называются «Экземплярами класса» или «Объектами».
Объект – конкретное представление какого-то предмета или живого существа из реального мира, созданное при помощи обобщенного описания – **класса**.
- С помощью одного класса можно создать любое количество объектов с разными свойствами.

Работа с классами и объектами

- ВАЖНО!!

В программировании работа с классами не ведется. Вся работа происходит с отдельными объектами, созданными на основе класса



Это кошка. Кошка - **объект**. С ней можно вести работу.

- У кошки есть:

Усы – много шт., Лапы – 4 шт., Хвост – 1 шт.

Кошка может делать:

Кушать(), Мурлыкать(), Ходить()

Это – **Класс**, который

описывает кошку. С ним

нельзя вести работу!

Создание объектов в Java.

- Создание объекта похоже на создание переменной. Нам также необходимо указать компьютеру тип, имя, а потом дописать еще несколько «волшебных действий».

тип – это имя класса – **Animal**.

Имя – **cat** – Кот – Может быть любым, но мы практикуемся на коте.

«Волшебные действия» после имени **cat** - `= new Animal();`

- Получается:

Animal cat = new Animal();

Работа с объектами

- Теперь мы полноценно можем работать с методами с свойствами конкретного **объекта**, обращаясь к ним через спецсимвол – « **•** »

```
cat.legs = 4; // Присваиваем свойству «legs» объекта «cat»
```

```
//значение = 4.(Т.к. у кошки 4 лапы)
```

```
cat.eat(); // Вызываем метод «eat()» у объекта «cat», который
```

```
//выполняет действие «Кушать»
```

```
System.out.println("У кошки " + cat.legs + " лапы");
```

```
// Выводим строку, которая сообщает, сколько лап у
```

```
//кошки
```



Модуль 1. Занятие 7



Модуль 1. Занятие 7