

Java SE8

Основы программирования.

Занятие 1. Введение. История. IDE среды. Термины ООП.

План занятий

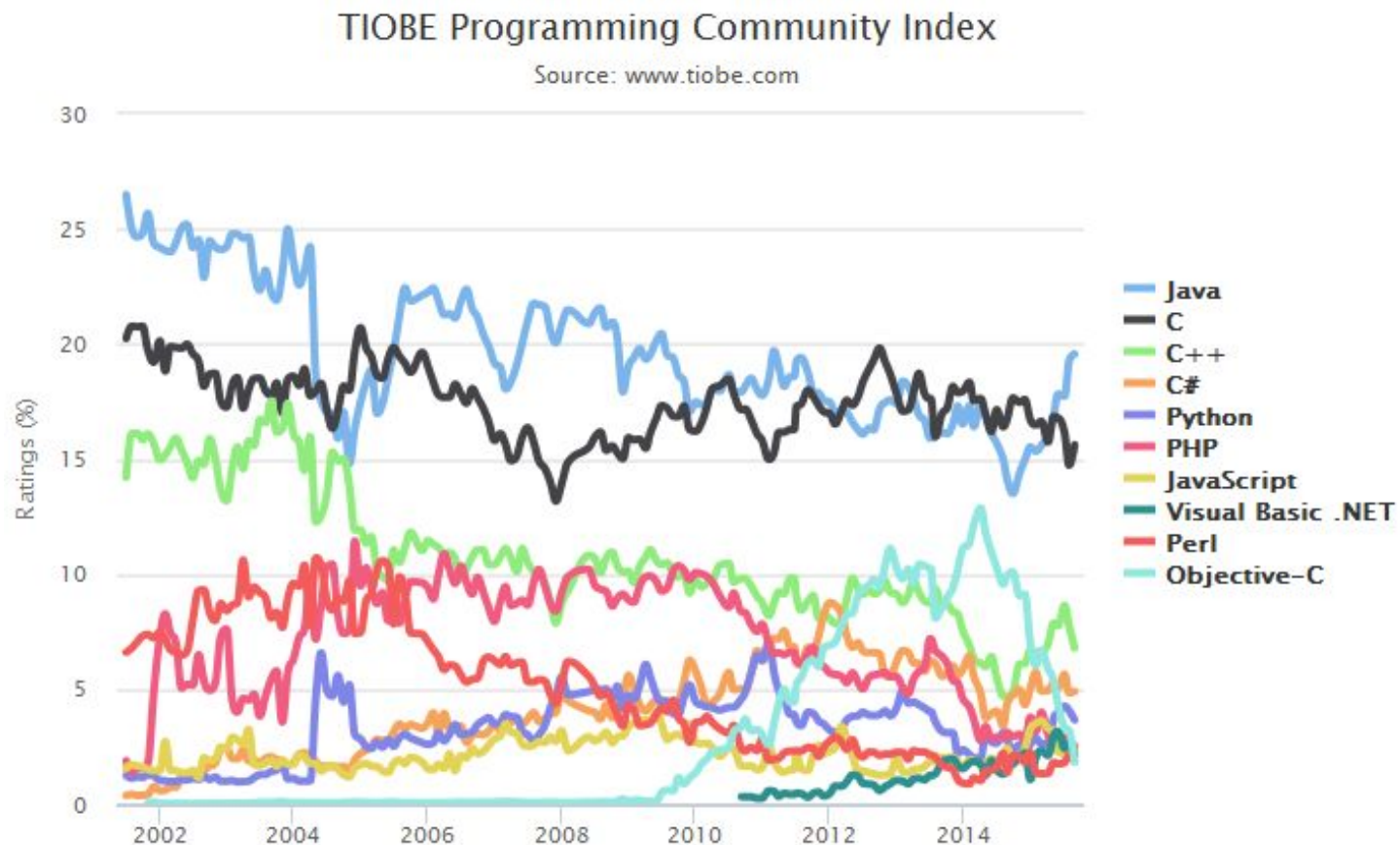
- | | |
|--|------------------------------------|
| 1. Введение в Java-технологии | 8. Классы Object и Class |
| 2. Введение в язык программирования Java | 9. Обработка ошибок |
| 3. Операции и операторы Java | 10. Потоки данных в Java |
| 4. Стандартные типы Java | 11. Дополнительные возможности ООП |
| 5. Разработка классов | 12. Коллекции |
| 6. Наследование и полиморфизм | 13. Сериализация |
| 7. Абстрактные классы и интерфейсы | 14. Работа с файловой системой |
| | 15. Шаблоны проектирования |

Самые популярные языки на сентябрь 15

Sep 2015	Sep 2014	Change	Programming Language	Ratings	Change
1	2	▲	Java	19.565%	+5.43%
2	1	▼	C	15.621%	-1.10%
3	4	▲	C++	6.782%	+2.11%
4	5	▲	C#	4.909%	+0.56%
5	8	▲	Python	3.664%	+0.88%
6	7	▲	PHP	2.530%	-0.59%
7	9	▲	JavaScript	2.342%	-0.11%
8	11	▲	Visual Basic .NET	2.062%	+0.53%
9	12	▲	Perl	1.899%	+0.53%
10	3	▼	Objective-C	1.821%	-8.11%
11	29	▲	Assembly language	1.806%	+1.22%
12	13	▲	Ruby	1.783%	+0.50%

www.tiobe.com

Процент использования языков на сентябрь 15



www.tiobe.com

Особенности языка Java

Язык высокого уровня. Особенности:

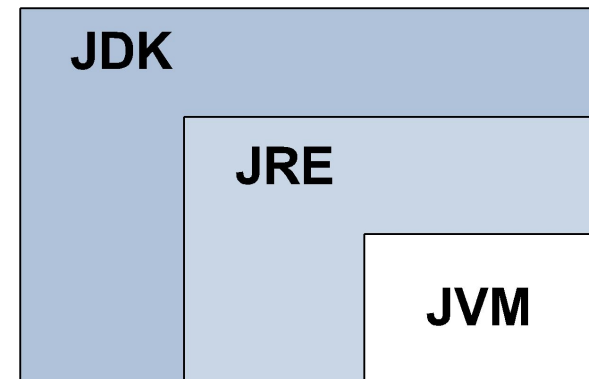
- Объектно-ориентированный подход
- Класс в основе всего
- Human-readable синтаксис
- Независимость от «железа»
- Огромный набор готовых API
- Безопасность и переносимость

Платформа Java

- **Платформа Java (Java Platform)** – программная среда, в которой работают приложения Java
 - Версии платформы Java для различных ОС (Windows, Linux, Solaris, Mac OS)
 - Включает в свой состав:
 - **Java Virtual Machine (JVM)** – *виртуальная машина Java* – программа, интерпретирующая приложения Java
 - **Java API** - библиотека программных компонентов (классов и интерфейсов), реализующих стандартный функционал
- **Java Platform, Standard Edition (Java SE)** – платформа широкого назначения для рабочих станций
 - **Java Platform, Enterprise Edition (Java EE)** – платформа для корпоративных приложений и приложений интернет
 - **Java Platform, Micro Edition (Java ME)** – платформа для устройств с ограниченными ресурсами и мобильных устройств
 - **Java Card** – платформа для смарт-карт

JRE и JDK

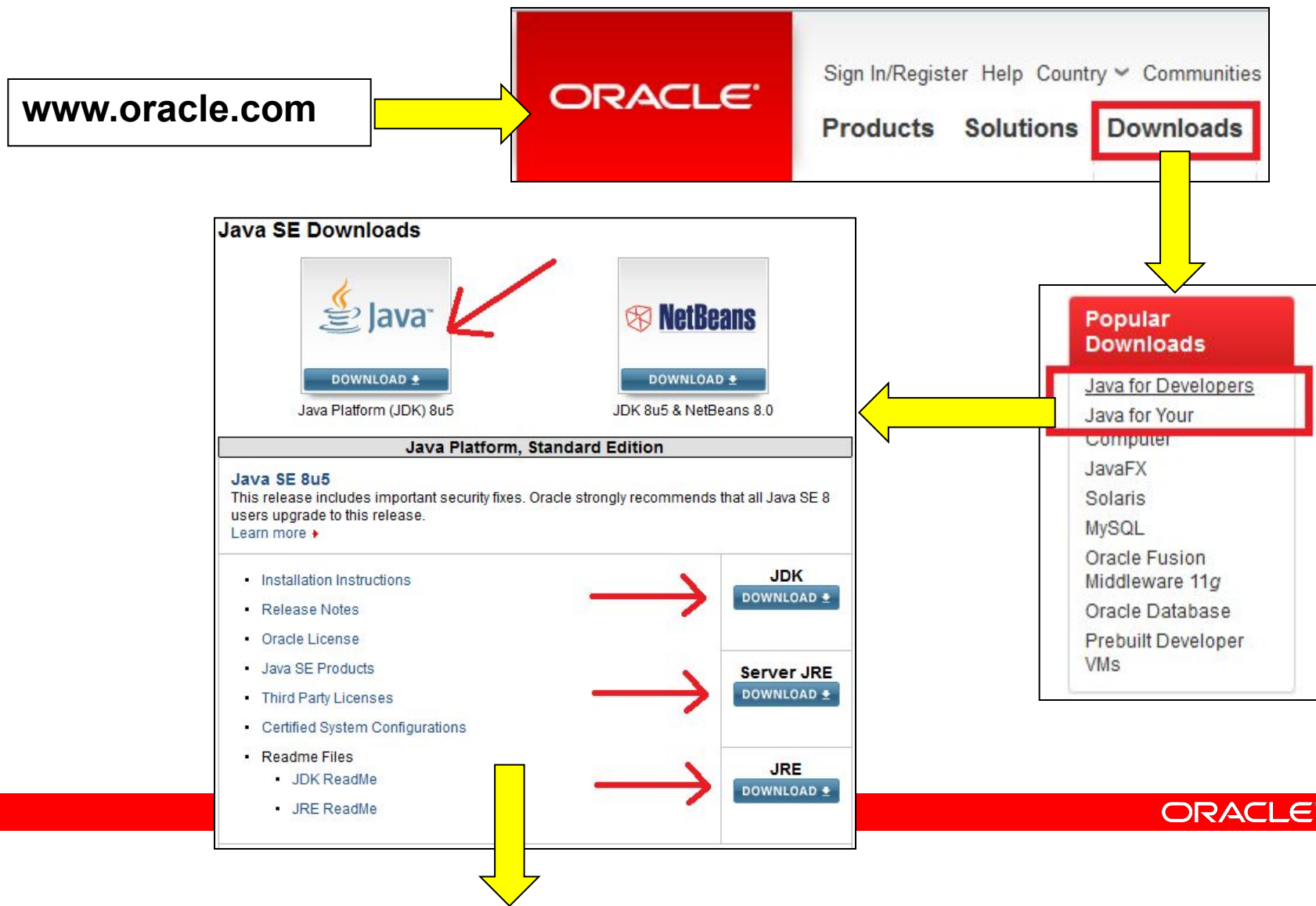
- **Java SE Runtime Environment (JRE)** - минимальная реализация платформы Java SE, необходимая для выполнения приложений
 - устанавливается на компьютеры конечных пользователей
 - включает в свой состав JVM и библиотеки, необходимые для выполнения программ
- **Java Development Kit (JDK)** – версия Java SE для разработки приложений
 - устанавливается на компьютеры разработчиков
 - включает в свой состав JRE, компилятор, отладчик, примеры программ, дополнительные библиотеки



История Java

- **1991** – начало работы над проектом Java
- **1995** – официальный релиз технологии Java компанией Sun Microsystems
- **1996** – выпуск JDK 1.0
- **1997** – выпуск JDK 1.1
- **1998** – выпуск J2SE 1.2
- **2000** – выпуск J2SE 1.3
- **2002** – выпуск J2SE 1.4
- **2004** – выпуск J2SE 5.0
- **2006** – выпуск Java SE 6.0
- **2010** – компания Sun вошла в состав корпорации Oracle
- **2011** – выпуск Java SE 7.0
- **2014** – выпуск Java SE 8.0

Загрузка и установка платформы Java SE



Загрузка и установка платформы Java SE

Java SE Runtime Environment 8 Downloads

Do you want to run Java™ programs, or do you want to develop Java programs? If you want to run Java programs, but not develop them, download the Java Runtime Environment, or JRE™.

If you want to develop applications for Java, download the Java Development Kit, or JDK™. The JDK includes the JRE, so you do not have to download both separately.

JRE MD5 Checksum

Java SE Runtime Environment 8u5

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

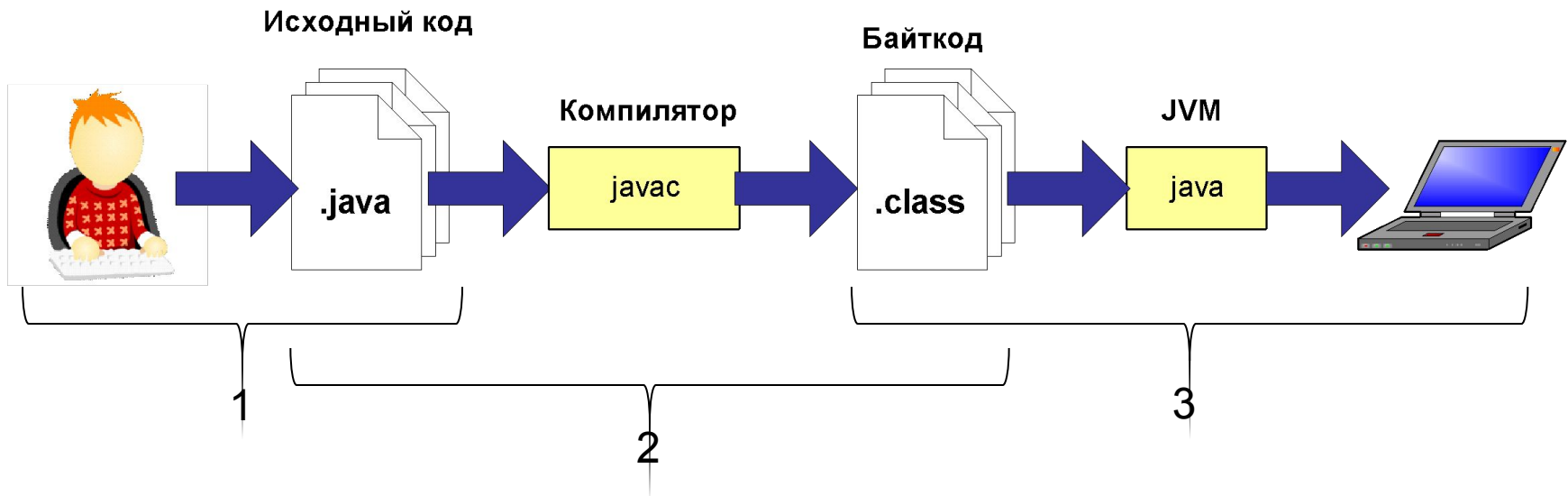
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	40.27 MB	 jre-8u5-linux-i586.rpm
Linux x86	55.46 MB	 jre-8u5-linux-i586.tar.gz
Linux x64	40.4 MB	 jre-8u5-linux-x64.rpm
Linux x64	54.41 MB	 jre-8u5-linux-x64.tar.gz
Mac OS X x64	56.61 MB	 jre-8u5-macosx-x64.dmg
Mac OS X x64	52.61 MB	 jre-8u5-macosx-x64.tar.gz
Solaris SPARC 64-bit	50.32 MB	 jre-8u5-solaris-sparcv9.tar.gz
Solaris x64	47.99 MB	 jre-8u5-solaris-x64.tar.gz
Windows x86 Online	1.53 MB	 jre-8u5-windows-i586-iftw.exe
Windows x86 Offline	29.67 MB	 jre-8u5-windows-i586.exe
Windows x86	45.87 MB	 jre-8u5-windows-i586.tar.gz
Windows x64	32.55 MB	 jre-8u5-windows-x64.exe
Windows x64	48.87 MB	 jre-8u5-windows-x64.tar.gz

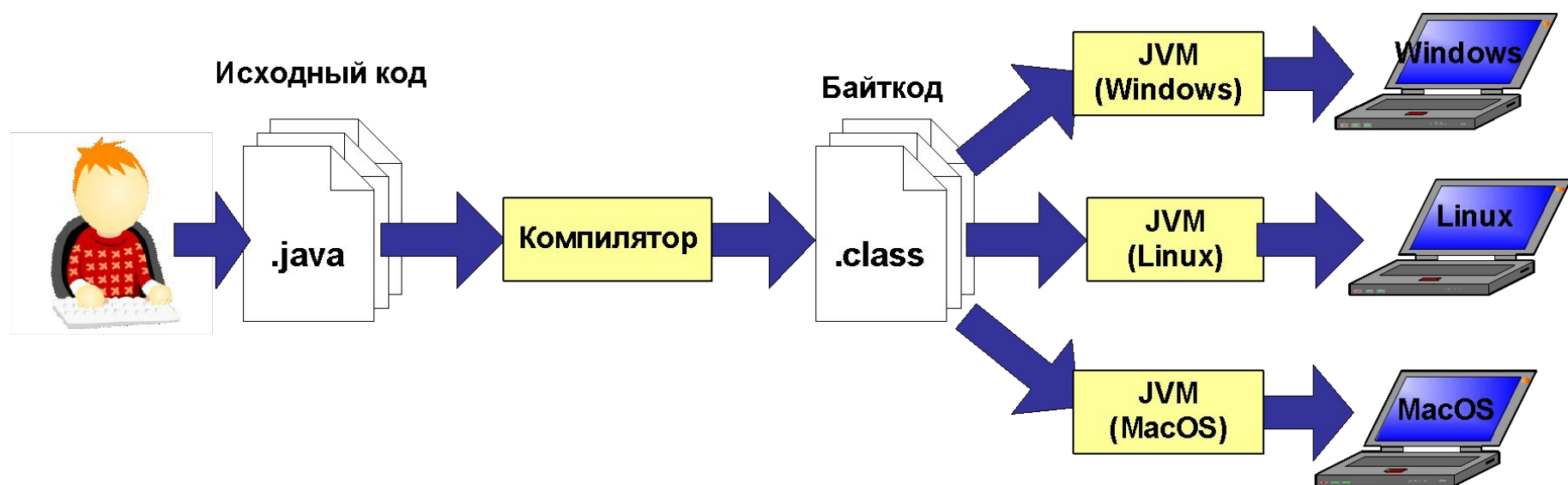
ORACLE

Этапы создания приложения Java

1. Разработка программного кода
 2. Компиляция исходного кода в байт-код
 3. Выполнение программы в JVM
- *Байткод (bytecode)* – машинно-независимый низкоуровневый язык виртуальной машины Java



Переносимость приложений Java



Hello, World!!!

HelloWorld.java

```
package hello;

public class HelloWorld{
    public static void main(String [] args) {
        System.out.println("Hello, Java world!!!");
    }
}
```



A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\System32\cmd.exe". The command prompt shows the command "c:\>javac HelloWorld.java" entered and executed. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.



A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\System32\cmd.exe". The command prompt shows the command "c:\>java hello.HelloWorld" entered and executed, followed by the output "Hello, Java world!!!". The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Преимущества программирования на Java

- Начать программировать на Java легко
- Маленький объем программного кода
- Высокая эффективность программного кода
- Приложения разрабатываются быстро
- Приложения не зависят от платформы и ОС



Среды разработки Java

IDE – Integrated Development Environment:

Среды разработки Java

Eclipse IDE

<http://www.eclipse.org>

NetBeans IDE

<https://netbeans.org>

IntelliJ IDEA

<http://www.jetbrains.com/idea/>

и т.д.

Eclipse. Загрузка Eclipse

Eclipse Downloads

[Packages](#) [Java™ 8 Support](#) [Developer Builds](#)

Eclipse Kepler (4.3.2) SR2 Packages for Windows



Eclipse Standard 4.3.2, 200 MB
Downloaded 4,664,642 Times [Other Downloads](#)
The Eclipse Platform, and all the tools needed to develop and debug it: Java and Plug-in Development Tooling, Git and CVS...

 [Windows 32 Bit](#)
[Windows 64 Bit](#)

Package Solutions

Filter Packages



Eclipse IDE for Java EE Developers, 250 MB
Downloaded 2,971,070 Times
Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...

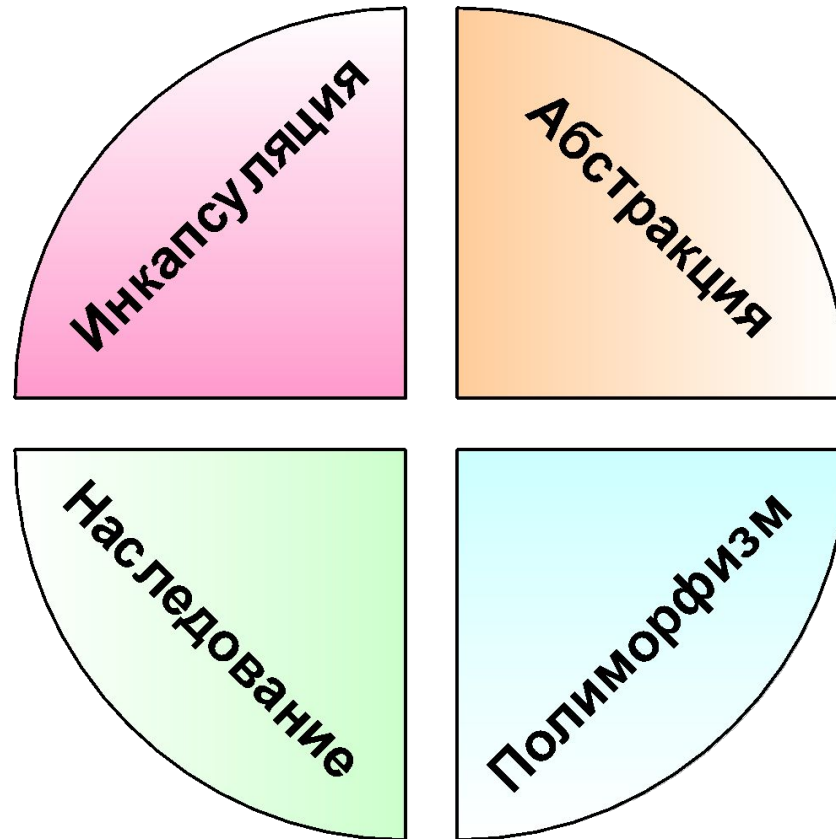
 [Windows 32 Bit](#)
[Windows 64 Bit](#)



Eclipse IDE for Java Developers, 153 MB
Downloaded 991,252 Times
The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration...

 [Windows 32 Bit](#)
[Windows 64 Bit](#)

Базовые принципы ООП



Понятие объекта. Инкапсуляция

Объект реального мира:

Состояние (state)

Линия поведения (behavior)

Программный объект (**Object**):

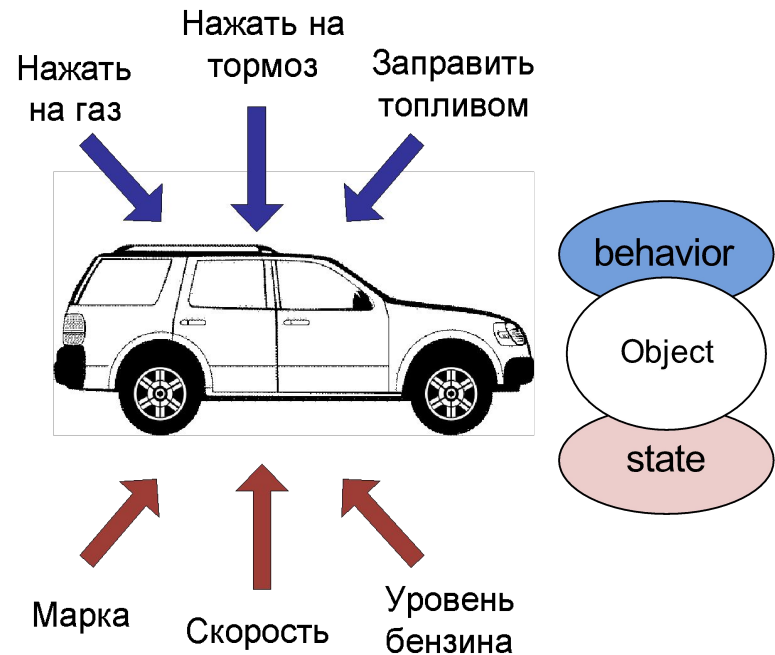
Поля (fields)

Методы (methods)

Инкапсуляция (encapsulation)

объединение данных и алгоритмов в рамках одной сущности (объекта)

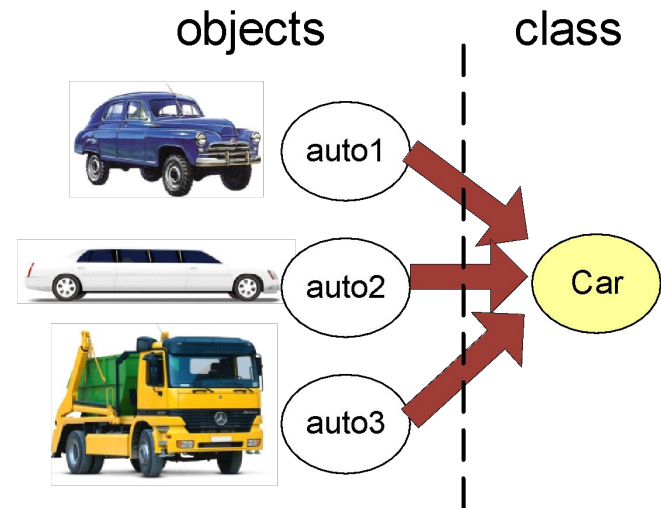
разграничение доступа к элементам объекта



Понятие класса

- **Класс** (*class*) описывает признаки состояния и поведение множества схожих объектов
- Класс – это пользовательский *тип данных*

```
class Car {  
    String name;  
    int speed;  
    int fuel;  
  
    void setName(String newName) {...}  
    void speedUp(int delta) {...}  
    void applyBrakes(int delta) {...}  
    void addFuel(int delta) {...}  
    void printState() {...}  
}
```



```
Car auto1 = new Car();  
Car auto2 = new Car();  
Car auto3 = new Car();
```

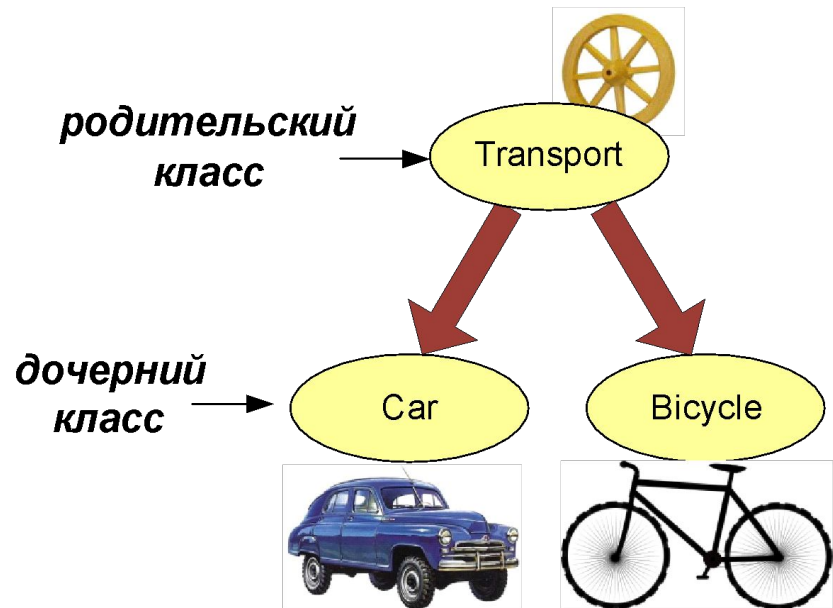
Наследование

Наследование (*inheritance*) – механизм создания новых классов на основе существующих

При наследовании **дочернему классу (*subclass*)** передаются поля и методы **родительского класса (*superclass*)**

У класса может быть один родитель и любое количество дочерних классов

```
class Transport {  
    ...  
}  
  
class Car  
    extends Transport {  
    ...  
}
```



Полиморфизм

- **Полиморфизм**
(*polymorphism*)

- имеется несколько реализаций алгоритма
- выбор реализации осуществляется в зависимости от типа объекта и типа параметров

- Механизмы реализации:

- **Перегрузка**
(*overloading*) методов
- **Переопределение**
(*overriding*) методов

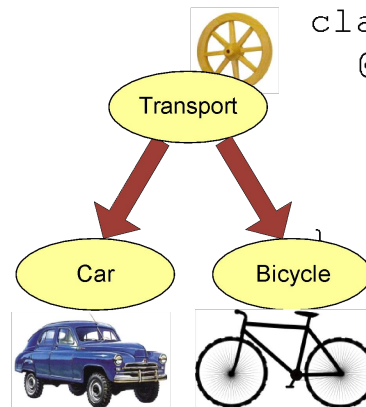
```
int max(int a, int b) {...}  
int max(int a, int b, int c) {...}  
int max(int[] arr) {...}
```

```
abstract class Transport{  
    abstract void beep();  
}  
  
class Car extends Transport{  
    @Override  
    void beep() {  
        System.out.println("Би-би");  
    }  
}
```

```
class Bicycle extends Transport{  
    @Override  
    void beep() {  
        System.out.println("Дзынь-дзынь");  
    }  
}
```

...

```
Transport tr1 = new Car();  
Transport tr2 = new Bicycle();  
tr1.beep(); tr2.beep();
```



Понятие интерфейса

Интерфейс (interface) определяет возможное поведение объектов
Интерфейс представляет собой совокупность методов без реализации

При объявлении класса можно указать, какие интерфейсы он будет поддерживать

```
interface Switchable {  
    void switchOn();  
    void switchOff();  
}  
  
class Lamp  
    implements Switchable {  
    ...  
}
```

