



# Java Essential

Основы объектно-ориентированного  
программирования



# Java Essential

Автор курса



Евгений Тихонов

# Java Starter

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)

Доступ можно получить через руководство вашего учебного центра



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://TestProvider.com)

## Основы объектно-ориентированного программирования

# Объектно-ориентированный подход

## Парадигмы ООП

ИНКАПСУЛЯЦИЯ

НАСЛЕДОВАНИЕ

ПОЛИМОРФИЗМ

АБСТРАКЦИЯ

# Парадигмы ООП

## Инкапсуляция

**Инкапсуляция** – механизм языка, позволяющий ограничить доступ одних компонентов программы к другим.

В Java существуют следующие модификаторы доступа:

- **private** – члены класса доступны только внутри класса;
- **default** (package-private, модификатор по умолчанию) – члены класса видны внутри пакета;
- **protected** – члены класса доступны внутри пакета и в наследниках;
- **public** – члены класса доступны всем.

```
class A {  
    private int field1;  
    int field2;  
    protected int field3;  
    public int field4;  
}
```

# Парадигмы ООП

## Наследование

**Наследование** – механизм объектно-ориентированного программирования (наряду с инкапсуляцией, полиморфизмом и абстракцией), позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом.

```
class A {  
    public int field1;  
    public void method() {  
        /* ... */  
    }  
}
```

```
class B extends A {  
    public int field2;  
}
```

```
public static void main(String[] args) {  
    B b = new B();  
    b.field1 = 5;  
    b.field2 = 8;  
    b.method();  
}
```

# Парадигмы ООП

## Полиморфизм

**Полиморфизм** – возможность объектов с одинаковой спецификацией иметь различную реализацию.

Формы полиморфизма:

1. Ad-hoc полиморфизм
2. Классический (принудительный) полиморфизм:
  - использование переопределенных членов (@Override).
  - приведение типов.

В случае одновременного использования двух форм классического полиморфизма, первая форма нейтрализует вторую (доминирует над второй).



# Парадигмы ООП

## Абстракция

**Абстракция** в ООП – это придание объекту характеристик, которые отличают его от всех других объектов, четко определяя его концептуальные границы.

**Абстрагирование** в ООП – это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые. Соответственно, абстракция – это набор всех таких характеристик.

# Объявление класса

## Содержание класса

```
public class Car {
```

```
    public String model;  
    private int maxSpeed;  
    private int speed;
```

```
    public Car(String model, int maxSpeed, int speed) {  
        this.model = model;  
        this.maxSpeed = maxSpeed;  
        this.speed = speed;  
    }
```

```
    public int getMaxSpeed() {  
        return maxSpeed;  
    }
```

```
    public void setMaxSpeed(int maxSpeed) {  
        this.maxSpeed = maxSpeed;  
    }
```

```
}
```

Объявление класса с именем Car

Объявление полей класса Car

Объявление конструктора класса Car

Объявление методов класса Car

# Члены класса

## Поля и Методы

**Поле класса** или **атрибут** (class field) – переменная, связанная с классом или объектом. Все данные объекта хранятся в его полях. Доступ к полям осуществляется по их имени.

```
private int speed;  
speed = 100;
```

**Метод класса** (class method) – процедуры и функции, связанные с классом. Определяют действия, которые можно выполнять над объектом такого типа, и которые сам объект может выполнять.

```
public int getSpeed(){  
    return speed;  
}
```

# Члены класса

## Конструкторы

**Конструктор** – специальный метод для создания экземпляра класса (инициализации объекта). Имя конструктора соответствует имени класса.

```
public Car(String model, int maxSpeed, int speed) {  
    this.model = model;  
    this.maxSpeed = maxSpeed;  
    this.speed = speed;  
}
```

Если программист явно не задал конструктор в классе, JVM при компиляции создаст **конструктор по умолчанию**

```
public Car() { }
```

# Объект

## Создание объекта

Чтобы создать объект используется ключевое слово **new**.

```
new Car();
```

При Этом JVM вызывает конструктор класса Car и собирает экземпляр класса Car. Чтобы использовать объект в дальнейшем, создадим указатель на него.

```
Car car = new Car();
```

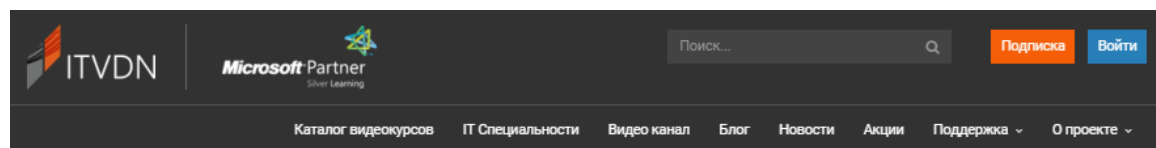
Теперь через ссылку мы можем получить доступ к полям и методам объекта.

```
car.model;  
car.setMaxSpeed(100);
```

Значения полей характеризуют **СОСТОЯНИЕ** экземпляра класса (объекта).  
Набор методов характеризует **ПОВЕДЕНИЕ** объекта.

# Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.

Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics

### 1 ITVDN 2015. Наши награды

ITVDN  
лучший ресурс для online обучения

### ITVDN 2015. Итоги года

В марте 2015 года ITVDN стал победителем конкурса IT Education Awards, который проходил в рамках IT Jam 2015 и награжден как лучший образовательный ресурс в номинации Online Education. Экспертное жюри, в состав которого вошли представители ведущих IT компаний, отметили такие преимущества ITVDN, как системный подход в обучении, позволяющий удаленно получить качественное образование по наиболее популярным специальностям, высокий профессионализм авторов видео курсов и использование современных методик оценки знаний.

### Новые видео

Исключения	0
Итераторы и генераторы	0

### Популярные видео курсы

Видео курс C# Стартовый (для начинающих)	9 уроков (16 ч. 3 мин.)
Видео курс по шаблонам проектирования	29 уроков (16 ч. 7 мин.)

### Теги

.NET Developer
Frontend Developer



# Проверка знаний

## TestProvider.com

TestProvider

Мы помогаем людям оценить себя

Регистрация Войти

Главная Каталог Сертификация Microsoft Поддержка О нас

### Тестирование

Языки программирования и информационные технологии

**Microsoft**

C# ASP.NET MVC JavaScript Patterns Of Design SQL Architecture Guide WCF HTML&CSS XML SEO WPF HTML5&CSS3 JQuery XNA SharePoint GUI for Android Windows Azure Platform Microsoft Patterns&Practices TFS SCRUM ReSharper TDD WWF LINQ Entity Framework Windows Forms Refactoring Microsoft Expression Blend 4 Windows Phone 8 Windows 8 AppStore Visual Studio Tips&Tricks MSF MEF SilverLight AJAX MEF Service Oriented Architecture

Пройти тест

Наши партнеры

Microsoft Partner CyberBionic ITVDN PROMETRIC TEST CENTER PEARSON VUE Authorized Test Center Windows Azure Cloud Partner EBA

Дополнительные ресурсы:

Очное обучение On-line обучение Видео обучение

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](http://TestProvider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



# Java Starter

Q&A



# Информационный видеосервис для разработчиков программного обеспечения

