

программирование

Лекция 1 Введение + ПО +
литература

Данные по предмету

Вид аттестации	Экзамен
Наличие КП	+
Дата сдачи КП	25.12.2017
Тематика КП	Десктоп приложения
Максимальный балл	60
Минимальный балл	40
Количество пар	л18+п18
ЛР	9 шт
Бонусные баллы	+

Программное обеспечение

Описание	Ссылка
Java	http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
Среда	https://netbeans.org/features/java/index_ru.html
Среда Андроид	https://developer.android.com/studio/index.html
Локальный java сервер	http://tomcat.apache.org/
Локальный сервер для бд	http://www.denwer.ru/
Альтернативная среда	https://www.jetbrains.com/idea/
Сборщик maven	https://maven.apache.org/

Литература

№	Название
1	«Java. Руководство для начинающих» Герберт Шилдт
2	«Философия Java» Брюс Эккель.
3	Head First Java, 2nd Edition
4	Java, A Beginner's Guide, 5th Edition
5	http://javarush.ru/ Интерактивное обучение java
6	https://vk.com/wall-76525381_30884 видео по java
7	http://startandroid.ru/ru/ Обучение андроид



Объектно-ориентированная парадигма

Методология объектно-ориентированного программирования

Увеличение размеров программ приводило к необходимости привлечения большего числа программистов, что, в свою очередь, потребовало дополнительных ресурсов для организации их согласованной работы.

В процессе разработки приложений заказчик зачастую изменял функциональные требования, что еще более усложняло процесс создания программного обеспечения.



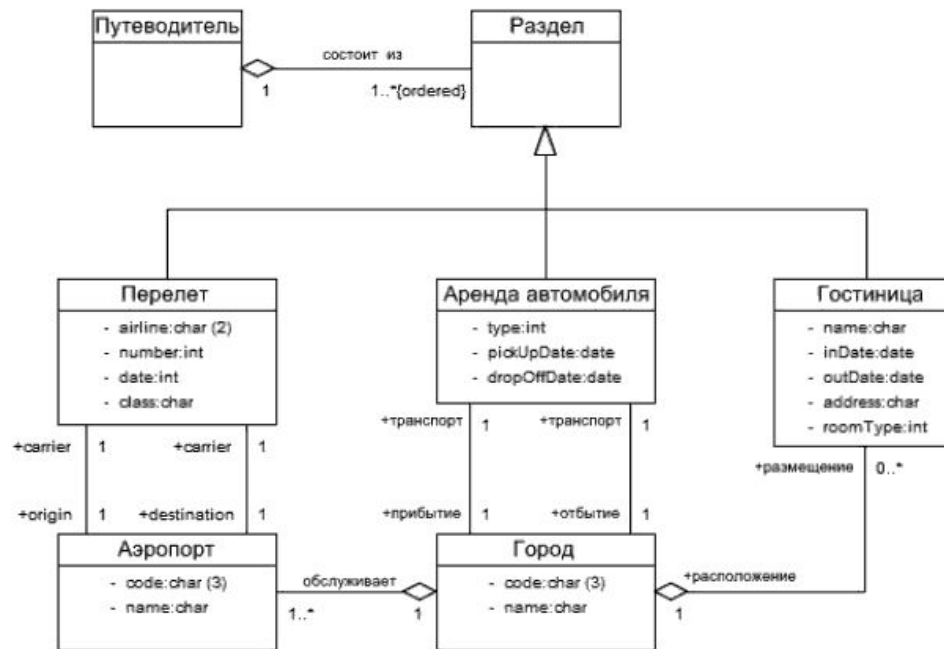
Методология объектно-ориентированного программирования

Традиционные методы процедурного программирования не способны справиться ни с нарастающей сложностью программ и их разработки, ни с необходимостью повышения их надежности.

Во второй половине 80-х годов возникла настоятельная потребность в новой методологии программирования, которая была бы способна решить весь этот комплекс проблем. Ею стало ООП.

Методология объектно-ориентированного программирования

Объектно-ориентированный подход к проектированию основан на представлении предметной области задачи в виде множества моделей для независимой от языка разработки программной системы на основе ее прагматики.



Методология объектно-ориентированного программирования

Прагматика определяется целью разработки программной системы, например, обслуживание клиентов банка, управление работой аэропорта, обслуживание чемпионата мира по футболу и т.п.

В формулировке цели участвуют предметы и понятия реального мира, имеющие отношение к создаваемой системе. При объектно-ориентированном подходе эти предметы и понятия заменяются моделями, т.е. определенными формальными конструкциями.

Объекты

По определению будем называть **объектом** понятие, абстракцию или любой предмет с четко очерченными границами, имеющий смысл в контексте рассматриваемой прикладной проблемы.

Введение объектов преследует две цели:

- понимание прикладной задачи (проблемы);
- введение основы для реализации на компьютере.

Объект - это мыслимая или реальная сущность, обладающая характерным поведением и отличительными характеристиками и являющаяся важной в предметной области .

Каждый объект имеет состояние, обладает четко определенным поведением и уникальной идентичностью.

Состояние

Состояние (state) - совокупный результат поведения объекта: одно из стабильных условий, в которых объект может существовать, охарактеризованных количественно; в любой момент времени состояние объекта включает в себя перечень (обычно статический) свойств объекта и текущие значения (обычно динамические) этих свойств.

Поведение

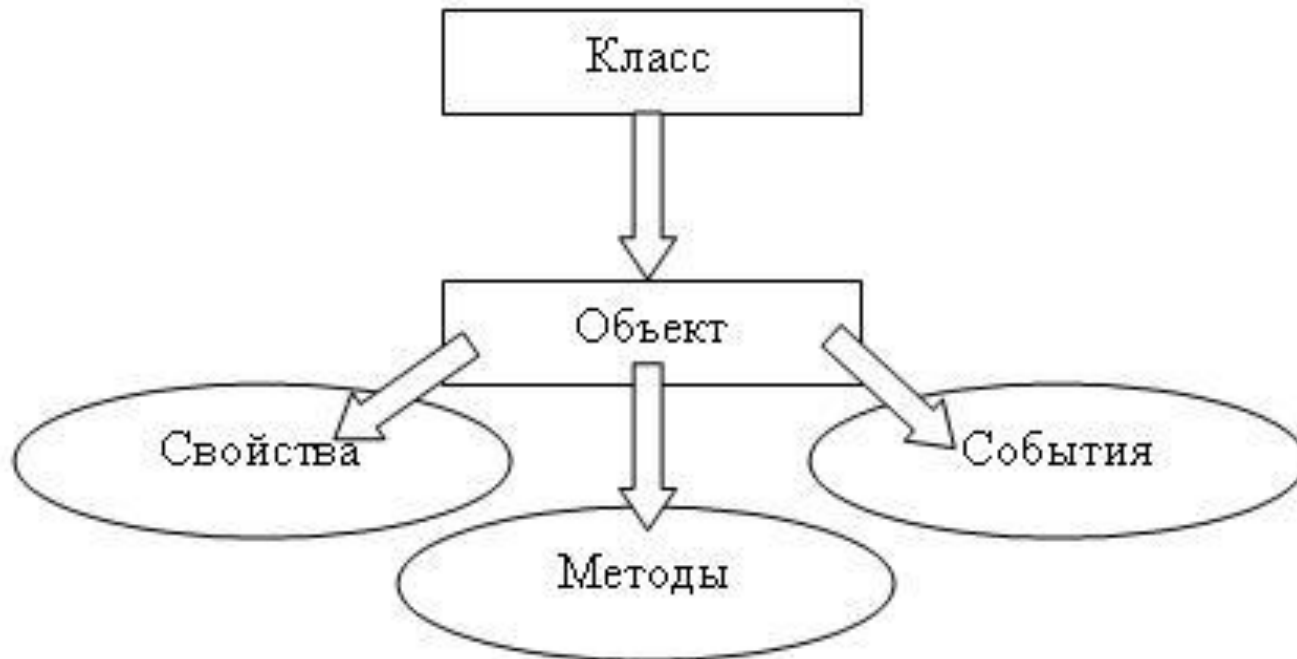
Поведение (behavior) - действия и реакции объекта, выраженные в терминах передачи сообщений и изменения состояния ;
видимая извне и воспроизводимая активность объекта

Уникальность

Уникальность (identity) -
свойство объекта; то, что отличает его от
других объектов.

Классы

Классы объектов являются «шаблонами», определяющими наборы свойств, методов и событий, по которым создается объект.





Абстракция

Абстрагирование – это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые.

Абстракция – это набор всех таких характеристик.

Инкапсуляция

Инкапсуляция - это механизм, который объединяет данные и код, манипулирующий этими данными, а также защищает и то, и другое от внешнего вмешательства или неправильного использования.

В объектно-ориентированном программировании код и данные могут быть объединены вместе; в этом случае говорят, что создаётся так называемый "чёрный ящик". Когда коды и данные объединяются таким способом, создаётся объект (object).

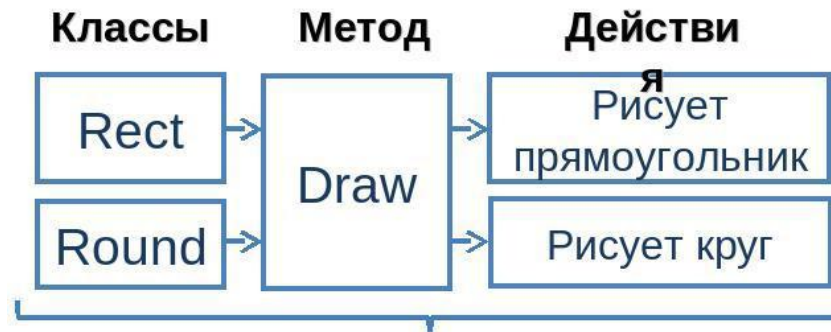
Наследование

Наследование (inheritance) - это процесс, посредством которого один объект может приобретать свойства другого. Точнее, объект может наследовать основные свойства другого объекта и добавлять к ним черты, характерные только для него. Наследование является важным, поскольку оно позволяет поддерживать концепцию иерархии классов (hierarchical class structure).



Полиморфизм

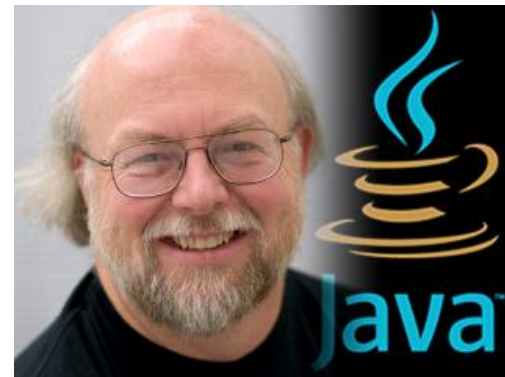
Полиморфизм (polymorphism) (от греческого polymorphos) - это свойство, которое позволяет одно и то же имя использовать для решения двух или более схожих, но технически разных задач. Целью полиморфизма, применительно к объектно-ориентированному программированию, является использование одного имени для задания общих для класса действий.



Полиморфизм

История Java

История Java восходит к 1991 году, когда группа инженеров из компании Sun под руководством Патрика Нотона (Patrick Naughton) и члена Совета директоров (и разностороннего компьютерного волшебника) Джеймса Гослинга (James Gosling) занялась разработкой небольшого языка, который можно было бы использовать для программирования бытовых устройств, например, контроллеров для переключения каналов кабельного телевидения (cable TV switchboxes).



ОСНОВЫ Java

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

ОСНОВЫ Java

Типы — примитивы (**primitives**):

- byte (число, 1 байт)
- short (число, 2 байта)
- int (число, 4 байта)
- long (число, 8 байтов)
- float (число с плавающей точкой, 4 байта)
- double (число с плавающей точкой, 8 байтов)
- char (символ, 2 байта)
- boolean (true (истина) или false (ложь), 1 байт)

ОСНОВЫ Java

Строки:

```
String s1 = new String("Who let the dogs out?");
```

```
String s2 = "Who who who who!";
```

```
String s3 = s1 + s2;
```

```
int num = 5;
```

```
String s = "I have " + num + " cookies";
```

ОСНОВЫ Java

Логические операции:

== равно

!= не равно

> больше

< меньше

>= больше или равно

<= меньше или равно

&& логическое и

|| логическое или

! Логическое не

ОСНОВЫ Java

Оператор if — else:

```
if (a == b) {  
    // a и b равны  
} else {  
    // a и b не равны  
}
```

```
if (a == b)  
    System.out.println("Yeah!");  
else  
    System.out.println("Ohhh...");
```

```
if (a == b)  
{  
    // a и b равны  
}
```

```
boolean a=true;  
if (a) {  
    // a true  
} else {  
    // a false  
}
```

ОСНОВЫ Java

Сокращенный оператор условия:

```
int a = 4;
```

```
int result = a == 4 ? 1 : 8;
```

```
// result будет равен 1
```

```
// Или обычная форма записи:
```

```
int result;
```

```
if (a == 4) {
```

```
    result = 1;
```

```
} else {
```

```
    result = 8;
```

```
}
```

Почитать и поделатать

Задачи:

<http://informatics.mccme.ru/mod/statements/view.php?id=2296>

Ввод-вывод, арифметические операции

<http://informatics.mccme.ru/mod/statements/view.php?id=276>

Условный оператор

Теория: Шилдт Java 8 Часть1- главы 1-5

Конец лекции 1