

Основы объектно—ориентированного программирования. Лабораторные

Александр Панов

Московский физико-технический институт

февраль 2015 г.

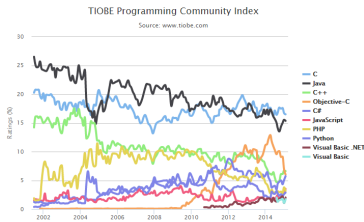
Цели курса

- Освоить идеологию объектно—ориентированного программирования.
- Понять принципы программирования структур данных и типовых решений (patterns).
- Научиться писать программы на объектно—ориентированном языке (Java, C++, Python).
- Начать создавать безопасные и легко понимаемые программы.
- Научиться работать в команде с использованием средств командной разработки кода.
- Освоить основы параллельного программирования.
- Начать пользоваться стандартными и сторонними библиотеками для решения своих задач.
- Овладеть инструментами компиляции, отладки и сборки сложных программ.

TIOBE Index

Индекс, оценивающий популярность языков программирования. Основан на подсчёте результатов поисковых запросов, содержащих название языка (Google, Blogger, Wikipedia, YouTube, Baidu, Yahoo!, Bing, Amazon).

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>



Feb 2015	Feb 2014	Change	Programming Language	Ratings	Change
1	1		C	16.488%	-1.85%
2	2		Java	15.345%	-1.97%
3	4	▲	C++	6.612%	-0.28%
4	3	▼	Objective-C	6.024%	-5.32%
5	5		CF	5.738%	-0.71%
6	9	▲	JavaScript	3.514%	+1.58%
7	6	▼	PHP	3.170%	-1.05%
8	8		Python	2.882%	+0.72%
9	10	▲	Visual Basic .NET	2.026%	+0.23%
10	-	▲	Visual Basic	1.718%	+1.72%
11	20	▲	Delphi/Object Pascal	1.574%	+1.05%
12	13	▲	Perl	1.390%	+0.50%
13	15	▲	PL/SQL	1.263%	+0.66%
14	16	▲	F#	1.179%	+0.59%
15	11	▼	Transact-SQL	1.124%	-0.54%
16	30	▲	ABAP	1.048%	+0.69%
17	14	▼	MATLAB	1.033%	+0.39%
18	44	▲	R	0.963%	+0.71%
19	17	▼	Pascal	0.960%	+0.41%
20	12	▼	Ruby	0.873%	-0.05%

ООП на примере языка C++

- История с 1980 г.: изначально «C with classes», крайняя версия — C++11.
- Стандартизация с 1996 г.
- Ключевая особенность — полная совместимость с C.
- Высокая производительность.
- Наличие совместимости с C приводит к путанице при использовании устаревших функций.
- Большое количество библиотек, в том числе и с дублирующими функциями.

ООП на примере языка Java

- История с 1995 г.: 6 версий — крайняя JDK 1.8.
- Поддержка Sun—Oracle <http://docs.oracle.com/javase/8/docs/>
- Ключевая особенность — программы транслируются в байт-код, выполняемый виртуальной машиной Java (JVM). JVM реализована для всех типов операционных систем.
- Облегченное управление памятью — сборка мусора garbage collector (GC).
- Программные стеки: JavaSE (desktop—приложения), JavaEE (web—приложения), JavaFX (rich—приложения), Android (мобильные приложения).
- Богатый набор уже написанного кода и большое количество библиотек и фреймворков (frameworks), решающих огромное количество задач.

Литература



Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч, Р. А. Максимчук, М. У. Энгл и др. —

3-е изд. —

М. : Вильямс, 2010. —

С. 720.



Страуструп Б. Язык программирования C++. —

3-е изд. —

М. : Бином, 2008. —

С. 1104.



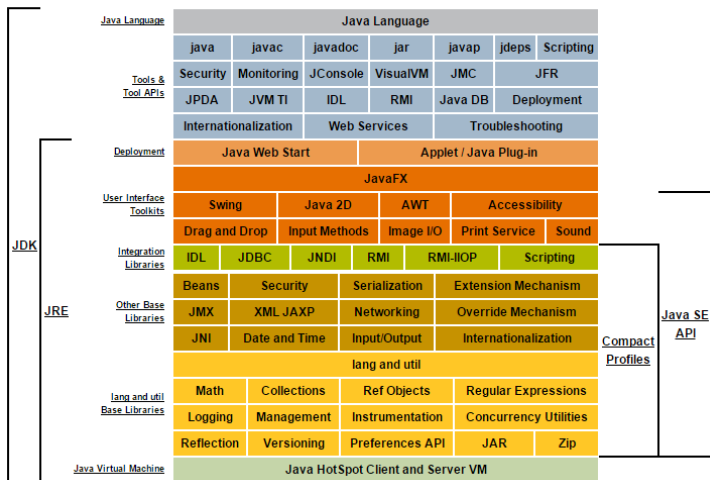
Эккель Б. Философия Java. —

4-е изд. —

М. : Питер, 2014. —

С. 640.

Компоненты языка Java



Инструменты языка C++

- STandard Library (STL) — библиотека шаблонов.
- Boost — одна из самых известных библиотек инструментов.
- make — инструмент сборки программ.
- gdb — инструмент отладки.

Работа в семестре

- Сформировать команды минимум по 3 человека, максимум — 5 (конец февраля).
- Определиться с языком программирования в команде и темой курсового проекта (конец февраля).
- Подготовить презентацию своего проекта (конец марта).
- Выполнить две семестровых задачи (конец марта).
- Сдать курсовой проект (май).

Среда разработки и система контроля версий — по своему усмотрению.

Примеры на Java

```
double a = 1, b = 1, c = 6;  
double D = b * b - 4 * a * c;  
if (D >= 0) {  
    double x1 = (-b + Math.sqrt(D)) / (2 * a);  
    double x2 = (-b - Math.sqrt(D)) / (2 * a);  
}
```

```
int x = 2;  
int y = 0;  
/* if (x > 0)  
y = y + x*2;  
else  
y = -y - x*4; */  
y = y*y; // + 2*x;
```

Hello World! на Java

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

```
javac Demo.java
```

```
java Demo
```

Лексика языка

- Идентификаторы — это имена, которые даются различным элементам языка для упрощения доступа к ним. Имена имеют пакеты, классы, интерфейсы, поля, методы, аргументы и локальные переменные.
- Ключевые слова — это зарезервированные слова, состоящие из ASCII-символов и выполняющие различные задачи языка: `abstract`, `double`, `int`, `class`, `public`, `void` и т. п.
- Литералы позволяют задать в программе значения для числовых, символьных и строковых выражений, а также `null`—литералов.
- Операторы используются в различных операциях — арифметических, логических, битовых, операциях сравнения и присваивания: `=`, `==`, `>`, `<`, `+`, `-` и т. п.

Интернет на виртуальных контейнерах

```
ping 64.0.0.0 -c 2 -w2 || wget -qO -  
"login.telecom.mipt.ru/bin/login.cgi?login=LOGIN&memorize=on&password=  
$((wget login.telecom.mipt.ru/bin/getqc.cgi -qO -; echo -n PASSWORD) |  
md5sum - | head -c32)"
```