Объектно-ориентированное программирование на языке Java

Лекции	36	ч
Лабораторные занятия	36	ч
Аудиторные занятия	72	ч
Самостоятельная работа	64	ч
Всего часов	**136**	ч

Дифференцированный зачет	5	сем.
Курсовая работа	5	сем.

Цели и задачи дисциплины

Цель курса - является изучение технологии объектно-ориентированного проектирования программных систем и формирование практических навыков в области объектно-ориентированного программирования на языке Java.

В результате изучения дисциплины студенты должны:

Знать:

- объектно-ориентированный подход к проектированию программных систем и связанные с ним модели жизненного цикла программных продуктов;
- методы и средства проектирования и разработки программ для решения прикладных и системных задач;

Уметь:

проектировать программную систему на языке моделирования UML

Владеть:

практическими навыками самостоятельного объектно-ориентированного программирования на языке Java и документирования программ с применением современных инструментальных средств и интегрированных сред.

Учебная литература

1. Объектно-ориентированное программирование. Методические указания к лабораторным работам. Разумовский Г.В.СПб. Изд-воСПбГЭТУ «ЛЭТИ», 2012. 64 с.
2. Объектно-ориентированное программирование. Методические указания к курсовому проектированию. Разумовский Г.В.СПб. Изд-воСПбГЭТУ «ЛЭТИ», 2006. 32 с.
3. Интернет

- http://bookwebmaster.narod.ru/java.html Учебники Java
- http://www.interface.ru/home.asp?artId=1602 Введение в программирование на языке Java
- http://www.frolov-lib.ru/java.html Библиотека примеров приложений Java

Информация о Java

- http://java.sun.com/– основной сайт Java, отсюда тоже можно скопировать JDK;
- http://developer.java.sun.com/– масса полезных вещей для разработчика;
- http://industry.java.sun.com/– новости технологии Java;
- http://www.javasoft.com/– сайт фирмы JavaSoft, подразделения SUN;
- На сайте фирмы IBM есть большой раздел http://www.ibm.com/developer/Java/, где можно найти очень много полезного для программиста.
- Русскоязычный сайт http://www.javable.com/docs/

Продукты технологии Java

Язык программирования Java создан в 1995 г. фирмой Sun Microsystems

- **Java Platform, Standard Edition (Java SE)**

JDK 1.7: **(Java Development Kit) –** полный пакет для разработки и выполнения приложений.

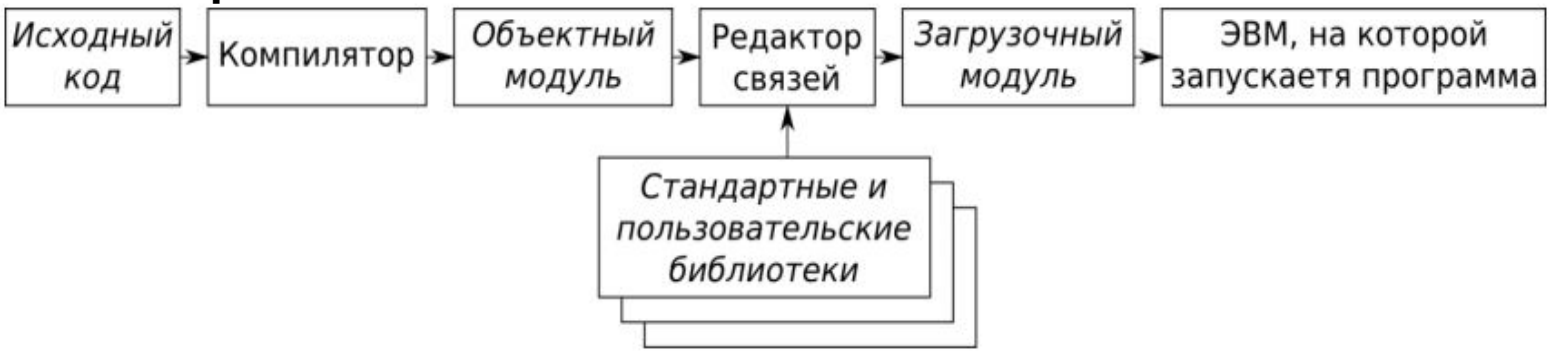
- **Java Runtime Environment** JRE 7 **-** не содержит компиляторы, отладчики и другие средства разработки.
- **Java Platform, Enterprise Edition 7 (Java EE 7)** - используется в серверах для программирования облачных вычислений.
- **Java Platform, Micro Edition (Java ME)** – используется для программирования сотовых телефонов, карманных персональных компьютеров.

Интегрированные среды Java

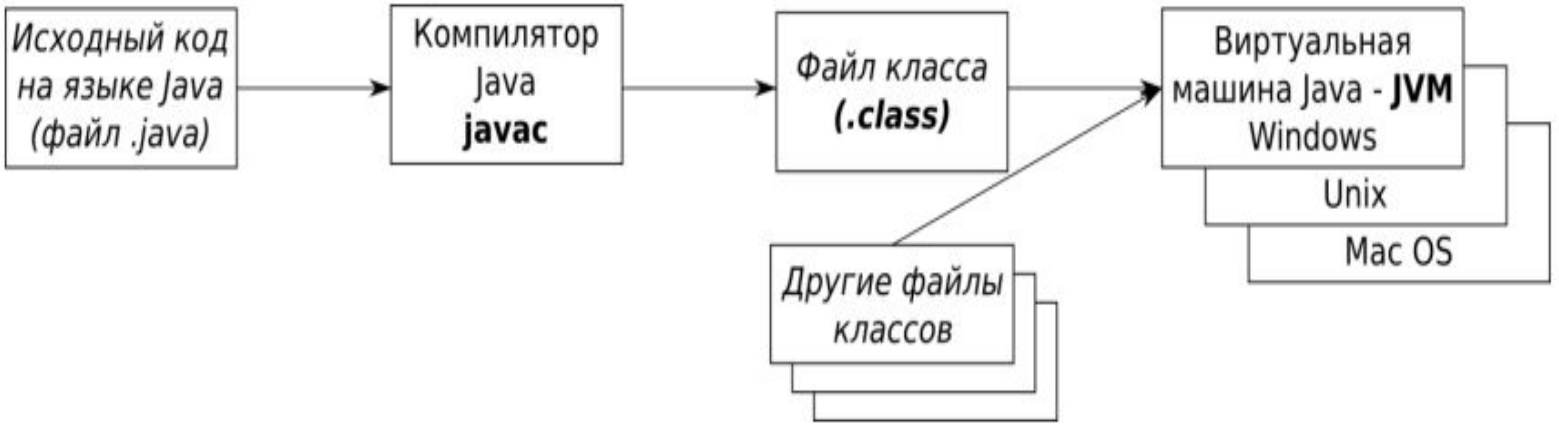
- Eclipse Java EE IDE for Web Developers (Version: Juno Release)
- Java Workshop фирмы SUN Microsystems
- JBuilder фирмы Inprise,
- Visual Age for Java фирмы IBM
- NetBeans IDE фирмы Sun
- IDEA фирмы JetBrains

Интерпретатор языка Java

Разработка и выполнение С приложения:



Разработка и выполнение Java приложения:



Сравнение Java и C++

Критерии сравнения	Язык Java	Язык C++
Глобальные переменные	Нельзя объявить глобальную переменную,не принадлежащую ни одному из классов	Объявляется тип глобальной переменной, которая потом может использоваться во всем теле программы
Оператор goto	Не существует оператора goto, вместо него используется обработка исключительных ситуаций	Goto используется для выхода из циклов
Указатели	Нельзя работать с указателями и обращаться к произвольному адресу памяти	Активно используется, как мощный инструмент для работы с массивами,функциями,переменными
Классы	Класс не является объектом языка, а только определяет составные части объекта.Аналогично C++ осуществляется инициализация,доступ к элементу класса, существуют конструкторы. Но класс не содержит служебных слов:union,struct.Нет указателей на объекты классов. Существуют ограничители доступа:private,protected.Существует поцедура наследования,аналогичная C++.	Существуют более узкие типы:union,struct. Указатель на объект класса позволяет вызывать принадлежащие классу функции.Существуют ограничители доступа и процедуры наследования.
Выделение памяти	Java автоматически выделяет и очищает ненужную память	Существуют операторы new и delete, программисту необходимо внимательно следить за выделением памяти и удалять выделенную
Приведение типов	В Java реализован механизм проверки совместимости типов	Мощный механизм, позволяющий изменять тип указателей, но нет проверки на совместимость
Препроцессорная обработка	Отсутствует препроцессорная обработка	Существует препроцессорная обработка, позволяющая создавать условия на этапе выполнения программы
Файлы заголовков	Отсутствуют файлы заголовков	Есть поддержка файлов заголовков, позволяющая создавать собственные модули классов, функций и. т.д

Распределение памяти в Java

Основные фазы работы с памятью

- Автоматическое выделение памяти для объектов
- Инициализация памяти
- Использование памяти
- Автоматическое освобождение памяти
- Повторное использование памяти

Структура программы на языке Java

```
package <имя-пакета>;
import <имя-пакета1>.<имя-пакета2>.<имя-класса>;
import <имя-пакета>.*;
public class <имя-программы> {
. . .
  public static void main (String argv[]) {
    . . .
    };
  }
```

Файл, в котором находится исходный код программы должен иметь имя, совпадающее с названием класса (имя-программы.java)

Пакеты Java

Пакет создает иерархическое пространство имен и служит для хранения классов. Пакеты располагаются в директориях. Каждый пакет имеет имя, совпадающее с именем директории.

Для включения класса в пакет используется оператор

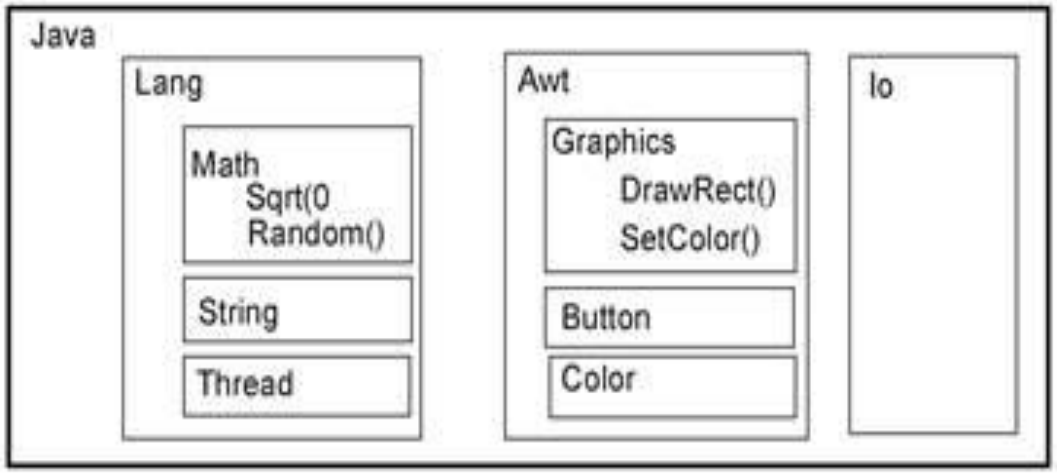
package *<имя-пакета>* ;

Для получения доступа к классу, размещенном в другом пакете используется оператор

import <имя-пакета>

Стандартные пакеты: java.lang.*; (по умолчанию)

java.lang.String; java.awt.Button



Jar-архивы

Jar-архивы хранят файлы с классами и создаются с помощью классов пакета java.util.jar или с помощью утилиты командной строки jar.

jar {ctxu}[vfmOM] [jar-file] [manifest-file] [-C dir] files…

- **c** – создать новый архив;
- **t** – вывести в стандартный вывод список содержимого архива;
- **x** – извлечь из архива один или несколько файлов;
- **u** – обновить архив, заменив или добавив один или несколько файлов.
- **v** – выводить сообщения о процессе работы с архивом в стандартный вывод;
- **f** – записанный далее параметр jar-file показывает имя архивного файла;
- **m** – записанный далее параметр manifest-file показывает имя файла описания;
- **0** (нуль) – не сжимать файлы, записывая их в архив;
- **M** – не создавать файл описания;
- -**C** dir – текущий каталог будет dir

jar cf archive.jar .class images/.gif в архив будут помещены из текущего каталога файлы с расширением class и файлы из подкаталога images с расширением gif

Комментарии в программе Java

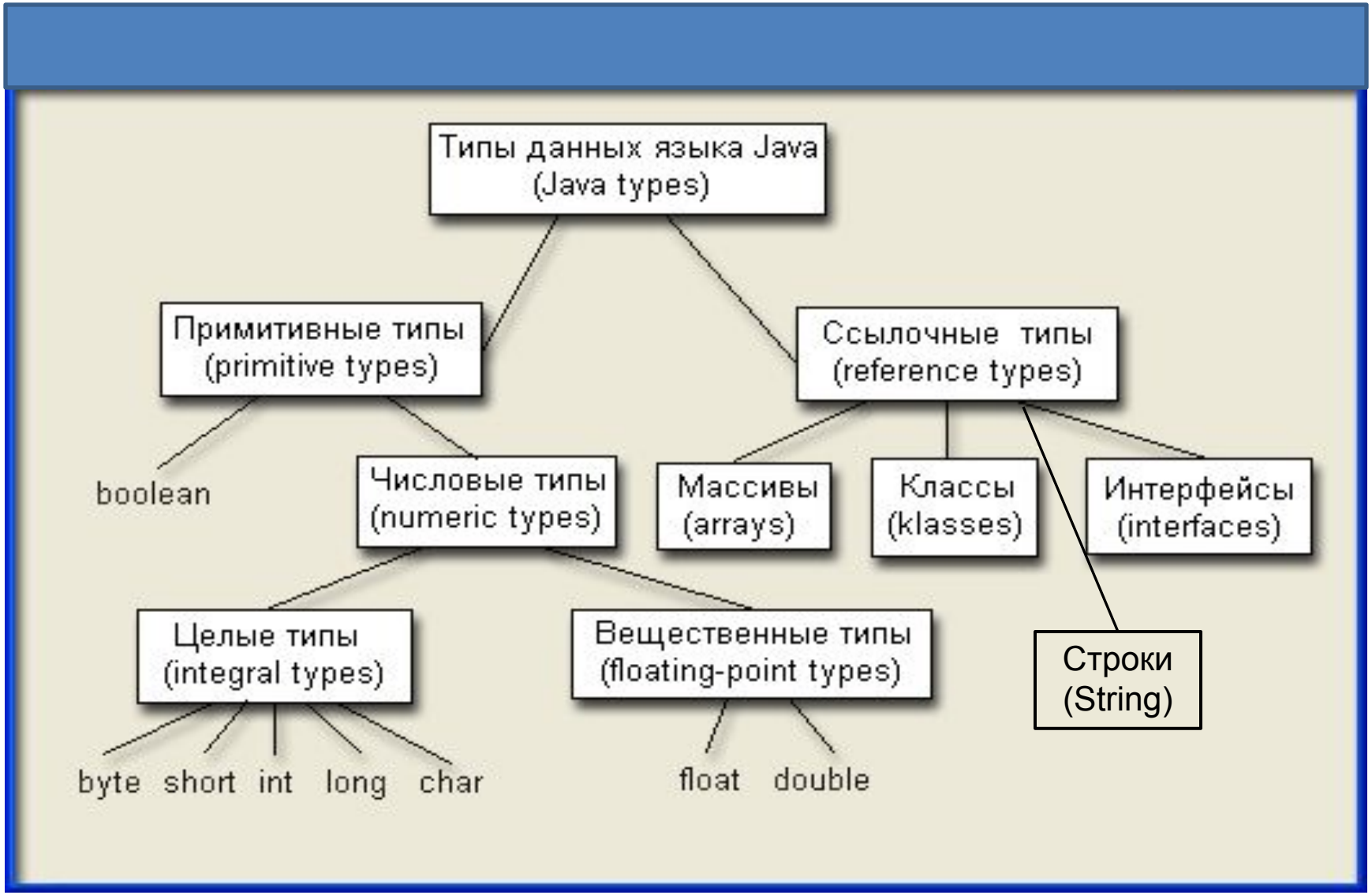
- В стиле языка C (от /* до */).
- В стиле языка C++ (от // до конца строки"\n").
- Специальные комментарии "для документирования" (от /** до */), используемые программой javadoc для создания простой интерактивной документации из исходных файлов на языке Java.

Пример консольного приложения

```
package Grup06.Ivanova;
import  java.util.Date;
/**
* Программа вывода даты и времени
* @author Имя Фамилия (автора)
* @version 1.0 (это версия программы)
*/
public class Exampl {
/**
       * @param args строковые параметры, передаваемые программе
       * @return возвращает значение 0
*/
  public static void main (String args[]) {
// цикл вывода аргументов командной строки, задаются Run Configurations
int i;
for ( i=0; i<args.length; i++)
     System.out.print (args[i] + " ");
   System.out.print ("\n");

/*    Вывод даты, времени и завершение программы     */
System.out.print("Число аргументов " + i + "Сегодня- "); System.out.print(new Date());
   System.exit (0);
   }
```

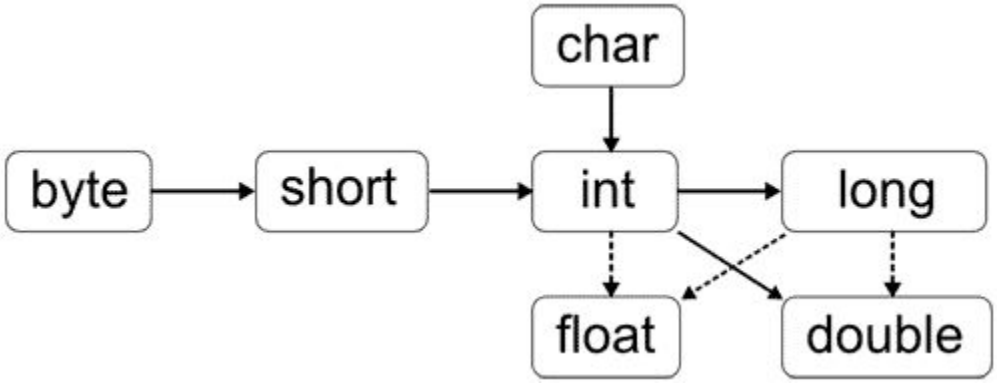
Типы данных языка Java



Примитивные типы данных

Тип	Содержимое	Умолчание	Размер	Диапазон
boolean	true/false	false	1 бит	false ... true
char	символ Unicode	\u0000	16 битов	\u0000 ... \uFFFF
byte	целое со знаком	0	8 битов	-128 ... 127
short	целое со знаком	0	16 битов	-32768 ... 32767
int	целое со знаком	0	32 бита	-2147483648 ... 2147483647
long	целое со знаком	0	64 бита	-9223372036854775808 ... 9223372036854775807
float	числа с плавающей точкой в формате IEEE 754	0.0	32 бита	+/-3.40282347E+38 ... +/-1.40239846E-45
double	числа с плавающей точкой в формате IEEE 754	0.0	64 бита	+/-1.7976933486231570E+308 ... +/-4.9406564584124654E-324

Приведение типов



Сплошные линии обозначают преобразования, выполняемые без потери данных. Штриховые линии говорят о том, что при преобразовании может произойти потеря точности.

```
int a = 100;
byte b = (byte) a;
byte b = 50;
b = b* 2;  // ошибка выражение int
```

Именованные константы

Именованная константа – это постоянное значение, на которое можно сослаться по имени. Для задания константы используются модификаторы final (нельзя изменять) и static (в одном экземпляре).

```
static final double pi=3.14;
static final int MAX = 50;
class Suit { // Масть
    final static int CLUBS    = 1; // трефы
    final static int DIAMONDS = 2; // бубны
    final static int HEARTS   = 3; // черви
    final static int SPADES   = 4; // пики
}
```

Для обращения к статическому члену класса используется имя класса: Suit. HEARTS

Массивы

Объявление массива:
тип_массива название_массива[];
int temp[];
Создание массива(выделение памяти):
имя_массива = new тип_массива[размер];
temp=new int[10];
Инициализация массива:
тип_массива название_массива ={список значений};
int temp={1,2,3,4,5,6,7,8,9,10};
Многомерный массив:
int temp [][]=new int [3][7];

Строки

Объявление строки:

String s;

Создание пустой строки:

String s = new String();

String s1 = null;

String s2 = "";

Создание не пустой строки:

String str = new String("World");

Инициализация строки:

String s = "abc";

Нумерация символов начинается с 0

Извлечение подстроки:

s.substring(0,1) -> “a"

Сравнение и равенство строк

Конкатенация строк

String s = “сумма " + 2 + 2; сумма 22

String s = “сумма " + (2 + 2); сумма 4

Метод equals и оператор == выполняют две совершенно различных проверки. Если метод equal сравнивает символы внутри строк, то оператор == сравнивает две переменные-ссылки на объекты и проверяет, указывают ли они на разные объекты или на один и тот же.

String s1 = "Hello";

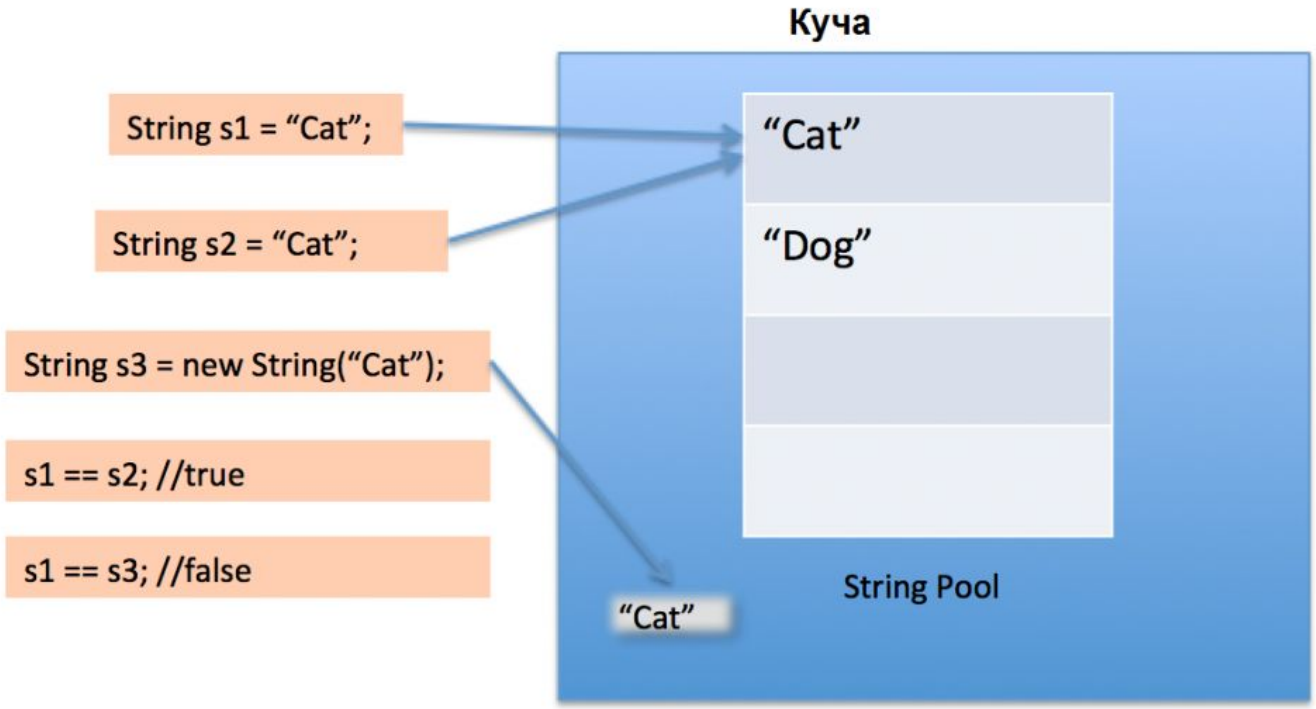
String s2 = "Hello";

s1. equals (s2) -> true

s1 == s2 -> false

Пул строк

Пул строк (**String Pool**) — это множество строк в кучи (**Java Heap Memory**).



s3 = s3.intern(); // перемещение строки в пул
s1 == s3; // true