***Класс String и его составляющие***

Класс String - Строка представляет собой последовательность символов. Для работы со строками в Java определен класс String, который предоставляет ряд методов для манипуляции строками. Физически объект String представляет собой ссылку на область в памяти, в которой размещены символы.

**Основные методы класса String**

**concat(): объединяет строки**

**valueOf(): преобразует объект в строковый вид**

**join(): соединяет строки с учетом разделителя**

**сompareTo(): сравнивает две строки**

**charAt(): возвращает символ строки по индексу**

**getChars(): возвращает группу символов**

**equals(): сравнивает строки с учетом регистра**

**equalsIgnoreCase(): сравнивает строки без учета регистра**

**regionMatches(): сравнивает подстроки в строках**

**indexOf(): находит индекс первого вхождения подстроки в строку**

**lastIndexOf(): находит индекс последнего вхождения подстроки в строку**

**startsWith(): определяет, начинается ли строка с подстроки**

**endsWith(): определяет, заканчивается ли строка на определенную подстроку**

**replace(): заменяет в строке одну подстроку на другую**

**trim(): удаляет начальные и конечные пробелы**

**substring(): возвращает подстроку, начиная с определенного индекса до конца или до определенного индекса**

**Операторы + и += для String**

На языке Java знак плюс (+) означает конкатенацию строк (con**cat**enation), иными словами - объединение строк.

String cat = "Кот";

String name = "Васька";

//складываем две строки и пробел между ними, чтобы слова не слиплись

String fullname = cat + "" + name; // получится Кот Васька

Если один из операндов в выражении содержит строку, то другие операнды также должны быть строками. Поэтому Java сама может привести переменные к строковому представлению, даже если они не являются строками.

int digit = 4;

String paws = "лапы";

String aboutcat = digit + paws; // хотя мы складываем число и строку, но все равно получим строк

## Форматирование строк

Предположим у нас есть строковый ресурс:

<string name="aboutcat">У кота по имени Барсик четыре лапы, один хвост. Ему 5 лет</string>

Чтобы вывести эту строку программно в элементе TextView, можно использовать код:

TextView tvCatsInfo = (TextView)findViewById(R.id.textView1);

tvCatsInfo.setText(R.string.aboutcat);

Представьте себе, что у вас несколько котов. Можно, конечно, завести для каждого кота свою строку. Но ведь строки очень похожи, меняются только имена и возраст. Также можно менять число лап и хвостов (что вы курите?).

В таких случаях можно применить форматирование строк. Нужно определить слова, которые мы будем менять и заменить их на специальный набор символов, которые начинаются с символа процента, затем идет число, увеличивающееся на единицу, далее $s для строк или $d для чисел. Итак, изменим наш строковой ресурс так:

<string name="aboutcat">У кота по имени %1$s %2$s лапы, %3$s хвост. Ему %4$d лет</string>

Внесём изменения в код:

String strBarsik = "Барсик";

String strPaws = "четыре";

String strTail = "один";

int year = 5;

String strCats = getResources().getString(R.string.aboutcat);

String strFinal = String.format(strCats, strBarsik, strPaws, strTail, year);

infoTextView.setText(strFinal);

Если вас есть кот Васька и ему шесть лет, то добавляем две переменные и форматируем строку

String strVaska = "Васька";

year = 6;

String strFinal = String.format(strCats, strVaska, strPaws, strTail, year);

infoTextView.setText(strFinal);

Здесь показан простейший пример с форматированием. Помните о нём и применяйте в нужных местах.

## Строковой ресурс

Строки желательно хранить в ресурсах (о ресурсах есть [отдельная статья](http://developer.alexanderklimov.ru/android/theory/resources.php)).

Программно доступ к строковому ресурсу делается так:

String catName = getResources().getString(R.string.barsik)

## Извлечь строки из строковых массивов в ресурсах

Предположим, у вас есть строковый массив, определённый в файле **strings.xml** под именем **cats\_array**. Тогда получить доступ к строкам из ресурсов можно так:

Resources res = getResources();

String[] cats = res.getStringArray(R.array.cats\_array);

## Методы

Возвращает символ с указанным смещением в этой строке. Отсчёт идёт от 0. Не надо использовать отрицательные и несуществующие значения, будьте серьёзнее. Для извлечения нескольких символов используйте getChars().

String testString = "Котёнок";

char myChar = testString.charAt(2);

tv.setText(Character.toString(myChar)); // выводит третий символ -