

4. OOP. JAVA

Strings

Класс String

```
public final class String extends Object  
    implements Serializable, Comparable<String>,  
        CharSequence
```

- общедоступный финальный (не может иметь подклассов);
- неизменяемым (immutable, созданный объект не может быть изменен).

Доступ к элементам строки

можно осуществить с помощью метода `charAt(int index)`

```
public char charAt(int index)
```

нумерация символов строки начинается с нуля.

Перегрузка операции "+" для строк

Если в операции сложения участвует строка, то второй операнд всегда приводится к строке.

При конкатенации (сложении) строк будет создан новый объект типа `String`, т.к. строки - неизменяемые объекты.

Получение строкового представления значений примитивных типов

Для того чтобы получить строковое представление значения примитивного типа следует воспользоваться статическим методом **valueOf**, который возвращает соответствующую строку.

Строковые литералы (строки-константы)

Строковые литералы (строки-константы) представляют из себя совокупность символов заключенных в двойные кавычки и являются объектными переменными.

Два строковых литерала, состоящих из одного и того же набора символов ссылаются на один и тот же объект-строку.

Сравнение строк посимвольно

Для сравнения строк посимвольно следует использовать метод `equals` объекта `String`.

```
public boolean equals(Object object)
```

Метод substring

Класс String содержит перегруженный метод **substring**, которой возвращает подстроку, определяемую значениями индексов начала и конца подстроки, передаваемых в метод в качестве параметров.

// возвр. часть строки, начиная с beginIndex:
public String substring(int beginIndex)

// возвр. часть строки от beginIndex до endIndex-1:
public String substring(int beginIndex, int endIndex)

Метод concat

Метод **concat** класса String *возвращает* строку this объединенную со строкой str.

```
public String concat(String str)
```

Метод `replace`

Метод **`replace`** *возвращает* строку `this` заменяя в ней все вхождения символа `oldChar` на `newChar`.

```
public String replace(char oldChar, char newChar)
```

StringBuffer

Является `mutable`, то есть объекты – изменяемые.

Может быть использован в многопоточных средах, так как все необходимые методы являются синхронизированными.

StringBuilder

Представляет изменяемую последовательность символов.

Был добавлен в Java 5 и во всем идентичен классу StringBuffer, но не синхронизирован, т.о. работает быстрее.