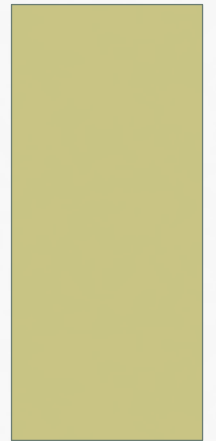


Interface again

JAVA ADVANCED (NOT SO)



INTERFACE

- ```
interface Instrument {
 int VALUE = 5; // public, static и final!
 void play(Note n); // public and abstract!
 void adjust(); // static methods are forbidden!
}
```
- ```
class Percussion implements Instrument {  
    public void play(Note n) {  
        ...  
    }  
    public void adjust() {  
        ...  
    }  
}
```

String

НЕ ПРИМИТИВНЫЕ ТИПЫ JAVA

СТРОКИ

- Последовательность символов произвольной длины
- Класс `java.lang.String`
- Не то же, что массив символов
- Никаких нулевых символов в конце, длина хранится отдельно
- Строки не изменяемы!

- `String hello = " Hello";`
- `String specialChars = "\r\n\t\"\\ \u0101\u2134\u03ff ";`
- `char[] charArray = {'a', 'b', 'c'};`
- `String string = new String (charArray);`

ДОСТУП К СОДЕРЖИМОМУ

- `int length()`
- `char charAt(int index)`
- `char[] toCharArray()`

- `String substring(int beginIndex)`
- `String substring(int beginIndex, int endIndex)`

СРАВНЕНИЕ СТРОК

- Оператор `==` сравнивает ссылки, а не содержимое строки
- `boolean` `equals(Object anotherObject)`
- `boolean` `equalsIgnoreCase(String anotherString)`
- `int` `compareTo(String anotherString)`
- `int` `compareToIgnoreCase(String anotherString)`

ОПЕРАЦИИ

- `boolean startsWith(String prefix)`
- `boolean endsWith(String suffix)`
- `int indexOf(String str)`
- `int lastIndexOf(String str)`

- `String trim()`
- `String replace(char oldChar, char newChar)`
- `String toLowerCase()`
- `String toUpperCase()`

КОНКАТЕНАЦИЯ СТРОК

- `String concat(String str)`
- Оператор `+` (работает, как `StringBuilder`)
`String helloWorld = " Hello " + " World!";`
- `java.lang.StringBuilder`
`StringBuilder buf = new StringBuilder();`
`buf.append ("Hello");`
`buf.append ("World");`
`buf.append ('!');`
`String result = buf.toString();`

РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ

- Регулярные выражения поддерживаются в стандартной библиотеке Java
- `boolean matches(String regex)`
- `String[] split(String regex)`
- `String replaceAll(String regex, String replacement)`
- `String replaceFirst(String regex, String replacement)`

ПРИМЕРЫ

```
String str = "a, b, c,d, e";  
String[] items = str.split(", *");  
// items -> {"a", "b", "c", "d", "e"}
```

```
String str = "(aa)(bb)(cccc)";  
String regex = "\\(([^)]*)\\)";  
String result = str.replaceAll(regex, "$1");  
// result -> "aabbcccc"
```