Fondamenti di Java

Continua...

Convenzioni

I nomi delle Classi iniziano con la MAIUSCOLA

I nomi degli Oggetti iniziano con la MINUSCOLA

```
Pila p=new Pila();
```

Passaggio di parametri

Le variabili dei tipi di dati primitivi sono sempre passati per copia.

Gli oggetti sono sempre passati per referenza.

(a pensarci, é ovvio: si *copia* l'identificatore dell'oggetto)

Passaggio di paramet


```
public class Numero {
  public int valore=0;
  Numero(int valore) {
   this.valore=valore;
  }
}
```

5 Programmazione 2 - Marco Ronchetti

Passaggio di paramet

```
public class Parametri {
  void incrementa(int x) {x++;}
  void incrementa(Numero x) {
     x.valore++;}
  public static void main(String a[]){
    Parametri p=new Parametri();
  }
  Parametri() {
    int z=5;
    incrementa(z);
    System.out.println(z);
    Numero n=new Numero(z);
    incrementa(n);
    System.out.println(n.valore);
  }
}
```

```
public class Numero {
  public int valore=0;
  Numero(int valore) {
   this.valore=valore;
  }
}
```

Output:
5
6

6 Programmazione 2 - Marco Ronchetti

package

Una collezione di classi correlate

```
package myclasses;
class A {...};
class B {...};


import myclasses.A;

import myclasses.*;
```

Fac.Scienze – Università di Trento

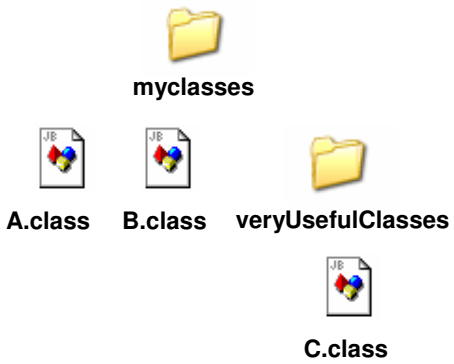
Annidamento di package


```
package myclasses;
class A {...};

package myclasses;
class B {...};

package myclasses.veryUsefulClasses;
class C {...};

import myclasses.*; // NON importa C!
```

Definizione suggerita di un nome univoco per i packages:
È basata sul nome internet (es.: it.unitn.science.mypackage)

Annidamento di package

I packages si riflettono in una struttura di directories



Class String

java.lang

Class String

java.lang.Object
  |
  +--**java.lang.String**

All Implemented Interfaces:
CharSequence, Comparable, Serializable

public final class **String**
extends Object
implements Serializable, Comparable, CharSequence

The `String` class represents character strings. All string literals in Java programs, such as `"abc"`, are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:

```
String str = "abc";
```

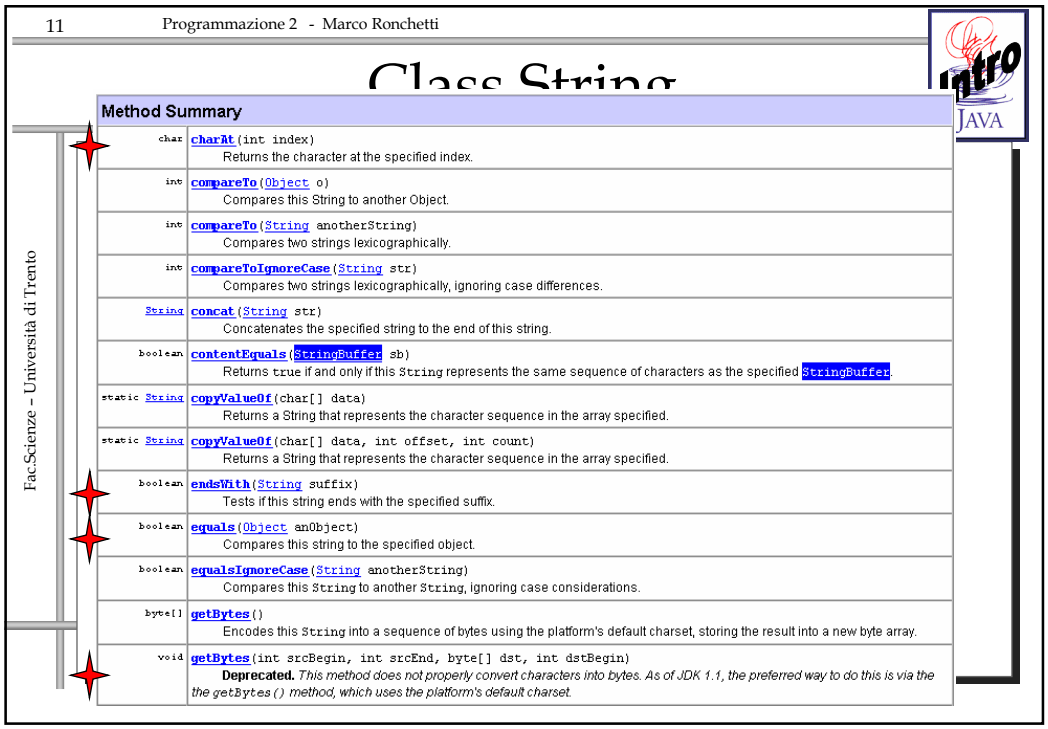
is equivalent to:

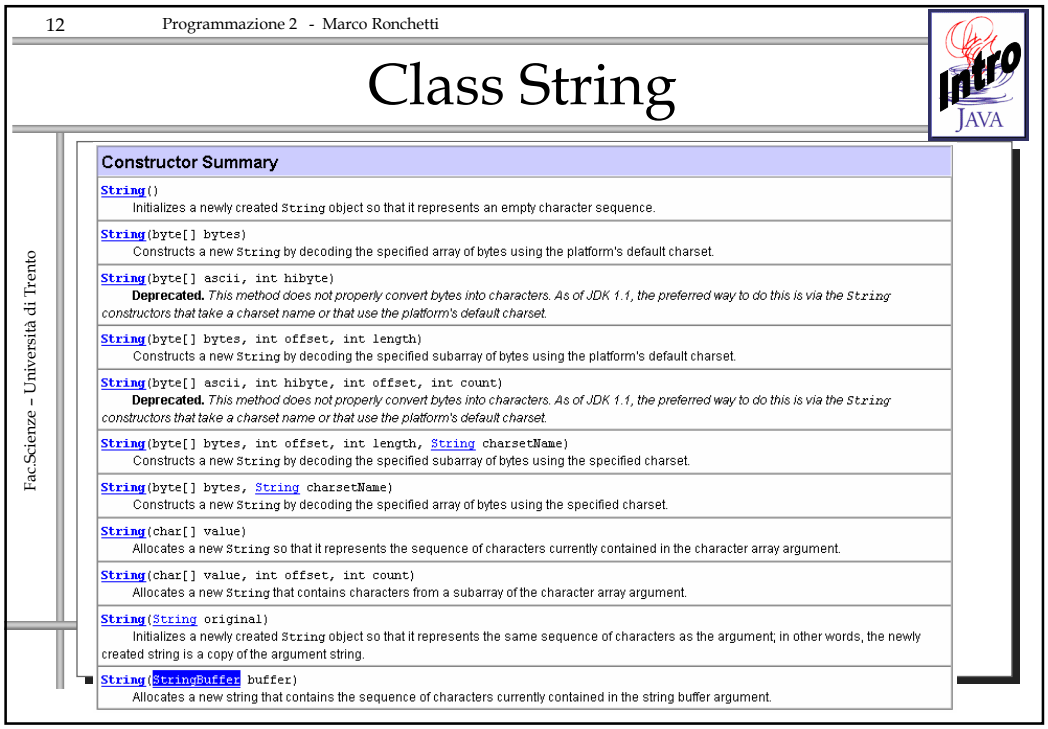
```
char data[] = {'a', 'b', 'c'};
String str = new String(data);
```


Class String

Constructor Summary

String() Initializes a newly created `String` object so that it represents an empty character sequence.
String(byte[] bytes) Constructs a new `String` by decoding the specified array of bytes using the platform's default charset.
String(byte[] ascii, int hibyte) **Deprecated.** *This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the `String` constructors that take a charset name or that use the platform's default charset.*
String(byte[] bytes, int offset, int length) Constructs a new `String` by decoding the specified subarray of bytes using the platform's default charset.
String(byte[] ascii, int hibyte, int offset, int count) **Deprecated.** *This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the `String` constructors that take a charset name or that use the platform's default charset.*
String(byte[] bytes, int offset, int length, String charsetName) Constructs a new `String` by decoding the specified subarray of bytes using the specified charset.
String(byte[] bytes, String charsetName) Constructs a new `String` by decoding the specified array of bytes using the specified charset.
String(char[] value) Allocates a new `String` so that it represents the sequence of characters currently contained in the character array argument.
String(char[] value, int offset, int count) Allocates a new `String` that contains characters from a subarray of the character array argument.
String(String original) Initializes a newly created `String` object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.
String(StringBuffer buffer) Allocates a new string that contains the sequence of characters currently contained in the string buffer argument.


11 Programmazione 2 - Marco Ronchetti
Class String
Intro JAVA
Fac.Scienze - Università di Trento
Method Summary
char charAt(int index) Returns the character at the specified index.
int compareTo(Object o) Compares this String to another Object.
int compareTo(String anotherString) Compares two strings lexicographically.
int compareToIgnoreCase(String str) Compares two strings lexicographically, ignoring case differences.
String concat(String str) Concatenates the specified string to the end of this string.
boolean contentEquals(StringBuffer sb) Returns true if and only if this String represents the same sequence of characters as the specified StringBuffer.
static String copyValueOf(char[] data) Returns a String that represents the character sequence in the array specified.
static String copyValueOf(char[] data, int offset, int count) Returns a String that represents the character sequence in the array specified.
boolean endsWith(String suffix) Tests if this string ends with the specified suffix.
boolean equals(Object anObject) Compares this string to the specified object.
boolean equalsIgnoreCase(String anotherString) Compares this String to another String, ignoring case considerations.
byte[] getBytes() Encodes this String into a sequence of bytes using the platform's default charset, storing the result into a new byte array.
void getBytes(int srcBegin, int srcEnd, byte[] dst, int dstBegin) Deprecated. This method does not properly convert characters into bytes. As of JDK 1.1, the preferred way to do this is via the the getBytes() method, which uses the platform's default charset.


12 Programmazione 2 - Marco Ronchetti
Class String
Intro JAVA
Fac.Scienze - Università di Trento
Constructor Summary
String() Initializes a newly created String object so that it represents an empty character sequence.
String(byte[] bytes) Constructs a new String by decoding the specified array of bytes using the platform's default charset.
String(byte[] ascii, int hibyte) Deprecated. This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the String constructors that take a charset name or that use the platform's default charset.
String(byte[] bytes, int offset, int length) Constructs a new String by decoding the specified subarray of bytes using the platform's default charset.
String(byte[] ascii, int hibyte, int offset, int count) Deprecated. This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the String constructors that take a charset name or that use the platform's default charset.
String(byte[] bytes, int offset, int length, String charsetName) Constructs a new String by decoding the specified subarray of bytes using the specified charset.
String(byte[] bytes, String charsetName) Constructs a new String by decoding the specified array of bytes using the specified charset.
String(char[] value) Allocates a new String so that it represents the sequence of characters currently contained in the character array argument.
String(char[] value, int offset, int count) Allocates a new String that contains characters from a subarray of the character array argument.
String(String original) Initializes a newly created String object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.
String(StringBuffer buffer) Allocates a new string that contains the sequence of characters currently contained in the string buffer argument.

Class String

Method Detail

length

```
public int length()
```

Returns the length of this string. The length is equal to the number of 16-bit Unicode characters in the string.

Specified by:
`length` in interface `CharSequence`

Returns:
the length of the sequence of characters represented by this object.

---

charAt

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

Returns the character at the specified index. An index ranges from 0 to `length() - 1`. The first character of the sequence is at index 0, the next at index 1, and so on, as for array indexing.

Specified by:
`charAt` in interface `CharSequence`

Parameters:
`index` - the index of the character.
Returns:
the character at the specified index of this string. The first character is at index 0.
Throws:
`IndexOutOfBoundsException` - if the `index` argument is negative or not less than the length of this string.

String

Per trasformare il contenuto di una stringa in un intero:

int Integer.parseInt(String s)

Per trasformare il contenuto di una stringa in un float:

float Float.parseFloat(String s)

Parametri di ingresso

I parametri del main sono inclusi in un vettore di String


```
/* sum and average command lines */
class SumAverage {
  public static void main (String args[]) {
    int sum = 0;
    float avg = 0;
    for (int i = 0; i < args.length; i++) {
      sum += Integer.parseInt(args[i]);
    }
    System.out.println("Sum is: " + sum);
    System.out.println("Average is: "
        + (float)sum / args.length);
  }
}
```