UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Fondamenti di Informatica (117IN)

A.A. 2022 / 2023

Lezione 7 - Enumerazioni, Array e String

Sylvio Barbon Junior
sylvio.barbonjunior@units.it

Sommario:

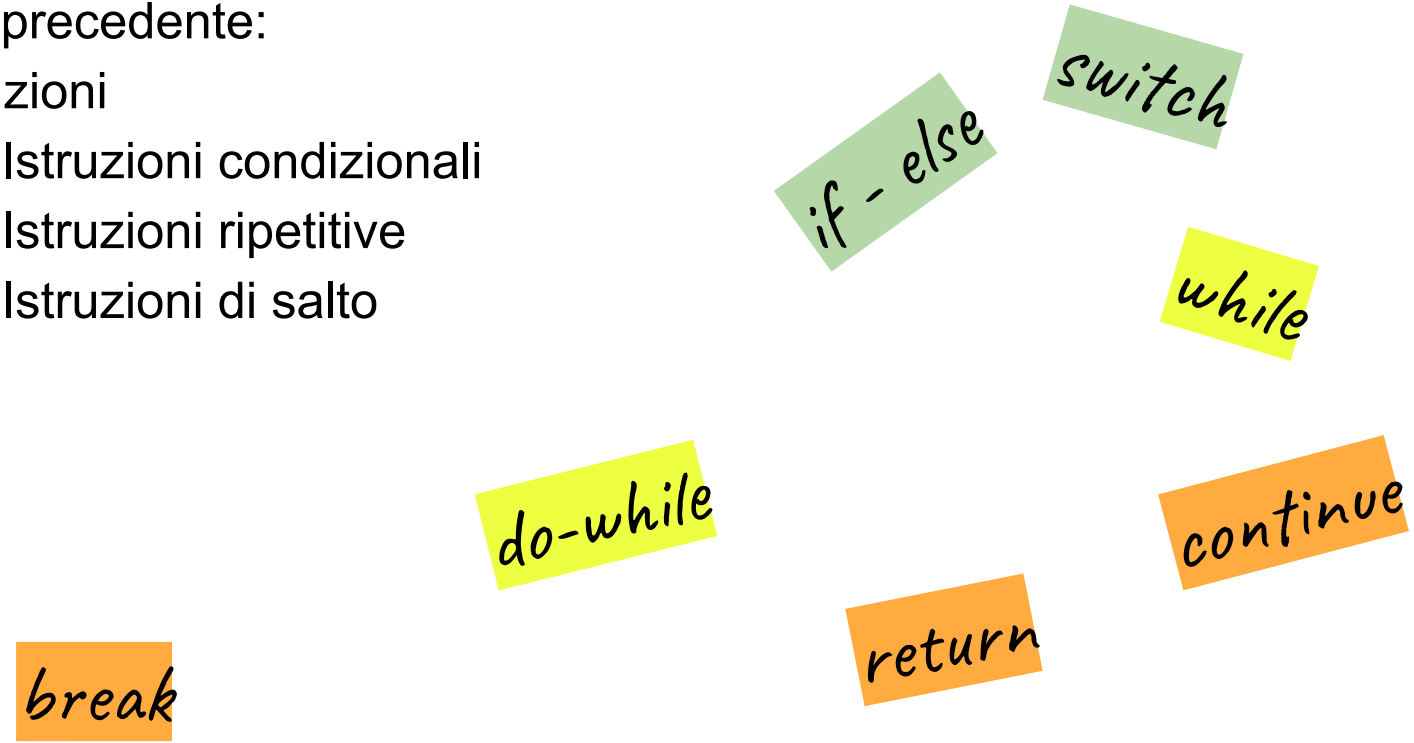
1) Tipi derivati, variabili e oggetti
   a) Enumerazione
2) Arrays
3) Classe String

Sylvio Barbon Junior - sylvio.barbonjunior@units.it

Lezione precedente:

Istruzioni

a) Istruzioni condizionali
b) Istruzioni ripetitive
c) Istruzioni di salto



Sylvio Barbon Junior - sylvio.barbonjunior@units.it

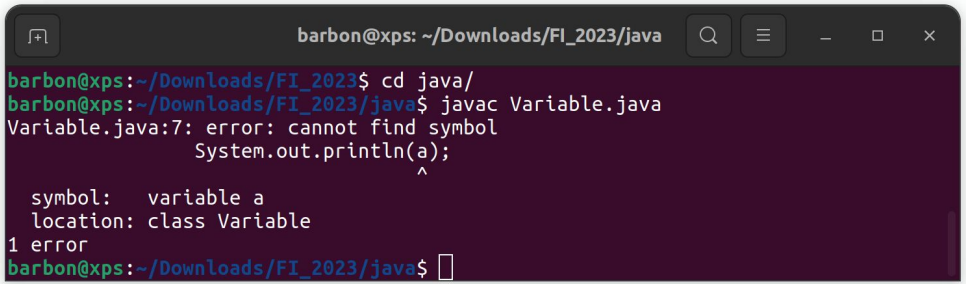
1) Tipi derivati, variabili e oggetti

- I tipi primitivi prevedono la definizione di variabili di un dato tipo;
- Una variabile di un tipo primitivo ha un **nome** e un **valore**;
- Il **tempo di vita** di una variabile dipende dal luogo in cui è definita;
- I tipi derivati (**enumerazioni, array, stringhe**, ecc) prevedono la definizione **del tipo**, la **definizione di variabili** di quel tipo e la creazione **di oggetti** di quel tipo;
- Un tipo derivato è costituito da **più componenti** raggruppate in un'unica entità;

Sylvio Barbon Junior - sylvio.barbonjunior@units.it

1) Tipi derivati, variabili e oggetti

```java
public class Variable{
    public static void main(String[] args) {
        if (args.length>0){
            int a = 2;
            System.out.println(a);
        }
        System.out.println(a);
    }
}
```



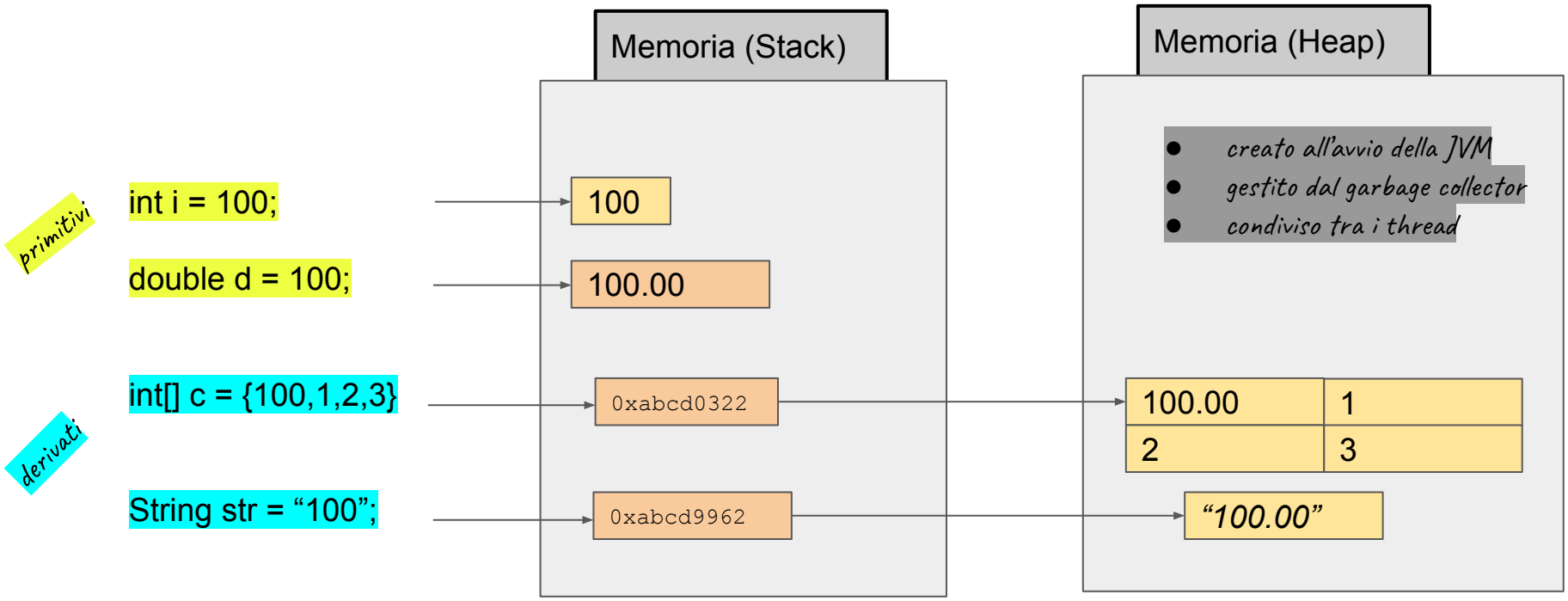
Sylvio Barbon Junior - sylvio.barbonjunior@units.it

1) Tipi derivati, variabili e oggetti

- Una variabile di un tipo derivato ha le stesse caratteristiche di una variabile di un tipo primitivo, con la differenza che il suo valore è un indirizzo, quello di un oggetto di quel tipo;
-
- Un oggetto di un tipo derivato ha un tempo di vita dinamico: viene creato esplicitamente quando serve (per mezzo dell'operatore new), allocati in una zona di memoria detta appunto **memoria dinamica**.

Sylvio Barbon Junior - sylvio.barbonjunior@units.it

1) Tipi derivati, variabili e oggetti



Sylvio Barbon Junior - sylvio.barbonjunior@units.it

1) Tipi derivati, variabili e oggetti

- Tipo enumerazione:
  - I tipi enumerazione vengono usati per rappresentare **un numero limitato di valori** associati a informazioni non numeriche;
  - Gli enumeratori sono identificatori di **costanti**;
  - Gli oggetti riferiti dagli enumeratori vengono **creati implicitamente** nel momento in cui viene definito il tipo.

```
1  public class Semaforo{
2      public static void main(String[] args){
3          enum SemaforoEnum {verde, giallo, rosso};
4
5          SemaforoEnum attuale = SemaforoEnum.verde;
6
7          System.out.println("Attualle:"+attuale);
8          for(SemaforoEnum i: SemaforoEnum.values())
9              System.out.println(i);
10     }
11 }
```

```
sylviobarbon@ws-piccola
Attualle:verde
verde
giallo
rosso
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

1) Tipi derivati, variabili e oggetti

- Tipo enumerazione:

```java
public class EnumEsempio{
    enum Uomo {habilis, erectus, sapiens};

    public static void main(String[] args) {
        Uomo primitivo = Uomo.habilis;
        System.out.println(primitivo);

        Uomo contemporaneo = Uomo.sapiens;
        System.out.println(contemporaneo);

        for(Uomo u : Uomo.values())
            System.out.println(u);
    }
}
```

```
barbon@xps: ~/Downloads/FI_2023/java
barbon@xps:~/Downloads/FI_2023/java$ java EnumEsempio
habilis
sapiens
habilis
erectus
sapiens
barbon@xps:~/Downloads/FI_2023/java$
```

2) Arrays

- Tipo array:
  - Rappresenta un **aggregato** di elementi **di uno stesso tipo**;
  - Ogni elemento si accede mediante un indice che ne individua la posizione (un numero intero positivo N di elementi, il cui indice va da 0 a N-1).
  - Gli elementi di un array vengono selezionati per mezzo dell'operatore di indicizzazione '[ ]'

```
1  public class EsempioArray{
2      public static void main(String[] args) {
3          int[] numArray1 = new int[5];
4          int numArray2[];
5          int numArray3[] = {1,2,3,4,5};
6
7          numArray1[1] = 300;
8          numArray2 = new int[100];
9
10         System.out.println(numArray1[1]);
11         System.out.println(numArray2[99]);
12         System.out.println(numArray3[0]);
13     }
14 }
```

Vari tipi di dichiarazioni.

Non aveva valori attributi, è stato iniziato con tutti valore come 0

```
sylviobarbon@ws-piccola:
300
0
1
```

2) Arrays

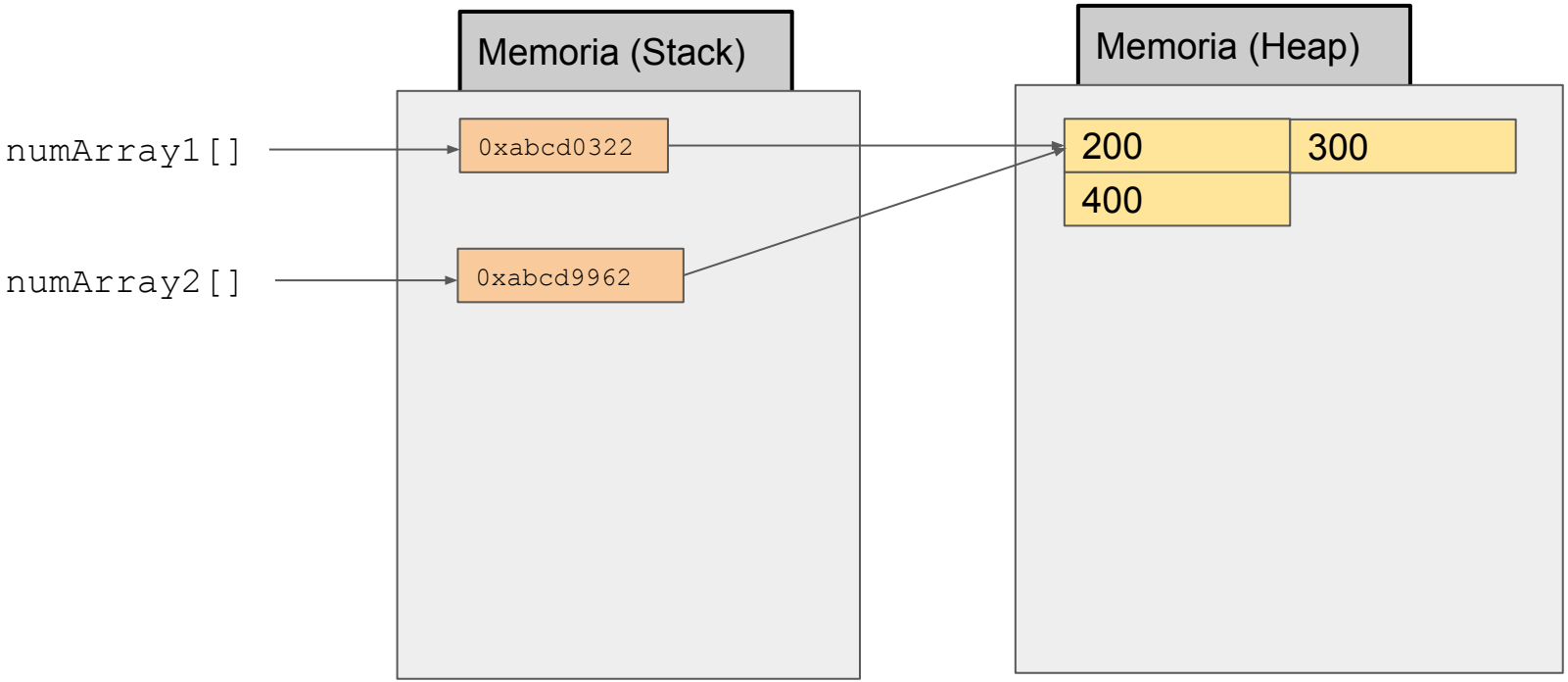
- Tipo array:
  - **L'operatore di assegnamento** (=) può essere applicato anche a variabili array dello stesso tipo: vengono però coinvolti i riferimenti, e non gli oggetti array.

```
1  public class EsempioArray2{
2      public static void main(String[] args) {
3          int[] numArray1 = {200,300,400};
4          int numArray2[];
5
6          numArray2 = numArray1;
7          System.out.println("numArray1:"+numArray1[0]);
8          System.out.println("numArray2:"+numArray2[0]);
9
10         numArray2[0] = 256;
11         System.out.println("numArray1:"+numArray1[0]);
12         System.out.println("numArray2:"+numArray2[0]);
13     }
14 }
```

```
sylviobarbon@ws-piccola:
numArray1:200
numArray2:200
numArray1:256
numArray2:256
```

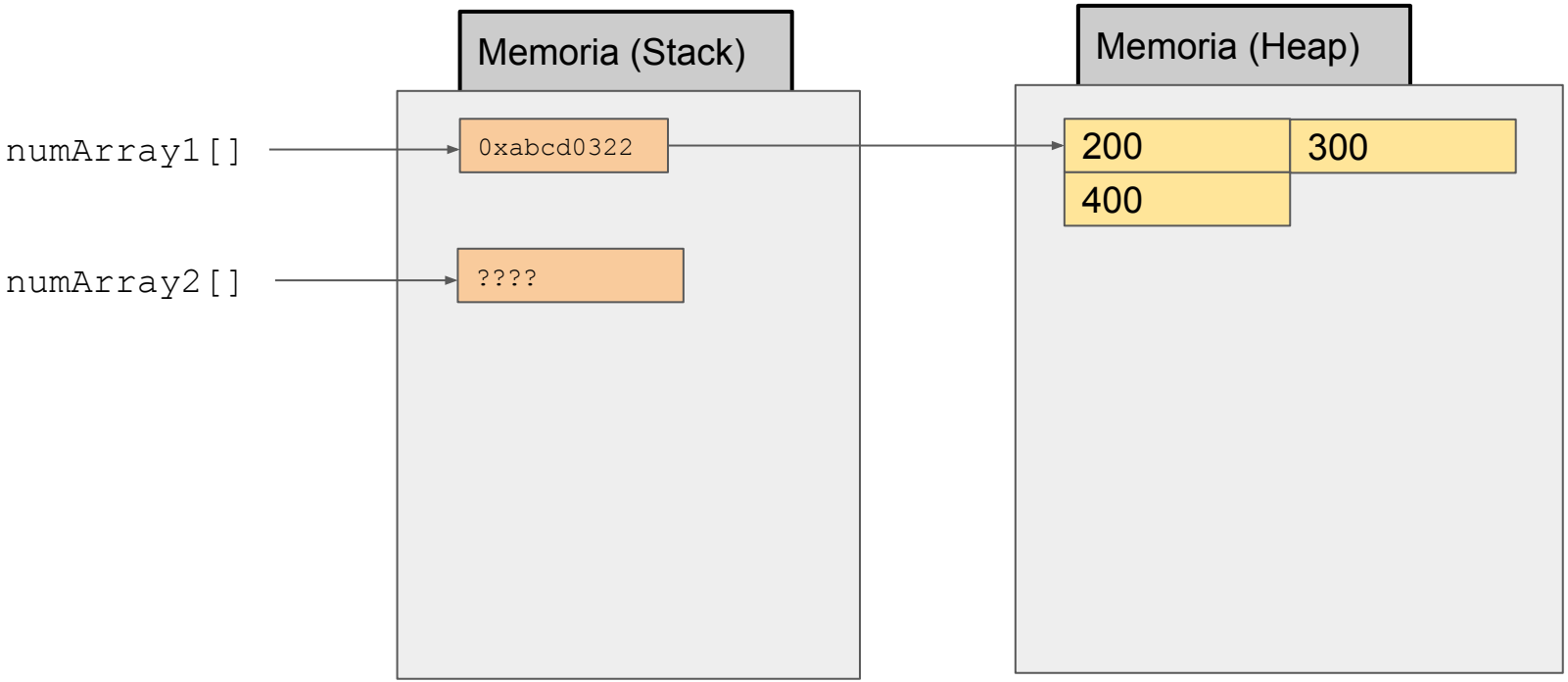

Sylvio Barbon Junior - sylvio.barbonjunior@units.it

2) Arrays



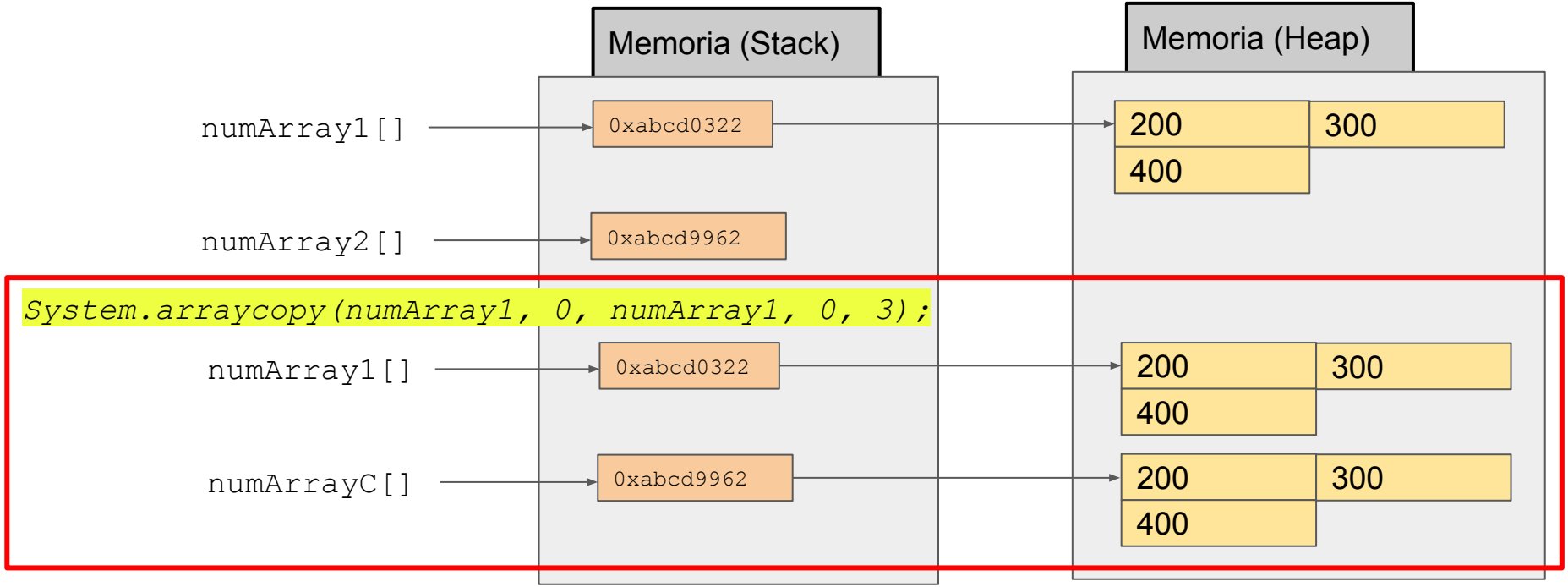
Sylvio Barbon Junior - sylvio.barbonjunior@units.it

2) Arrays



Sylvio Barbon Junior - sylvio.barbonjunior@units.it

2) Arrays



Sylvio Barbon Junior - sylvio.barbonjunior@units.it

2) Arrays

- Tipo array (multidimensionali):
  - Un oggetto array (**array primario**) può avere come elementi altri oggetti array (**array secondari**): in questo caso negli elementi dell'array primario sono memorizzati i riferimenti degli array secondari (e non direttamente gli array secondari).

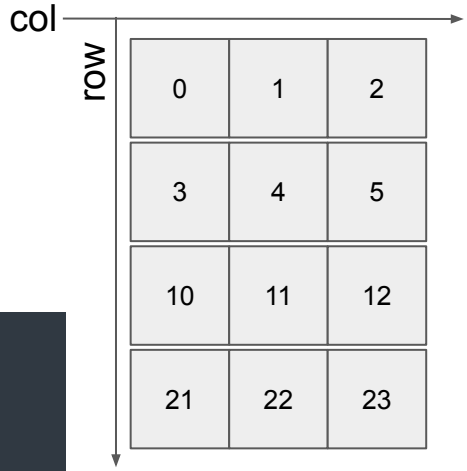
```
1  public class EsempioArray3{
2      public static void main(String[] args) {
3          int[][] numArray2D = {{1,2,3},
4                                {100,200,300},
5                                {1000,2000,3000},
6                               };
7
8          for (int i=0;i<3;i++) {
9              for(int j=0; j<3; j++){
10                 System.out.println(numArray2D[i][j]);
11             }
12         }
13
14     }
15 }
```

1	2	3
100	200	300
1000	2000	3000

2) Arrays

- Tipo array (multidimensionali):

[0][0]	[0][1]	[0][2]
[1][0]	[1][1]	[1][2]
[2][0]	[2][1]	[2][2]
[3][0]	[3][1]	[3][2]




```java
public class EsempioArray4{
    public static void main(String[] args) {
        int row = 4;
        int col = 3;
        int[][] numArray2D = new int[row][col];

        int aux = 1;
        for (int i=0;i<row;i++) {
            for(int j=0; j<col; j++){
                numArray2D[i][j] = j+(i*aux);
                System.out.println("["+i+"]["+j+"]:"+numArray2D[i][j]);
            }
            aux+=2;
        }

    }
}
```

```
sylviobarbon@ws-piccola:
[0][0]:0
[0][1]:1
[0][2]:2
[1][0]:3
[1][1]:4
[1][2]:5
[2][0]:10
[2][1]:11
[2][2]:12
[3][0]:21
[3][1]:22
[3][2]:23
```

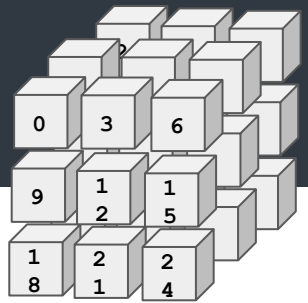
2) Arrays

- Tipo array (multidimensionali):

```java
public class EsempioArray5{
    public static void main(String[] args) {
        int size = 3;
        int[][][] cube = new int[size][size][size];

        int i = 0;
        for (int x=0;x<size;x++) {
            for(int y=0; y<size; y++){
                for(int z=0; z<size; z++){
                    cube[x][y][z] = i++;
                    System.out.println("cube["+x+"]["+y+"]["+z+"]="+cube[x][y][z]);
                }
            }
        }

    }
}
```




```
sylviobarbon@ws-piccola:~
cube[0][0][0]=0
cube[0][0][1]=1
cube[0][0][2]=2
cube[0][1][0]=3
cube[0][1][1]=4
cube[0][1][2]=5
cube[0][2][0]=6
cube[0][2][1]=7
cube[0][2][2]=8
cube[1][0][0]=9
cube[1][0][1]=10
cube[1][0][2]=11
cube[1][1][0]=12
cube[1][1][1]=13
cube[1][1][2]=14
cube[1][2][0]=15
cube[1][2][1]=16
cube[1][2][2]=17
cube[2][0][0]=18
cube[2][0][1]=19
cube[2][0][2]=20
cube[2][1][0]=21
cube[2][1][1]=22
cube[2][1][2]=23
cube[2][2][0]=24
cube[2][2][1]=25
cube[2][2][2]=26
```

2) Arrays

- Tipo array:
  - Per **ricercare** se un dato elemento è presente all'interno di un array, ed eventualmente **conoscerne la posizione**, occorre precisare se l'array è o meno ordinato.
  - Se l'array non è **ordinato**, non si può fare altro che **una ricerca completa.**

```java
public class EsempioArray6{
    public static void main(String[] args) {
        int size = 5;
        int[][] matrix = new int[size][size];

        matrix[1][3] = 300;
        for (int x=0;x<size;x++) {
            for(int y=0; y<size; y++){
                if(matrix[x][y]!=0)
                    System.out.println("Pos:["+x+"]["+y+"]");
            }
        }

    }
}
```

```
sylviobarbon@ws-piccola:
Pos:[1][3]
```

3) Classe String

- Una stringa è una sequenza di caratteri, che può anche non avere alcun carattere (stringa vuota);
- In Java esiste come classe il tipo String (package java.lang): una variabile appartenente a questo tipo, detta variabile stringa, rappresenta il riferimento di un oggetto stringa;
- Un **oggetto** stringa viene creato (**e allocato in memoria dinamica**) per mezzo dell'operatore **new** seguito da un costruttore.

```
1  public class EsempioString1{
2      public static void main(String[] args) {
3              String str1 = "Sylvio Barbon";
4              String str2 = new String();
5              str2 = "La mia stringa";
6              String str3 = new String("Altra stringa.");
7
8              System.out.println(str1+" \n"+str2+" \n"+str3);
9          }
10 }
```

```
sylviobarbon@ws-piccola
Sylvio Barbon
La mia stringa
Altra stringa.
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

3) Classe String

- Concatenazione di stringhe:

```java
public class EsempioString2{
    public static void main(String[] args) {
            String str1 = new String("ABC");
            String str2 = new String("DEF");
            String concatenazione = str1+str2;

            System.out.println(concatenazione);

            int i = 100;
            concatenazione += i;
            System.out.println(concatenazione);
        }
}
```

```
sylviobarbon@ws-piccola:
ABCDEF
ABCDEF100
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

3) Classe String

- Lunghezza di una stringa:
  - La lunghezza di una stringa è determinata dal valore restituito dalla funzione `length();`

```
1  public class EsempioString3{
2      public static void main(String[] args) {
3              String str1 = new String("Ciao a tutti!");
4
5              System.out.println("La lunghezza della stringa è:");
6              System.out.println(str1.length());
7          }
8  }
```

```
sylviobarbon@ws-piccola:~/Scaricati/Fi/CodiciSorgenti$ java EsempioString3
La lunghezza della stringa è:
13
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

3) Classe String

- Uguaglianza fra stringhe:
  - Per confrontare le stringhe memorizzate in due oggetti stringa si utilizza la funzione `equals();`

```
1  public class EsempioString4{
2      public static void main(String[] args) {
3              String str1 = new String("Italia");
4              String str2 = new String("Italia");
5              String str3 = new String("Brasile");
6
7              System.out.println(str1==str2);
8              System.out.println(str1.equals(str2));
9
10             System.out.println(str1==str3);
11             System.out.println(str1.equals(str3));
12         }
13 }
```

```
sylviobarbon@ws-piccola:
false
true
false
false
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

3) Classe String

- Selezione di un carattere
  - Per avere il carattere in posizione indice si usa la funzione `charAt();`

```
1  public class EsempioString5{
2      public static void main(String[] args) {
3              String str1 = new String("012345a78910");
4
5              System.out.println("Qual è il sesto carattere?\n"+str1.charAt(6));
6
7          }
8  }
```

```
sylviobarbon@ws-piccola:~/Scaricati/Fi/CodiciSorgenti$ java EsempioString5
Qual è il sesto carattere?
a
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

3) Classe String

- Confronto fra stringhe
  - Le stringhe memorizzate in due oggetti stringa possono essere confrontate fra loro (lessicograficamente) per mezzo della funzione `compareTo();`
  - Il risultato è negativo, zero o positivo a seconda del risultato del confronto.

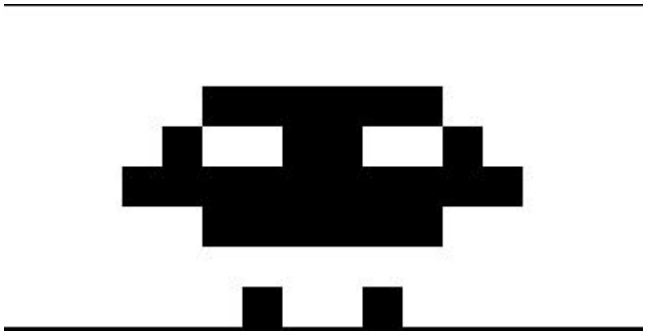
```
1  public class EsempioString6{
2      public static void main(String[] args) {
3              String strP = new String("abc");
4              String strM = new String("abcd");
5              String strL = new String("abcdef");
6
7              System.out.println(strP.compareTo(strM));
8              System.out.println(strP.compareTo(strL));
9              System.out.println(strL.compareTo(strM));
10             System.out.println(strL.compareTo(strP));
11             System.out.println(strP.compareTo("123"));
12         }
13 }
```

```
sylviobarbon@ws-piccola:
-1
-3
2
3
48
```

Esempio

```java
import java.io.File;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;

public class Image{
    public static void main(String[] args)  throws Exception {
      File file= new File("Alien.png");
      BufferedImage img = ImageIO.read(file);
      System.out.println("Image (height):"+img.getHeight());
      System.out.println("Image (width):"+img.getWidth());
      for (int y = 0; y < img.getHeight(); y++) {
        System.out.println("");
         for (int x = 0; x < img.getWidth(); x++) {
            int pixel = img.getRGB(x,y);
            if(pixel==-1)
                System.out.print("B");
            else
                System.out.print("N");
         }
      }
      System.out.println("");
    }
}
```



```
barbon@xps: ~/Downloads/FI_2023/java
barbon@xps:~/Downloads/FI_2023/java$ java Image
Image (height):8
Image (width):16

BBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBB
BBBBBNNNNNNBBBBB
BBBBNBBNNBBNBBBB
BBBNNNNNNNNNNBBB
BBBBBNNNNNNBBBBB
BBBBBBBBBBBBBBBB
BBBBBBNBBNBBBBBB
barbon@xps:~/Downloads/FI_2023/java$
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it