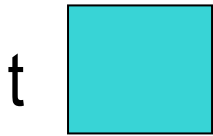
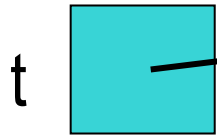


Assegnamento non è uguaglianza

String t;



t = "Cash";

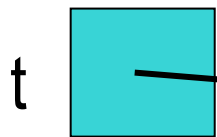


si riferisce a

Cash

oggetto String

t = "Credit";



si riferisce a

Cash

oggetto String

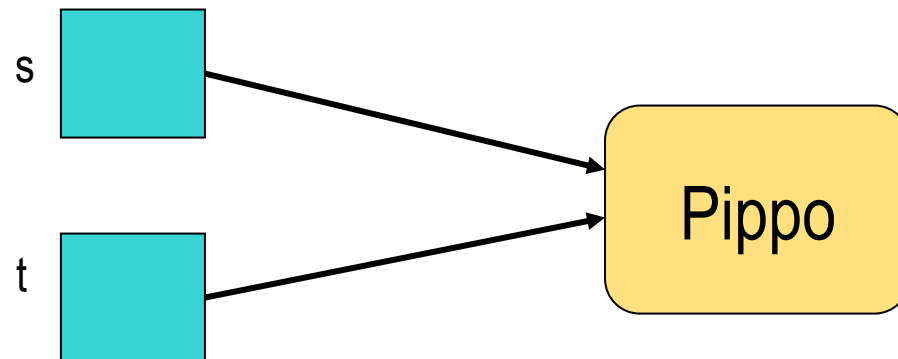
Credit

oggetto String

Variabili e oggetti

- Una variabile di riferimento si riferisce ad un solo oggetto alla volta
- Un oggetto può essere referenziato da più variabili simultaneamente

```
String s, t;  
s="Pippo"  
t=s;
```





Ruoli delle variabili

- Salvataggio o recupero a seconda della posizione

```
String s, t;  
s = "Pippo"; // s, variabile; "Pippo", valore  
t = s;       // t, variabile; s, valore
```

- Indipendenza:

```
s="Pluto"; //cambia il valore di s NON di t
```



Dichiarazione

(variazioni)

- Più variabili sulla stessa linea

```
String x,y,z;
```

- Con inizializzazione

```
String nome = "Marco", cognome  
= "Rossi";
```

- Possono essere distribuite nel codice ma devono precedere l'uso



Esempio (I)

```
public class Esempio {  
    public static void main(String arg[]) {  
        String      greeting;  
        String      bigGreeting;  
        greeting = "Hi, World";  
        bigGreeting = greeting.toUpperCase();  
        System.out.println(greeting);  
        System.out.println(bigGreeting);  
    }  
}
```



Esempio (II)

```
public class Esempio {  
    public static void main(String arg[]) {  
        String      greeting;  
        greeting = "Hi, World";  
        String      bigGreeting;  
        bigGreeting = greeting.toUpperCase();  
        System.out.println(greeting);  
        System.out.println(bigGreeting);  
    }  
}
```



Esempio (III)

```
public class Esempio {  
    public static void main(String arg[]) {  
        String  greeting = "Yo, World";  
        String  bigGreeting = greeting.toUpperCase();  
        System.out.println(greeting);  
        System.out.println(bigGreeting);  
    }  
}
```



Ancora sulle stringhe

- Metodi della classe String

<u>method</u>	<u>returns</u>	<u>arguments</u>
toUpperCase	ref. String object	none
toLowerCase	ref. String object	none
length	a number	none
trim	ref. String object	none
concat	ref. String object	ref. String object
substring	ref. String object	number
substring	ref. String object	two numbers

Posizioni nelle stringhe

- Le posizioni dei caratteri in una stringa sono numerate a partire da 0

H	a	m	b	u	r	g	e	r
0	1	2	3	4	5	6	7	8

✂ Stringhe e sottostringhe

```
String big = "hamburger";  
String small = big.substring(3,7);  
String medium = big.substring(3);  
String bigInCaps = big.toUpperCase();  
String order = big.concat(" with onions");
```



Esempio

```
import java.io.*;
class Esempio {
    public static void main(String arg[]) {
        String      first = "John";
        String      middle = "Fitzgerald";
        String      last = "Kennedy";
        String      initials;
        String      firstInit, middleInit, lastInit;
        firstInit = first.substring(0,1);
        middleInit = middle.substring(0,1);
        lastInit = last.substring(0,1);
        initials = firstInit.concat(middleInit);
        initials = initials.concat(lastInit);
        System.out.println(initials);
    }
}
```



Proprietà delle stringhe

- **Immutabilità**: una volta creato un oggetto String NON può cambiare
 - Es: l'invio di un messaggio `toUpperCase` comporta la creazione di un nuovo oggetto `String`
- Stringa vuota
 - Lunghezza 0
 - Nessun carattere
 - Reference: ""



Meccanismi

- Dato

```
String w, x, y, z, s;  
w = "ab";  
x = "cd";  
y = "ef";  
z = "gh";
```

- Assegnare ad **s** la concatenazione delle stringhe referenziate da **w**, **x**, **y**, **z**
 - "abcdefgh"

Cascata di messaggi

- `s=w.concat(x).concat(y).concat(z);`
- Il messaggio `concat(x)` è inviato a `w`
 - L'espressione `w.concat(x)` si riferisce alla stringa risultante
 - `w.concat(x)` → `abcd`
- Il messaggio `concat(y)` è inviato alla nuova stringa "abcd"
 - L'espressione `w.concat(x).concat(y)` si riferisce alla stringa risultante
 - `w.concat(x).concat(y)` → `abcdef`
- Il messaggio `concat(z)` è inviato alla nuova stringa "abcdef"
 - L'espressione `w.concat(x).concat(y).concat(z)` si riferisce alla stringa risultante
 - `w.concat(x).concat(y).concat(z)` → `abcdefgh64`



Cascata di messaggi

- `s=w.concat(x).concat(y).concat(z)`
- E' il processo di invio di un messaggio ad un oggetto per creare un nuovo oggetto, che a sua volta riceve un messaggio per creare un nuovo oggetto, che ...

Composizione di messaggi

- `s=w.concat(x.concat(y.concat(z)))`
- Il messaggio `concat(z)` è inviato a `y`
 - `y.concat(z)` si riferisce alla stringa risultante
 - `y.concat(z)` → efgh
- Un messaggio `concat` con tale nuovo oggetto come argomento è inviato a `x`
 - `x.concat(y.concat(z))` si riferisce alla stringa risultante
 - `x.concat(y.concat(z))` → cdefgh
- Un messaggio `concat` con tale nuovo oggetto come argomento è inviato a `w`
 - `w.concat(x.concat(y.concat(z)))` si riferisce alla stringa risultante
 - `w.concat(x.concat(y.concat(z)))` → abcdefgh



Altri metodi della classe String

- `String s = "parola"`
- `length()`: ritorna la lunghezza della stringa

```
int len = s.length(); // len == 6
```
- `charAt(int i)`: ritorna il carattere in posizione i-esima

```
char c=s.charAt(0); // c == 'p'
```
- `indexOf(char c)`: ritorna l'indice della prima occorrenza del carattere indicato

```
int i=s.indexOf('o'); // i == 3
```
- `lastIndexOf(char c)`: come sopra ma per l'ultima occorrenza di c



Esercizi

- Scrivere un programma che stampa il carattere centrale di una stringa.
- Scrivere un programma che data una stringa di almeno 2 caratteri, ne costruisce un'altra dove il primo carattere è scambiato con l'ultimo, che viene poi stampata a video.



Esercizi (2)

- Contare il numero di caratteri tra la prima occorrenza e l'ultima occorrenza del carattere i nella stringa arrivederci
- Invertire la prima parte con la seconda parte della sequenza di caratteri arrivederci