



Università
degli Studi di Palermo

Corso di Laurea
Ingegneria Gestionale

Lezione 13

Numeri e Caratteri

Ing. Massimo Cossentino

Sistemi Informativi Aziendali

a.a. 2008/2009

Operazioni aritmetiche

- $/$ è l'operatore per la divisione
- Se entrambi gli argomenti dell'operatore $/$ sono di tipo intero, il risultato è un numero intero e il resto viene ignorato.
 - $7.0 / 4$ restituisce 1.75
 - $7 / 4$ restituisce 1
- L'operatore $\%$ calcola il resto di una divisione.
 - $7 \% 4$ è 3



Operazioni aritmetiche

```
final int PENNIES_PER_NICKEL = 5; //1 Penny = 1/100 di dollaro
final int PENNIES_PER_DIME = 10;
final int PENNIES_PER_QUARTER = 25;
final int PENNIES_PER_DOLLAR = 100;

// Calcola il valore totale in centesimi
int total = dollars * PENNIES_PER_DOLLAR +
           quarters * PENNIES_PER_QUARTER +
           nickels * PENNIES_PER_NICKEL +
           dimes * PENNIES_PER_DIME + pennies;

// Usa la divisione intera per convertire in dollari e centesimi
int dollars = total / PENNIES_PER_DOLLAR;
int cents = total % PENNIES_PER_DOLLAR;
```

3

La classe Math

- La classe `Math` contiene i metodi `sqrt` e `pow` per calcolare radici quadrate e potenze.
- Per calcolare x^n , si usa il metodo `Math.pow(x, n)`
 - tuttavia, per calcolare x^2 è molto più efficiente scrivere semplicemente $x * x$
- Per estrarre la radice quadrata di un numero, si usa il metodo `Math.sqrt` scrivendo `Math.sqrt(x)`
- In Java, $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$ diventa

```
(-b + Math.sqrt(b * b - 4 * a * c)) / (2 * a)
```

Metodi matematici in Java

<code>Math.sqrt(x)</code>	radice quadrata di x (≥ 0)
<code>Math.pow(x, y)</code>	x^y ($x > 0$, o $x = 0$ e $y > 0$, o $x < 0$ e y è un intero)
<code>Math.sin(x)</code>	seno di x (x in radianti)
<code>Math.cos(x)</code>	coseno di x
<code>Math.tan(x)</code>	tangente di x
<code>Math.asin(x)</code>	arcoseno ($\sin^{-1} x \in [-\pi/2, \pi/2]$, $x \in [-1, 1]$)
<code>Math.acos(x)</code>	arcocoseno ($\cos^{-1} x \in [0, \pi]$, $x \in [-1, 1]$)
<code>Math.atan(x)</code>	arcotangente ($\tan^{-1} x \in (-\pi/2, \pi/2)$)
<code>Math.atan2(x, y)</code>	arcotangente ($\tan^{-1} (y/x) \in [-\pi, \pi]$, x può essere 0)
<code>Math.toRadians(x)</code>	converte x da gradi a radianti (cioè restituisce $x \cdot \pi/180$)
<code>Math.toDegrees(x)</code>	converte x da radianti a gradi (cioè restituisce $x \cdot 180/\pi$)
<code>Math.exp(x)</code>	e^x
<code>Math.log(x)</code>	logaritmo naturale ($\log(x)$, $x > 0$)
<code>Math.round(x)</code>	il numero intero (di tipo <code>long</code>) più prossimo a x
<code>Math.ceil(x)</code>	il più piccolo numero intero (di tipo <code>double</code>) che sia $\geq x$
<code>Math.floor(x)</code>	il più grande numero intero (di tipo <code>double</code>) che sia $\leq x$
<code>Math.abs(x)</code>	valore assoluto, $ x $
<code>Math.max(x, y)</code>	il numero maggiore tra x e y
<code>Math.min(x, y)</code>	il numero minore tra x e y

Priorità di esecuzione delle operazioni

$$\begin{array}{c} (-b + \text{Math.sqrt}(b * b - 4 * a * c)) / (2 * a) \\ \underbrace{\qquad\qquad\qquad} \quad \underbrace{\qquad\qquad\qquad} \quad \underbrace{\qquad\qquad\qquad} \\ \qquad\qquad b^2 \qquad\qquad 4ac \qquad\qquad 2a \\ \underbrace{\qquad\qquad\qquad} \\ \qquad\qquad b^2 - 4ac \\ \underbrace{\qquad\qquad\qquad} \\ \qquad\qquad \sqrt{b^2 - 4ac} \\ \underbrace{\qquad\qquad\qquad} \\ \qquad\qquad -b + \sqrt{b^2 - 4ac} \\ \underbrace{\qquad\qquad\qquad} \\ \qquad\qquad \frac{-b + \sqrt{b^2 - 4ac}}{2a} \end{array}$$



Università
degli Studi di Palermo

Metodi Statici nella classe Math

Invocare metodi statici

- Un metodo `static` non agisce su un oggetto. Quindi NON si può scrivere

```
double x = 4;  
double root = x.sqrt(); // Errore
```

- I metodi statici sono sempre definiti all'interno di classi
- Per convenzione: i nomi delle classi iniziano con una lettera maiuscola, quelli di oggetti e metodi con una lettera minuscola (le invocazioni di metodi sono seguite da parentesi)

```
Math  
System.out
```


Invocazione di un metodo statico

NomeClasse.nomeMetodo(parametri)

Esempio:

`Math.sqrt(4)`

Obiettivo:

Invocare un metodo statico (un metodo che non agisce su un oggetto) e fornire i suoi parametri.



Università
degli Studi di Palermo

Stringhe

Stringhe

- Una stringa è una sequenza di caratteri

```
"Hello, World!"
```

- Le stringhe sono esemplari della classe `String`
- Per calcolare la lunghezza di una stringa:

```
int n = message.length();
```

- Stringa vuota:

```
" "
```

Concatenazione

- Usando l'operatore + si possono concatenare più stringhe:

```
String name = "Dave";  
String message = "Hello, " + name;  
// message è "Hello, Dave"
```

- Ogni volta che uno degli argomenti dell'operatore + è una stringa, l'altro argomento viene convertito in una stringa.

```
String a = "Agent";  
int n = 7;  
String bond = a + n; // bond è "Agent7"
```

Concatenazione negli enunciati print

- La concatenazione è molto utile per ridurre il numero degli enunciati

`System.out.print.`

- Per esempio,

```
System.out.print("The total is ");  
System.out.println(total);
```

può essere combinato nella singola invocazione:

```
System.out.println("The total is " + total);
```

Conversione tra stringhe e numeri

- Se una stringa contiene le cifre di un numero, viene usato il metodo `Integer.parseInt` o `Double.parseDouble` per ottenere il valore numerico.

substring

- Per estrarre una parte di una stringa, viene usato il metodo `substring`.

- La

```
String greeting = "Hello, World!";
```



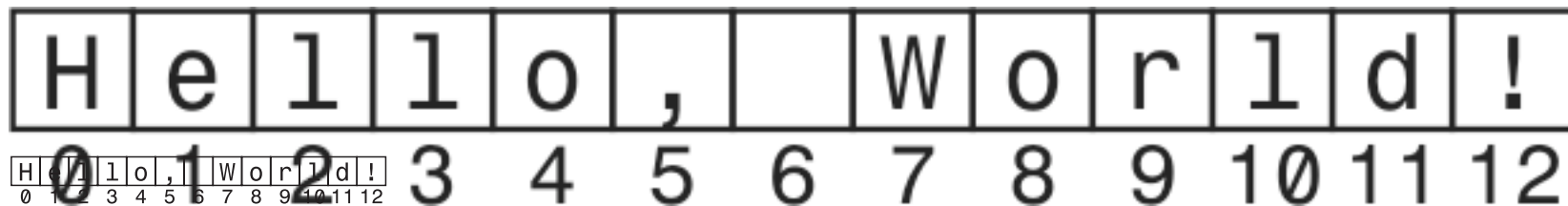
```
String sub = greeting.substring(0, 5); // sub è "Hello"
```

 da 1 e così via.

- Le posizioni dei caratteri in una stringa si contano a partire da zero.

- Sintassi:

`mystring.substring(posprimocarattere, num_caratteri)`



Continua



- All'oggetto `System.in` è associato un insieme di funzionalità minimali: può leggere solo un byte alla volta
- In Java 5.0, è stata aggiunta una classe `Scanner` che consente la lettura semplice e comoda dei dati inseriti in ingresso da tastiera.

```
Scanner in = new Scanner(System.in);  
System.out.print("Enter quantity: ");  
int quantity = in.nextInt();
```

- Quando viene invocato il metodo `nextInt` o `nextDouble`, il programma si arresta in attesa che l'utente digiti un numero e prema il tasto "Enter" ("Invio"):
- Il metodo `nextLine` restituisce la successiva riga di testo fornita in ingresso (fino alla pressione del tasto "Enter"), sotto forma di oggetto di tipo `String`.
- Il metodo `next` restituisce, invece, la parola successiva, una sequenza di caratteri terminata da un carattere di spaziatura ("white space")



```
import java.util.Scanner;

public class CashRegisterSimulator
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        CashRegister register = new CashRegister();
        System.out.print("Enter price: ");
        double price = in.nextDouble();
        register.recordPurchase(price);
        System.out.print("Enter dollars: ");
        int dollars = in.nextInt();
        System.out.print("Enter quarters: ");
        int quarters = in.nextInt();
        System.out.print("Enter dimes: ");
        int dimes = in.nextInt();
        System.out.print("Enter nickels: ");
        int nickels = in.nextInt();
    }
}
```

Continua



File CashRegisterSimulator.java

```
System.out.print("Enter pennies: ");
    int pennies = in.nextInt();
    register.enterPayment(dollars, quarters,
        dimes, nickels, pennies);

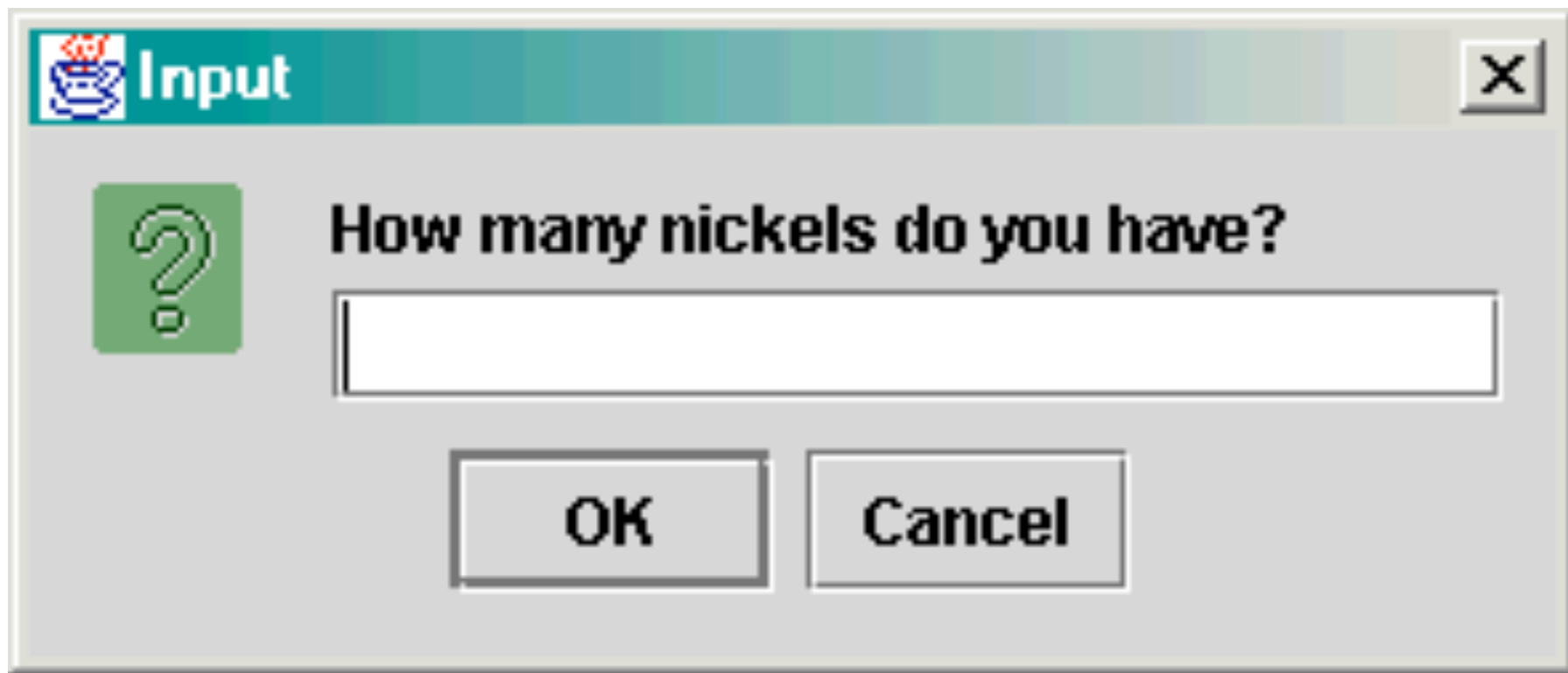
    System.out.print("Your change: ");
    System.out.println(register.giveChange());
}
}
```

File CashRegisterSimulator.java

Visualizza

```
Enter price: 7.55  
Enter dollars: 10  
Enter quarters: 2  
Enter dimes: 1  
Enter nickels: 0  
Enter pennies: 0  
Your change is 3.05
```

Lettura di dati da una finestra di dialogo





Lettura di dati da una finestra di dialogo

```
String input = JOptionPane.showInputDialog  
("How many nickels do you have?");
```

- La finestra di dialogo viene mostrata dalla classe JOptionPane che appartiene alle librerie Java
- Bisogna esplicitamente dire al compilatore che si vuole usare ogni determinata parte delle librerie
- Si usa il comando import

```
import javax.swing.JOptionPane;
```

Conversione da Stringhe a Interi

- La finestra JOptionPane vista prima riporta in ingresso una stringa.
- Allora è necessario convertire la stringa in un numero:

```
int nickels = Integer.parseInt(input);
```

- Va aggiunto `System.exit(0)` alla fine del metodo `main` ogniqualvolta venga invocato `showInputDialog` o `showMessageDialog`



```
import java.util.Scanner;
import javax.swing.JOptionPane;
public class CashRegisterSimulator
{
    public static void main(String[] args)
    {
        //Scanner in = new Scanner(System.in);
        String input = JOptionPane.showInputDialog ("Enter price: ");
        CashRegister register = new CashRegister();
        //System.out.print("Enter price: ");
        //double price = in.nextDouble();
        double price = Double.parseDouble(input);
        register.recordPurchase(price);
        System.out.print("Enter dollars: ");
        Scanner in = new Scanner(System.in);
        int dollars = in.nextInt();

        ...

        System.out.print("Your change: ");
        System.out.println(register.giveChange());
        System.exit(0);
    }
}
```