Programmazione in Java
e
gestione della grafica

Lezione 7

La scorsa lezione…..

- Metodi della classe **`Math`**
- Definizione e chiamata di metodi **`static`**
- Esempi

Esercizio della lezione scorsa

- Esercizio 6.3

  Scrivere un programma che contiene al suo interno come metodi i codici (gia' scritti) rispettivamente per calcolare il max , il min, la somma e il prodotto di 10 elementi inseriti dall'utente.

  Il programma propone all'inizio un menu con 4 scelte e chiede all'utente quale delle quattro funzioni vuole eseguire. In base alla scelta inserita poi esegue uno dei quattro calcoli.

Metodi che restituiscono valori

- Ricordiamo la sintassi per la definizione di un metodo:

`public static` <*tipo-restituito*> NomeMetodo**(** <*lista parametri*>**)**
{
    <*istruzioni*>
}

uso istruzione: **`return`** <*variabile*>

`return` <*valore*>

Esercizio della lezione passata

- Esercizio 6.2

  Scrivere un programma che prende in input un valore a compreso tra 2 e 20 (2<= a <=20) e stampa sullo schermo un rettangolo di base 2a+1 e altezza a tutto fatto di simboli R contenente un triangolo di simboli S come in .figura (per a=3).:

  RRRSRRR

  RRSSSRR

  RSSSSSR

Sul compilatore :

`Rettangolo1.java`

Il metodo `Math.random()`

Chiamato anche **generatore di numeri pseudo casuali**, restituisce un numero x di tipo double tale che 0.0 <= x < 1.

Nelle applicazioni in genere servono numeri casuali interi in un intervallo definito.

Esempi:

simulare lancio dado =numero tra 1 e 6,

estrarre biglietto lotteria = numero da 33 a 3333

Normalizzazione

Per avere un numero intero b tra min e max :

Sia: **`a = Math.random()`**

`b = Math.floor(a *(max-min+1)) + min`

`Math.floor` = calcola l'intero inferiore

(max –min +1) = numero di elementi su cui estrarre

Normalizzazione (bis)

Per avere un numero intero b tra num interi a partire da min :

Sia: **`a = Math.random()`**

`b = Math.floor(a * num) + min`

`Math.floor` = calcola l'intero inferiore

Estrazione dei numeri in una lotteria

ALGORITMO

- Si richiede il numero del primo biglietto venduto
- Si richiede quanti biglietti sono stati venduti
- Si estrae il biglietto vincitore
- Si stampa il numero del biglietto vincitore

VARIANTE

- Si estraggono tre biglietti vincitori

Sul compilatore :

`SelectWinner.java`

Gioco “Indovina il numero”: computer vs utente

ALGORITMO

- Il computer “sceglie” un numero x tra 1 e 100 e chiede all’utente di indovinarlo
- L’utente propone un numero y
- Finchè y diverso da x, il computer:
  - risponde che y non è esatto dicendo se è troppo grande oppure troppo piccolo
  - Richiede un nuovo valore t
  - L’utente inserisce un nuovo valore
- Se y=x il computer si congratula con l’utente riportando il numero di tentativi fatti per indovinare il numero

VARIANTE

- Se l’utente non indovina entro 6 tentativi, ha perso (PERCHE’?)

Sul compilatore :

`IndovinaNumero.java`

Gioco “Indovina il numero”: computer vs utente

ALGORITMO (variante)

- Il computer “sceglie” un numero x tra 1 e 100 e chiede all’utente di indovinarlo
- L’utente propone un numero y
- Finchè y diverso da x, il computer:
  - risponde che y non è esatto dicendo se è troppo grande oppure troppo piccolo
  - Richiede un nuovo valore t
  - L’utente inserisce un nuovo valore
- Se entro **6** tentativi viene inserito y corretto (y=x)
  - il computer si congratula con l’utente riportando il numero di tentativi fatti per indovinare il numero

  altrimenti
  - Il computer comunica all’utente che ha perso e termina il gioco

Nuovo esercizio:

- Esercizio 7.1 (variante di IndovinaNumero)

  Modificare il programma IndovinaNumero come descritto nell'algoritmo con variante e cioè in modo che l'utente vince solo se indovina entro 6 tentativi.

- DOMANDA: perché proprio 6 tentativi?
- RISPOSTA: Esiste una semplice strategia di gioco che permette di indovinare sicuramente entro 7 tentativi.
- DOMANDA 1: Qual'e' la strategia?
- DOMANDA2: Come dimostriamo che 7 tentativi sono *sufficienti*?
- DOMANDA 3: Con tale strategia, quanti tentativi mi servono per indovinare un numero tra 1 e 1000?

Gioco “Indovina il numero”: utente vs computer

ALGORITMO

- Il computer “chiede” all’utente di pensare un numero x tra 1 e 100 e chiede di inserire 0 per iniziare il gioco
- Il computer “propone” un numero y e chiede all’utente di valutarlo (0= ok, 1= troppo piccolo, 2= troppo grande)
- Finchè l’utente non scrive 0, il computer
  - “propone” un nuovo numero
  - Chiede nuovamente all’utente di valutarlo
  - (mantiene memoria del numero dei tentativi)
- Se *entro* **6** tentativi viene inserito y corretto (y=x)
  - il computer “si vanta” con l’utente della sua vittoria riportando il numero di tentativi fatti per indovinare il numero

  altrimenti
  - Il computer comunica all’utente che di essere stato battuto e termina il gioco

NOTA: L’utente non deve barare…. Il computer potrebbe accorgersene!

NOTA 1: Il computer si accorge se l’utente sta imbrogliando solo se il programma è scritto adeguatamente

Gioco “Indovina il numero”: utente vs computer

ALGORITMO

- Il computer “chiede” all’utente di pensare un numero x tra 1 e 100 e chiede di inserire 0 per iniziare il gioco
- Il computer “propone” un n… …tarlo (0= ok, 1= troppo piccolo, 2=
- Finchè l’utente non scrive 0,
  - **“propone” un nuovo numero**
  - Chiede nuovamente all’utente di valutarlo
  - (mantiene memoria del numero dei tentativi)
- Se *entro* **6** tentativi viene inserito y corretto (y=x)
  - il computer “si vanta” con l’utente della sua vittoria riportando il numero di tentativi fatti per indovinare il numero

  altrimenti
  - Il computer comunica all’utente che di essere stato battuto e termina il gioco

Occore implementare una strategia efficiente di gioco!

Esercizi

- Esercizio 7.0

  Descrivere nei particolari un algoritmo per implementare una strategia efficace per giocare a IndovinaNumero

Esercizi

- Esercizio 7.1 (variante di IndovinaNumero)

  Modificare il programma IndovinaNumero come descritto nell'algoritmo con variante e cioè in modo che l'utente vince solo se indovina entro 6 tentativi.

Esercizi

- Esercizio 7.2

  Un numero n>1 si dice *primo* se non ammette altri divisori oltre 1 e se stesso.

  Scrivere un metodo **`Primo`** che prende come parametro un numero intero positivo e restituisce un valore booleano true se tale numero e' primo e false altrimenti.

Esercizi

- Esercizio 7.3

  Scrivere un programma (che utilizza il metodo **`Primo`**) che prende in input dall'utente un intero positivo e dice se tale numero e' o no primo.

  (L'utente potrebbe essere invitato ad inserire sempre nuovi numeri finche' non inserisce lo 0)

Esercizi

- Esercizio 7.3

  Scrivere un programma (che utilizza il metodo **`Primo`**) che scrive sullo schermo tutti i numeri primi <=1000.

Esercizi

- Esercizio 7.4

  Scrivere un programma (che utilizza il metodo **`Primo`**) che prende in input un valore intero positivo X e scrive sullo quanti numeri primi ci sono fra 2 e X .