Esercizi di Fondamenti d'Informatica I

Prof. Giuseppe Manco

A cura di
Ing. Antonella Guzzo

Esercizi

Esercizio 1

Scrivere il programma **`Media`** che inizialmente chiede all'utente un numero intero `num` maggiore di zero. Se il numero fornito dall'utente non è maggiore di zero, il programma richiede il numero, finché necessario. Quindi il programma chiede all'utente `num` numeri decimali, e alla fine ne stampa il minimo, il massimo e la media, con una opportuna intestazione.

```
import corejava.*;
class Media {
public static void main (String args[])
{	int num;
	do{
	num=Console.readInt("Inserisci un numero intero positivo");
	if(num<=0)
		System.out.println("Errore!il numero non e' positivo");
	} while(num<=0);
	int count=0;
	double somma, minimo, massimo, numero;
	somma=0;
	minimo=0;
	massimo=0;
	while(count<num){
		count++;
		numero=Console.readDouble(" inserisci il numero decimale N"+(count));
		somma+=numero;
		if(count==1){
			minimo=numero;
			massimo=numero;
		}
		else{
			if(numero<minimo)
				minimo=numero;
			if(numero>massimo)
				massimo=numero;
		}

	}
	System.out.println("la media dei valori inseriti e': "+somma/num);
	System.out.println("il minimo dei valori inseriti e': "+minimo);
	System.out.println("il massimo dei valori inseriti e': "+massimo);
	}
}
```

Soluzione

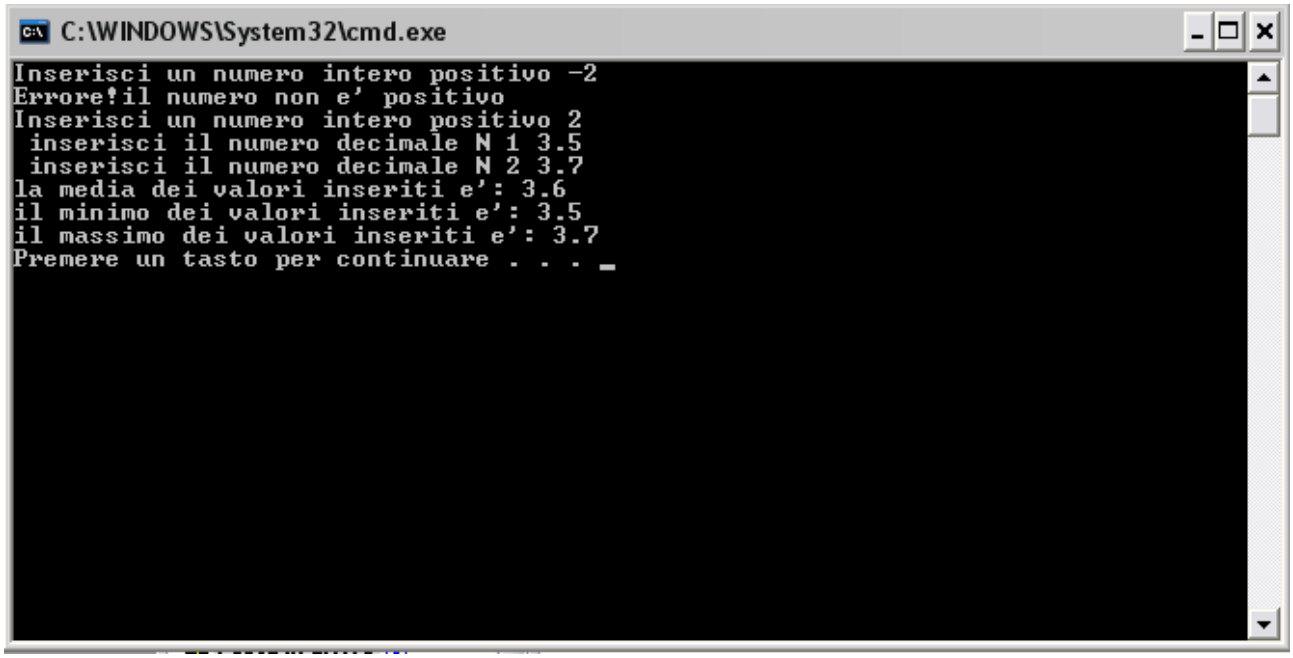
La prima riga importa il package corejava in cui e' contenuta la classe *Console*.java . Tale classe verrà utilizzata per leggere da input il numero intero inserito dall'utente e assegnarlo alla variabile intera **num**. Poiche' il programma richiede che il numero sia intero e positivo (>=0) viene utilizzata l'istruzione **do{…} while(num<=0)** per leggere l'intero, se il numero inserito dall'utente non e' positivo il programma stampa

un messaggio di errore (**if** all'interno del **do..while**) e ripropone all'utente l'inserimento del numero finche' necessario.
A questo punto si procede alla lettura di num numeri decimali. Il ciclo di **while** e' definito tramite una variabile contatore **count** che serve a tenere traccia del numero di letture fatte (se **count** =3 ho letto 3 valori). Ogni ciclo di while esegue le seguenti operazioni:

1. aggiorna il contatore **count**;
2. legge un nuovo numero decimale e lo memorizza nella variabile **numero;**
3. somma **numero** ai valori precedentemente letti (NOTA: se il numero e' il primo inserito, la somma dei precedenti è zero);
4. verifica se il numero e' il primo ad essere letto esso sarà massimo e minimo, altrimenti lo si puo' confrontare con il massimo e il minimo attualmente memorizzati.

Alla fine il programma stamperà la media, la somma e il minimo.

Di seguito viene mostrato il risultato dell'esecuzione dal prompt di DOS



Esercizio2

Scrivere il programma **`Scomposizione`** che inizialmente chiede all'utente un numero intero positivo e quindi stampa su video la scomposizione in fattori di quel numero. Ad esempio, se il numero letto è 150, il programma stamperà in sequenza i valori

2

3

5

5

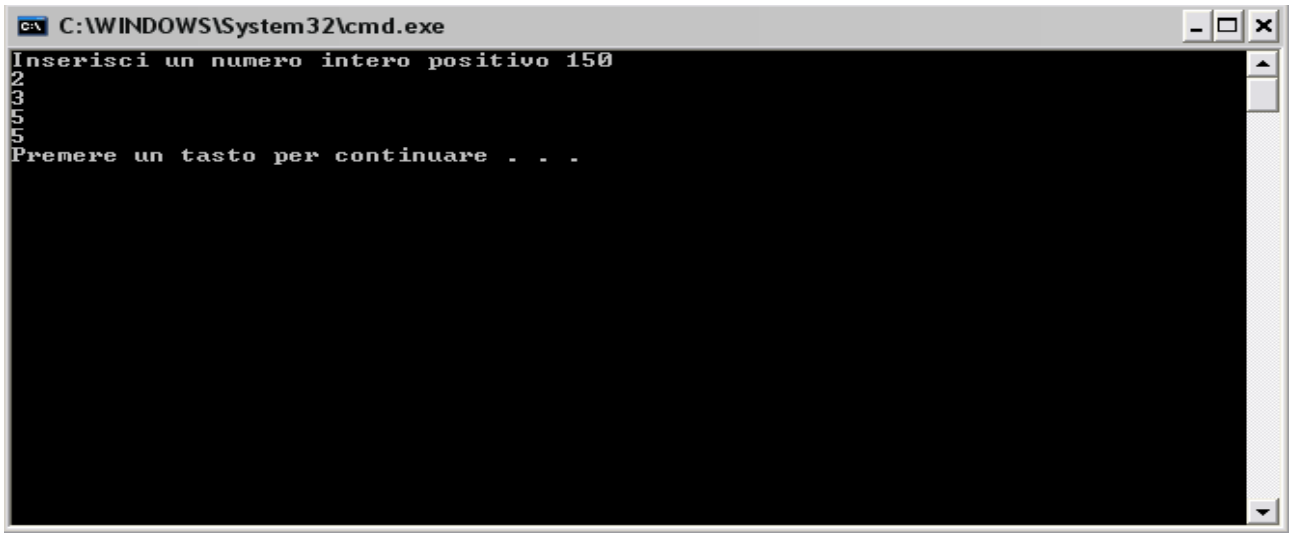
Infatti, 150 = 2×3×5×5

Soluzione

La prima riga importa la classe *Console*.java contenuta nel package corejava . Tale classe verrà utilizzata per leggere da input il numero intero inserito dall'utente e assegnarlo alla variabile intera **num**. Poiche' il programma richiede che il numero sia intero e positivo (>=0) viene utilizzata l'istruzione **do{…} while(num<=0)** per leggere l'intero, se il numero inserito dall'utente non e' positivo il programma stampa un messaggio di errore (**if** all'interno del **do..while**) e ripropone all'utente l'inserimento del numero finche' necessario.
Il ciclo while calcola i divisori del numero a partire da due.

```
import corejava.Console;
class Scomposizione {
public static void main (String args[])
{       int num;
        do{
        num=Console.readInt("Inserisci un numero intero positivo");
        if(num<=0)
                System.out.println("Errore!il numero non e' positivo");
        }  while(num<=0);
        int quoziente=num, divisore;
        divisore=2;
        while(quoziente>=divisore){
                        if(quoziente%divisore==0){
                                System.out.println(divisore);
                                quoziente=quoziente/divisore;
                }
                else divisore++;
        }
  }
}
```

Di seguito viene mostrato il risultato dell'esecuzione dal prompt di DOS



Esercizio3

Scrivere il programma **`NumeriPrimi`** che inizialmente chiede all'utente un numero intero e quindi stampa tutti i numeri primi fino a quel numero stesso. Ad esempio, se l'utente immette 20, la sequenza stampata sarà

1

2

3

5

7

9

11

13

17

19

Soluzione

La prima riga importa la classe *Console*.java contenuta nel package corejava . Tale classe verrà utilizzata per leggere da input il numero intero inserito dall'utente e assegnarlo alla variabile intera **numero**.
Poiche' il programma richiede che il numero sia intero e positivo (>=0) viene utilizzata all'interno del **main** l'istruzione **do{…} while(numero<=0)** per leggere l'intero, se il numero inserito dall'utente non e' positivo il programma stampa un messaggio di errore (**if** all'interno del **do..while**) e ripropone all'utente l'inserimento del numero finche' necessario. L'istruzione di for scandisce tutti i numeri interi (variabile **i**) fino a **numero** e se verifica che è primo lo stampa (**System.out.println(i);**). Per fare la verifica utilizza il metodo **boolean ePrimo(int n)** che riceve un intero n come parametro e restituisce **true** o **false** a seconda se l'intero n e' primo o no.
Il metodo ePrimo restituisce true se n=2, restituisce false se n e' pari (se il numero è pari e diverso da due sicuramente non è primo), se è dispari va a cercare nel ciclo while se esiste un divisore di n.
In realtà non vengono provati tutti i divisori, ma è sufficiente cercare nei dispari fino ad un massimo della radice quadrata di n (il limite della ricerca è l'intero più vicino alla radice quardata del numero **(int)Math.round(Math.sqrt(n))**.
Uscire dal while significa due possibilità:

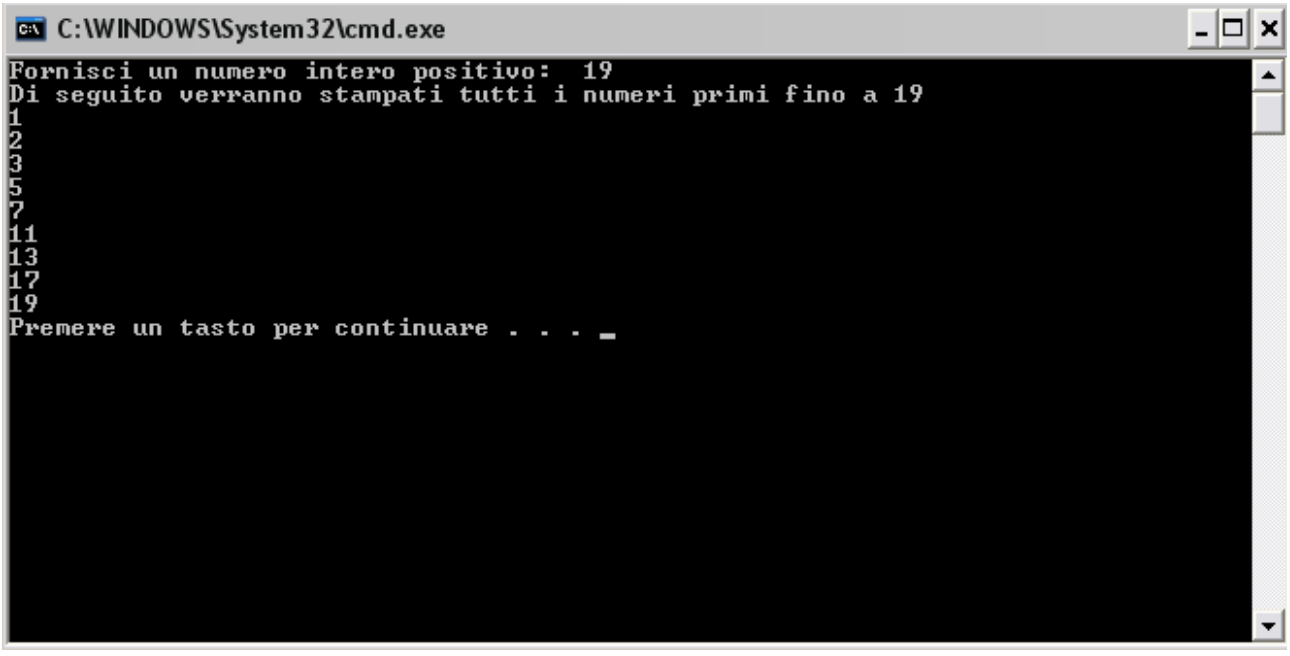
1. ho superato il limite, quindi non ho trovato divisori ed il numero è primo (istruzione **if(divisore>limite) return true;**)
2. ho trovato un divisore **n%divisore == 0** il numero non è primo.

```
import corejava.*;
class NumeriPrimi{

   static boolean ePrimo( int n ){
      if( n==2 ) return true;
      if( n%2==0 ) return false; //numero pari
      /*numero certamente dispari*/
      int limite=(int)Math.round(Math.sqrt(n));
      int divisore=3;
      while( divisore<=limite && n%divisore != 0 ) divisore+=2;
      if( divisore>limite ) return true;
      return false;
   }

   public static void main(String []args){
       int numero;
       do{
                  numero=Console.readInt("Fornisci un numero intero positivo: ");
                  if( numero <= 0 )
                     System.out.println("Attenzione! Il numero deve essere"+
                       "strettamente positivo!");
       } while(numero<=0);
       System.out.println("Di seguito verranno stampati tutti i numeri primi fino a "+numero);
       for(int i=0; i<=numero; i++)
             if( ePrimo(i) )
                       System.out.println(i);
         }
```

Di seguito viene mostrato il risultato dell'esecuzione dal prompt di DOS



Esercizio4

Un metodo per calcolare l'approssimazione della radice quadrata di un numero x (con un'approssimazione $\in$) è il seguente:

- Inizialmente, si identifica un'approssimazione iniziale (intera) y tale che $y^2 \leq x < (y+1)^2$
- Se y è un'approssimazione, la media tra y e x/y è un'approssimazione migliore.

Scrivere un programma che legge in input due valori razionali **`x`** e **`epsilon`** minore di 1 e calcola e stampa il valore **`y`** tale che

$$|y^2 - x| < epsilon$$

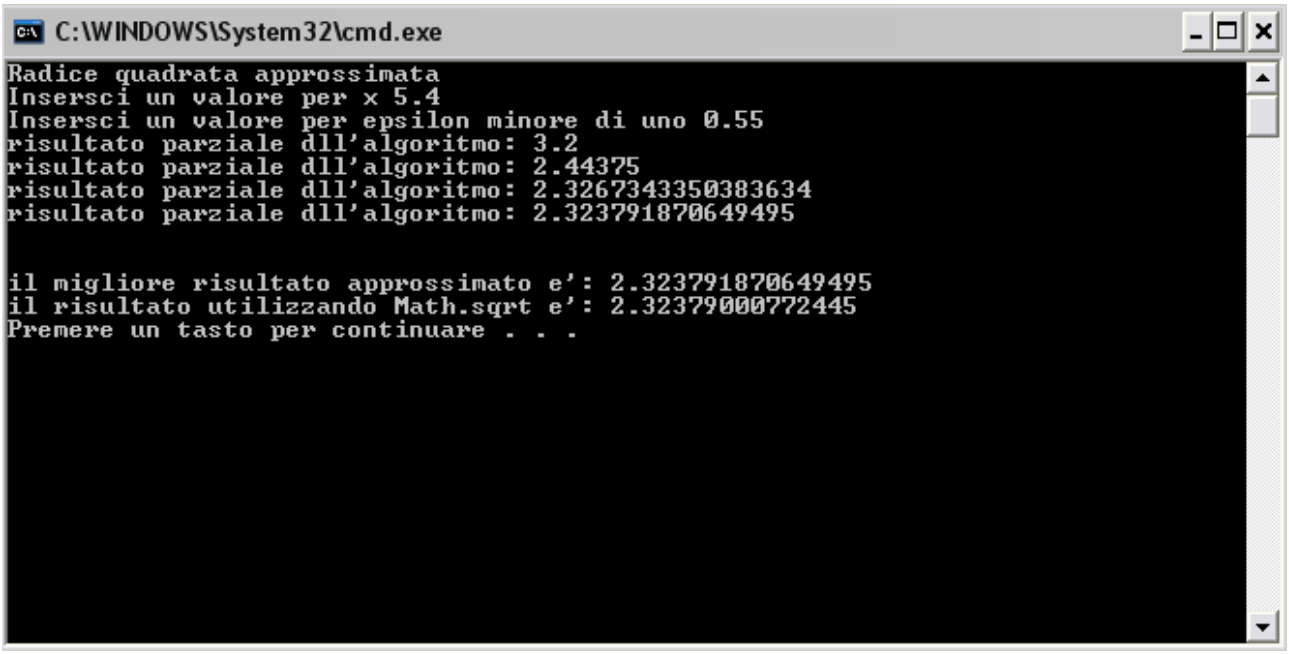
sfruttando le idee sopra descritte.

Soluzione

La prima riga importa la classe *Console*.java contenuta nel package corejava . Tale classe verrà utilizzata per leggere da input il numero reale inserito dall'utente e assegnarlo alla variabile double **x**. Il programma utilizza eps come valore di tolleranza nel ciclo do while (infatti la ricerca continua finchè l'approssimazione è >= eps) . Continuando il ciclo finchè la condizione non è falsa garantiamo che la risposta abbia precisione epsilon. La variabile y (contenete il valore della radice approssimato) viene inizializzata a 1 ed ad ogni iterazione del do..while il valore corrente viene sostituito dalla media tra y e x/y. Questo oscillare intorno alla media costituisce la chiave dell'algoritmo. Il programma stampa sia i risultati parziali (cioè tutti quelli generati all'interno del ciclo), il risultato finale (approssimato secondo ipsilon) e per finire il risultato ottenuto utilizzando la classe Math.

```
import corejava.*;
public class SqrtApprox{
  public static void main(String []args){
    System.out.println("Radice quadrata approssimata");
    double x=Console.readDouble("Inserisci un valore per x");
    double eps=Console.readDouble("Insersci un valore per epsilon minore di uno");
    double y=1.0, oldy;
    do{
                System.out.print("risultato parziale dll'algoritmo: ");
      oldy=y;
      y=(y+x/y)/2;
      System.out.println(y);
    } while( Math.abs(oldy*oldy-x )>=eps );
    System.out.println();
    System.out.println();
    System.out.println("il migliore risultato approssimato e': "+y);
        System.out.println( "il risultato utilizzando Math.sqrt e': "+ Math.sqrt(x) );
  }
}
```

Di seguito viene mostrato il risultato dell'esecuzione dal prompt di DOS



Esercizio 5

Scrivere il programma **`Tabelline`** che inizialmente chiede all'utente un numero intero `num` compreso tra 2 e 10 (estremi inclusi). Se il numero fornito dall'utente non soddisfa questa condizione, il programma richiede il numero, finché necessario. Quindi il programma stampa le tabelline da 1 a `num` con una opportuna intestazione.

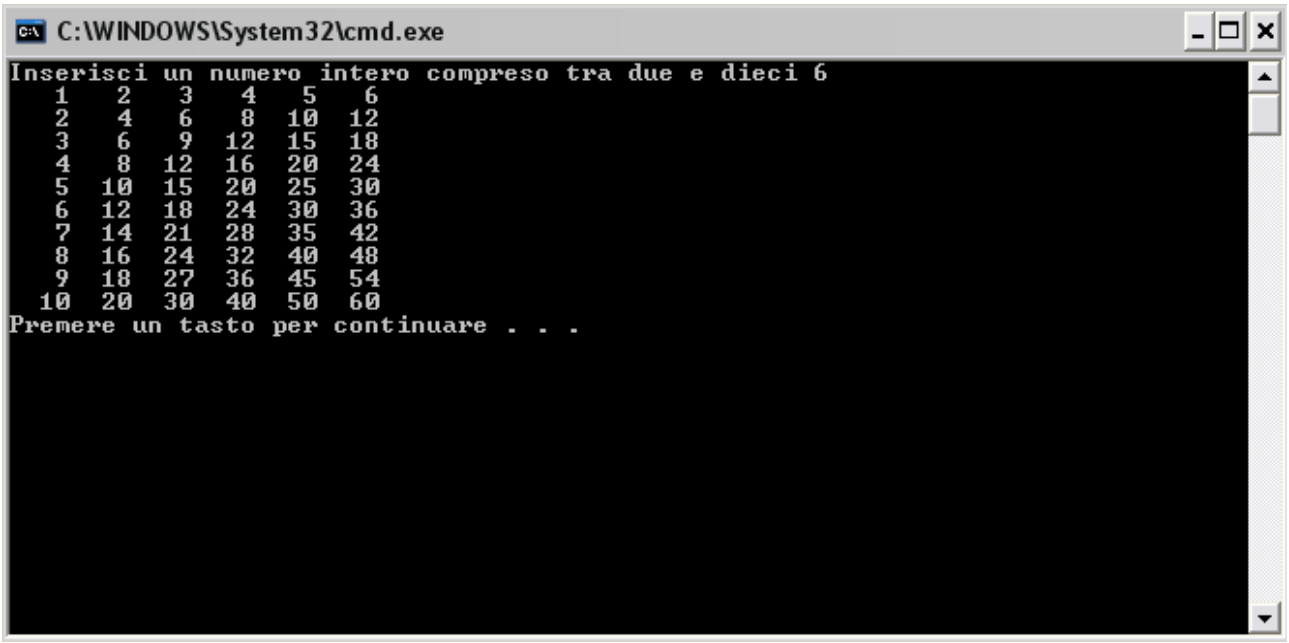
```
import corejava.*;
class Tabelline {
public static void main (String args[])
{       int num;
        boolean fuoriBound;
        do{
        num=Console.readInt("Inserisci un numero intero compreso tra due e dieci");
        fuoriBound=num<2||num>10;
        if(fuoriBound)
                System.out.println("Errore!il numero non e' compreso tra due e dieci");
                }
        while(fuoriBound);
        int count=1;
        int numero=count;
        do{
        //inizio la prima colonna a partire dal primo numero della tabellina
                while(numero<=num){
                int ris=numero*count;
                if(ris<10)
                System.out.print(" ");
                if(ris<100)
                        System.out.print(" ");
                        System.out.print(" "+ris);
                        numero++;
                }
        System.out.println(); //passo alla riga successiva
                count++;
                numero=1;//ricomincio dalla prima colonna
                }
        while(count<=10);
}

}
```

Soluzione

La prima riga importa il package corejava in cui e' contenuta la classe *Console*.java . Tale classe verrà utilizzata per leggere da input il numero intero inserito dall'utente e assegnarlo alla variabile intera **num**. Poiche' il programma richiede che il numero appartenga all'intervallo >=2 e <=10 viene utilizzata l'istruzione **do{…} while(fuoriBound)** per leggere l'intero, se il numero inserito dall'utente e' fuori dell'intervallo richiesto (cioe' la variabile booleana **fuoriBound=true**), il programma stampa un messaggio di errore (**if** all'interno del **do..while**) e ripropone all'utente l'inserimento del numero finche' necessario.
A questo punto bisogna stampare le prime **num** tabelline (per esempio se num =6 stamperà le prime 6 tabelline del 10). Poichè la stampa avviene per righe sono necessari due cicli di **while** innestati, il ciclo piu' esterno (istruzione **do..while**) stampa la riga (se sono sulla riga 2 moltiplico tutti i numeri per due, ecc..) , mentre il ciclo interno (istruzione **while**) mi definisce in quale colonna mi trovo. NOTA: la stampa degli spazi (istruzione **System.out.print(" ");)** mi serve per mettere in colonna i risultati.

Di seguito viene mostrato il risultato dell'esecuzione dal prompt di DOS



Esercizio 6

Scrivere il programma **`SommaQuadrati`** che ripete 5 volte le seguenti azioni: chiede all'utente una sequenza di numeri interi, quindi ne stampa la somma dei quadrati, e la media aritmetica. Ogni sequenza si considera terminata quando l'utente fornisce uno zero in input (lo zero NON fa parte della sequenza).

Soluzione

La prima riga importa la classe *Console*.java contenuta nel package corejava . Tale classe verrà utilizzata per leggere da input il numero intero inserito dall'utente e assegnarlo alla variabile intera **num**. Poiche' il programma richiede che siano lette 5 sequenze di numeri, si utilizza il ciclo do..while esterno per leggere 5 volte la sequenza, internamente al do while c'è un do while interno che definisce la singola sequenza. Il ciclo interno infatti e' eseguito fin quando il numero letto e' diverso da zero (la sequenza termina con uno zero) ed esegue le seguenti operazioni:

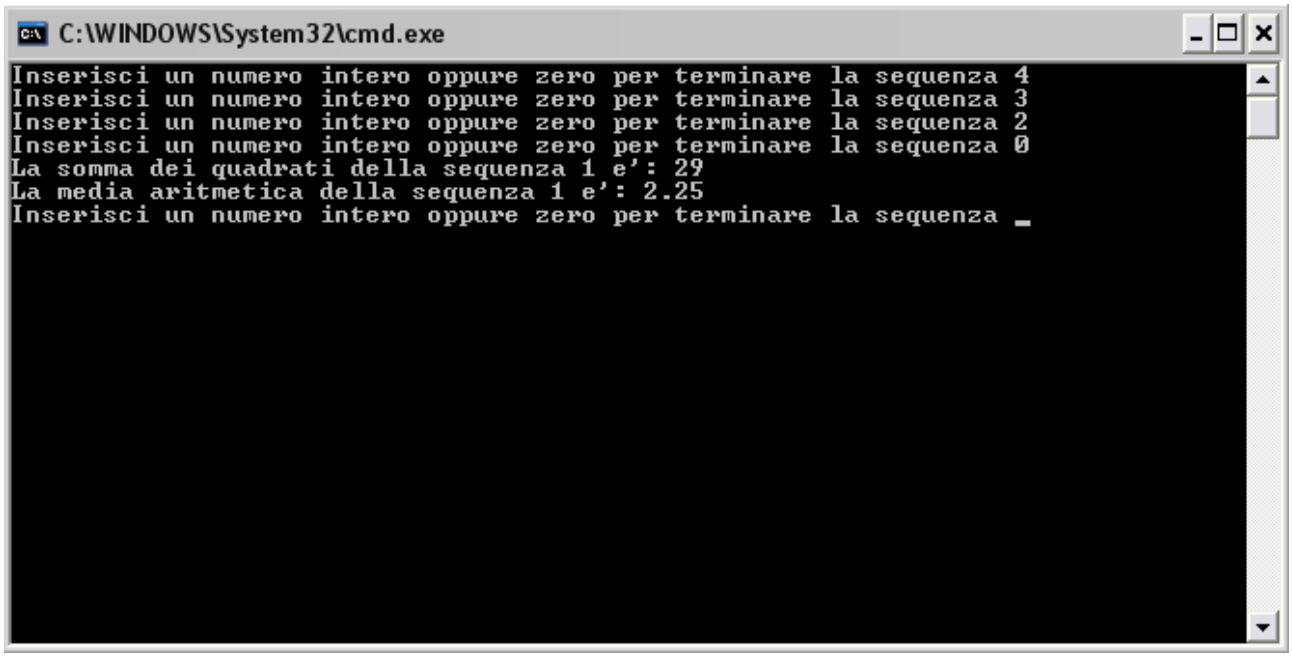
1. legge il numero della sequenza e lo somma ai precedenti;
2. calcola il quadrato (con il metodo **Math.pow(numero,2**);) e lo somma ai precedenti;

Usciti dal while interno, stampa la somma dei quadrati e la media aritmetica calcolata prima e passa alla sequenza successiva.

```
import corejava.*;
class SommaQuadrati {
public static void main (String args[])
{       final int num=5;
        int nVolte=0;
        int sommaQuad, numero, count;
        double somma;
        do{ //inizio di una sequenza
                nVolte++;
                sommaQuad=0;
                somma=0;
                count=0;
                do{
                        numero=Console.readInt("Inserisci un numero intero oppure zero per terminare la
                        sequenza");
                        count++;
                        somma+=numero;
                        sommaQuad+=(int)Math.pow(numero,2);
                }  while(numero!=0);
        System.out.println("La somma dei quadrati della sequenza " + nVolte+" e': "+sommaQuad);
        System.out.println("La media aritmetica della sequenza " + nVolte+" e': "+ somma/count);
        } while(nVolte<=num);
}

}
```

Di seguito viene mostrato parte dell'esecuzione dal prompt di DOS



Esercizio 7

Scrivere il programma **`DoppioTriangolo`** che inizialmente chiede all'utente un numero pari `num` compreso tra 2 e 20 (estremi inclusi). Se il numero fornito dall'utente non soddisfa questa condizione, il programma chiede di nuovo il numero, finché necessario. Quindi il programma stampa un quadrato di caratteri di lato `num`, la cui diagonale (dal vertice inferiore sinistro a quello superiore destro) e' costituita dal carattere `'%'`, il triangolo superiore e' costituito da `'+'`, e quello inferiore da `'='`. Ad esempio, se `num=5`, allora il programma stampa il quadrato mostrato in seguito:

```
++++%
+++%=
++%==
+%===
%====
```

Soluzione

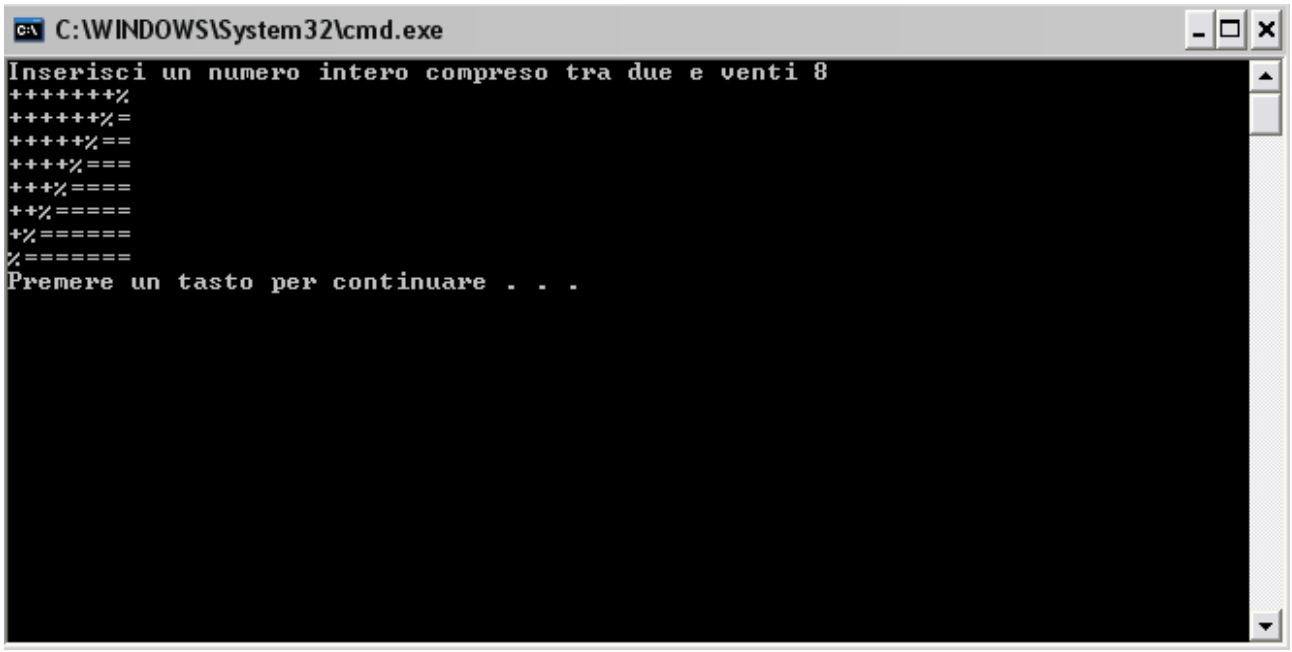
```
import corejava.Console;
class DoppioTriangolo {
public static void main (String args[])
{       int num;
        boolean fuoriBound;
        do{
        num=Console.readInt("Inserisci un numero intero compreso tra due e venti");
        fuoriBound=num<2||num>20;
        if(fuoriBound)
                System.out.println("Errore!il numero non e' compreso tra due e venti");
                }
        while(fuoriBound);
        int righe=1;
        int colonne=1;
        do{
                while(colonne<=num){
                if(colonne<num-righe+1)
                        System.out.print("+");
                else
                        if(colonne==num-righe+1)
                        System.out.print("%");
                        else
                                System.out.print("=");

                colonne++;
                }
                System.out.println();
                righe++;
                colonne=1;
                } while(righe<=num);
}
}
```

La prima riga importa il package corejava in cui e' contenuta la classe *Console*.java . Tale classe verrà utilizzata per leggere da input il numero intero inserito dall'utente e assegnarlo alla variabile intera **num**. Poiche' il programma richiede che il numero appartenga all'intervallo >=2 e <=20 viene utilizzata l'istruzione **do{…} while(fuoriBound)** per leggere l'intero, se il numero inserito dall'utente e' fuori

dell'intervallo richiesto (cioe' la variabile booleana **fuoriBound=true**), il programma stampa un messaggio di errore (**if** all'interno del **do..while**) e ripropone all'utente l'inserimento del numero finche' necessario.
A questo punto bisogna stampare il quadrato. La stampa avviene per righe quindi sono necessari due cicli di **while** innestati, il ciclo piu' esterno (istruzione **do..while**) stampa la riga (se **righe** =2 sono sulla riga 2) , mentre il ciclo interno (istruzione **while**) mi definisce in quale colonna mi trovo. In particolare se il numero di righe e' maggiore del numero di colonna và stampato **System.out.print("+"),** se e' uguale (mi trovo sulla diagonale) và stampato **System.out.print("%"),** altrimenti (mi trovo sotto la diagonale) và stampato **System.out.print("="),**

Di seguito viene mostrato il risultato dell'esecuzione dal prompt di DOS



Esercizio8

Scrivere il programma PariDispari che chiede ripetutamente una sequenza di numeri interi all'utente. Il programma termina quando l'utente fornisce un numero negativo, e stampa un messaggio dicendo quanti numeri pari e quanti numeri dispari l'utente ha fornito.

Soluzione

La prima riga importa la classe *Console*.java contenuta nel package corejava . Tale classe verrà utilizzata per leggere da input il numero intero inserito dall'utente e assegnarlo alla variabile intera **num**. Poiche' il programma richiede che la sequenza temina quando l'utente inserisce un numero negativo si esegue ciclicamente l'istruzione **do..while** fin quando il numero letto e' positivo (la sequenza termina con uno zero) ed esegue le seguenti operazioni:

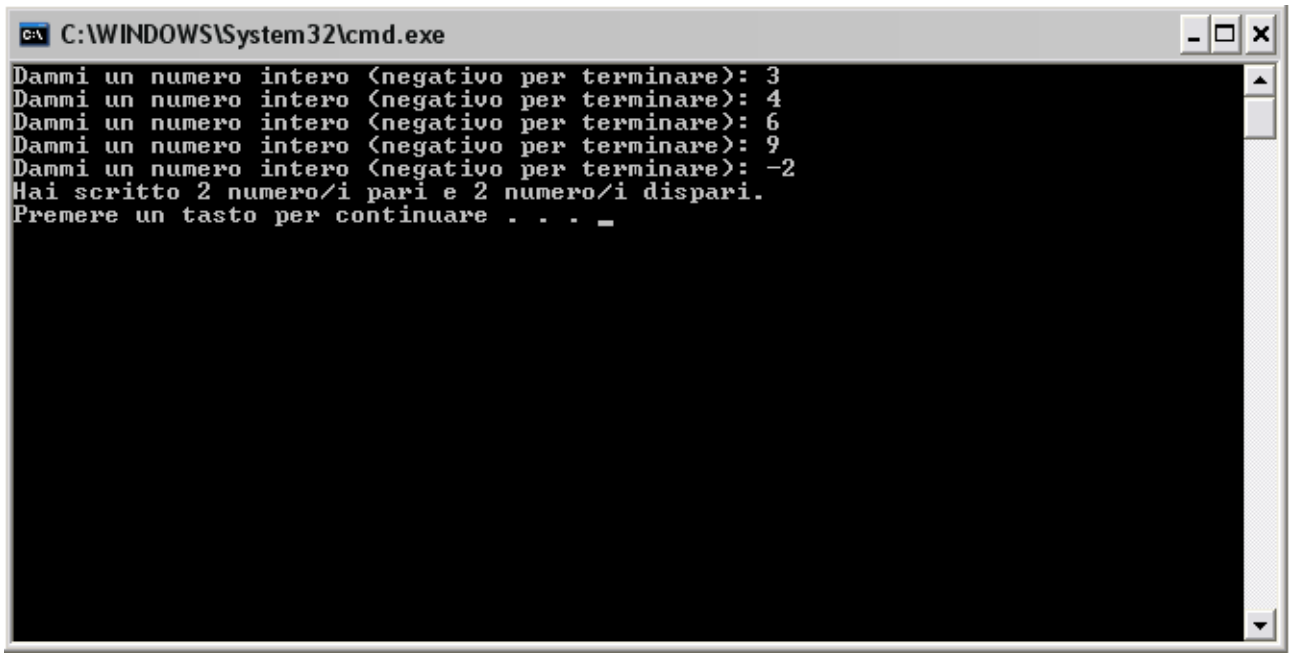
1. legge il numero della sequenza;
2. se e' positivo verifica se e' pari (e aggiorna il contatore dei **pari**) o dipari (e aggiorna il contatore dei **dispari**)

All'uscita del while (e' terminata la sequenza) stampa il numero di pari e dispari **System.out.println("Hai scritto "+ pari+" numero/i pari e " + dispari+ " numero/i dispari. ");**

```
Import corejava.*;
class PariDispari {
   public static void main(String[] args) {
      int numero;       // elemento corrente della sequenza
      int pari=0;     // numero degli elementi pari
      int dispari=0;      // numero degli elementi dispari
      int i;            // per contare gli elementi letti

      /* scrive un messaggio */
      do{
      numero=Console.readInt("Dammi un numero intero (negativo per terminare):");
                  if(numero>=0){ // devo aggiornare i pari e i dispari
                  if (numero%2==0)
                        pari++;  //il numero letto e' pari
                  else
                        dispari++;
                        }
                  }while(numero>=0);
      /* visualizza il risultato */
      System.out.println(
         "Hai scritto "+ pari+" numero/i pari e " + dispari+ " numero/i dispari. ");
   }
}
```

Di seguito viene mostrato il risultato dell'esecuzione dal prompt di DOS



Esercizio 9

Scrivere un programma per ricevere dall'utente un dato che descrive una carta da gioco, mediante le seguenti abbreviazioni:

A	Asso
2…10	Punteggio carta
J	Jack
D	Donna
R	RE
Q	Quadri
C	Cuori
P	Picche
F	Fiori

Soluzione

La prima riga importa la classe *Console*.java contenuta nel package corejava . Tale classe verrà utilizzata per leggere da input la Stringa inserita dall'utente che descrive la carta secondo la convenzione definita nella traccia e l'assegna alla variabile **card**. Il ciclo di while scandisce la stringa carattere per carattere (per esempio se l'utente inserisce la stringa **card=10P,** il primo simbolo (si parte dalla posizione zero) è **card.charAt(0)=1;**). L'istruzione di swicth, fornisce per ogni **case** la descrizione del simbolo. (NOTA: osserva l'utilizzo del break nel caso il simbolo sia un numero)

```
Import corejava.*;
class Carte {
public static void main (String args[])
{
     char simbolo;
     String card=Console.readString("Inserisci la descrizione della carta");
     int count=0;
     while(count<card.length()){
          simbolo=card.charAt(count);
          switch(simbolo)
     { case 'A':System.out.print("Asso");break;
       case 'J':System.out.print("Jack");break;
       case 'D':System.out.print("Donna");break;
       case 'R':System.out.print("Re");break;
       case 'Q':System.out.print(" di quadri");break;
       case 'P':System.out.print(" di picche");break;
       case 'F':System.out.print(" di fiori");break;
       case '0':
       case '1':
       case '2':
       case '3':
       case '4':
       case '5':
       case '6':
       case '7':
       case '8':
       case '9':System.out.print(simbolo);break;
       default: System.out.print("simbolo inserito ERRATO");
     }
     count++;
     }
     System.out.println();
}
}
```

Di seguito viene mostrato il risultato dell’esecuzione dal prompt di DOS