Reti e Laboratorio
Modulo Laboratorio 3
AA. 2025-2026

docente: Laura Ricci

laura.ricci@unipi.it

Correzione 1 Esercizio di Ripasso

02/10/2025

ESERCIZIO RIPASSO N.1

- considerare un'azienda nella cui organizzazione sono coinvolte diverse persone, con i seguenti diversi ruoli
  - impiegati livello 1: hanno uno stipendio mensile base
  - impiegati livello 2: ottengono un bonus da aggiungere allo stipendio mensile base degli impiegati di primo livello
  - lavoratori a ore: vengono pagati una cifra standard per ogni ora lavorata
  - volontari: non percepiscono alcuna paga
- tutte le persone coinvolte nella organizzazione sono caratterizzate dal nome, l'indirizzo ed il numero di telefono.
- per tutti gli impiegati di livello 1 e di livello 2 livello e per i lavoratori a ore viene registrato anche il codice fiscale e un campo che riporta lo stipendio base per gli impiegati di livello 1 e di livello 2 e la paga oraria per i lavoratori a ore

ESERCIZIO RIPASSO N.1

- si scriva un programma JAVA che
  - crei uno staff contenente un numero prefissato (dato in input) di lavoratori per ognuno dei tipi precedenti, impostando, in fase di creazione, lo stipendio mensile standard per gli impiegati di livello 1 e 2, il bonus per quelli di livello 2 e la cifra oraria per i lavoratori a ore
  - quindi calcoli e stampi la paga per tutte le persone coinvolte nell'azienda, stampando per i volontari la stringa "`Grazie!`", invece dello stipendio
  - si deve considerare il valore dello stipendio mensile, calcolato in base ai valori registrati in fase di creazione, e alle ore lavorate per i lavoratori a ore
- il programma
  - deve essere strutturato in un insieme di classi organizzate gerarchicamente
  - deve utilizzare il polimorfismo per la definizione del metodo che calcola lo stipendio di ogni membro dell'organizzazione

I COLLABORATORI

```
package Assignment_1;
abstract public class StaffMember
   { protected String name;
     protected String address;
     protected String phone;
//-----------------------------------------------------------------
// costruttore: crea un membro dello staff della azienda
// inserisce le informazioni comuni a tutti i collaboratori (anche i
volontari): nome,indirizzo e telefono
// ma il modo di calcolare il salario è diverso per ogni dipendente
//-----------------------------------------------------------------
    public StaffMember(String Name, String Address, String Phone)
      {
        this.name = Name;
        this.address = Address;
        this.phone = Phone;
      }
    //prosegue nella pagina successiva
```

I COLLABORATORI

```
//-----------------------------------------------------------------------
// restituisce una stringa con le informazioni base del collaboratore
// toString definito per overriding del metodo toString della classe
   Object
// quel metodo offre solo informazioni generiche sul nome della classe
// è bene sempre ridefinirlo per dare informazioni più specifiche
//-----------------------------------------------------------------------
public String toString()
   {
     String result = "Name: " + name + "\n";
     result += "Address: " + address + "\n";
     result += "Phone: " + phone;
return result;
  }
// prosegue nella pagina successiva
```

I COLLABORATORI

```
//-------------------------------------------------------------------
// metodo comune a tutti i collaboratori è il metodo che stabilirà il loro
// salario
// ogni classe di collaboratori avrà poi la propria procedura per
   calcolare il salario
// per i volontari si restituirà un salario pari a 0
// il metodo deve essere definito quindi come astratto, solo segnatura del
   metodo, niente codice
//-------------------------------------------------------------------
   public abstract double pay();
   }
```

GLI IMPIEGATI

```
package Assignment_1;
public class Impiegato extends StaffMember
   { protected String codfisc;
     protected double payRate;
//--------------------------------------------------------------------
// costruttore: per un lavoratore dipendente (non volontario)
// si registra anche codice fiscale e un valore che serve per il
calcolo del suo salario
// questo valore assume significati diversi per impiegati e lavoratori
   a ore, per i primi è il salario, per i secondi la paga oraria
//--------------------------------------------------------------------
     public Impiegato(String Name, String Address, String Phone,
                      String codfisc, double rate)
          {
           super(Name, Address, Phone);
           this.codfisc = codfisc;
           payRate = rate;
          }
// prosegue nella pagina successiva
```

GLI IMPIEGATI

```
public String toString()
    {
     String result = super.toString();
     result += "\nSocial Security Number: " + codfisc;
     return result;
    }
//-----------------------------------------------------------------------
// implementazione del metodo astratto pay()
// il salario per questo impiegato è il valore passato al costruttore
//-----------------------------------------------------------------------
public double pay()
    {
       return payRate;
    }
}
```

I DIRIGENTI

```
package Assignment_1;
   public class Dirigente extends Impiegato
    { private double bonus;
//-----------------------------------------------------------------------
// Costruttore: un dirigente è un impiegato particolare
// Aggiunge allo stipendio un bonus, che viene messo a 0 e verrà indicato al
   momento del calcolo dello stipendio
//-----------------------------------------------------------------------
    public Dirigente(String Name, String Address, String Phone,
                     String codfisc, double rate)
      {
       super(Name, Address, Phone, codfisc, rate);
       bonus = 0; // bonus non ancora determinato
      }
// prosegue nella pagina successiva
```

I DIRIGENTI

```
//-----------------------------------------------------------------
// Invocato per attribuire un bonus a un dirigente
//-----------------------------------------------------------------
public void awardBonus(double execBonus)
   { bonus = execBonus; }
//-----------------------------------------------------------------
// metodo per il calcolo del salario del dirigente
// overriding del metodo pay() di Impiegato
//-----------------------------------------------------------------
public double pay()
{
   double payment = super.pay() + bonus;
   bonus = 0;
   return payment;
}
}
```

I LAVORATORI AD ORE

```
package Assignment_1;

public class A_Ore extends Impiegato
  { private int hoursWorked;
//---------------------------------------------------------------
// Costruttore: registra le informazioni generali + inizializza il numero
  di ore lavorate
//---------------------------------------------------------------
   public A_Ore(String Name, String Address, String Phone,
       String codfisc, double rate)
     {
        super(Name, Address, Phone, codfisc, rate);
        hoursWorked = 0;
     }
// prosegue nella pagina successiva
```

I LAVORATORI AD ORE

```
//-----------------------------------------------------------------
// Aggiorna le ore lavorate
//-----------------------------------------------------------------
public void addHours(int moreHours)
   { hoursWorked += moreHours; }
//-----------------------------------------------------------------
// Calcola il salario per un lavoratore a ore
//-----------------------------------------------------------------
public double pay()
   { double payment = super.pay() * hoursWorked;
    hoursWorked = 0;
    return payment; }
public String toString()
   {  String result = super.toString();
   result += "\nCurrent hours: " + hoursWorked;
   return result; }
```

I VOLONTARI

```
package Assignment_1;

public class Volontario extends StaffMember
   {
//---------------------------------------------------------------------
// Construttore: solo nome, indirizzo e numero telefono
//---------------------------------------------------------------------
   public Volontario(String Name, String Address, String Phone)
     {
       super(Name, Address, Phone);
     }
//---------------------------------------------------------------------
// Restituisce la paga uguale a 0 per il volontario
//---------------------------------------------------------------------
   public double pay()
      { return 0.0; } }
```

CREAZIONE DELLO STAFF E PAGAMENTI

```
package Assignment_1;
public class Staff
{ private StaffMember[] staffList;
//-----------------------------------------------------------------
// Constructor: costruisce una lista di collaboratori
//-----------------------------------------------------------------
public Staff()
  {staffList = new StaffMember[6];
   staffList[0] = new Dirigente("Laura","Via Roma 5","070707","123456789",2500);
   staffList[1] = new Impiegato("Carla","Via Liguria 20","050505","987654321",1250);
   staffList[2] = new Impiegato("Mario", "Via Milano 120","060606","010203040",1500);
   staffList[3] = new A_Ore("Alessia","Via Calabria 231","010101","958473625",15.55);
   staffList[4] = new Volontario("Giovanni","Via Campania 4","020202");
   staffList[5] = new Volontario("Anna","Via Cagliari 27","090909");
   ((Dirigente)staffList[0]).awardBonus(500.00);
   ((A_Ore)staffList[3]).addHours(40);
  }
// prosegue nella pagina successiva
```

CREAZIONE DELLO STAFF E PAGAMENTI

```
public void payday()
    {
     double amount;
     for (int count=0; count < staffList.length; count++)
       {
        System.out.println(staffList[count]);
        // ATTENZIONE: in questa chiamata si utilizza il polimorfismo
        amount = staffList[count].pay();
        if (amount == 0.0)
           System.out.println("Grazie!");
        else
          System.out.println("Pagato: " + amount);
          System.out.println("-----------------------------------");
       }
     }
}
```

IL MAIN

```
package Assignment_1;

public class Azienda {

//-----------------------------------------------------------------
// Creates un insieme di collaboratori e ne calcola il salario
//-----------------------------------------------------------------


public static void main(String[] args)
  {
    Staff personnel = new Staff();
    personnel.payday();
   }
}
```