

Programmazione ad Oggetti:

JAVA

Esercitazione Classi Estese

Docente

Prof. Sandro Scirocco

- Scrivere la classe **Motorino** che ha i seguenti attributi **colore**: una stringa indicante il colore del motorino, **velocità**: un numero con la virgola indicante la velocità in Km/h che possiede il motorino, **tipo**: una stringa indicante la marca e il modello del motorino es. "Piaggio scarabeo", l'attributo **antifurto** un boolean che indica se è stato inserito l'antifurto (ha un valore iniziale pari a false). Il costruttore ha come parametri una stringa per il colore, una stringa per il tipo, un numero con la virgola per la velocità ed assegna opportunamente i valori dei parametri agli attributi. Scrivere il metodo **getVelocità** che restituisce la velocità del motorino, scrivere inoltre il metodo **accelera** che ha come parametro un numero con la virgola indicante i Km/h che si vogliono aggiungere alla velocità, il metodo verifica il valore dell'attributo **antifurto** se è false aggiunge il valore del parametro all'attributo velocità, altrimenti non fa nulla. Scrivere il metodo **inserisciAntifurto** che assegna un valore true all'attributo **antifurto**.

Ricapitolando: Classe **Motorino** {colore, velocità, tipo, antifurto}
Metodi (**costruttore**), (**getVelocità**), (**accelera** se non c'è l'antifurto), (**inserisciAntifurto true**)

- Scrivere la classe **Motorino** che ha i seguenti attributi **colore**: una **stringa** indicante il colore del motorino, **velocità**: un **numero con la virgola** indicante la velocità in Km/h che possiede il motorino, **tipo**: una **stringa** indicante la marca e il modello del motorino es. “Piaggio scarabeo”, l’attributo **antifurto** un **boolean** che indica se è stato inserito l’antifurto (ha un valore iniziale pari a **false**).

```
class Motorino {  
    String colore;  
    float velocità;  
    String tipo;  
    boolean antifurto =false;  
}
```

- Il costruttore ha come parametri una **stringa** per il **colore**, una **stringa** per il **tipo**, un **numero con la virgola** per la **velocità** ed assegna opportunamente i valori dei parametri agli attributi.

```
Motorino(String c, String t, float x){  
    colore=c;  
    velocità=x;  
    tipo=t;  
}
```

Attenti al float !!!!!!!!!!!!!

- Il costruttore ha come parametri una **stringa** per il **colore**, una **stringa** per il **tipo**, un **numero con la virgola** per la **velocità** ed assegna opportunamente i valori dei parametri agli attributi.

Alternativamente il costruttore poteva essere:

```
Motorino(String colore, String tipo, float velocità){  
    this.colore=colore;  
    this.velocità=velocità;  
    this.tipo=tipo;  
}
```

- Scrivere il metodo **getVelocità** che restituisce la velocità del motorino

Il metodo restituisce la velocità occorre capire il tipo di dato per tale attributo. Nell'esercizio:

Scrivere la classe **Motorino** che ha i seguenti attributi:

.... **velocità**: un **numero con la virgola** indicante la velocità in Km/h che possiede il motorino

ho definito l'attributo come
float velocità;

```
float getVelocità(){  
    return velocità;  
}
```

- Scrivere inoltre il metodo **accelera** che ha come parametro un numero con la virgola indicante i Km/h che si vogliono aggiungere alla velocità, il metodo verifica il valore dell'attributo **antifurto** se è false aggiunge il valore del parametro all'attributo velocità, altrimenti non fa nulla.

```
void accelera(float x){  
    if(antifurto == false) {velocità = velocità+x; }  
}
```

- Scrivere il metodo **inserisciAntifurto** che assegna un valore true all'attributo **antifurto**.

```
void inserisciAntifurto(){  
    antifurto=true;  
}
```

```

class Motorino {
    String colore;
    float velocità;
    String tipo;
    boolean antifurto =false;
    Motorino(String c, String t, float x){
        colore=c;
        velocità=x;
        tipo=t;
    }
    float getVelocità(){
        return velocità;
    }
    void accelera(float x){
        if(antifurto == false) velocità = velocità+x;
    }
    void inserisciAntifurto(){
        antifurto=true;
    }
}

```

- Scrivere la classe **MotorinoImmatricolato** sottoclasse della classe Motorino che ha in più 2 attributi: **maxVelocità** un numero con la virgola (coerente con la scelta fatta per l'attributo velocità) indicante la velocità massima in Km/h che il motorino può raggiungere; **targa** una stringa indicante la targa del motorino (ad entrambi gli attributi viene assegnato un valore nel costruttore). Aggiungere il metodo **getMax** il metodo stampa il valore dell'attributo maxVelocità. Ridefinire il metodo **accelera** in modo che prima di modificare la velocità effettui un controllo sulla velocità massima raggiunta. Il metodo definisce una variabile s (dello stesso tipo di velocità) ed assegna ad s la somma tra il valore del parametro del metodo ed il valore dell'attributo velocità; se s è minore del valore dell'attributo maxVelocità assegna il valore di s all'attributo velocità altrimenti assegna all'attributo velocità il valore dell'attributo maxVelocità.

- Scrivere la classe **MotorinoImmatricolato** sottoclasse della classe **Motorino**

```
class MotorinoImmatricolato extends Motorino {  
  
}
```

- Scrivere la classe **MotorinoImmatricolato** sottoclasse della classe **Motorino** che ha in più 2 attributi:
maxVelocità un numero con la virgola (coerente con la scelta fatta per l'attributo velocità) indicante la velocità massima in Km/h che il motorino può raggiungere;
targa una stringa indicante la targa del motorino (a entrambi gli attributi viene assegnato un valore nel costruttore).

```
class MotorinoImmatricolato extends Motorino {  
  
    float maxVelocità;  
    String targa;  
}
```

La sottoclasse non eredita il costruttore!

- Nel testo: (a entrambi gli attributi viene assegnato un valore nel costruttore).

Costruttore
superclasse



```
Motorino(String c, String t, float x){  
    colore=c;  
    velocità=x;  
    tipo=t;  
}
```

```
MotorinoImmatricolato(String c, String t, float x,  
float y, String targa){  
    super(c,t,x);  
    maxVelocità = y;  
    this.targa = targa;  
} //costruttore sottoclasse
```

- Aggiungere il metodo **getMax** il metodo stampa il valore dell'attributo maxVelocità.

```
void getMax(){  
    System.out.println(maxVelocità);  
}
```

- Ridefinire il metodo **accelera** in modo che prima di modificare la velocità effettui un controllo sulla velocità massima raggiunta.

Metodo della superclasse Motorino

```
void accelera(float x){  
    if(antifurto == false) velocità = velocità+x;  
}
```

- Il metodo definisce una variabile s (dello stesso tipo di velocità) ed assegna ad s la somma tra il valore del parametro del metodo e il valore dell'attributo velocità; se s è minore del valore dell'attributo maxVelocità assegna il valore di s all'attributo velocità altrimenti assegna all'attributo velocità il valore dell'attributo maxVelocità.

```
void accelera(float x){  
    float s=velocità+x;  
    if(antifurto == false) {  
        if(s<=maxVelocità) velocità=s;  
        else velocità=maxVelocità; }  
}
```



```

class MotorinoImmatricolato extends Motorino {
    float maxVelocità;
    String targa;
    MotorinoImmatricolato(String c, String t, float x, float y,
        String targa){
        super(c,t,x);
        maxVelocità = y;
        this.targa = targa;
    }
    void getMax(){
        System.out.println(maxVelocità);
    }
    void accelera(float x) {
        float s=velocità+x;
        if(antifurto == false)
            if(s<=maxVelocità) velocità=s;
            else velocità=maxVelocità;
    }
}

```

- Scrivere la classe **UsoMotorino** che ha il metodo main in cui si istanzia un oggetto di tipo Motorino di colore *grigiometallizzato* con velocità *40,5* e tipo *Piaggio Liberty* , richiamare il metodo *getVelocità* memorizzandone il valore in una variabile. Istanziare un altro oggetto di tipo *MotorinoImmatricolato* con colore *rosso*, velocità *30,5* tipo *Aprilia Scarabeo*, maxvelocità *60* targa *CV1234*, e richiamare il metodo *getMax*. Infine richiamare il metodo *accelera* con parametro pari a *30,7* per entrambi gli oggetti e quindi richiamare il metodo *getVelocità* per entrambi gli oggetti stampando i due valori.

- Scrivere la classe **UsoMotorino** che ha il metodo main

```
class UsoMotorino {  
  
    public static void main( String args[]){  
  
    }  
  
}
```

- Scrivere la classe **UsoMotorino** che ha il metodo main in cui si istanzia un oggetto di tipo **Motorino** di colore *grigiometallizzato* con velocità *40,5* e tipo *Piaggio Liberty*

```
Motorino m;  
m= new Motorino("grigiometallizzato","Piaggio Liberty",40.5);
```

```
Motorino m;  
m= new Motorino("grigiometallizzato", "Piaggio Liberty", 40.5);
```

Scrivendo in tal modo ho un errore

```
C:\esercizi>javac UsoMotorino.java  
UsoMotorino.java:5: cannot resolve symbol  
symbol : constructor Motorino (java.lang.String,java.lang.String,double)  
location: class Motorino  
m= new Motorino("grigiometallizzato", "Piaggio Liberty", 40.50);  
  ^
```

Il compilatore avvisa che nella classe usoMotorino trova la chiamata ad un costruttore che vuole come parametri una stringa, una stringa, un double (40.5) ma non esiste

il costruttore
è stato definito

```
Motorino(String c, String t, float x){  
    colore=c;  
    velocità=x;  
    tipo=t;  
}
```

- Scrivere la classe **UsoMotorino** che ha il metodo main in cui si istanzia un oggetto di tipo **Motorino** di colore *grigiometallizzato* con velocità *40,5* e tipo *Piaggio Liberty*

```
class UsoMotorino {  
  
    public static void main( String args[]){  
        Motorino m;  
        m= new Motorino("grigiometallizzato", "Piaggio Liberty", 40.5f)  
  
    }  
}
```

- Scrivere la classe **UsoMotorino** che ha il metodo main in cui si istanzia un oggetto di tipo Motorino di colore *grigiometallizzato* con velocità *40,5* e tipo *Piaggio Liberty* , richiamare il metodo `getVelocità` memorizzandone il valore in una variabile.

```
class UsoMotorino {  
  
    public static void main( String args[]){  
        Motorino m;  
        m= new Motorino("grigiometallizzato","Piaggio Liberty",40.5f)  
        float x = m.getVelocità();  
    }  
}
```

- Istanziare un altro oggetto di tipo **MotorinoImmatricolato** con colore *rosso*, velocità *30,5* tipo *Aprilia Scarabeo*, maxvelocità *60* targa *CV1234*,

```
MotorinoImmatricolato mI;  
mI= new  
MotorinoImmatricolato("rosso","Aprilia  
Scarabeo",30.5f, 60.0f, "CV1234");
```

- richiamare il metodo getMax

Il metodo nella classe è definito

```
void getMax(){  
    System.out.println(maxVelocità);  
}
```

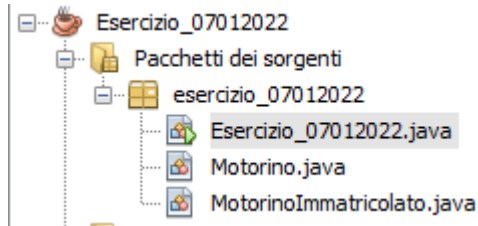
```
mI.getMax();
```

- Infine richiamare il metodo accelera con parametro pari a 30,7 per entrambi gli oggetti e quindi richiamare il metodo getVelocità per entrambi gli oggetti stampando i due valori.

```
mI.accelera(30.7f);  
m.accelera(30.7f);  
System.out.println(mI.getVelocità());  
System.out.println(m.getVelocità());
```

```
class UsoMotorino {  
  
    public static void main( String args[]){  
        Motorino m;  
        m= new Motorino("grigiometallizzato", "Piagio Liberty",40.5f);  
        float x = m.getVelocità();  
  
        MotorinoImmatricolato mI;  
        mI= new Motorino("rosso", "Aprilia Scarabeo",30.5f,  
            60.0f,"CV1234");  
        mI.getMax();  
        mI.accelera(30.7f);  
        m.accelera(30.7f);  
        System.out.println(mI.getVelocità());  
        System.out.println(m.getVelocità());  
  
    }  
}
```

Esercizio commentato



```
package esercizio_07012022;
```

```
public class Esercizio_07012022 {
```

```
    public static void main(String[ ] args) {
```

```
        System.out.println("abbiamo creato una nuova istanza del motorino");
```

```
        Motorino m; // istanza della classe motorino (si crea un motorino prima della immatricolazione)
```

```
        m= new Motorino("grigiometallizzato","Piagio Liberty",40.5f);
```

```
        float x = m.getVelocità(); // si legge la velocità e la si assegna alla variabile x
```

```
        System.out.println("velocità "+x); // si stampa la parola velocità ed il valore della variabile x
```

```
        // diamo un accelerata di +120,5 km/h
```

```
        System.out.println("acceleriamo di ben 120,5 km/h con il metodo accelera");
```

```
        m.accelera(120.5f); // chiamiamo il metodo accelera e gli passiamo 120,5
```

```
        System.out.print("nuova velocità dopo accelerata ");
```

```
        System.out.println(m.getVelocità()); // stampiamo la nuova velocità
```

```
        //passiamo al motorino targato
```

```
        System.out.println("abbiamo creato una nuova istanza del motorino Immatricolato con limite di velocità");
```

```
        MotorinoImmatricolato mI; // istanza della sottoclasse motorinoImmaricolato (si crea un motorino targato)
```

```
        mI= new MotorinoImmatricolato("rosso","Aprilia Scarabeo",30.5f, 60.0f,"CV1234");
```

```
        System.out.print("velocità motorino targato ");
```

```
        float y = mI.getVelocità();
```

```
        System.out.println("velocità "+y); // si stampa la parola velocità ed il valore della variabile y
```

```
        System.out.print("velocità massima del motorino targato ");
```

```
        mI.getMax(); //chiama il metodo getmax che stampa la velocità massima
```

```
        System.out.println("lo acceleriamo di 30,7 dovrebbe andare a 61,2 ma siccome la massima è 60.... ");
```

```
        mI.accelera(30.7f); // chiamiamo il metodo accelera e gli passiamo 30,7
```

```
        System.out.print("nuova velocità dopo accelerata ");
```

```
        mI.getMax(); //chiama il metodo getmax che stampa la nuovavelocità
```

```
    }
```

```
}
```

```
public class Motorino {
```

```
    // attributi
```

```
    String colore;
```

```
    float velocità;
```

```
    String tipo;
```

```
    boolean antifurto =false;
```

```
    //costruttore
```

```
    Motorino(String c, String t, float x){
```

```
        colore=c;
```

```
        velocità=x;
```

```
        tipo=t;
```

```
    }
```

```
    //metodo che restituisce la velocità
```

```
    float getVelocità(){
```

```
        return velocità;
```

```
    }
```

```
    //metodo per aumentare la velocità
```

```
    void accelera(float x){
```

```
        if(antifurto == false) velocità = velocità+x;
```

```
    }
```

```

// metodo per inserire vero sull'antifurto
void inserisciAntifurto(){
    antifurto=true;
}
}

public class MotorinoImmatricolato extends Motorino {
    float maxVelocità;
    String targa;
    MotorinoImmatricolato (String c, String t, float x, float y, String targa){
        super(c,t,x);
        maxVelocità = y;
        this.targa = targa;
    }
    void getMax(){
        System.out.println(maxVelocità);
    }
    void accelera(float x) {
        float s=velocità+x;
        if(antifurto == false)
            if(s<=maxVelocità) velocità=s; else velocità=maxVelocità;
    }
}

```