



Università  
degli Studi  
di Ferrara

**DE** Department of  
Engineering  
Ferrara

Esercitazioni di  
**FONDAMENTI DI  
INFORMATICA  
MODULO B**

**ESERCITAZIONE 10 - JAVA**

Bertagnon Alessandro: <sup>3</sup>[alessandro.bertagnon@unife.it](mailto:alessandro.bertagnon@unife.it)  
Azzolini Damiano: [damiano.azzolini@unife.it](mailto:damiano.azzolini@unife.it)

# Esercizio su Classi Astratte e Interfacce - 1

---

Si crei la seguente classe **astratta** **Veicolo**:

```
private double velocita;
```

```
private double accelerazione;
```

```
public Veicolo() { ... }
```

```
public Veicolo(double velocita, double accelerazione) { ... }
```

```
// GETTERS
```

```
public double getVelocita() { ... }
```

```
public double getAccelerazione() { ... }
```

## Esercizio su Classi Astratte e Interfacce - 2

---

// SETTERS

**public void** setVelocita(**double** velocita) { ... }

**public void** setAccelerazione(**double** accelerazione) { ... }

// prende in input un oggetto di classe VEICOLO e ne

// stampa la velocità

**static public void** printVelocita(**Veicolo** v){ ... }

// stampa il numero di ruote del veicolo, METODO ASTRATTO

**public abstract void** printNRuote();

## Esercizio su Classi Astratte e Interfacce - 3

---

Si crei la seguente classe **Automobile** che **estende** la classe Veicolo:

```
private String targa;
```

```
private boolean avviata;
```

```
public Automobile(String targa){ ... }
```

```
// "accende" e "spegne" la macchina, agendo sul boolean avviata
```

```
public void avvia() { ... }
```

```
public void spegni() { ... }
```

## Esercizio su Classi Astratte e Interfacce - 4

---

```
// accelera controlla se la macchina è avviata, nel
// qual caso calcola i nuovi valori
// di accelerazione e velocità (vel. Precedente +
// acc*sec)
public void accelera(double accelerazione, int secondi) { ... }

// stampa il numero di ruote del veicolo => 4
public void printNRuote();
```

# Esercizio su Classi Astratte e Interfacce - 5

---

Si crei la seguente classe **Bicicletta** che estende la classe Veicolo

```
// nel costruttore, impostare velocità e
// accelerazione a 0
public Bicicletta(){ ... }

// pedala imposta i nuovi valori nel seguente
// modo:
// - accelerazione (num. Pedalate / sec ^2)
// - velocità (vel. Precedente + acc*sec)
public void pedala(int numeroPedalate, int secondi) { ... }

// stampa il numero di ruote del veicolo => 2
public void printNRuote();
```

# Esercizio su Classi Astratte e Interfacce - main

---

- Creare una classe **VeicoloMain** contenente il metodo main.
- Nel metodo main definire un oggetto **a** istanza di **Automobile** e un oggetto **b** istanza di **Bicicletta**.
- Richiamare i metodi **pedala** e **accelera** rispettivamente per **b** e **a**.
- Stampare la velocità dei due veicoli e il loro numero di ruote, usando i metodi `printVelocita(Veicolo v)` e `printNRuote()`.

# Esercizio su Classi Astratte e Interfacce - 1

---

- Si supponga di avere delle carte da gioco. Ogni carta rappresenta un **Personaggio** che può essere di tipo o **Umano** o **Mostro**.
- Un Personaggio di tipo **Umano** ha una forza fisica iniziale pari al numero 10 e può **combattere**, un Personaggio di tipo **Mostro**, invece, ha una forza fisica iniziale pari a 15 e può **azzannare**.

# Esercizio su Classi Astratte e Interfacce - 2

---

- I Personaggi del gioco sono Eroe, Vampiro e Licantropo:
  - **Eroe** è solo della categoria **Umano**,
  - **Vampiro** è solo della categoria **Mostro**,
  - **Licantropo** è di tipo **Mostro** nelle notti di luna piena, altrimenti di tipo **Umano**.
- In particolare, la forza fisica dei personaggi diminuisce di un valore pari a:
  - 3 per l'Eroe ad ogni combattimento
  - 2 per il Vampiro ad ogni azzanno
  - 2 per il Licantropo nelle notti di luna piena, 3 nelle altre

# Esercizio su Classi Astratte e Interfacce - 3

---

- Si dovranno definire tutte le classi e le **interfacce** necessarie per realizzare il gioco.
- Si dovrà inoltre definire una classe principale che conterrà il main. Il metodo main:
  - Istanza tre oggetti: uno di tipo Eroe, uno di tipo Licantropo e uno di tipo Vampiro
  - Fa combattere tre volte l'Eroe
  - Fa combattere una volta il Vampiro
  - Fa combattere il Licantropo due volte.
  - Stampa al termine la forza fisica rimasta a ciascun personaggio

# Esercizio su Classi Astratte e Interfacce - 4

---

- Cosa conviene usare: **subclassing** o **interfacce**?
- Cosa modelliamo con le classi e cosa con le **interfacce**?
- La scelta **CORRETTA** è usare:
  - le **interfacce** per modellare i **comportamenti**
  - le **classi** per modellare le **entità concrete**

# Esercizio su Classi Astratte e Interfacce - 5

---

