



Programmazione Object Oriented

Realizzazione di Classi

Dr. Dario Di Nucci - 30/09/2022

Cosa impariamo oggi?

| | Martedì (ore: 9-11 Aula: F/1) | Giovedì (ore: 9-11 Aula: F/1) | Venerdì (ore: 9-13 aula: P/13) |
|----------------------------------|--|---------------------------------------|---|
| Settimana 1 (20-23 Settembre) | Introduzione al corso Programmazione OO | Utilizzare gli Oggetti | Lab: Introduzione Lab: Scrittura primi programmi in Java |
| Settimana 2 (27-30 Settembre) | Utilizzare le Classi | Realizzare Classi | Lab: Creare classi ed oggetti |
| Settimana 3 (4-6 Ottobre) | Tipi di Dati Fondamentali | Decisioni/Iterazioni | |
| Settimana 4 (11-14 Ottobre) | Vettori e Array | Progettazione di Classi | Lab: Creare classi |
| Settimana 5 (18-21 Ottobre) | Progettazione di Classi | Interfacce e Polimorfismo | Lab: Classi con array |
| Settimana 6 (25-27 Ottobre) | Interfacce e Polimorfismo | Ereditarietà | Lab: Packages e variabili statiche |
| Settimana 7 (3-4 Novembre) | | Ereditarietà | Lab: Interfacce |
| Settimana 8 (8-11 Novembre) | Testing e Debugging | Collezioni Programmazione Generica | Lab: Interfacce |
| Settimana 9 (15-18 Novembre) | Gestione delle Eccezioni | IO e Flussi | Lab: Ereditarietà |
| Settimana 10 (22-25 Novembre) | Programmazione Grafica | Gestione degli Eventi | Lab: Eccezioni |
| Settimana 11 (29 Nov - 2 Dic) | Interfacce Grafiche Utente | Lambda Expressions | Lab: Esercizi su eventi e interfacce grafiche |
| Settimana 12 (6-9 Dicembre) | Reflections | | Lab: Esercizi su eventi e interfacce grafiche |

Esercizio

Scrivere la classe **SavingsAccount** che è del tutto simile alla classe **BankAccount**, tranne che per una variabile di istanza aggiuntiva, **interestRate** (tasso di interesse annuo). Fornire, oltre a tutti i metodi della classe **BankAccount**:

1. Un metodo costruttore che imposti sia il saldo iniziale che il tasso di interesse;
2. Un metodo **addInterest** che aggiunge gli interessi trimestrali al conto.

Esercizio

Un distributore di bibite contiene lattine di bibite. Per comprare una bibita il cliente deve inserire un gettone. Quando il gettone è inserito, una lattina cade nel contenitore dove può essere presa dal cliente. Il distributore può essere riempito con ulteriori lattine. Vogliamo determinare quante lattine e gettoni sono presenti nella macchina ad un dato istante.

1. Quali metodi sono necessari per la classe **VendingMachine**?
2. Tradurre le descrizioni informali in Java signatures.
3. Aggiungere le variabili di istanza (i.e., attributi) necessarie.
4. Implementare e testare la classe ottenuta.



Esercizi

Esercizio come il precedente rispetto però ad una classe **Student** dove siamo interessati a registrare i voti degli esami di uno studente e recuperare il voto medio.

Esercizio come il precedente rispetto però ad una classe **Infected** dove siamo interessati a recuperare il numero totale di infetti per una malattia e questi possono aumentare o diminuire in base al parametro R_t .



Esercizio

La classe **Car** è caratterizzata dalla resa del carburante (litri/chilometro) e la quantità di carburante nel serbatoio. La resa è specificata nel costruttore e inizialmente il serbatoio è vuoto.

La classe contiene un metodo **drive** per simulare il percorso di un'automobile per una certa distanza, riducendo conseguentemente il livello di carburante nel serbatoio, un metodo **getGas** per ispezionare il livello del carburante e un metodo **addGas** per fare rifornimento.

Realizzare e testare la classe.

Esercizio

Implementare e testare una classe **InputChecker** che contenga un metodo che stampi

1. **"OK"** se legge "S" ,"SI", "OK", "certo" oppure "perché no?".
2. **"Fine"** se legge "N" o "No"
3. **"Dato non corretto"** altrimenti



Esercizio

Realizzare e testare una classe **FirstAndLastWord** che contenga due metodi.

Il primo metodo chiede all'utente quante parole voglia inserire, le prenda in input da tastiera e stampi la prima e l'ultima parola secondo l'ordine alfabetico.

Il secondo metodo è uguale al precedente ma ignora la differenza tra maiuscole e minuscole

Esercizio

Realizzare e testare una classe **Locker** per rappresentare una serratura con combinazione di tre lettere maiuscole

I metodi sono

- **void unlock(String)** che apre la serratura se il codice passato è quello della serratura
- **boolean isOpen()** che verifica se la serratura è aperta
- **void lock()** che chiude la serratura
- **void newComb(String)** che setta una nuova combinazione se la serratura è aperta

Cosa facciamo la prossima volta?

| | Martedì (ore: 9-11 Aula: F/1) | Giovedì (ore: 9-11 Aula: F/1) | Venerdì (ore: 9-13 aula: P/13) |
|----------------------------------|--|---------------------------------------|---|
| Settimana 1 (20-23 Settembre) | Introduzione al corso Programmazione OO | Utilizzare gli Oggetti | Lab: Introduzione Lab: Scrittura primi programmi in Java |
| Settimana 2 (27-30 Settembre) | Utilizzare le Classi | Realizzare Classi | Lab: Creare classi ed oggetti |
| Settimana 3 (4-6 Ottobre) | Tipi di Dati Fondamentali | Decisioni/Iterazioni | |
| Settimana 4 (11-14 Ottobre) | Vettori e Array | Progettazione di Classi | Lab: Creare classi |
| Settimana 5 (18-21 Ottobre) | Progettazione di Classi | Interfacce e Polimorfismo | Lab: Classi con array |
| Settimana 6 (25-27 Ottobre) | Interfacce e Polimorfismo | Ereditarietà | Lab: Packages e variabili statiche |
| Settimana 7 (3-4 Novembre) | | Ereditarietà | Lab: Interfacce |
| Settimana 8 (8-11 Novembre) | Testing e Debugging | Collezioni Programmazione Generica | Lab: Interfacce |
| Settimana 9 (15-18 Novembre) | Gestione delle Eccezioni | IO e Flussi | Lab: Ereditarietà |
| Settimana 10 (22-25 Novembre) | Programmazione Grafica | Gestione degli Eventi | Lab: Eccezioni |
| Settimana 11 (29 Nov - 2 Dic) | Interfacce Grafiche Utente | Lambda Expressions | Lab: Esercizi su eventi e interfacce grafiche |
| Settimana 12 (6-9 Dicembre) | Reflections | | Lab: Esercizi su eventi e interfacce grafiche |