

## Prova Pratica - Livello Base Compito B1

---

COGNOME E NOME: \_\_\_\_\_ MATRICOLA: \_\_\_\_\_

**Tempo a disposizione: 3 ore**

### Dipinti in Java

E' necessario scrivere un'applicazione in linguaggio Java che gestisca i dati relativi ad una collezione di dipinti. Per ciascun dipinto è necessario rappresentare il titolo e l'anno di produzione.

All'avvio dell'applicazione viene creata una collezione che non contiene dipinti.

Successivamente l'applicazione deve consentire di svolgere i seguenti caso d'uso:

**“Utente inserisce dipinto”**

- l'utente fornisce i dati di un nuovo dipinto
- l'applicazione aggiunge il dipinto alla collezione

**“Utente visualizza dati dipinti”**

- il sistema stampa sullo schermo l'elenco dei dipinti

**“Utente cerca dipinto più recente”**

- il sistema cerca il dipinto con l'anno di produzione più recente
- il sistema stampa i dati del dipinto sullo schermo

Scenario alternativo: nessun dipinto nella collezione

- l'applicazione stampa un messaggio di errore e il caso d'uso si conclude

**“Utente verifica anni dipinti”**

- l'utente fornisce il valore di un anno
- il sistema verifica che tutti i dipinti della collezione siano successivi all'anno fornito dall'utente
- il sistema stampa un messaggio sullo schermo per segnalare se la condizione è verificata o meno

Scenario alternativo: nessun dipinto nella collezione

- l'applicazione stampa un messaggio di errore e il caso d'uso si conclude

Sviluppare l'applicazione che implementa i casi d'uso elencati. Al termine dello sviluppo, produrre i seguenti diagrammi UML:

- modello concettuale
- diagramma delle classi

## Prova Pratica - Livello Base Compito B2

---

COGNOME E NOME: \_\_\_\_\_ MATRICOLA: \_\_\_\_\_

**Tempo a disposizione: 3 ore**

### Francobolli in Java

E' necessario scrivere un'applicazione in linguaggio Java che gestisca i dati relativi ad una collezione di francobolli. Per ciascun francobollo è necessario rappresentare il nome (es: "Gronchi Rosa") e il valore in euro.

All'avvio dell'applicazione viene creata una collezione che non contiene francobolli.

Successivamente l'applicazione deve consentire di svolgere i seguenti caso d'uso:

**“Utente inserisce francobollo”**

- l'utente fornisce i dati di un nuovo francobollo
- l'applicazione aggiunge il francobollo alla collezione

**“Utente visualizza dati francobolli”**

- il sistema stampa sullo schermo l'elenco dei francobolli

**“Utente cerca francobolli economici”**

- l'utente fornisce un valore in euro
- il sistema stampa sullo schermo i dati di tutti i francobolli il cui valore è inferiore a quello fornito dall'utente

**“Utente verifica condizione sui francobolli”**

- il sistema verifica se esistono due francobolli con lo stesso nome che appaiono l'una dopo l'altra nella collezione
  - il sistema stampa un messaggio sullo schermo per segnalare se la condizione è verificata o meno
- Scenario alternativo: nessun dipinto nella collezione
- l'applicazione stampa un messaggio di errore e il caso d'uso si conclude

Sviluppare l'applicazione che implementa i casi d'uso elencati. Al termine dello sviluppo, produrre i seguenti diagrammi UML:

- modello concettuale
- diagramma delle classi

## Prova Pratica - Livello Base Compito B3

---

COGNOME E NOME: \_\_\_\_\_ MATRICOLA: \_\_\_\_\_

**Tempo a disposizione: 3 ore**

### Elezioni in Java

E' necessario scrivere un'applicazione in linguaggio Java che gestisca i dati relativi ai candidati alle elezioni politiche. Per ciascun candidato è necessario rappresentare il nome e i voti riportati.

All'avvio dell'applicazione viene creata una collezione che non contiene dipinti.

Successivamente l'applicazione deve consentire di svolgere i seguenti caso d'uso:

**“Utente inserisce candidato”**

- l'utente fornisce i dati di un nuovo candidato
- l'applicazione aggiunge il candidato alla collezione

**“Utente visualizza dati candidati”**

- il sistema stampa sullo schermo l'elenco dei candidati

**“Utente cerca candidato più votato”**

- il sistema cerca il candidato con il più alto numero di voti
- il sistema stampa i dati del candidato sullo schermo

Scenario alternativo: nessun candidato nella collezione

- l'applicazione stampa un messaggio di errore e il caso d'uso si conclude

**“Utente visualizza dati dei primi n candidati nella collezione”**

- l'utente fornisce un numero intero n (es: n=3)
- il sistema visualizza i dati dei primi n candidati contenuti nella collezione (es: i dati dei candidati nelle prime 3 posizioni della collezione)

Scenario alternativo: non ci sono candidati sufficienti nella collezione

- il sistema stampa un messaggio di errore e invita a ripetere l'immissione

Sviluppare l'applicazione che implementa i casi d'uso elencati. Al termine dello sviluppo, produrre i seguenti diagrammi UML:

- modello concettuale
- diagramma delle classi

## Prova Pratica - Livello Base Compito B4

---

COGNOME E NOME: \_\_\_\_\_ MATRICOLA: \_\_\_\_\_

**Tempo a disposizione: 3 ore**

### Dipendenti in Java

E' necessario scrivere un'applicazione in linguaggio Java che gestisca i dati relativi ai dipendenti di un'azienda. Per ciascun dipendente è necessario rappresentare il nome e lo stipendio.

All'avvio dell'applicazione viene creata una collezione che non contiene dipendenti.

Successivamente l'applicazione deve consentire di svolgere i seguenti caso d'uso:

#### “Utente inserisce dipendente”

- l'utente fornisce i dati di un nuovo dipendente
- l'applicazione aggiunge i dati del dipendente alla collezione

#### “Utente visualizza dati dipendenti”

- il sistema stampa sullo schermo l'elenco dei dipendenti

#### “Utente calcola stipendio medio”

- l'utente fornisce un valore di stipendio
- il sistema verifica che lo stipendio medio dei dipendenti sia superiore allo stipendio fornito dall'utente
- il sistema stampa lo stipendio medio calcolato e un messaggio per segnalare se la condizione precedente è verificata o meno

Scenario alternativo: nessun dipendente nella collezione

- l'applicazione stampa un messaggio di errore e il caso d'uso si conclude

#### “Utente verifica stipendi dipendenti”

- il sistema verifica che tutti i dipendenti abbiano uno stipendio superiore a 2000 Euro
- il sistema stampa un messaggio sullo schermo per segnalare se la condizione è verificata o meno

Scenario alternativo: nessun dipendente nella collezione

- l'applicazione stampa un messaggio di errore e il caso d'uso si conclude

Sviluppare l'applicazione che implementa i casi d'uso elencati. Al termine dello sviluppo, produrre i seguenti diagrammi UML:

- modello concettuale
- diagramma delle classi