

# Esercitazione

1. Quali sono le possibili cause di un'eccezione?
2. Descrivere il costrutto formale per la gestione di una eccezione fornendo un esempio.
3. Descrivere i ruoli che esistono all'interno di un progetto
4. Descrivere la differenza tra class diagram e object diagram descrivendo gli elementi di cui possono essere composti.

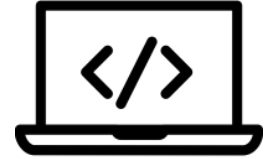
# **Esercitazione – Esercizio Pratico**

# Esercitazione



- Si vuole realizzare un'applicazione console per la gestione di un conto bancario caratterizzato da:
  - *Numero di Conto*
  - *Nome della Banca*
  - *Saldo*
  - *Data Ultima Operazione*

# Esercitazione



- Per il conto corrente deve essere possibile realizzare dei movimenti, (siano essi di prelievo o di versamento). Ciascun movimento bancario è caratterizzato da

- *Importo*
- *Data del Movimento*

*I movimenti potranno essere poi così distinti:*

- *CashMovement*, con la proprietà aggiuntiva *Esecutore*
- *TransfertMovement*, con le proprietà aggiuntive *Banca d'Origine* e *Banca Destinazione*
- *CreditCardMovement*, con le proprietà aggiuntive *Tipo* (enum con i valori AMEX, VISA, MASTERCARD, OTHER) e *Numero di Carta*

*Continua ...*

# Esercitazione



La Console app da realizzare consentirà

- Di creare un nuovo Account
- Di inserire diversi tipi di Movimenti (input dall'utente)
- Stampare i dati del conto e i movimenti

Realizzare class diagram, object diagram e use case diagram per l'applicazione appena descritta.

Controllare l'input dell'utente valutando tutte le opzioni di programmazione difensiva. Realizzare inoltre un'opportuna eccezione da scatenare quando l'utente sfora il saldo con un prelievo.

# Esercitazione – Consegna



- Creare un repository su GitHub con il nome nel formato:
  - *TestWeek6*
- All'interno del file aggiungere:
  - Il file DOCX con le risposte alle domande
  - La solution sviluppata per l'esercitazione pratica