Appunti teoria di informatica  
L’**override** (sovrascrittura) è un meccanismo che consente a una sottoclasse di fornire una nuova implementazione di un metodo ereditato dalla superclasse. Serve per personalizzare il comportamento di un metodo.

L’**overload** è quando un metodo ha lo stesso nome ma firme diverse nella stessa classe. Serve per implementare comportamenti diversi a seconda dei parametri.

L’**overhead** è il “lavoro extra” che la **Java Virtual Machine (JVM)** deve fare per decidere quale metodo usare **quando il programma è in esecuzione**.

Usando il polimorfismo (per esempio, una variabile di tipo “generico” che può contenere oggetti di classi diverse), la JVM deve controllare in runtime quale versione del metodo è giusta per l’oggetto reale.

**Esempio**:

Hai una variabile Animale animale che potrebbe contenere un oggetto Cane o un oggetto Gatto.

Quando chiami il metodo animale.verso(), la JVM deve capire se deve eseguire Cane.verso() o Gatto.verso().

Un’**interfaccia** è un contratto che definisce un insieme di metodi (tutti astratti, fino a Java 7). A differenza delle classi astratte:

• Una classe può implementare più interfacce.

• I metodi di un’interfaccia sono implicitamente public e abstract.

Una **classe astratta** è una classe i cui oggetti non possono essere istanziati direttamente e serve come “modello” per altre classi. Può contenere:

• Metodi concreti (con corpo).

• Metodi astratti (senza corpo, definiti con la parola chiave abstract).

Un’**exception** (eccezione) in Java è un evento che si verifica durante l’esecuzione di un programma, che interrompe il normale flusso di esecuzione. Le eccezioni possono essere causate da errori come l’accesso a risorse non disponibili, divisioni per zero o operazioni non valide. Java gestisce le eccezioni attraverso un sistema di blocchi try-catch, che permette di intercettarle e gestirle in modo controllato.