



# PIANO DI LAVORO CONSUNTIVO

## INFORMATICA Classe 4Aii 2020-21

### **Informazioni generali**

**Docenti:** Luigi Ferrari, Gaetano Sberna

#### **Materiale didattico e testi consigliati**

- [1] Dispense fornite in classe a cura del docente, reperibili sull'ambiente e-learning della scuola
- [2] C.S. Horstmann – Concetti di informatica e fondamenti di Java; Apogeo Education, 5.a ed 2007.
- [3] ambiente e-learning della scuola, con altro materiale.

#### **Altre fonti utilizzate dal docente per le spiegazioni in classe:**

- Documentazione ufficiale di Java in rete, sul sito [java.oracle.com](http://java.oracle.com) (in particolare <http://docs.oracle.com/javase/tutorial/tutorialLearningPaths.html>)
- [www.horstmann.com](http://www.horstmann.com) sito contenente in particolare un corso organizzato a slide con esempi molto precisi (in inglese)
- E. K. Koffmann, P. A. T. Wolfgang - Object, abstraction, data structures and design using Java 5.0 - Wiley, 2005 (in inglese)
- Articoli di approfondimento reperibili su internet

### **Argomenti trattati**

#### **1. La programmazione ad oggetti**

Confronto tra programmazione strutturata e programmazione ad oggetti.  
Concetti di oggetto, classe, variabile, riferimento, metodo, interfaccia.  
Terminologia dei linguaggi ad oggetti. Concetto di ereditarietà e polimorfismo.  
Programmazione agile ed extreme programming: concetti di base.

- [1] modulo 1
- [2] cap. 1, 2

#### **2. Il linguaggio di programmazione Java: fondamentali**

Storia di Java. Java come linguaggio interpretato. Java virtual machine.  
Sviluppo di applicazioni Java con il JDK, con l'ambiente BlueJ.  
Sintassi del linguaggio. Classi pubbliche e private (classi interne); metodi pubblici e metodi statici.  
Variabili di classe. Costruttori.  
Scopo delle variabili. Modificatori di accesso (public, protected, private). Tipi di metodi: per l'accesso agli attributi, modificatori. La classe String. Array in Java. Classe Stack. Possibile realizzazione di array dinamici. Variabili e costanti di classe (static).

Documentazione del codice.

Il problema del controllo del codice: criteri per la scelta dei casi di test, test semiautomatici. Il metodo main, le classi di test.

- [1] modulo 2
- [2] cap. 3, 4, 6, 7, 8 per la parte sugli array

#### **1. Relazioni tra gli oggetti**

Progetto di programmi che coinvolgono più classi. Criteri per la scelta delle classi. Interazione tra le classi: basso accoppiamento, elevata coesione. Programmazione guidata dalla responsabilità. Metodi accessori e metodi modificatori. Variabili e metodi statici. Visibilità delle variabili.  
Esempi di ereditarietà e di polimorfismo. Classi astratte, interfacce per supplire alla mancanza di ereditarietà multipla. La superclasse Object. Binding dinamico dei dati. Le interfacce come specifiche. Ereditarietà di variabili e metodi; costruzione di sottoclassi; controllo di accesso agli attributi e ai metodi delle sottoclassi; conversioni tra superclasse e sottoclasse. Esempi di interazioni tra gli oggetti:



composizione, ereditarietà, implementazione (di interfacce). Cenni al controllo degli errori: eccezioni in java.

[1] modulo 3

[2] cap. 8, 9, 10

## 2. Grafica in java

(solo pratico e solo per alcuni) Realizzazione di interfacce grafiche in Java: uso di IntelliJ e VisualStudio Code per la realizzazione di interfacce utenti grafiche (GUI)

Interfacce grafiche. Le librerie Swing. Concetto di programmazione basata sugli eventi. Contenitori: Frame e Pannelli. Componenti: bottoni, caselle di testo, combo box...

[2] cap. 5 (grafica di base), 11 (gestione degli eventi), 13 (interfacce utente grafiche)

## 3. Laboratorio

Installazione del JDK di Java e di Bluej. Uso di Greenfoot per la realizzazione di semplici programmi grafici. Realizzazione di semplici classi e sistemi di classi con Bluej. Uso di base (per alcuni) dei sistemi di sviluppo Visual Studio Code o IDEA IntelliJ. Documentazione del software con Javadoc.

### ***Obiettivi minimi per la sufficienza***

Sono considerate conoscenze assolutamente necessarie per conseguire la sufficienza e per il passaggio alla classe successiva:

- Conoscere e saper implementare in C e in Java i principali algoritmi di ordinamento, ricerca ed elaborazione numerica (numeri primi, MCD, mcm...)
- Conoscere i principi base della OOP (astrazione, incapsulamento, ereditarietà, polimorfismo..)
- Conoscere la sintassi di base di Java: attributi di tipo primitivo e oggetto, array di tipo primitivo e di oggetti; costruttori; metodi osservatori e metodi modificatori, la classe Object.
- Saper realizzare una classe, documentando e definendo gli attributi (anche array), i costruttori, i metodi osservatori e modificatori della classe; saper testare in modo automatico le classi realizzate.
- Conoscere i principali metodi di aggregazione tra le classi (composizione, associazione, ereditarietà, implementazione di interfacce...)

Saranno indicati sulla piattaforma google classroom attività di recupero e rinforzo sia per gli alunni con sospensione del giudizio che senza.

Genova Sestri, 3 Giugno 2021

Docenti: \_\_\_\_\_

Alunni: \_\_\_\_\_