

# UNIVERSITÀ DELLA CALABRIA



Facoltà di Ingegneria

Corso di laurea in Ingegneria Informatica

## “Polinomio GUI”

Professore del corso di POO:

Prof. Libero Nigro

Studente:

Carmelo Gugliotta

ANNO ACCADEMICO 2020-2021

# Progetto: Polinomio GUI

Il progetto consente di manipolare polinomi tramite operazioni visualizzabili a schermo.

La gerarchia di classi utilizzata per lo sviluppo del progetto è la seguente:

1. PolinomioGUI (**Classe Concreta, GUI**)
2. RegexPolinomi (**Classe di Utilità, String to Polinomio**)

## 1. PolinomioGUI (Classe Concreta, GUI)

La classe concreta presenta un main, in grado di istanziare la Classe FinestraGUI e di visualizzare a schermo una vera e propria Graphical User Interface, che permette all'utente di effettuare operazioni sui polinomi aggiunti.

La classe FinestraGUI estende JFrame e presenta i seguenti metodi:

- FinestraGUI ()
  - Costruttore che si occupa del corretto istanziamento della Finestra, con cui l'utente dovrà successivamente interagire.  
Si noti che durante l'istanziamento della Finestra viene istanziato anche un oggetto AscoltatoreEventiAzione (la classe implementa ActionListener) che consente la gestione delle seguenti componenti:
    - **JMenuItem**: tipoLL, tipoSet, tipoLC, apri, salva, salvaConNome, esci, about, aggiungiPolinomio, rimuoviPolinomio, modificaPolinomio, svuota.
    - **JBUTTON**: Box\_Val, Box\_add, Box\_mul, Box\_D.
- AggiungiPolinomio(String Polinomio)
  - Riceve un oggetto String il cui contenuto è un Polinomio. Si occupa di aggiungere il polinomio alla lista di JCheckBox, e di permettere la sua visualizzazione su schermo.
- AggiornaFrame()
  - Si occupa della visualizzazione corretta dei componenti di cui è composta la finestra.  
Ogni qualvolta si inseriscano dei componenti alla finestra è bene rivalidare il Frame e attuare un repaint dello stesso.
- consensoUscita()
  - Si occupa di gestire il caso in cui l'utente volesse uscire dal programma.  
N.B: nel costruttore abbiamo ridefinito l'operazione di chiusura aggiungendo un WindowListener con parametro un WindowAdapter() (Per ridefinire solo i metodi interessati).
- DisattivaComandi()
  - Si occupa di interrompere la possibilità di interagire con alcune componenti della Finestra.
- AttivaComandi()
  - Si occupa di dare la possibilità all'utente di interagire con alcune componenti della finestra.
- PreMenu() e menuAvviato()
  - Si occupano di disabilitare e attivare alcune componenti, prima della scelta del tipo (LL,Set,LC) e dopo la scelta di quest'ultimo.

- `ripristina(String nomeFile)`  
Riceve come parametro un oggetto stringa il cui contenuto è caratterizzato dal nome di un file. Durante la sua esecuzione può lanciare `IOException`. Si occupa di ripristinare polinomi utilizzati in una sessione precedente tramite l'utilizzo di un `BufferedReader`.
- `salva(String nomeFile)`  
Riceve come parametro un oggetto stringa il cui contenuto è caratterizzato dal nome di un file. Durante la sua esecuzione può lanciare `IOException`. Si occupa di salvare i polinomi visualizzati a schermo.

**N.B:** Quando l'utente aggiunge un nuovo polinomio, tramite il `JMenuItem` "aggiungiPolinomio", viene istanziato un nuovo `Frame`, che si occuperà di ricevere il polinomio, valutare il suo contenuto e aggiungere quest'ultimo al `Pannello_Polinomi`.

**N.B:** Importante è la classe `RegexPolinomi` (`poo.util`), che permette, di ricevere una stringa e un tipo (`LL,LC,SET`), e di istanziare un `Polinomio` che esprima il valore dell'oggetto stringa. Lancia `IllegalArgumentException()` se il tipo passato non è riconosciuto, o se l'oggetto stringa passato come parametro non esprime un polinomio ben formato.