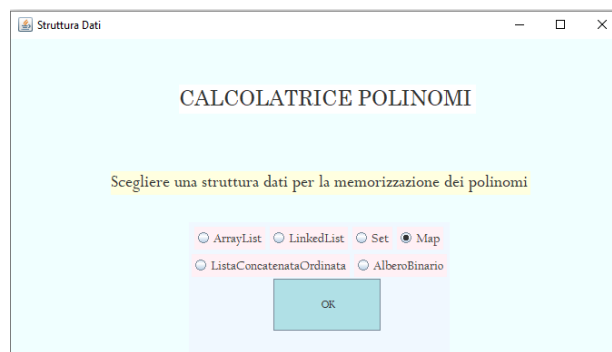


## Progetto Polinomi

Il progetto *Polinomi* si compone di due package: *poo.util*, munito delle classi e delle interfacce necessarie per l'utilizzo delle strutture dati *Albero Binario* e *Lista Concatenata Ordinata*, e *poo.polinomi*, composto dalle classi necessarie per l'implementazione dei polinomi nelle varie strutture dati richieste; nello stesso package sono inoltre presente le due classi principali "*PolinomioGUI.java*" che implementa l'interfaccia grafica e "*PolinomioUtil.java*", che rappresenta una classe di utilità contenente i metodi adibiti al controllo della validità dei polinomi e alla conversione dei polinomi in stringhe e viceversa.

### Punto 1

La scelta del tipo di implementazione dei polinomi è consentita attraverso un frame apposito "*RadioButtonFrame*", il quale consente di memorizzare il polinomio *attraverso ArrayList, LinkedList, Map, Set, Albero binario, Lista Concatenata Ordinata*: a tal proposito si utilizzando le rispettive classi *PolinomioAR, PolinomioLL, PolinomioMap, PolinomioSet, PolinomioABR, PolinomioLCO*. All'interno del frame, la scelta del tipo di implementazione è gestita attraverso dei *JRadioButton*.



### Punto 2

Una volta scelta l'implementazione è possibile iniziare ad inserire polinomi attraverso due opzioni:

"*Inserisci*" ed "*InserimentoDiretto*" che portano alla comparsa dei rispettivi frame, ovvero "*FrameInserisciPolinomio*" e "*FrameInserisciPolinomioDiretto*", dotati ciascuno di un' apposita area di inserimento nella quale scrivere il polinomio.

Il primo consente di costruire il polinomio, inserendo un monomio per volta, di cui ne verrà verificata la correttezza per ogni inserimento attraverso il metodo "*monomioValido*"; il secondo consente di inserire direttamente un intero polinomio di cui ne verrà verificata la correttezza con l'apposito metodo "*polinomioValido*".

### Punto 3

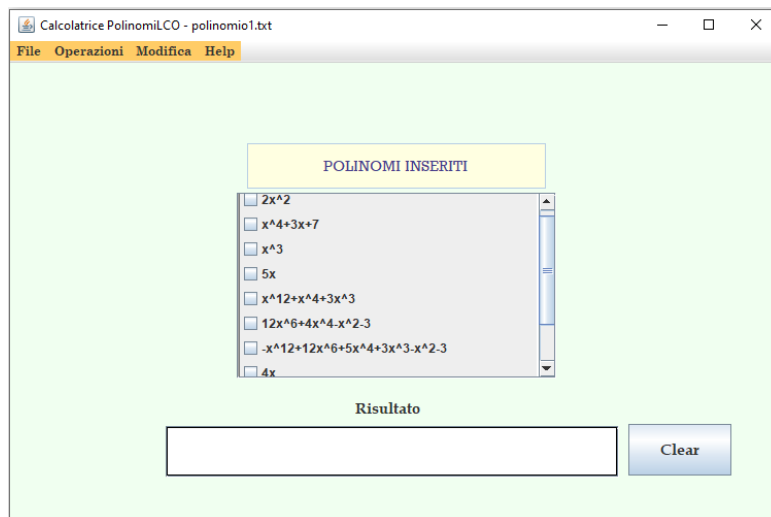
I polinomi vengono memorizzati come testo all'interno delle checkbox, e di volta in volta, all'occorrenza, con le dovute verifiche, vengono convertiti in oggetti Polinomi. In questo modo, anche lavorando con tantissimi polinomi si lavorerà soltanto con 2 oggetti Polinomi alla volta in quanto i restanti polinomi risiederanno come contenuto testuale all'interno degli oggetti checkbox.

L'idea alla base della memorizzazione dei polinomi è l'utilizzo di una variabile statica "*polinomio*" dichiarata globalmente, utilizzata per creare altri polinomi vuoti diversi, attraverso il metodo factory, grazie al quale diviene possibile mantenere la struttura dati scelta.

I polinomi sono memorizzati ciascuno in una differente checkbox, il cui contenuto è allocato dinamicamente, è possibile svolgere anche operazioni di modifica su un polinomio già creato.

Il metodo *“gestioneCheckbox”* si occupa di creare la nuova checkbox, aggiornare l’elenco delle checkbox, di inserirle in un JScrollPane e gestire la selezione di esse.

Le checkbox sono a loro volta memorizzate all’interno di una TreeMap *“elenco Checkbox”*, che avrà come chiave il numero della checkbox e come valore l’oggetto checkbox: le checkbox all’interno della mappa sono disposte per ordine cronologico di inserimento.



#### Punto 4

Attraverso il metodo salvataggio è possibile salvare all’interno di un file di testo il contenuto di ogni checkbox contenente i polinomi creati. Il metodo in questione itera sulla mappa delle checkbox scrivendo per ogni riga di un file, un polinomio.

Attraverso il metodo ripristino è possibile, invece, prelevare righe da un file di testo, convertirle in polinomi e memorizzarle sottoforma di checkbox. Questo metodo crea un ArrayList di backup, crea un polinomio  $p$  attraverso il metodo factory, che sia del tipo scelto, e per ogni riga del file converte la stringa in polinomio attraverso il metodo *“conversioneStringPolinomio”*.

#### Punto 5

Il metodo *“gestioneCheckbox”* si occupa anche di tenere traccia, attraverso la variabile *“selezionate”*, della selezione e della deselection delle checkbox, stabilisce inoltre il vincolo di non poter selezionare più di 2 polinomi alla volta.

Attraverso la selezione di un solo polinomio è possibile eseguire le operazioni di derivata, valutazione e cambio di un polinomio.

La variabile *“selezionate”* permette di verificare che non più di un polinomio sia stato selezionato.

Viene creato un nuovo polinomio  $p$  ed un ciclo itera sulla mappa delle checkbox verificandone lo stato, se la checkbox risulta selezionata il suo contenuto verrà prelevato e trasformato in polinomio attraverso il metodo *conversioneStringPolinomio* e restituito all’interno della variabile  $p$ . In base alla scelta verrà applicata l’operazione opportuna.

## Punto 6

Stesso discorso vale per le operazioni binarie, in questo caso la variabile *selezionate* si assicurerà che non più di 2 polinomi vengano selezionati e nuovamente un ciclo sulla mappa si occuperà di convertire da stringa a polinomio il contenuto delle due checkbox in questione.

Si noti che le operazioni di derivata, addizione e moltiplicazione porteranno alla creazione di un nuovo oggetto checkbox che avrà come contenuto il risultato della operazione appena eseguita, che sarà quindi disponibile per nuove operazioni.

## Punto 7

L'eliminazione dei polinomi creati e quindi delle checkbox associate, è permessa sia attraverso la selezione dei polinomi (1 o 2 per volta) che attraverso lo svuotamento totale del pannello in uso.

Nel primo caso iterando sulla mappa si eliminano solo le checkbox che risultato selezionate e si utilizza l'oggetto checkbox fornito dall'iteratore per passarlo alla remove del pannelloCheckbox.

Nel secondo caso si itera sulla mappa delle checkbox e si elimina ogni elemento, e si usa sempre l'oggetto checkbox fornito dall'iteratore, per eliminare la checkbox anche dal pannello.

Successivamente il pannello viene aggiornato per apportare le modifiche

### Elementi basilari dell'interfaccia grafica

I comandi di modifica, inserimento, salvataggio e scelte delle operazioni sui polinomi sono garantiti ed organizzati grazie all'utilizzo combinato di JMenuBar e JMenuItem. Le checkbox sono memorizzate all'interno di un JPanel inserito a sua volta in un JScrollPane e la console è costituita da una JTextArea.

I frame principali utilizzati per l'interfaccia sono:

*FrameAL*: utilizzato per l'inserimento della capacità nel caso di ArrayList

*FrameInserisciPolinomio*: usato per l'inserimento del polinomio tramite monomi

*FrameInserisciPolinomiodiretto* utilizzato per l'inserimento diretto di un polinomio

*FrameCambiaPolinomio*: utilizzato per la modifica di un polinomio già esistente

*FrameValuta*: utilizzato per la valutazione di un polinomio.

*FrameRadioButton*: utilizzato per la scelta dell'implementazione.

Vengono inoltre utilizzate svariate JLabel indicative e JOptionPane per la segnalazione di eventuali errori o conferma delle operazioni effettuate.

*Di seguito si propone una schematizzazione dei metodi, già citati, che costituiscono il fulcro principale del funzionamento del progetto:*

Metodi di utility appartenenti alla classe PolinomioUtil.java

```
public static Monomio isolaMonomio(String Monomio)  
public static boolean monomioValido(String Monomio)  
public static boolean polinomioValido(String polinomioString)  
public static boolean numeroRealeValido(String reale)
```

Metodi di supporto appartenenti alla classe contenente l'interfaccia grafica

```
private static void gestioneCheckbox(Polinomio p)  
private static Polinomio conversioneStringPolinomio(String polinomioString)  
private static void elimina()  
private static void salvataggio(String nomeFile)  
private static void ripristino(String nomeFile)
```