Facoltà di Ingegneria

Corso di laurea in Ingegneria Informatica

“Espressione Aritmetica”

Professore del corso di POO:
Prof. Libero Nigro

Studente:
Carmelo Gugliotta

ANNO ACCADEMICO 2020-2021

Progetto: Espressione Aritmetica

Il progetto consente di effettuare la valutazione di un'espressione aritmetica intera, con le priorità degli operatori della matematica.

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

1. EspressioneAritmetica (**Classe Concreta, Immutabile (final))**
2. EspressioneAritmeticaGUI (**Classe Concreta, GUI**)

1. EspressioneAritmetica (Classe Concreta, Immutabile (final))

La seguente classe, dichiarata con il suffisso **final** (**Classe di utilità**), consente di raggiungere l'obbiettivo proposto, tramite 6 metodi, di cui uno solo pubblico (**Risolvi ()**) che dà il via all'algoritmo di valutazione discusso nel pdf.

N.B: il modificatore final intende una classe finale ovvero una classe completa che non necessita di specializzazioni o estensioni e dunque è più che logico che non sia possibile ereditarla.

I 9 metodi **definiti** sono i seguenti:

- EspressioneAritmetica ()
  - Il costruttore della seguente classe è reso privato, poiché una classe dichiarata final non può essere istanziata.
- Risolvi (String Espressione)
  - Riceve come parametro un oggetto stringa, che è l'espressione che deve essere valutata, e ritorna un intero che il valore dell'espressione. **Lancia IllegalArgumentException** se l'oggetto stringa corrisponde a un'espressione malformata (Il controllo avviene tramite la Regex).
    Il valore dell'espressione ritornato è calcolato tramite il metodo ValutaEspressione.

- valutaEspressione (**String Tokenizer** st, **boolean** flag)
  - Riceve come **parametri** un **oggetto StringTokenizer** (Già istanziato sull'oggetto stringa Espressione) e un valore **booleano** che indica lo stato d'incontro di una parentesi aperta/chiusa (Il funzionamento è spiegato più avanti). Restituisce un valore intero che corrisponde al risultato dell'espressione sulla quale è stato istanziato **st**. Lancia **IllegalStateException** durante la sua esecuzione se l'espressione è malformata.
    Il seguente metodo è stato implementato seguendo il ragionamento, spiegato nel pdf **Progetto-ValutazioneEspressione**, nella sezione **Algoritmo di valutazione**.

    Flag: Durante lo scorrimento della stringa e il push degli operandi/operatori all'interno dei rispettivi Stack, si può incontrare una parentesi aperta/chiusa. Nel caso in cui si incontri una parentesi aperta si invoca ricorsivamente la procedura di valutaEspressione cambiando però il valore della **flag** a **false** (Che indica quindi l'apertura della parentesi). Successivamente quando si incontrerà la rispettiva parentesi chiusa, se lo

stato del **flag** è a **false**, quest'ultima corrisponde alla parentesi aperta incontrata e non vi è nessun problema di malformazione, quindi si riporta la **flag** a **true**, si esce dal ciclo, si svuotano gli stack e in assenza di ulteriori malformazioni viene ritornato il risultato della **sottoEspressione**.
Se non si incontra una parentesi chiusa e lo stato della **flag** è a **false**, significa che a una parentesi aperta non corrisponde una chiusa e quindi l'espressione è malformata e viene lanciata l'eccezione **IllegalStateException ()**.
Se si incontra una parentesi chiusa e lo stato della **flag** è a **true**, significa che a una parentesi chiusa non corrisponde una precedente parentesi aperta e quindi l'espressione è malformata e viene lanciata l'eccezione **IllegalStateException ()**.

- valutaOperando (**String** opc, **StackConcatenato<String>** Operatori, **StackConcatenato<Integer>** Operandi)
    - Riceve come parametri: un oggetto stringa che esprime il valore di un Operando, il riferimento dello stack ove sono allocati gli Operatori, e il riferimento dello stack ove sono allocati gli Operandi. Si occupa del corretto push degli Operandi.
      Il suo funzionamento logico è conforme a quanto discusso a lezione.

- EseguiOperazione (int o1, int o2, String op, StackConcatenato<String> Operatori, StackConcatenato<Integer> Operandi)
    - Riceve come parametri: due interi, un oggetto stringa che indica l'operazione da effettuare e il riferimento dello stack Operandi. Si occupa di effettuare l'operazione tra due Operandi e di allocare il risultato ottenuto nella struttura dati (Stack).
      La scelta dell'operazione da effettuare è indicata dal valore dell'oggetto Stringa op, che viene controllato tramite uno switch.

- Compara (String x, String x1)
    - Riceve due oggetti stringa, che contengono rispettivamente due operatori e restituisce, seguendo le precedenze usali della matematica, un intero che indica la precedenza di x rispetto a x1.
      -1 se x1 ha priorità su x, 0 se caratterizzati dalla stessa priorità (operatori uguali), 1 se x ha priorità su x1. (Usali valori che restituisce un CompareTo).
      Innanzitutto, il metodo controlla che gli operatori passati come parametro sono uguali. Dopodiché se non è stato già restituito il valore della priorità dell'uno rispetto all'altro, si controlla il gruppo di appartenenza di x, e se x1 non appartiene a un gruppo con priorità più bassa viene restituito -1.Alla fine dei controlli si ha la certezza che x ha priorità maggiore rispetto a x1.

All'interno della classe è presente anche un main di prova, che permette di effettuare test per controllare il corretto funzionamento dell'algoritmo.

2. EspressioneAritmeticaGUI (Classe Concreta, GUI)

La classe concreta presenta un main, in grado di istanziare la Classe FinestraExp e di visualizzare a schermo una vera e propria Graphical User Interface che tramite determinate interazioni permette l'elaborazione di una espressione.

La classe FinestraExp estende JFrame e implementa ActionListener e presenta i seguenti metodi:

- FinestraExp ()
  - Costruttore che si occupare del corretto istanziamento della Finestra, con cui l'utente dovrà successivamente interagire.
- actionPerformed (ActionEvent e)
  - L'implementazione dell'action listener, ci obbliga a definire tale metodo callback che riceve come argomento l'evento scatenante e permette quindi la sua gestione.
    In particolare, se successivamente all'inserimento di una espressione (tramite **JTextField**), quest'ultima risulterà ben formata, verrà mostrato a schermo il risultato. In caso contrario l'utente sarà avvisato tramite un JOptionPane che l'espressione inserita è malformata.

GUI: