**Scientific calculator project**

**Membri del gruppo 6:**

* Luca Taddeo [l.taddeo5@studenti.unisa.it](mailto:l.taddeo5@studenti.unisa,it) (referente)
* Andrea Landi [a.landi88@studenti.unisa.it](mailto:a.landi88@studenti.unisa.it)
* Antonio Scaldaferri [a.scaldaferri4@studenti.unisa.it](mailto:a.scaldaferri4@studenti.unisa.it)
* Filippo Somma [f.somma13@studenti.unisa.it](mailto:f.somma13@studenti.unisa.it)

**Fase 1: Pre-game:**

**Link bacheca Trello:** https://trello.com/b/SAiN1zvs/calcolatrice

**Definition of Formatting:**

* Blank Space prima e dopo gli operandi;
* Blank Space dopo la virgola e non prima;
* Compact control readability style;
* Convenzione *ClassNameTest* per i nomi delle class test (e.g. la class test della classe *Adder* si chiamerà *AdderTest*);
* Convenzione *testMethodName* per i nomi dei test methods (e.g. il test method del metodo *add* si chiamerà *testAdd*).

**Definition of Done:**

* Acceptance criteria rispettati;
* Superamento degli unit tests;
* Superamento dei test funzionali (di integrazione);
* Codice formattato correttamente;
* Documentazione delle classi completa.

**Linguaggio /ambiente di sviluppo:** Java/ Apache Netbeans.

**Design architecture of the program:**

|  |  |
| --- | --- |
| Come pattern architetturale è stato scelto il Model-View-Controller in modo da separare la logica di presentazione dei dati (view) dalla logica di business (model). Viene utilizzato il framework JavaFX, per cui:   * La classe **StackPerGliperandi** corrisponde al **Model**; * La classe **FXMLDocumentController** corrisponde al **Controller**; * La **View** è rappresentata dal file **FXML.fxml**. |  |

**Primo Sprint Planning:**

**Stima iniziale della Project Velocity:** 11

**Tasks scelti per questo Sprint:**

* *#1 Costruzione stack*:   
  Creazione classe per lo stack che implementi la tecnologia LIFO.
* *#2 Notazione operandi*:  
  Implementare notazione cartesiana per i numeri complessi.

Implementare notazione per i numeri reali

* *#3 Scelta operandi*:  
  Utilizzare gli ultimi operandi per eseguire un'operazione.
* *#4 Allocazione del risultato*:  
  Allocare il risultato nello stack.
* #7 Somma:

Implementare la somma degli ultimi due operandi.

**Tasks assegnati ai membri del Team:**

* Luca Taddeo: *#4 Allocazione del risultato, #3 Scelta operandi.*
* Andrea Landi: *#2 Notazione operandi.*
* Antonio Scaldaferri: *#1 Costruzione stack.*
* Filippo Somma: *#7 Somma.*

**Goal 1° sprint:**

Visualizzazione del risultato di una somma tra due operandi inseriti mediante un’interfaccia a riga di comando.

**Primo Sprint - Sviluppo**

**Tasks completati dai membri del Team:**

* Luca Taddeo: *#7 Somma (Implementazione mediante il pattern Factory Method),*

*#8 Differenza (Implementazione mediante il pattern Factory Method),*

*#9 Prodotto (Implementazione mediante il pattern Factory Method),*

*#10 Rapporto (Implementazione mediante il pattern Factory Method),*

*#11 Radice quadrata (Implementazione mediante il pattern Factory Method),*

*#12 Inversione di segno (Implementazione mediante il pattern Factory Method);*

* Andrea Landi: *#2 Notazione operandi (Implementazione della classe ComplexNumber),*

*#7 Somma (Implementazione mediante il pattern Factory Method),*

*#8 Differenza (Implementazione mediante il pattern Factory Method),*

*#9 Prodotto (Implementazione mediante il pattern Factory Method),*

*#10 Rapporto (Implementazione mediante il pattern Factory Method),*

*#11 Radice quadrata (Implementazione mediante il pattern Factory Method),*

*#12 Inversione di segno (Implementazione mediante il pattern Factory Method);*

* Antonio Scaldaferri: *#1 Costruzione stack,*

*#5 Visualizzazione elementi,*

*#6 Interfaccia grafica,*

*#7 Somma (Implementazione mediante il pattern Factory Method),*

*#8 Differenza (Implementazione mediante il pattern Factory Method),*

*#9 Prodotto (Implementazione mediante il pattern Factory Method),*

*#10 Rapporto (Implementazione mediante il pattern Factory Method),*

*#11 Radice quadrata (Implementazione mediante il pattern Factory*

*Method),*

*#12 Inversione di segno (Implementazione mediante il pattern Factory Method);*

* Filippo Somma: *#2 Notazione operandi (Implementazione della notazione cartesiana per i*

*numeri complessi e della notazione per i numeri reali),*

*#3 Scelta operandi,*

*#4 Allocazione del risultato,*

*#5 Visualizzazione elementi,*

*#6 Interfaccia grafica,*

*#7 Somma (Implementare la somma degli ultimi due operandi),*

*#8 Differenza Implementare la differenza degli ultimi due operandi),*

*#9 Prodotto (Implementare il prodotto degli ultimi due operandi),*

*#10 Rapporto (Implementare il rapporto degli ultimi due operandi),*

*#11 Radice quadrata (Implementazione mediante il pattern Factory Method),*

*#12 Inversione di segno (Implementare l'inversione di segno dell'ultimo elemento inserito nello stack);*

**Sprint review:**

* **User stories complete alla fine di questo sprint:** *#1 Costruzione stack,*

*#2 Notazione operandi,*

*#3 Scelta operandi,*

*#4 Allocazione del risultato,*

*#5 Visualizzazione elementi,*

*#6 Interfaccia grafica,*

*#7 Somma,*

*#8 Differenza,*

*#9 Prodotto,*

*#10 Rapporto,*

*#11 Radice quadrata,*

*#12 Inversione di segno.*

* **Project Velocity misurata:** 33

**Retrospective:**

* **Keep doing:**

1. Continuare ad utilizzare Trello sfruttando i suoi Power Up in modo da esplicitare tutte le informazioni associate ad ogni user story per organizzare meglio il lavoro;
2. Continuare ad utilizzare GitHub in modo da dover effettuare poche revisioni in seguito ai vari merge.

* **Less of:** approfondire meglio il design delle classi prima di mettere mano al codice.
* **More of**: impiegare più tempo a ragionare sulle singole user stories poiché la stima effettuata in fase di pre-game non è stata accurata.

A parità di story points, infatti, i seguenti task han richiesto più lavoro rispetto ad altri:

1. Notazione operandi ( > 2 SP);
2. Scelta operandi ( < 2 SP);
3. Radice quadrata ( > 3 SP).

* **Start doing:** è necessario essere più dettagliati nei daily meeting nello specificare i task giornalieri.



