**BARBARA RATTO S4952202**

**1 Documentazione / Commenti**

1.a Generare la documentazione usando Doxygen e verificare se la documentazione prodotta permette di orientarsi nella struttura del progetto software. Indicare i punti che potrebbero essere migliorati.

Andando su ‘Class List’ vedo:

**Classe polygon**

Sono presenti due documentazioni diverse per la classe **polygon**: *Polygon* e *polygon*.

* *Polygon*: manca un breve riassunto che descriva la classe, però è possibile visualizzare la documentazione senza problemi (la documentazione viene generata correttamente)
* *polygon*: è presente il breve riassunto non è possibile vedere la documentazione

Quest’errore potrebbe essere dovuto al fatto che nel file polygon.cpp, alla riga 3 del codice, c’è scritto:

@file rectangle.cpp

invece che

@file polygon.cpp

La descrizione della funzione GetArea è *GETFUNCTIONS*, sarebbe da migliorare, scrivendo *get the area of the object.*

**Classe rectangle**

In generale la documentazione è fatta in modo preciso e ordinato, è sempre presente una breve spiegazione del compito svolto da ogni funzione.

Riporto però alcune imprecisioni

Nella breve descrizione della classe c’è scritto *the class rectangle is a poligonal figure with 4 sides and*… avrei scritto che la classe rectangle permette di lavorare con figure con 4 lati…

La breve spiegazione delle funzioni SetDim e GetDim è SETFUNCIONS e GETFUNCIONS, un po’ generica.

Avrei separato in modo più evidente le funzioni di tipo ‘virtual = 0’ dalle altre *access functions.*

Nell’ottica di una possibile aggiunta di altre funzioni virtual o virtual=0 penso sarebbe utile averle tutte insieme, per fare poi altri eventuali sottogruppi all’interno di questo blocco

Per il resto la documentazione è fatta bene

**Classe rhombus**

Stesse considerazioni fatte per la classe rectangle.

**Classe isoTriangle**

Stesse considerazioni fatte per la classe rectangle.

Andando invece in ‘File List’

**Main**

La documentazione è stata fatta in modo corretto.

1.b Utilizzando la documentazione e i commenti inseriti nei file .ccp e .h verificare se il progetto software aderisce alle specifiche assegnate durante la prima settimana. Indicare le eventuali specifiche non rispettate.

1.c Verificare che il file README introduca correttamente lo scopo del progetto software e che dia sufficienti informazioni per un corretto uso dell’interfaccia a riga di comando. Indicare eventuali mancanze e/o possibili migliorie.

**2 Compilazione e prima sessione di test**

2.a Verificare se è possibile compilare il progetto

2.b Test dell’interfaccia a riga di comando: l’interfaccia funziona correttamente? L’interfaccia è di facile utilizzo? Quali prove sono state eseguite per fare il test?

**3 Seconda sessione di test**

3.a Definire una procedura di test che permetta di identificare possibili bug nel codice prodotto per le singole classi e i singoli metodi di ogni classe

3.b Riportare i risultati del test

devo scrivere io il main