**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.

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

**Risultati del test**

1.a) La documentazione Doxygen è stata generata correttamente, ogni file presenta una descrizione per ogni funzione, grazie alla quale è possibile orientarsi più facilmente all’interno del progetto software. L’unica eccezione da considerarsi proprio una sottigliezza, consiglio una maggior attenzione alla personalizzazione della pagina html (magari inserendo il nome del progetto software). Per il resto trovo che sia documentata correttamente.

1.b) Le specifiche sono state rispettate correttamente: la classe RightTriangle() presenta i parametri base e altitude nella sezione protected, e la funzione GetHypotenuse() nella sezione public in maniera tale che possa essere chiamata all’occorrenza ( e non essendo un attributo come base e altitude ma una funzione non c’è pericolo che il parametro in se venga modificato). La funzione Draw() funziona correttamente e sfrutta a pieno il polimorfismo permettendo di visualizzare a schermo l’area e il perimetro del poligono richiesto, inoltre il codice implementa la possibilità di poter scegliere il poligono generato. È stata definita virtual =0 nella classe base astratta polygon, in maniera che venga obbligatoriamente ridefinita in ogni rispettiva classe derivata e ciò permetterà di implementare il polimorfismo (classe base e astratta).

1.c) Il file README, risulta essere preciso e puntuale sul definire lo scopo del progetto però risulta essere poco esaustivo sul fornire più informazioni per quanto riguarda il corretto uso dell’interfaccia a riga di comando. Consiglio di aggiungere informazioni più dettagliate sull’utilizzo del Doxygen e la compilazione del progetto software tramite riga di comando, in modo tale da renderlo più esaustivo e di facile utilizzo anche per utenti alle prime armi con c++ e la programmazione ad oggetti.

2.a) Il progetto al primo tentativo non si riesce a compilare per colpa dell’utilizzo della scanf\_s e la riga #pragma once all’interno dei file .h. Solo dopo aver modificato scanf\_s con scanf e dopo aver tolto #pragma once dai file .h, il progetto può essere compilato.

2.b) L’interfaccia risulta funzionante e di facile utilizzo. Come test ho selezionato la possibilità di scelta che il progetto software fornisce una volta eseguito, ovvero la possibilità di creare vari poligoni. L’unica miglioria da poter effettuare per quanto riguarda il controllo del carattere/numero selezionato da tastiera che non rientra nel range di scelta. Per un utente trovo che sia più usufruibile la possibilità di riselezionare una scelta senza interrompere definitivamente il progetto software, perciò, costringere l’utente a ricompilare tutto il progetto. Un ulteriore consiglio di miglioria riguarda la mancata implementazione del codice che gestisce la distruzione degli oggetti creati (manca implementazione del codice che chiama i distruttori delle classi).

3.a) Possibile procedura di test per identificare possibili bug del progetto software preso in analisi:

1. scrivo un main che andrà a testare la corretta creazione degli oggetti e testerà la corretta esecuzione delle funzioni implementate e verificare se non sono presenti bug dopo la creazione e test delle stesse.

2. un ulteriore possibilità di test riguarda la modifica dei parametri impostati nel codice e verificare se il codice crea dei bug.

Ho svolto il primo punto dove mi sono limitato a testare il corretto funzionamento della classe righttriangle. Lo stesso tipo di test è stato effettuato per tutte le classi e si è verificato il correttto funzionamento delle stesse.

NOTA🡪 sul main caricato su github non è stato riportato lo stesso procedimento di test per le altri classi poiché ricorsivo.