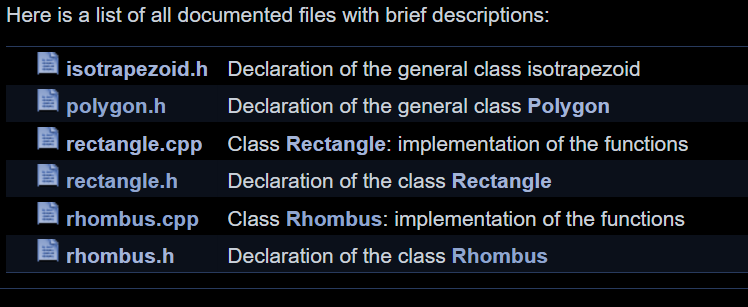
**1 Documentazione / Commenti**

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.

Le descrizioni delle classi mi sembra di facile comprensione e utilizzo. Guardando la lista dei file manca il file isotrapezoid.cpp per cui non è possibile visualizzarne il codice:



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.

Il progetto aderisce alle richieste della prima settimana in quanto:  
-presenta i necessari setters e getters per l’interazione con l’oggetto trapezio isoscele  
-presenta due funzioni Area e Perimeter (e di conseguenza GetPerimeter e GetArea) che permettono di ritornare i due valori correttamente e di impostare i valori delle variabili di classe base area e perimeter dell’oggetto

-presenta una funzione per l’impostazione della variabile e il “return” dalla funzione del valore del lato obliquo.

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.

Il file README.md è molto chiaro nella spiegazione dello scopo e delle classi. Non esplica l’utilizzo dell’interfaccia implementata dal main, poiché si concentra sull’applicazione e sul come utilizzare le classi in un proprio programma. Sarebbe quindi conveniente aggiungere una breve spiegazione dell’interfaccia (per quanto intuitivo) e, volendo rimanere nell’ottica usata da lui (cioè di utilizzare le classi) avrei forse aggiunto qualche elemento più specifico alla classe, anche se presente nella documentazione (quali funzioni utili all’utilizzatore della classe).

**2 Compilazione e prima sessione di test**

2.a Verificare se è possibile compilare il progetto

Non compila inizialmente per i seguenti motivi:

🡪L’utilizzo della scanf non è accettata dall’ambiente, ho quindi semplicemente modificato con scanf\_s.

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?

L’interfaccia funziona correttamente nell’interazione e nella stampa dei vari “draw” degli oggetti. Inoltre l’interfaccia è di facile utilizzo, eccetto per l’inserimento dei dati durante la creazione di una forma: Quando viene chiesto di inserire i dati è incerta la modalità di inserimento (problema minimale perché funziona correttamente in ogni caso)  
Provando a generare oggetti e a stamparli non sembrano apparire grandi problemi.  
Nel momento della creazione di forme, provando a creare oggetti inserendo dati in modi diversi (uno spazio tra un numero e un altro, oppure tornando più volte a capo o inserendo lettere al posto di numeri) si nota che spesso l’inserimento di dati non coerenti non è gestita propriamente. Per esempio l’inserimento di lettere causa il crash istantaneo del programma dopo la stampa, ripetuta numerose volte, del menù con diversi codici di errore, mentre per la creazione di trapezi isosceli bisogna inserire 3 dati al posto che due per far si che il programma accetti i primi 2 inseriti:

Immagine che contiene testo, schermata, Carattere, nero

Descrizione generata automaticamente

Il problema si presenta in maniera più grave nel caso in cui inserissimo una base non accettabile (bottom < top), caso in cui entra in un loop di warning da cui non si può uscire  
Immagine che contiene testo, schermata, Carattere, nero

Descrizione generata automaticamente

Immagine che contiene testo, schermata, software, schermo

Descrizione generata automaticamenteAlla chiusura del programma tramite l’opzione 5 del menù, gli oggetti creati se presenti vengono eliminati dalla memoria e si viene riportati nel menù. Se però dopo ciò si preme nuovamente l’opzione 5, viene generata un eccezione:

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamenteLa parte che lo generà sta nell’eliminazione da memoria degli oggetti creati:

Questo perché la terminazione del   
programma non è prevista: la prima volta che viene premuto 5 l’array viene azzerato ma il programma non termina. La seconda volta che 5 viene premuto, l’operazione del delete verrà compiuta su un oggetto che però non esiste, perciò su una parte di memoria già liberata dalla presenza di oggetti. Se si prova ad usare l’opzione 5 senza aver mai creato ne cancellato nulla (cioè all’avvio), il problema non si presenta perché l’array è stato inizializzato e l’opzione 5 ,anche se usata più volte, non intacca l’array, lasciandolo inizializzato come in partenza. Se questo viene invece fatto dopo aver creato qualcosa e successivamente cancellato, il delete verrà fatto su un elemento dell’array non inizializzato (appena “uscito” da un delete).  
Questo probabilmente causa l’eccezione presente.

C’è inoltre un errore nella gestione della variabile “choice” poiché, se si preme 5 per uscire, vengono eliminati tutti gli oggetti ma non si esce di per se dal programma, bensì si torna al menù. Per uscire sarebbe necessario ri-premere 5, ma se si prova a schiacciare un altro numero, essendo in quei casi presenti riferimenti a oggetti ormai eliminati in memoriaImmagine che contiene testo, schermata, Software multimediale, software

Descrizione generata automaticamente   
(Per esempio in questo caso dopo aver premuto 5 ed essere tornato nel menù, ho provato a mandare un draw che, non essendoci più objects[j], crea una eccezione) e causa il crash del programma

Immagine che contiene testo, schermata, Carattere, menu

Descrizione generata automaticamente

**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

N.B🡪 Nella classe isotrapezoid manca l’implementazione nel .cpp della funzione GetDim() presente invece nel .h, ho quindi commentato nel mio main(test) la parte di verifica della funzione GetDim();  
  
Il programma main(test) è quello che si occupa del check della classe IsoTrapezoid

La procedura di test è quella di testare ogni funzionalità in tutti i possibili casi di funzionamento (inserire valori non validi, aggiornare i parametri della classe e calcolare l’area, etc…)  
  
Commenti sono presenti direttamente nel main(test).

3.b Riportare i risultati del test

Polygon - constructor – default  
isotrapezoid - constructor - default

---isotrapezoid---

Base1 = 0

Base2 = 0

Height = 0

Perimeter = 0

Area = 0

🡪Il costruttore di default del trapezio crea correttamente l’oggetto poligono e trapezio.

🡪L’inizializzazione avviene correttamente

Polygon - constructor - default

isotrapezoid - constructor

WARNING: isotrapezoid - constructor: Bottom Side value should be > 0

WARNING: isotrapezoid - constructor: Top Side value should be > 0

WARNING: isotrapezoid - constructor: Height value should be > 0

---isotrapezoid---

Base1 = -1.07374e+08

Base2 = -1.07374e+08

Height = -1.07374e+08

Perimeter = 0

Area = 1.15292e+16

Polygon - constructor - default

isotrapezoid - constructor

Polygon - copy constructor

---isotrapezoid---

Base1 = 2

Base2 = 2

Height = 2

Perimeter = 8

Area = 4

WARNING: isotrapezoid - SetBase2: Base2 (Top Side) should be < than Base1 (Bottom Side)

---isotrapezoid---

Base1 = 1

Base2 = 0

Height = 3

Perimeter = 7.08276

Area = 1.5

---isotrapezoid---

Base1 = 3

Base2 = 2

Height = 1

Perimeter = 7.23607

Area = 2.5

WARNING: isotrapezoid - SetBase2: Base2 (Top Side) should be < than Base1 (Bottom Side)

A s' Base1 measures: --> 3

A s' Base1 measures: --> 5

A s' Base2 measures: --> 5

A s' Height measures: --> 5

🡪La costruzione con parametri NON validi del trapezio costruisce per default il poligono correttamente, ma (poiché i valori non vengono inizializzati data la loro invalidità) il dump successivo suggerisce che i valori mostrati sono il contenuto randomico in quelle locazioni di memoria non inizializzate.

--------------------------------------------------------

🡪Il costruttore di copia avviene correttamente in quanto i valori coincidono

🡪L’uso di funzione SetDim (che richiama direttamente le funzioni SetBase1 e 2 e Set Height) con valori non validi porta alla notificazione dell’errore dalla funzione SetBase2 (deve essere minore della base1).

Di conseguenza la stampa

🡪L’uso corretto di SetDim è da questo dump confermato.

🡪Il set base1 con valore non valido riporta correttamente l’errore (nelle due funzioni nella stampa del messaggio di errore c’è un errore perché nella funzione SetBase1 il warning restituisce “WARNING: isotrapezoid – SetBase2…, l ‘ho modificato per migliore comprensione”. Il valore quindi non viene aggiornato ma rimane invariato.

🡪L’uso corretto di SetBase1 e 2 e di height è gestito correttamente e inoltre le Get funzionano anch’esse nel modo corretto.

A s' side measures: --> 5

A s' perimeter and area --> 20 & 25

drawing isotrapezoid:

Top Side = 5

Bottom Side = 5  
Diagonal Side = 5  
Area = 25

Perimeter = 20

---isotrapezoid---

Base1 = 5

Base2 = 5

Height = 5

Perimeter = 20

Area = 25

WARNING: isotrapezoid – SetBase1: Base2 (Top Side) should be < than Base1 (Bottom Side)

---isotrapezoid---

Base1 = 2

Base2 = 2

Height = 2

Perimeter = 8

Area = 4

isotrapezoid - operator =

---isotrapezoid---

Base1 = 6

Base2 = 1

Height = 1

Perimeter = 12.3852

Area = 3.5

isotrapezoid - operator ==

Eseguito D=A in modo corretto

isotrapezoid - destructor

Polygon - destructor

isotrapezoid - destructor

Polygon - destructor

isotrapezoid - destructor

Polygon - destructor

🡪GetSide funziona correttamente

🡪GetPerimeter e GetArea corretti

🡪Il check della funzione draw riconferma i valori osservati prima attraverso le funzioni get e, a loro volta, coincidono con quelle del dump

🡪Questo errore è dato dal fatto che (se svolgiamo una SetDim: Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente  
la Base1 viene (per controllare che sia maggiore della B2) viene controllata con quella PRECEDENTE (nel nostro esempio A.Base2 misura 5).Immagine che contiene testo, Carattere, linea, schermata

Descrizione generata automaticamente

Facilmente risolvibile gestendo diversamente SetDim.

🡪 Viene prima stampata, per il check degli operatori = e ==, l’oggetto D.  
Questo viene successivamente modificato tramite una copia D=A. Si nota che

I Dati stampati dal secondo Dump coincidono con quelli settati in A.SetDim(6,1,1).  
Il 6 è presente per ovviare all’errore precedentemente trovato.

Il confronto avviene infine correttamente.

🡪IL distruttore funziona infine correttamente.

isotrapezoid - destructor

Polygon - destructor