

Projecto Manipulação de números complexos

Relatório do projecto

Nº aluno: a21000662

Elaborado por: Paulo Fonseca da Silva

Em: Junho 2011

Revisão: 0

Contactos:

Telf.: +351 919861565

Email:

paulosilva@biglevel.pt

Site:

—

Manipulação de números complexos

Índice

Introdução:	3
1. Requisitos Funcionais:	4
2. Requisitos não Funcionais:	5
2.1 Módulo main.c:	5
2.2 Módulo calculadora.c:	5
2.3 Módulo complexo.c:	6
2.4 Módulo complexo.h:	6
3. Referências Bibliográficas:	7
Documentos relacionados:	8

Manipulação de números complexos

Introdução:

O objectivo do presente documento é descrever os requisitos funcionais e não funcionais, necessários para o desenvolvimento do projecto de programação ("Manipulação de números complexos").

Pretende-se a codificação de um programa em linguagem C para manipulação de números complexos. Genericamente o programa deverá implementar uma calculadora para números complexos que, além das operações usuais, terá ainda a funcionalidade de testar conhecimentos.

O programa tem uma parte que é uma calculadora de números complexos. A calculadora deve ter o número complexo actual (que representa o visor das calculadoras "normais") e deve ser possível indicar a operação pretendida e um número complexo. O resultado da operação fica no número complexo actual (visor). As operações que a calculadora sabe realizar são: limpar o visor, simétrico, conjugado, potenciação, inverso, adição, subtracção, multiplicação e divisão. O programa tem ainda uma parte que é o teste de conhecimentos. Aqui são feitas várias perguntas sobre operações com números complexos (as mesmas definidas para a calculadora) e o utilizador responde indicando um número complexo. O utilizador deve poder escolher o número de perguntas do teste. No fim do teste o utilizador é informado do número de respostas certas e erradas, e das respectivas percentagens.

Revisão Nº: 0 Data:13-06-2011	Elaborado Por: Paulo Silva _____	Aprovado Por: _____	Página:3/8
----------------------------------	-------------------------------------	------------------------	------------

Manipulação de números complexos

1. Requisitos Funcionais:

- ❑ Módulo Principal com menu de acesso às várias opções:

```
***** ULP - PROJECTO DE PROGRAMACAO I *****
*   N U M E R O S   C O M P L E X O S   *
      1 - Modo calculadora.
      2 - Teste de conhecimentos.
      3 - AJUDA ?
      0 - SAIR.
DIGITE SUA OPCAO:
```

Fig. 1 – Exemplo do menu do projecto

- Gerir um menu, com acesso às várias opções dos outros módulos (com uma tecla de acesso), e com uma determinada tecla para sair (Zero) – (ver fig. 1.)

- ❑ Módulo de Calculadora

```
***** ULP - PROJECTO DE PROGRAMACAO I *****
                      C A L C U L A D O R A
-----
                      Z1 = 2 + 2i
-----
1 - Soma.                2 - Subtracao.
3 - Multiplicacao.       4 - Divisao.
5 - Simetrico.           6 - Conjugado.
7 - Inverso.             8 - Potenciacao.
9 - Limpar visor.
0 - Retroceder.
-----
DIGITE SUA OPCAO:
```

Fig. 2 – Exemplo do menu de calculadora

- Permite realizar operações aritméticas simples (soma, subtração, multiplicação, divisão etc..) de números complexos (modo rectangular)
- Limpar visor

- ❑ Teste de conhecimentos

```
***** ULP - PROJECTO DE PROGRAMACAO I *****
          T E S T E   C O N H E C I M E N T O S
BOA SORTE.: Paulo Silva : pergunta.: 1 de 2
-----
                      Z1 = 2 + 2i
-----
1 - Soma.                2 - Subtracao.
3 - Multiplicacao.       4 - Divisao.
5 - Simetrico.           6 - Conjugado.
7 - Inverso.             8 - Potenciacao.
9 - Limpar visor.
-----
DIGITE SUA OPCAO:
```

Fig. 3 – Exemplo do menu de calculadora (teste de conhecimentos)

- Permite realizar operações aritméticas simples (soma, subtração, multiplicação, divisão etc..) de números complexos (modo rectangular)
- Neste ecrã o utilizador introduz as respostas para cada operação escolhida, e no fim obtém o resultado das respostas dadas (media aritmética)
- Limpar visor

Manipulação de números complexos

2. Requisitos não Funcionais:

- ❑ A organização do projecto (no DevC++) teve em conta:
 - A criação de 4 módulos (source files) principais: (ver fig. 4.)
 - o main.c
 - o calculadora.c
 - o complexo.c
 - o complexo.h

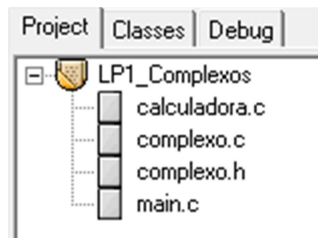
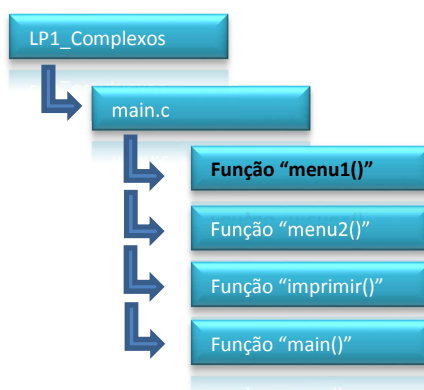


Fig. 4 – Exemplo da organização do projecto no DevC++

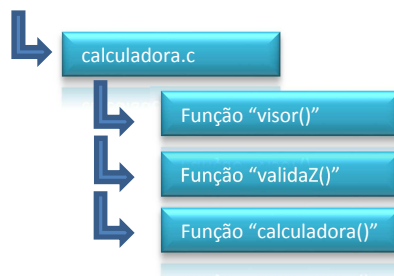
- ❑ Cada um destes módulos contem as Funções C necessárias para responder aos Requisitos Funcionais: uma função para cada opção do menu:

2.1 Módulo main.c:



Resumo: Este módulo resolve a primeira parte do problema, que é disponibilizar ao utilizador um menu1 (menu inicial) do programa com as opções descritas anteriormente (fig.1) onde pode avançar para um segundo menu, com as opções aritméticas já referidas, repetidamente até este escolher zero, opção de saída da aplicação. É composto também por uma estrutura condicional que dependendo da escolha do utilizador ira invocar uma função por opção, para responder aos requisitos funcionais de cada uma. É composto por uma função imprimir, responsável por ir actualizando os resultados no visor da calculadora.

2.2 Módulo calculadora.c:



Resumo: É neste módulo que encontramos a chamada a todas as funções de cálculo descritas no menu2. É composto também por uma estrutura condicional que dependendo da escolha do utilizador ira invocar uma função por opção, para responder aos requisitos funcionais de cada uma. Controla também a introdução do numero complexo na forma rectangular por parte do utilizador.

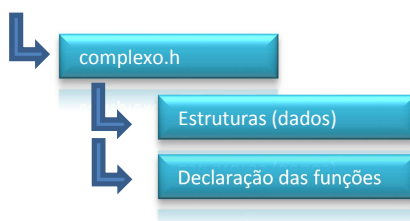
Manipulação de números complexos

2.3 Módulo complexo.c:



Resumo: É neste módulo que encontramos a todas as funções de cálculo descritas no menu2. Tem também uma função que faz a notação matemática de todos os números complexos na forma rectangular.

2.1 Módulo complexo.h:



Resumo: Este módulo é o header do programa, nele encontramos a declaração de todas as funções de cálculo descritas no menu2. Também é nele que defini as estruturas de dados que utilizo em toda aplicação (complexo e resultado).

- Complexo
 - typedef struct{
 - double r; // Parte Real
 - double i; // Parte imaginária
 - }complexo;
- Resultado – para guardar a resposta do utilizador na opção de auto-avaliação
 - typedef struct{
 - char resposta[80];
 - int pontos;
 - }resultado;

Manipulação de números complexos

Conclusões:

Em suma a solução deste desafio, passou pela capacidade de dividir o problema em vários sub-problemas, desta forma foi possível tratar cada um individualmente e avaliar a relação uns com os outros; depois a forma como organizarmos esses sub-algoritmos em módulos no C, passa também mais uma vez por uma estruturação do código por forma a ser mais simples resolver problemas que surjam ao nível da sintaxe. A utilização de funções, passa também por essa reestruturação e também a reutilização do código em várias zonas do programa sem termos de voltar a reescrever a rotina toda novamente.

3. Referências Bibliográficas:

http://editthis.info/images/takai/5/50/EDI_Aula06.pdf

Manipulação de números complexos

Documentos relacionados: