



PCS3111

Laboratório de Programação Orientada a Objetos para Engenharia Elétrica

Teste e depuração

Escola Politécnica da Universidade de São Paulo

Testes e depuração

- Definições
 - **Teste**: processo de executar um programa com o objetivo de encontrar defeitos
 - **Depuração**: processo de localizar um *suposto* defeito e corrigi-lo
- Saídas corretas em um teste não *garantem* que o software não tem defeitos
 - O teste pode não ter sido bom o suficiente!
- Existem abordagens de teste e de depuração
 - Engenharia de Software

Exemplo: total de uma compra

- Se a soma dos preços dos produtos for maior que "limite" reais, o frete não deve ser cobrado

```
11 double totalDaCompra (double produtos[], int quantidade,  
12                        double frete, double limite) {  
13     double total = 0;  
14     for (int i = 0; i < quantidade; i++)  
15         total = total + produtos[i];  
16     if (total > limite)  
17         total = total + frete;  
18     return total;  
19 }
```

EX08

- Como fazer testes?

Exemplo: total de uma compra

- Se a soma dos preços dos produtos for maior que "limite" reais, o frete não deve ser cobrado

```
11 double totalDaCompra (double produtos[], int quantidade,  
12                        double frete, double limite) {  
13     double total = 0;  
14     for (int i = 0; i < quantidade; i++)  
15         total = total + produtos[i];  
16     if (total > limite)  
17         total = total + frete;  
18     return total;  
19 }
```

EX08

```
21 int main() {  
22     double produtos[] = {5, 10, 5, 10};  
23     double total = totalDaCompra (produtos, 4, 5.5, 25);  
24  
25     if (total != 30) cout << "Erro: " << total << endl;  
...
```

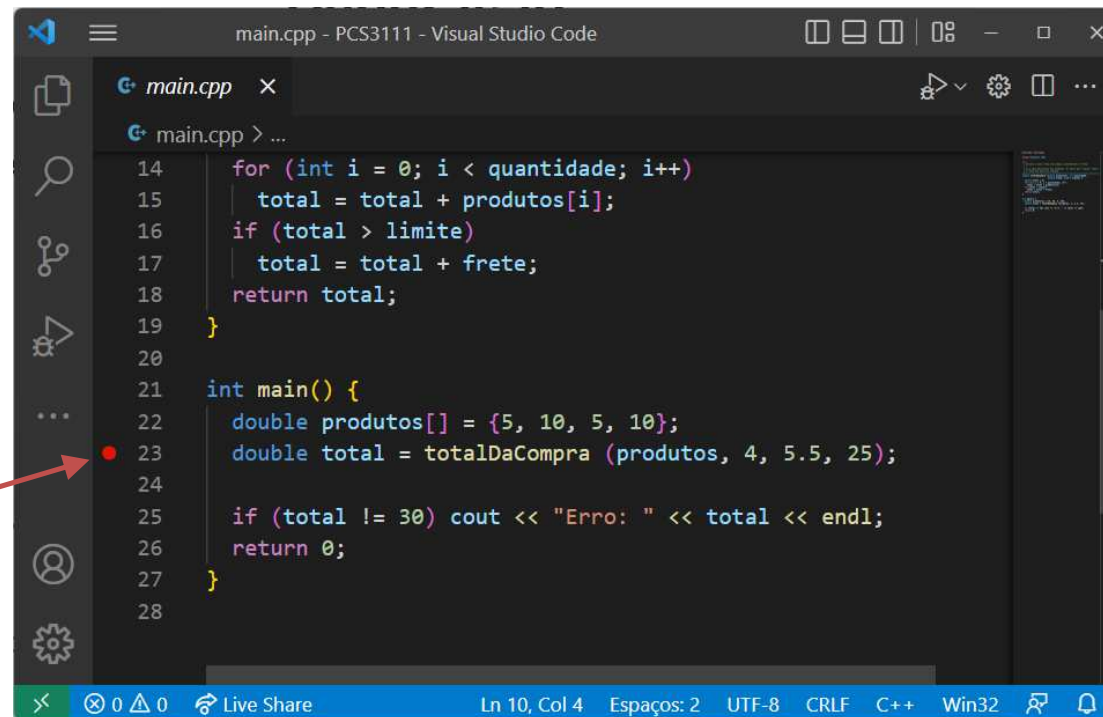
Saída

Erro: 35.5

Teste

Depuração

- *Breakpoint*
 - Ponto de parada no programa para depuração



```
main.cpp - PCS3111 - Visual Studio Code
main.cpp x
main.cpp > ...
14   for (int i = 0; i < quantidade; i++)
15       total = total + produtos[i];
16   if (total > limite)
17       total = total + frete;
18   return total;
19 }
20
21 int main() {
22     double produtos[] = {5, 10, 5, 10};
23     double total = totalDaCompra (produtos, 4, 5.5, 25);
24
25     if (total != 30) cout << "Erro: " << total << endl;
26     return 0;
27 }
28
```

Clique à esquerda do 23

Depuração

■ Controle da depuração

Executar até o próximo *breakpoint*

Ir para a próxima linha

Sair da função atual

Parar a execução

Valores das variáveis

```
main.cpp - PCS3111 - Visual Studio Code
EXECUTAR E DE... g++ build and debug active file (configuraçõ... Live Share CRLF C++ Win32

VARIÁVEIS
  Locals
    > produtos: [4]
    total: 3.9525251667299724e-323
  Registers

INSPEÇÃO

PILHA DE CHAMADAS
  [1]
    main()
      EM PAUSA EM BREAKPOINT
      main.cpp 23:1

19 }
20
21 int main() {
22     double produtos[] = {5, 1
23     double total = totalDaCon
24
25     if (total != 30) cout <<
26     return 0;
27 }
28
```

Observação

- Nunca compare `double` com `==` ou `!=`
 - Pontos flutuantes tem *problemas de precisão*

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
..
21
22 int main() {
23     double produtos[] = {5, 10, 5, 10};
24     double total = totalDaCompra (produtos, 4, 5.5, 25);
25
26     if (abs(total - 30) > 0.01) cout << "Erro: " << total << endl;
27     return 0;
28 }
```

Inclua `cmath` para usar o `abs`
e faça `using namespace std`

EX08b

Se a diferença entre o valor esperado e o obtido for maior que um *épsilon*, há um erro

Bibliografia

- MYERS, G. J. **The Art of Software Testing**. John Wiley & Sons, 2ª edição, 2004.
- SAVITCH, W. **C++ Absoluto**. Pearson, 1st ed. 2003. Seção 10.1.