

Pré-condições e Pós-condições

- ❑ Nesta aula são descritas **pré-condições** e **pós-condições**
 - ❑ Elas permitem que um programador especifique o que um procedimento deve realizar
-

Pré-condições e Pós-condições

- ❑ Frequentemente um programador necessita comunicar precisamente o que um procedimento deve realizar, sem qualquer indicação de como esta tarefa deve ser realizada
- ❑ Você pode imaginar uma situação onde isto pode ocorrer?

Exemplo

- ❑ Você é o chefe de um grupo de programadores e você deseja que um dos seus programadores escreva um procedimento como parte de um projeto



*Aqui estão os requisitos
para um procedimento
que eu desejo que você
escreva*

*Eu não me importo
qual método o
procedimento utilizará,
desde que os
requisitos
sejam respeitados*

O que são Pré-condições e Pós-condições?

- ❑ Uma forma de especificar tais requisitos é por meio de um par de declarações sobre o procedimento
- ❑ A declaração de **pré-condição** indica o que deve ser verdade antes que o procedimento seja chamado
- ❑ A declaração de **pós-condição** indica o que será verdade quando o procedimento terminar o seu trabalho

Exemplo

```
void writeSquareRoot(float Number)
// Pré-condição: Number >= 0
// Pós-condição: A raiz quadrada de Number deve
// ser escrita
{
    ...
}
```

Exemplo

```
void writeSquareRoot(float Number)
// Pré-condição: Number >= 0
// Pós-condição: A raiz quadrada de Number deve
// ser escrita
{
```

- A pré-condição e pós-condição aparecem como comentários no seu programa
- Elas são frequentemente colocadas após o cabeçalho do procedimento

Exemplo

```
void writeSquareRoot(float Number)
// Pré-condição: Number >= 0
// Pós-condição: A raiz quadrada de Number deve
// ser escrita
{
```

Neste exemplo, a pré-condição
requer que

Number >= 0

seja verdade sempre que
o procedimento for chamado

Exemplo

Quais destas chamadas de procedimento respeitam a pré-condição?

```
writeSquareRoot( -10 );  
writeSquareRoot( 0 );  
writeSquareRoot( 5.6 );
```


Exemplo

Quais destas chamadas de procedimento respeitam a pré-condição?



```
writeSquareRoot( -10 );
```






```
writeSquareRoot( 0 );
```

```
writeSquareRoot( 5.6 );
```

A segunda e terceira chamadas estão corretas, uma vez que seus argumentos são maiores ou iguais a zero

Exemplo

Quais destas chamadas de procedimento respeitam a pré-condição?

-  `writeSquareRoot(-10);`
-  `writeSquareRoot(0);`
-  `writeSquareRoot(5.6);`

Mas a primeira chamada viola a pré-condição, uma vez que seu argumento é menor que zero

Exemplo

```
void writeSquareRoot(float Number)
// Pré-condição: Number >= 0
// Pós-condição: A raiz quadrada de Number deve
// ser escrita
{
```

A pós-condição sempre indica qual trabalho deve ser realizado pelo procedimento. Neste caso, quando o procedimento retorna, a raiz quadrada de **Number** deve ser escrita

Outro Exemplo

```
bool vowel(char Letter)
// Pré-condição: Letter é uma letra maiúscula ou
// uma letra minúscula (no intervalo 'A' .. 'Z' ou 'a' .. 'z')
// Pós-condição: O valor retornado pela função
// é verdade se Letter é uma vogal; caso-contrário
// o valor retornado pela função é falso
{
    ...
}
```

Outro Exemplo

Quais valores serão retornados por estas chamadas da função?

```
vowel( 'A' )  
vowel( 'Z' )  
vowel( '?' )
```

Outro Exemplo

Quais valores serão retornados por estas chamadas da função?

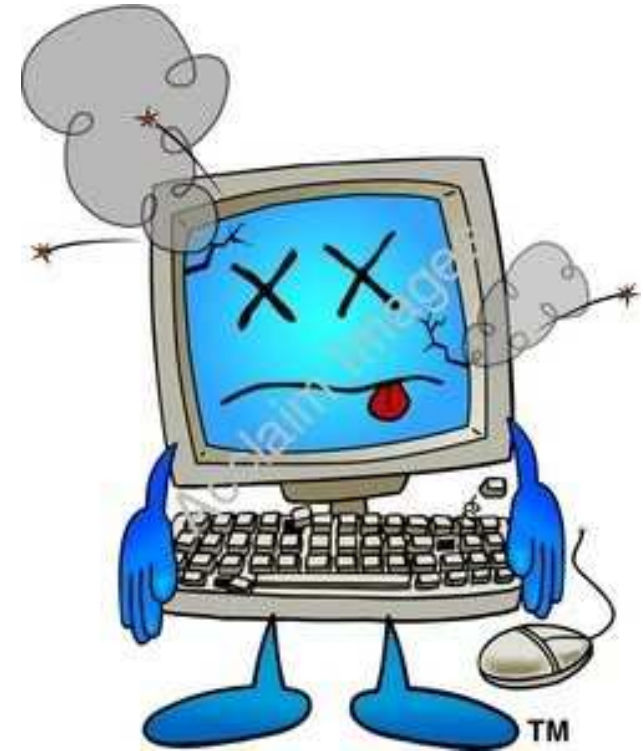
vowel('A')	true
vowel('Z')	false
vowel('?')	

Ninguém sabe, pois a pré-condição está sendo violada.

Outro Exemplo

Quais valores serão retornados por estas chamadas da função?

vowel('?')



A violação de uma pré-condição pode até mesmo resultar em efeitos inesperados

Sempre certifique que a pré-condição seja válida...

- ❑ O programador que chama a função ou procedimento é responsável por **certificar que a pré-condição é válida** quando o procedimento é chamado

Neste ponto, meu programa chama o seu procedimento e eu tenho certeza que a pré-condição é válida



...então a pós-condição se torna verdade ao fim do procedimento

- ❑ O programador que escreve a função ou procedimento assume que a pré-condição é válida, e certifica que a pós-condição se torna verdade ao fim do procedimento

Então meu procedimento será executado e quando isto é feito, a pós-condição será verdade, eu garanto



Questão

☐ Suponha que você chame um procedimento, e você não se certifica que a pré-condição é verdadeira. Quem é responsável caso esta falha cause algum tipo de desastre?

- ❶ Você
- ❷ O programador que escreveu o procedimento
- ❸ Ninguém

Questão

❑ Suponha que você chame um procedimento, e você não se certifica que a pré-condição é verdadeira. Quem é responsável caso esta falha cause algum tipo de desastre?

① Você

O programador que chama um procedimento é responsável por certificar que a pré-condição é válida

Por outro lado, programadores cuidadosos também seguem as seguintes regras:

- ❑ Quando você escreve um procedimento, você deve detectar em quais condições uma pré-condição será violada
- ❑ Se você detectar que uma pré-condição foi violada, então imprima uma mensagem de erro e aborte o programa...

Por outro lado, programadores cuidadosos também seguem as seguintes regras:

- ❑ Quando você escreve um procedimento, você deve detectar em quais condições uma pré-condição será violada
- ❑ Se você detectar que uma pré-condição foi violada, então imprima uma mensagem de erro e aborte o programa...
- ❑ ...em vez de causar um desastre



Exemplo

```
void writeSquareRoot(float Number)
// Pré-condição: Number >= 0
// Pós-condição: A raiz quadrada de Number deve
// ser escrita
{
    if (Number < 0)           // Pré-condição falhou
    { cout << "ERROR: Number is negative" << endl;
      abort(); // Isto para um programa em C++
    }

    ...
}
```

Vantagens do Uso de Pré-condições e Pós-condições

- ❑ Resumidamente descrevem o comportamento de uma função ou procedimento...
 - ...sem entrar em detalhes de implementação
- ❑ Posteriormente, você pode reimplementar o procedimento de uma nova forma...
 - ...entretanto os programas (os quais dependem exclusivamente da pré-condição/pós-condição) continuarão a funcionar sem alterações

Resumo

Pré-condição

- ❑ O programador que chama o procedimento certifica que a pré-condição é válida
- ❑ O programador que escreve um procedimento assume que a pré-condição é verdade quando o procedimento inicia sua execução

Pós-condição

- ❑ O programador que escreve um procedimento certifica que a pós-condição é verdadeira ao término do procedimento