Lista 8 (EXERCÍCIOS)

1) Qual a diferença entre o mecanismo de exceção implementado na linguagem C++ e o demonstrado em sala de aula na linguagem C?

Dica: o mecanismo em C é incompleto.

2) Escreva um main genérico que consiga capturar qualquer tipo de exceção.

3) Implemente uma classe própria que gerencie exceções, especializando a classe std::exception.

4) Prove que os objetos de exceções (aqueles que são lançados) são propriamente destruídos.

Dica: é possível aproveitar o exercício anterior.

5) Prove que os objetos de exceções criados na heap e lançados pelos seus ponteiros não são propriamente destruídos.

6) Escreva uma função que pode lançar quatro tipos de exceções: um char, um int, um bool e uma classe criada por você. No main, chame a função de maneira a lançar exceções e, utilizando catch, verifique essas exceções. Deriva sua classe de exceções da classe padrão. Escreva a função de tal maneira que o sistema se recupere e tente novamente.

7) Modifique sua solução para o exercício anterior para lançar um double de dentro da função, violando a especificação. Capture a exceção com seu gerenciador de exceções inesperadas e finalize o programa graciosamente (sem chamar abort()).

8) Reescreva a sua solução de lista encadeada simples para lançar exceções quando não for possível remover um elemento (inexistente). Utilize a exceção std::out_of_range.

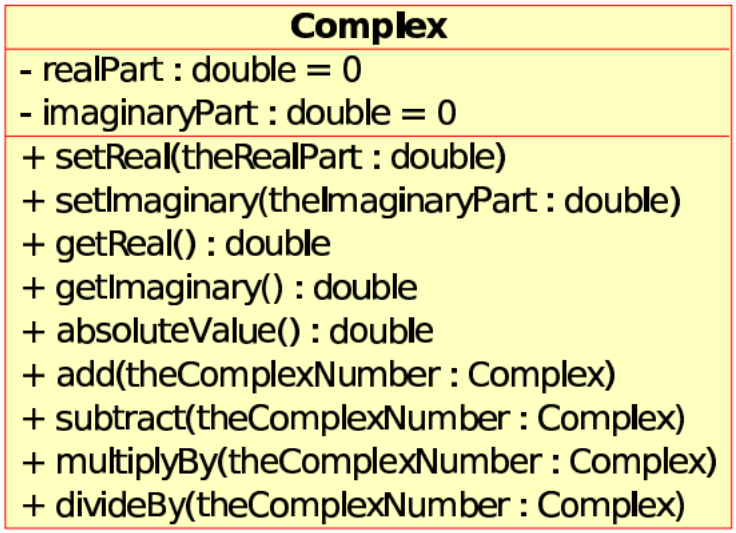
9) Crie uma classe UInt32 para representar um inteiro não assinalado de 32 bits. Sua classe deve se comportar como o tipo unsigned int e deverá lançar exceção quando operações de soma ou subtração incorrerem em overflow ou underflow.

10) Realize o teste de custo computacional das exceções. Implemente uma função que realize alguma tarefa onde é possível ocorrer um erro. Escreva duas implementações dessa função. Na primeira, realize o tratamento do erro sem utilizar exceções. Na segunda, utilize exceções.

11) É possível a ocorrência de uma exceção dentro de um catch?

12) Escreva um esboço do diagrama UML de sua APS. O seu projeto atende os princípios S.O.L.I.D. ?

13) Para o projeto de uma calculadora é necessário que esta trabalhe com números complexos. Bob, o engenheiro de software responsável pela arquitetura do projeto, designou as tarefas e passou para você a implementação da classe base que representa números complexos. Implemente esta classe de acordo com seu diagrama:



Bob chamou sua atenção para o caso em que a divisão possa ter denominador nulo.

14) A partir do diagrama de classes de exemplo UML para veículos, implemente o código das classes. Compare com o código do exemplo.

15) Represente em UML a relação entre uma classe triângulo e a classe pontos.

16) Em UML represente os fatos: a) gato TEM quatro pernas; b) gato É um animal; c) gato CONHECE cachorro; d) gato DEPENDE de comida.