Programação Orientada a Objetos – POO

Prof. Pedro Carlos da Silva Lara
pcslara@lncc.br
home page: www.lncc.br/∼pcslara
Instituto Superior de Tecnologia em Ciências da Computação de Petrópolis

Lista de Exercícios 4

Sobrecarga de Operadores e Templates

1) Crie uma classe denominada `Racional` que abstrai a aritmética de números racionais. Sobrecarregue os seguintes operadores: `+, -, *, /, ==, !=, +=, -=, <<, >>`.

2) Crie uma classe denominada `String` que implementa uma conjunto de caracteres de forma conveniente. Sobrecarregue os operadores: `+, +=, ==, !=, [], <<, >>`.

3) Desenvolva a classe `Polinomio`. A representação interna de um `Polinomio` é um *array* de **termos**. Cada termo contém um **coeficiente** e um **expoente**. Por exemplo, o termo

$$5x^3$$

tem coeficiente 5 e expoente 3. Desenvolva uma classe completa contendo funções construtoras e destrutoras adequadas, bem como as funções `get` e `set`. A classe também deve fornecer as seguintes capacidades de operador sobrecarregado:

- Sobrecarregar o operador adição: `+`.
- Sobrecarregar o operador subtração: `-`.
- Sobrecarregar o operador multiplicação: `*`.
- Sobrecarregar o operador comparação: `==`.
- Sobrecarregar o operador comparação: `!=`.
- Sobrecarregar o operador de avaliação: `()`.

4) Usando o conceito de *templates*, crie uma classe que implementa uma matriz denomnada `Matriz`. O tipo de dado dos elementos da matriz será definido usando templates. Sobrescreva os operadores `+, -, *, ==, !=, ()`. O operador `()` será usado para acessar o elemento em uma determinada posição `i,j`. Por exemplo:

```
Matriz < float > M(4,4);   // Cria uma matriz de float (4x4)
M(0,1) = 2.3;              // M[0][1] = 2.3;
M(0,2) = 3.7;              // M[0][2] = 2.7;
.
.
.
```

5) Usando a classe `map` feita em sala de aula crie uma classe denominada `Agenda` que herda de `map` usando os templates: `Pessoa` e `std::string`. Assim a chave do `map` será uma pessoa e o seu telefone será uma `std::string`. Crie os métodos que possam inserir contato, buscar contato, remover contato, modificar contato. A herança será realizada da seguinte forma:

```
class Agenda : public map< Pessoa, string> {
...
};
```