

Programação Orientada a Objetos

Trabalho Prático 2

Data da entrega: 05/11/2017

Valor: 25 pts

1. Orientações Gerais

O trabalho pode ser realizado em grupos de dois alunos ou individualmente e deve ser entregue por e-mail até às 23:59 da data definida para entrega através do Moodle. É permitida a discussão de estratégias com os colegas, porém a implementação somente deve ser feita pelos integrantes do grupo. Cópias de programas (ainda que parciais) não serão aceitas. Entregas com atraso poderão ser aceitas pelo professor, porém terão descontos progressivos. Deve ser postado apenas um arquivo compactado contendo as soluções e documentação de todos os exercícios.

2. Objetivo

Implementar as classes e programas a seguir em linguagem C++ ANSI. Para cada questão deve ser fornecido um arquivo compactado contendo códigos e documentação, se necessário.

2.1. A Good Programmers lançou no mercado uma classe com o nome Ypto. Os engenheiros da Good Programmers optaram por declarar os atributos da classe como privados, sendo o acesso feito usando métodos get()/set():

```
// Ypto.h
class Ypto {
private:
    int a;
    int b;
    int c;
public:
    Ypto (int a1=0, int b1=0, int c1=0) : a(a1), b(b1),
    c(c1) {};
    int produto() const { return (a*b*c); };
    int soma() const { return (a+b+c); };
    int getA() const { return a; };
    int getB() const { return b; };
    int getC() const { return c; };
    void setA(int a1) { a=a1; };
    void setB(int b1) { b=b1; };
    void setC(int c1) { c=c1; };
};
```

Vários programadores adquiriram a classe Ypto e utilizaram-na nos seus programas. Um desses programas, designado por Exemplo2, é indicado a seguir:

```
// Arquivo: Exemplo2.cpp
#include "Ypto.h"
#include <iostream>
using namespace std;
int main()
{
    Ypto x1;
    Ypto x2(5, 6, 7);
    x1.setA(1);
    x1.setB(2);
    cout << x1.soma();
    cout << x2.soma();
    x1.setC(x2.getA());
    x2.setA(x2.getA()+1);
    cout << x1.getC();
    cout << x2.getA();
    //.....
}
```

Os engenheiros da Good Programmers concluíram que seria muito melhor se os 3 valores fossem guardados num array, de maneira tal que seja possível incluir quantos valores forem necessários, removendo assim a restrição a 3 valores, e por isso lançaram um update da classe Ypto.

- Tendo em conta esta alteração modifique a classe Ypto para mantê-la funcional.
- Indique quais alterações o cliente teria de efetuar no seu programa Exemplo2.
- Implemente um tratamento de exceções adequado para a classe Ypto modificada. Para isso você deve definir (pelo menos) uma classe de exceção.

2.2. Sabe-se que a linguagem C++ não possui em seu arcabouço uma classe para tratamento de datas. Eventualmente programadores necessitam manipular datas em seus programas. Sabendo disso implemente uma classe **data**, herdada a partir da classe DataInterface fornecida a seguir, que atenda aos seguintes requisitos:

- Deve ser possível a definição do formato da data no construtor da classe;
- O separador dos campos da data pode ser tanto / quanto – e também ser definidos no construtor da classe;
- A classe deve instanciar objetos de data nos seguintes formatos:
 - dd/mm/aaaa (dia/mês/ano)
 - mm/dd/aaaa (mês/dia/ano)
 - aaaa-mm-dd (ano/mês/dia)
- A classe deve implementar os seguintes serviços:
 - Somar/subtrair dias a uma data
 - Somar/subtrair meses a uma data
 - Somar/subtrair anos a uma data
 - Calcular a diferença em dias entre duas datas
 - Retornar a data em um formato passado como parâmetro

- Retornar data por extenso
- A classe deve também sobrecarregar os operadores abaixo:
 - Operator= (verifica se duas datas são iguais independentemente do formato)

Após a implementação da classe data defina um programa que executa todas as funcionalidades da mesma.

```
// DataInterface.h
class DataInterface {
protected:
    int dia;
    int mes;
    int ano;
    string formato;
    char separador = '/';
public:
    DataInterface(string p_formato, string p_data) :
    formato(p_formato) {
        int i = 0;
        string sDia, sMes, sAno;
        while ( (i<p_data.size()) && (p_data[i]!=separador) ) {
            sDia.push_back(p_data[i]);
            i++;
        }
        dia = stoi(sDia);
        i++;
        while ( (i<p_data.size()) && (p_data[i]!=separador) ) {
            sMes.push_back(p_data[i]);
            i++;
        }
        mes = stoi(sMes);
        i++;
        while ( (i<p_data.size()) && (p_data[i]!=separador) ) {
            sAno.push_back(p_data[i]);
            i++;
        }
        ano = stoi(sAno);
    }
    ~DataInterface() { };
    virtual string getData() = 0;
};
```