

 Universidade do Minho	Escola de Engenharia da Universidade do Minho Mestrado Integrado em Eng. Electrónica Industrial e Computadores Complementos de Programação de Computadores	2014/2015 MIEEIC (1º Ano) 2º Sem
Luís Paulo Reis		
Aula Prática 2: Exercícios de Introdução ao C++		

Objectivos:

Esta Folha de Exercícios destina-se a:

- Facilitar a transição do ambiente de programação da linguagem C para o da linguagem C++;
- Construir as primeiras aplicações em linguagem C++.

Os exercícios aqui propostos deverão ser realizados no mais simples ambiente de desenvolvimento possível para a linguagem C: editor de texto de programação ou editor DevC++ e ferramentas da GCC (GNU Compiler Collection) e afins.

Exercício 1

Pretende construir-se uma classe **CAlunoProg** em C++ que permita guardar a identificação e gerir as notas de um aluno de Programação.

A classe deve definir atributos privados que permitam:

- Guardar o nome, `string`, e o número, `int`, do aluno (`nomeAluno` e `numAluno`);
- Guardar as notas, `double` (entre 0..20), da avaliação intercalar e final do trabalho (`trabInt`, `trabFin`);
- Guardar as notas, `double` (entre 0..20), dos 2 mini-testes. Guarde esta informação num vector `notasMiniTestes` de modo a ser fácil estender a classe para um número superior de mini-testes.
- Guardar a nota, `double` (entre 0..20), do exame final (`exame`) caso o aluno o realize.

A classe também deve ter métodos públicos que permitam:

- Construir e destruir os objectos da classe;
- Métodos `set` e `get` para todos os atributos privados da classe;
- Calcular a nota final (0-20) de acordo com a seguinte fórmula de avaliação: $0,15 \cdot \text{trabInt} + 0,25 \cdot \text{trabFin} + 0,2 \cdot (\text{soma das notas dos mini-testes}) + 0,4 \cdot \text{exame}$;
- Apresentar no ecrã a ficha do aluno (nome, número, notas das várias avaliações, nota final e se o aluno está aprovado).

a) Implemente o ficheiro header (`CAlunoProg.h`) com a definição da classe.

b) Implemente, no ficheiro de código “`CAlunoProg.cpp`”, os seguintes métodos:

b1. o construtor vazio `CAlunoProg::CAlunoProg()`

- b2.** o construtor `CAlunoProg::CAlunoProg(int numero, string nome);`
- b3.** Os métodos `set` e `get` para todos os atributos privados, excluindo o vetor `notasMiniTestes` (e.g., `string CAlunoProg::getNomeAluno()`, `void CAlunoProg::SetNumAluno(int num)`);
- b4.** os métodos `set` e `get` relativos ao vetor `notasMiniTestes` (e.g., `void CAlunoProg::setNotaMiniTeste(int numMiniTeste, double nota)` que atribui o valor `nota` à prova prática `numMiniTeste` (1 ou 2) - faça código flexível que permita alterar o número de mini-testes da disciplina).
- b5.** o método `float CAlunoProg::calculaNotaFinal()` que calcula a nota final do aluno de acordo com as regras de avaliação da disciplina;
- b6.** O método `void CAlunoProg::printStatsAluno()` que imprime na consola a ficha do aluno, a nota final e informa se o aluno está aprovado ou reprovado.

c) Teste todas as funcionalidades da classe **CAlunoProg** com um programa (**gestaoProg**) que permita a introdução dos dados dos alunos de uma turma e a sua gestão.

```
class CAlunoProg
{
public:
    CAlunoProg();
    CAlunoProg(int, string);
    ~CAlunoProg();
    void setNumAluno(int);
    void setNomeAluno(string);
    void setTrabInt(double);
    void setTrabFin(double);
    void setExame(double);
    void setNotaMiniTeste(int, float);
    void printFichaAluno();
    int getNumAluno();
    string getNomeAluno();
    double getTrabInt();
    double getTrabFin();
    double getExame();
    float getNotaMiniTeste(int numMiniTeste);
    float getNotaFinal();
    float calculaNotaFinal();
private:
    int numAluno;
    string nomeAluno;
    double trabInt, trabFin, notasMiniTestes[2], exame;
    float notaFinal;
};
```