



Programação 2

2º Semestre

Trabalho Prático P2A

Efectue as tarefas de programação descritas abaixo, usando a linguagem C++ em ambiente Linux.

Grupo 1

Pretende-se escrever um programa em C++, denominado "CentroComercial", para gerir um centro comercial. O programa deve conter uma classe denominada `CCentroComercial` com a seguinte estrutura:

```
class CCentroComercial {
public:
    typedef struct {
        string nome;
        string tipo;
        int renda;
    } LOJA;
private:
    vector<LOJA> lojas;
public:
    void ler_lojas(const char *nome);
    void imprimir_lojas(ostream &output, string tipo);
    void imprimir_lojas(ostream &output);
    void ordenarLojas ();
};
```

Execute as tarefas seguintes, tendo sempre o cuidado de testar no programa principal (função `main`), todas as funções criadas. O código fonte do programa deve ser escrito num ficheiro com o nome "CentroComercial.cpp" e deve ser compilado com o comando: **g++ CentroComercial.cpp -o CentroComercial**.

a) Implemente o membro-função:

```
void CCentroComercial::ler_lojas(const char * filename)
```

que lê a informação de um conjunto de lojas de um ficheiro de texto para o vector `lojas`.

O ficheiro está organizado em blocos de 3 linhas, contendo o nome da loja, o tipo da loja e a renda (euros/mês), por exemplo:

```
MacDonalds
Fast Food
1000
Zara
Roupa
5000
```

b) Implemente os membros-função

```
void imprimir_lojas(ostream &output, string tipo);
void imprimir_lojas(ostream &output); //todas
```

que escreve no stream output os elementos do vector `lojas`, 1 por cada linha. Em cada linha os campos de cada loja deverão ser separados por ponto-e-vírgula.

c) Implemente o membro-função

```
void ordenarLojas ();
```

que ordene o vector `lojas` por renda através do método ordenação por selecção.



Programação 2

2º Semestre

Trabalho Prático P2A

Efectue as tarefas de programação descritas abaixo, usando a linguagem C++ em ambiente Linux.

Grupo 2

Pretende-se escrever um programa em C++, denominado "JogoMedieval". O nosso jogo vai ter dois tipos de personagens: os jogadores e os monstros. O programa deve então conter uma classe denominada `CJogoMedieval` com a seguinte estrutura:

```
class CJogoMedieval {
public:
    typedef struct {
        string tipo; // "monster" or "player"
        int health;
        int strength;
        double x;
        double y;
    } CHARACTER;

private:
    vector<CHARACTER> characters;
    int activeCharacter;

public:
    CJogoMedieval (int nPlayers, int nMonsters);
    CJogoMedieval ();

    void saveGame (const char *filename);
    void loadGame (const char *filename);

    CHARACTER findWeakestMonster(int maxHealth);
};
```

Execute as tarefas seguintes, tendo sempre o cuidado de testar no programa principal (função `main`), todas as funções criadas. O código fonte do programa deve ser escrito num ficheiro com o nome "JogoMedieval.cpp" e deve ser compilado com o comando: **g++ JogoMedieval.cpp -o JogoMedieval**.

a) Implemente o construtor

```
CJogoMedieval (int nPlayers, int nMonsters);
```

que gera para o vector `characters` um conjunto de `nPlayers` e `nMonsters`. A posição de cada um deve ser aleatória entre 0 e 1, a `strength` gerada aleatoriamente entre 40 e 100, e a `health` inicializada a 100. Deve ainda colocar o índice do personagem activo (`activeCharacter`) a 0.

O construtor `CJogoMedieval()`; apenas coloca o índice do personagem activo a 0.

b) Implemente os membros-função

```
saveGame (const char * filename);
loadGame (const char * filename);
```

para guardar e carregar o estado do jogo a partir de um ficheiro.

O ficheiro está organizado em linhas, cada linha contendo o tipo de personagem, a saúde, força e a posição `x` e `y`, separados por espaços, por exemplo:

```
player 98 56 0.4 0.7
monster 77 34 0.1 0.4
```

```
monster 9 18 0.7 0.2
```

c) Implemente o membro-função

```
CHARACTER findWeakestMonster(int maxHealth);
```

que retorna o monstro mais fraco, tendo no máximo `maxHealth` de `health`.



Programação 2

2º Semestre

Trabalho Prático P2A

Efectue as tarefas de programação descritas abaixo, usando a linguagem C++ em ambiente Linux.

Grupo 3

Pretende-se escrever um simulador de um radio-despertador em C++, denominado "RadioDespertador". O programa deve então conter uma classe denominada `CRadioDespertador` com a seguinte estrutura:

```
class CRadioDespertador
{
public:
    typedef struct {
        int h, m, s;
    } TIME;
private:
    TIME currTime;
    TIME alarm;
    int alarmDuration;
    string stationAlarm;
    vector<string> stationsName;
    vector<double> stationsFrequency;

public:
    int addStation (string name, const char * stationDatabase);
    int removeStation (string name);
    int removeStation (double freq);

    void setTime (int hr, int mn, int sg);
    void setAlarm (int hr, int mn, int sg, int duration, string station);
    int isRinging();
};
```

Execute as tarefas seguintes, tendo sempre o cuidado de testar no programa principal (função `main`), todas as funções criadas. O código fonte do programa deve ser escrito num ficheiro com o nome "RadioDespertador.cpp" e deve ser compilado com o comando: **`g++ RadioDespertador.cpp -o RadioDespertador`**.

a) Implemente o membro-função

```
int CRadioDespertador::addStation (string name, const char *
stationDatabase);
```

que adiciona a estação com nome `name` à lista de estações pre-definidas. A frequência da estação é lida de um ficheiro `stationDatabase` com a informação de todas as estações conhecidas. O ficheiro está organizado em blocos de 2 linhas, contendo o nome da estação e a frequência:

```
Rádio Onda Viva
96.1
RFM
104.1
```

Se a estação não for encontrada no ficheiro ou já existir na lista de estações pre-definidas a função retorna -1; caso contrário retorna 0.

b) Implemente os membros-função

```
int CRadioDespertador::removeStation (string name);  
int CRadioDespertador::removeStation (double freq);
```

Se a estação não for encontrada as funções retornam -1; caso contrário retornam 0;

c) Implemente o membro-função

```
void CRadioDespertador::setTime (int h, int m, int s);  
void CRadioDespertador::setAlarm (int h, int m, int s, int duration,  
    string station);  
int CRadioDespertador::isRinging();
```

em que `duration` é a duração do alarme em minutos e `station` é a estação que deve tocar quando o alarme tocar.

A função `int isRinging()`; retorna -1 se o alarme ainda não começou a tocar, 0 se se encontra a tocar e 1 se já tocou.

A função `void setAlarm(...)`; verifica, no vector `stationsName`, se a estação escolhida está programada; caso não esteja deve adicioná-la à lista com o auxílio da função:

```
int CRadioDespertador::addStation (string name, const char *  
    stationDatabase);
```



Programação 2

2º Semestre

Trabalho Prático P2A

Efectue as tarefas de programação descritas abaixo, usando a linguagem C++ em ambiente Linux.

Grupo 4

Pretende-se escrever um programa em C++, denominado “Videoteca”, para gerir uma videoteca particular. O programa deve conter uma classe `CVideo` com a seguinte estrutura:

```
class CVideo
{
    string nome;
    string realizador;
    int ano;
public:
    CVideo (string nome, string realizador, string ano);

    int getAno ();
    string getNome();
    string getRealizador();
};
```

Execute as tarefas seguintes, tendo sempre o cuidado de testar no programa principal (função `main`), todas as funções criadas. O código fonte do programa deve ser escrito num ficheiro com o nome “Videoteca.cpp” e deve ser compilado com o comando:

g++ Videoteca.cpp -o Videoteca.

a) Implemente a função

```
int lerVideoteca (vector<CVideo *>& vt, const char * filename);
```

para ler de um ficheiro a informação sobre os nossos filmes.

O ficheiro está organizado em blocos de 3 linhas, contendo o nome do filme, o realizador e o ano:

```
O Bom Pastor
Robert de Niro
2006
Ghost Rider
Mark Steven Johnson
2007
```

b) Implemente os membros-função `getAno()`, `getNome()` e `getRealizador()` da classe `CVideo` e a função global

```
void mostraVideos (vector<CVideo *>& vt, int minAno, int maxAno);
```

que imprime na consola todos os videos realizados entre os anos `minAno` e `maxAno`.

c) Implemente o operador

```
bool CVideo::operator< (CVideo& v2)
```

que retorna o resultado da comparação dos anos e a função global

```
int comparaPorAno (const CVideo &v1, const CVideo &v2)
```

que retorna `-1` se o ano de `v1` é inferior ao ano de `v2`, `0` se o ano de `v1` é igual ao ano de `v2` e `1` se o ano de `v1` é superior ao ano de `v2`.



Programação 2

2º Semestre

Trabalho Prático P2A

Efectue as tarefas de programação descritas abaixo, usando a linguagem C++ em ambiente Linux.

Grupo 5

Pretende-se escrever um programa em C++, denominado “medidas”, que processa um conjunto de medidas (números inteiros). O programa deve conter uma classe denominada `CMedidas` com a seguinte estrutura:

```
class CMedidas {
    int nmed;
    vector <int> medidas;

public:
    void ler_medidas(const char *nome);
    void imprimir_medidas(ostream &output);
    int pesquisa_seq(const vector<int> &medPesq);
    void bubble_sort(int ordem);
};
```

Execute as tarefas seguintes, tendo sempre o cuidado de testar no programa principal (função `main`), todas as funções criadas. O código fonte do programa deve ser escrito num ficheiro com o nome “medidas.cpp” e deve ser compilado com o comando: **g++ medidas.cpp -o medidas**.

a) Implemente a função-membro:

```
void CMedidas::ler_medidas (const char *nome)
```

que lê um conjunto de números inteiros de um ficheiro de texto para o vector `medidas` e coloca em `nmed` o número total de medidas. Caso ocorra algum erro na leitura `nmed` deve ficar com o valor -1. Construa um ficheiro, designado “medidas.txt” com os seguintes valores para realizar o teste do programa:

```
9 10 3 5 0 1 2 3 4 5 6 7 8 20 0 15 3 4 5 10 0 3 4 5 3 0 23 14 0
```

b) Implemente a função-membro:

```
void CMedidas::imprimir_medidas(ostream &output)
```

que escreve no stream `output` os elementos do vector `medidas`, separados por um espaço – a medida de valor zero indica mudança de linha.

c) Construa a função-membro:

```
void CMedidas::bubble_sort(int ordem)
```

que ordene o vector por ordem crescente (caso `ordem` seja um inteiro positivo) ou decrescente (caso `ordem` seja um inteiro negativo), utilizando o método “bubble_sort”. Caso `ordem` seja 0, o vector deve permanecer inalterado.



Programação 2

2º Semestre

Trabalho Prático P2A

Efectue as tarefas de programação descritas abaixo, usando a linguagem C++ em ambiente Linux.

Grupo 6

Pretende-se escrever um programa em C++, denominado "JardimZoologico", para gestão de um Jardim Zoológico. O programa deve conter uma classe denominada CJardimZoologico com a seguinte estrutura:

```
class CJardimZoologico {
public:
    typedef struct {
        string categoria; // urso, girafa, etc
        string nome;
        string veterinario;
        int idade; // anos
    } ANIMAL;
private:
    vector<ANIMAL> animais;
public:
    bool animalJovem(const char *nome);
    vector<ANIMAL> animaisJovens();
    void loadAnimals(istream& is);
    vector<string> veterinarioComMaisTrabalho();
};
```

Execute as tarefas seguintes, tendo sempre o cuidado de testar no programa principal (função main), todas as funções criadas. O código fonte do programa deve ser escrito num ficheiro com o nome "JardimZoologico.cpp" e deve ser compilado com o comando:

g++ JardimZoologico.cpp -o JardimZoologico

a) Implemente o membro-função

```
bool CJardimZoologico::animalJovem(const char *nome);
```

que verifica se um dado animal é jovem e

```
vector<ANIMAL> CJardimZoologico::animaisJovens();
```

que retorna o conjunto de todos os animais jovens. Um animal é jovem se não tiver mais de 5 anos.

b) Implemente a função

```
void CJardimZoologico::loadAnimals(istream& is);
```

que carrega a informação sobre os animais existente num ficheiro. Esta deve ir por linhas da seguinte forma:

```
urso
Mikas
João Carvalho
5
girafa
Margarida
Tiago Santos
8
```

c) Implemente o operador

```
bool CJardimZoologico::operator> (CJardimZoologico& zoo2);
```

para comparar dois jardins zoológicos. Um jardim zoológico é maior que um segundo se a soma das idades dos seus animais for superior à soma das idades dos animais no segundo.