



PROGRAMAÇÃO POR OBJETOS | 21093

Solução-tipo

Abaixo colocamos uma solução mínima que atenda aos desafios lançados no enunciado, chamando a atenção para as palavras que agora destacamos a negrito, e que correspondem aos pedidos que mais frequentemente não foram inteiramente respeitados nas respostas dos alunos...

Tem associado a este enunciado o ficheiro dados.csv para usar.

1. O ficheiro dados.csv, em anexo, tem as estatísticas da pandemia COVID-19 em Portugal fornecidas pela Direção-Geral da Saúde, no formato recolhido diariamente pela Data Science for Social Good Portugal (<https://github.com/dssg-pt>) Há dois tipos de linhas no ficheiro: a de cabeçalhos e as de dados.
 - a. Defina uma classe **abstrata** “Linha”, destinada a representar um conjunto de dados, com um **método** “**Ler**” que recebe como parâmetro um stream; e um **método** “**Escrever**”, destinado ao ecrã.

em Linha.h:

```
#ifndef LINHA_H_
#define LINHA_H_
```

```

#include <iostream>
using namespace std;

class Linha {
public:
    Linha();
    virtual ~Linha();
    virtual void Ler(std::istream&) = 0;
    virtual void Escrever() = 0;
};

#endif /* LINHA_H_ */

```

- b. Defina classes **derivadas** LinhaDeCabecalho e LinhaDeDados, para guardar **dados textuais e numéricos**, respetivamente.

em LinhaDeCabecalho.h

```

#ifndef LINHADECABECALHO_H_
#define LINHADECABECALHO_H_

#include <string>
#include <vector>
#include <iostream>
#include "Linha.h"

class LinhaDeCabecalho: public Linha {
private:
    vector<string> columnHeader;
public:
    LinhaDeCabecalho();
    ~LinhaDeCabecalho();
    void Ler(std::istream&);
    void Escrever();
};

#endif /* LINHADECABECALHO_H_ */

```

em LinhaDeDados.h

```

#ifndef LINHADEDADOS_H_
#define LINHADEDADOS_H_

```

```

#include <string>
#include <vector>
#include <iostream>
#include "Linha.h"

class LinhaDeDados: public Linha {
private:
    vector<double> columnData;
public:
    LinhaDeDados();
    ~LinhaDeDados();
    void Ler(std::istream&);
    void Escrever();
};

#endif /* LINHADEDADOS_H_ */

```

- c. Implemente os métodos "Ler" e "Escrever", considerando que o "Ler" só lê **uma linha do stream fornecido**; e que o "Escrever" apresenta os dados **na horizontal no ecrã, separados por "|"** (barra vertical).

em LinhaDeCabecalho.cpp

```

#include "LinhaDeCabecalho.h"

LinhaDeCabecalho::LinhaDeCabecalho() {
}

LinhaDeCabecalho::~LinhaDeCabecalho() {
}

void LinhaDeCabecalho::Ler(std::istream& myStream){
    //std::cout << "inside Cabecalho -- " <<
myStream.rdbuf() << endl;
    string word;
    while(getline(myStream, word, ',')) {
        columnHeader.push_back(word);
    }
}

void LinhaDeCabecalho::Escrever(){

```

```

        for (auto word: columnHeader) {
            std::cout << word << " | ";
        }
        std::cout << endl;
    }
}

```

em LinhaDeDados.cpp

```

#include "LinhaDeDados.h"

LinhaDeDados::LinhaDeDados() {}

LinhaDeDados::~LinhaDeDados() {}

void LinhaDeDados::Ler(istream& myStream){
    string valueStr;
    while(getline(myStream, valueStr, ',')) {
        double value = 0;
        if (!valueStr.empty()) {
            value = stod(valueStr); // poderia
            ser uma boa politica um try...catch, mas com os dados
            disponibilizados funciona.
        }
        columnData.push_back(value);
    }
}

void LinhaDeDados::Escrever(){
    for (auto word: columnData) {
        cout << word << " | ";
    }
    cout << endl;
}

```

2. Defina uma classe DadosTabelados que recorra às classes da pergunta anterior:
 - a. Tem uma **linha de cabeçalho do tipo Linha**, à qual atribui uma LinhaDeCabecalho.
 - b. Tem um **conjunto** de dados do tipo Linha, ao qual atribui objetos LinhaDeDados.

- c. Lê a **primeira linha do ficheiro dados.csv e as linhas restantes para os objetos respectivos**, usando os métodos desses objetos.

em DadosTabelados.h

```
#ifndef DADOSTABELADOS_H_
#define DADOSTABELADOS_H_

#include <string>
#include <vector>
#include <fstream>
#include <sstream>
#include <iostream>
#include "Linha.h"
#include "LinhaDeCabecalho.h"
#include "LinhaDeDados.h"

class DadosTabelados {
private:
    LinhaDeCabecalho header;
    LinhaDeDados dataLine;
public:

    DadosTabelados();
    ~DadosTabelados();
};

#endif /* DADOSTABELADOS_H_ */
```

em DadosTabelados.cpp

```
#include "DadosTabelados.h"

DadosTabelados::DadosTabelados() {
    string path = "./src/dados.csv";
    ifstream fileIn;
    fileIn.open(path, ios::in);
    string data;
    getline(fileIn, data);

    stringstream myStream(data);
    header.Ler(myStream);

    while (getline(fileIn, data)) {
        stringstream myStream(data);
        dataLine.Ler(myStream);
    }
}
```

```
    }  
}  
DadosTabelados::~~DadosTabelados() {}
```