



Exercício (2 pontos)

Data de entrega: 24/08/2019 até às 23:59

1. Implemente o código abaixo apresentado em sala de aula como um Projeto: Pessoa.hpp

```
1 #ifndef __PESSOA_HPP
2 #define __PESSOA_HPP
3
4 #include <string>
5
6 using namespace std;
7
8 class Pessoa
9 {
10     public:
11         float peso, altura; //variável
12         char sexo;
13         string nome;
14
15         void setNome(string nome);
16         string getNome();
17         void setPeso(float p);
18         float getPeso();
19
20         void setSexo(char sexo);
21         char getSexo();
22
23         void setAltura(float altura);
24         float getAltura();
25
26         void imprimir();
27 };
28
29 #endif
30
31 Pessoa.cpp
32
33 1 #include <iostream> //biblioteca
34 2 #include "Pessoa.hpp"
35 3
36 4 using namespace std;
37 5
38 6 void Pessoa::setNome(string nome)
39 7 {
40 8     this->nome = nome;
41 9 }
42 10 string Pessoa::getNome()
43 11 {
44 12     return nome;
45 13 }
46 14
47 15 void Pessoa::setPeso(float p)
48 16 {
49 17     this->peso = p;
50 18 }
51 19 float Pessoa::getPeso()
52 20 {
53 21     return peso;
54 22 }
55 23
56 24 void Pessoa::setSexo(char sexo)
57 25 {
58 26     this->sexo = sexo;
59 27 }
60 28 char Pessoa::getSexo()
61 29 {
62 30     return sexo;
63 31 }
64 32
65 33 void Pessoa::setAltura(float altura)
66 34 {
67 35     this->altura = altura;
68 36 }
69 37 float Pessoa::getAltura()
70 38 {
71 39     return altura;
72 40 }
73 41
74 42 void Pessoa::imprimir()
75 43 { //Imprime os dados da Pessoa
76 44     cout << "-----\n";
77 45     cout << "Nome: " << nome << endl;
78 46     cout << "sexo: " << sexo << endl;
79 47     cout << "Altura: " << altura << endl;
80 48     cout << "Peso: " << peso << endl;
81 49 }
```



Exercício (2 pontos)

Data de entrega: 24/08/2019 até às 23:59

Main.cpp

```
1 #include <iostream>
2 #include "Pessoa.hpp"
3 int main(){
4     Pessoa p,z; // chama o construtor-padrão
5     float peso,altura;
6     char s;
7     string nome;
8
9     cout << "Digite o nome: ";
10    getline(cin,nome);
11    //cin >> nome;
12    p.setNome(nome);
13    cout << "Digite o sexo: ";
14    cin >> s;
15    p.setSexo(s);
16    cout<< "Digite a altura: ";
17    cin >> altura;
18    p.setAltura(altura);
19    cout << "Digite o peso: ";
20    cin >> peso;
21    p.setPeso(peso);
22
23    p.imprimir();
24 } //fim
```

Makefile

```
1 CXX      := -g++
2 CXXFLAGS := -pedantic-errors -Wall -Wextra -Werror
3 LDFLAGS  := -lstdc++ -lm
4 BUILD    := ./build
5 OBJ_DIR  := $(BUILD)/objects
6 APP_DIR  := $(BUILD)/
7 TARGET   := app
8 INCLUDE  := -Iinclude/
9 SRC      := $(wildcard src/*.cpp)
10 # $(wildcard src/module1/*.cpp) \
11 # $(wildcard src/module2/*.cpp) \
12 # $(wildcard src/module3/*.cpp) \
13
14 OBJECTS := $(SRC:%.cpp=$(OBJ_DIR)/%.o)
15
16 all: build $(APP_DIR)/$(TARGET)
17
18 $(OBJ_DIR)/%.o: %.cpp
19     @mkdir -p $(@D)
20     $(CXX) $(CXXFLAGS) $(INCLUDE) -o $@ -c $<
21
22 $(APP_DIR)/$(TARGET): $(OBJECTS)
23     @mkdir -p $(@D)
24     $(CXX) $(CXXFLAGS) $(INCLUDE) $(LDFLAGS) -o $(APP_DIR)/$(TARGET) $(OBJECTS)
25
26 .PHONY: all build clean debug release run
27
28 build:
29     @mkdir -p $(APP_DIR)
30     @mkdir -p $(OBJ_DIR)
31
32 debug: CXXFLAGS += -DDEBUG -g
33 debug: all
34
35 release: CXXFLAGS += -O3
36 release: all
37
38 clean:
39     -@rm -rvf $(OBJ_DIR)/*
40     -@rm -rvf $(APP_DIR)/*
41
42 run:
43     ./$(BUILD)/$(TARGET)
```



Exercício (2 pontos)

Data de entrega: 24/08/2019 até às 23:59

2. Utilizando como exemplo o Exercício 01, crie um projeto para uma empresa de Taxi contendo uma classe corrida com:

Atributos:

- Modelo do Carro
- Motorização do Carro
- Fabricante do Veículo
- Nome do Motorista
- Quilometragem Inicial
- Quilometragem Final

Implemente os getters e setters de cada atributo.

Implemente o main para ler os dados do teclado e mostrar os dados de uma corrida com o valor final a ser pago (considere R\$ 3,75 por KM rodado).