



Prova 3

Professor: Gustavo Henrique Borges Martins

Aluno: _____ Matrícula: _____

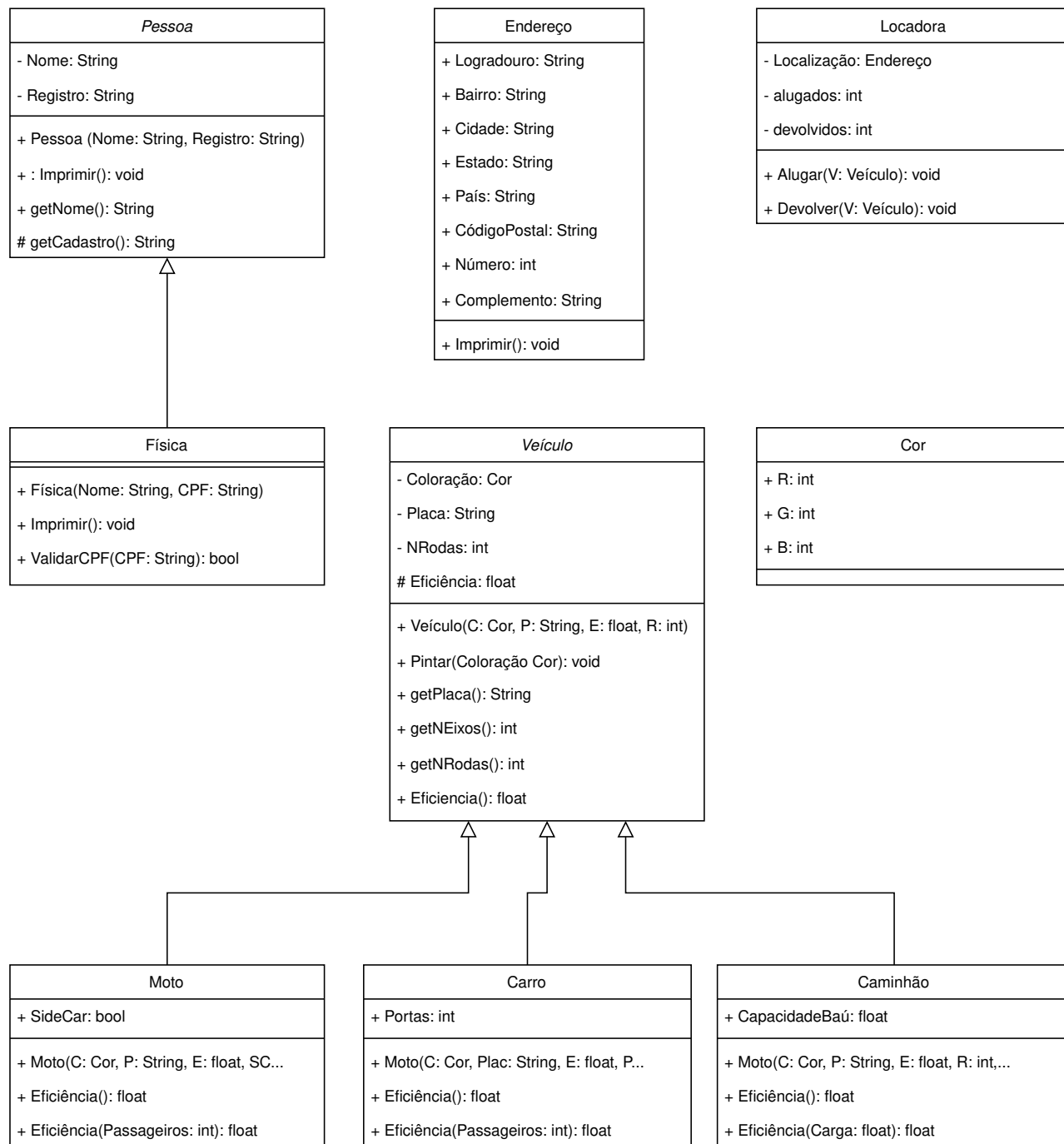
Instruções para a prova:

- Preencha todas as folhas desta prova com seu nome e sua matrícula.
- Leia atentamente a todas as questões antes de resolvê-las.
- Resolva as questões desta prova em linguagem C.
- Não deixe de responder nenhuma questão.
- Deixe comentários sobre as questões, eles podem ser considerados na correção.

1. O código a seguir implementa uma **Pessoa Jurídica**:

```
1 #include <string>
2 class Juridica: public Pessoa {
3     public:
4         Juridica(std::string, std::string);
5         void Imprimir();
6         static bool ValidarCNPJ(std::string);
7     }
8     Juridica::Juridica(std::string Nome, std::string CNPJ) : Pessoa(Nome,CNPJ) {}
9     void Juridica::Imprimir() {
10         std::cout << "Pessoa Jurídica: " << getNome() << " inscrita no CNPJ: " <<
            getRegistro() << std::endl;
11     }
12     bool Juridica::ValidarCNPJ(std::string CNPJ) {
13         int CNPJ[14], digitos[2];
14         for (int i = 0; i < CNPJ.length(); i++) {
15             CNPJ[i] = std::stoi(CNPJ[i]);
16         }
17         digitos[0] = 5*CNPJ[0] + 4*CNPJ[1] + 3*CNPJ[2] + 2*CNPJ[3];
18         digitos[0] += 9*CNPJ[4] + 8*CNPJ[5] + 7*CNPJ[6] + 6*CNPJ[7];
19         digitos[0] += 5*CNPJ[8] + 4*CNPJ[9] + 3*CNPJ[10] + 2*CNPJ[11];
20         digitos[0] = 11-(digitos[0]%11);
21         if (digitos[0] >= 10)
22             digitos[0] = 0;
23         digitos[1] = 6*CNPJ[0] + 5*CNPJ[1] + 4*CNPJ[2] + 3*CNPJ[3];
24         digitos[1] += 2*CNPJ[4] + 9*CNPJ[5] + 8*CNPJ[6] + 7*CNPJ[7];
25         digitos[1] += 6*CNPJ[8] + 5*CNPJ[9] + 4*CNPJ[10] + 3*CNPJ[11] + 2*CNPJ[12];
26         digitos[1] = 11-(digitos[1]%11);
27         if (digitos[1] >= 10)
28             digitos[1] = 0;
29         if ((digitos[0] == CNPJ[12]) && (digitos[1] == CNPJ[13]))
30             return true;
31         return false;
32     }
```

Analise o diagrama de classes UML seguinte:



Dado o diagrama de classes UML e a classe **Pessoa Jurídica** implementada, resolva:

- (a) (5 pontos) Desenhe o diagrama de classes UML para a classe **Pessoa Jurídica**.
- (b) (5 pontos) Para cada uma das classes no diagrama, indique quais classes tem relações de herança, quais métodos estão sobrecarregados, quais classes tem o polimorfismo e quais classes são abstratas, indicando se são puramente abstratas.
- (c) (5 pontos) Implemente o esqueleto da classe **Veículo**.
- (d) (5 pontos) Escolha uma classe filha de **Veículo** e implemente-a, desde o esqueleto até os métodos.
 - Uma moto possui 2 rodas, enquanto um carro possui 4 rodas, já um caminhão pode ter um número diferente de rodas.
 - A eficiência do veículo é dada pelo conteúdo do atributo eficiência no método que não possui parâmetro.
 - Para o método que possui parâmetro, a eficiência é dada da seguinte forma:
 - Para qualquer caso inválido, retorne zero.
 - Numa moto, podem haver 0, 1 ou 2 passageiros p , então a eficiência será dada pela relação: $\frac{E}{1+p}$. Só existe a possibilidade de terem 2 passageiros se o **SideCar** estiver marcado como verdadeiro.
 - Num carro, podem haver 0, 1, 2, 3 ou 4 passageiros p e a relação da eficiência é dada por: $\frac{E}{1+p/2}$.
 - Num caminhão, podem haver cargas c de 0 à 5000 kg e a relação da eficiência é dada por: $\frac{E}{1+c/1000}$.
- (e) (5 pontos) Passe o argumento "**10001000100033**" para o método **Juridica::ValidarCNPJ** e execute-o, indicando o resultado de saída. Realize as contas e deixe-as indicadas na prova.

2. (5 points (bonus)) Implemente as classes **Cor**, **Locadora** e **Endereço**.

- O método **Endereço::Imprimir** deverá colocar o texto na saída padrão da seguinte forma: "Logradouro Número Complemento Bairro Cidade Estado País CódigoPostal".
- O método **Locadora::Alugar** deverá acrescentar 1 em **Locadora::alugados** e o método **Locadora::Devolver** deverá acrescentar 1 em **Locadora::devolvidos**.

Questões	1	2	Total
Total de pontos	25	0	25
Pontos obtidos			