

 PUC Minas	Curso: Ciência da Computação	
	Disciplina: Algoritmos e Estrutura de Dados I	
	Nome: Pedro Henrique Lopes Costa	
	Código de pessoa: 1362465	Campus: Coração Eucarístico
	Período: 1º	Turma: Manhã

9

Questão 01

```
#include <iostream>
#include <math.h>
#include <string.h>
#include <locale.h>

using namespace std;

//
----- //

class Funcionario
{
    string nome;
    float salario;

public:

    Funcionario() {}

    Funcionario(string c_nome, float c_salario)
    {
        setNome(c_nome);
        setSalario(c_salario);
    }

    void setNome(string newName) { nome = newName; }
    void setSalario(float newSalario) { salario = newSalario; }
    string getNome() { return nome; }
    float getSalario() { return salario; }

    virtual void exibeDados() { printf("[Funcionário]\nNome: %s\nSalário: R$%.2f\n\n", getNome().c_str(), getSalario()); }
};
```

```
// ----- //
```

```
class Gerente: public Funcionario
{
    public:

    Gerente(string c_nome, float c_salario)
    {
        setNome(c_nome);
        setSalario(c_salario);
    }

    virtual void exibeDados() { printf("[Gerente]\nNome: %s\nSalário:
R$%.2f\n\n", getNome().c_str(), getSalario()); }
};

// ----- //
```

```
class AssistTec: public Funcionario
{
    int matricula;
    float aumento;

    public:

    AssistTec(string c_nome, float c_salario, int c_matricula, float
c_aumento)
    {
        setNome(c_nome);
        setSalario(c_salario);
        setMatricula(c_matricula);
        setAumento(c_aumento);
    }

    void setMatricula(int newMatricula) { matricula = newMatricula; }
    int getMatricula() { return matricula; }
    void setAumento(float newAumento) { aumento = newAumento; }
    float getAumento() { return aumento; }

    virtual void exibeDados()
    {
        printf("[Assistente Técnico]\nNome: %s\nMatrícula: %i\nSalário:
R$%.2f (=%.0f%%)\n\n", getNome().c_str(), getMatricula(), getSalario() * (1 +
(aumento / 100)), aumento);
    }
}
```

```

};

// ----- //

class AssistADM: public Funcionario
{
    // 0 = Dia | 1 = Noite

    int matricula, turno;
    float adicional;

public:

    AssistADM(string c_nome, float c_salario, int c_matricula, int c_turno)
    {
        setNome(c_nome);
        setSalario(c_salario);
        setMatricula(c_matricula);
        setTurno(c_turno);

        if(c_turno == 0) adicional = 0;
        else adicional = 20;
    }

    void setMatricula(int newMatricula) { matricula = newMatricula; }
    int getMatricula() { return matricula; }
    void setTurno(int newTurno) { turno = newTurno; }
    int getTurno() { return turno; }
    void setAdicional(float newAdicional) { adicional = newAdicional; }
    float getAdicional() { return adicional; }

    virtual void exibeDados()
    {
        float result = 0;

        if(getTurno() == 1) result = getSalario() * (1 + (adicional / 100));
        else result = getSalario();

        printf("[Assistente ADM]\nNome: %s\nMatrícula: %i\nTurno: %s (+%.0f%% de adicional)\nSalário: R$%.2f\n\n", getNome().c_str(), getMatricula(), getTurno() == 0 ? "Dia" : "Noite", getAdicional(), result);
    }
};

```

```

//
----- //

int main()
{
    setlocale(LC_ALL, "Portuguese");

    // ----- //

    Funcionario *funcs[5];

    funcs[0] = new Funcionario("Maria", 3000);
    funcs[1] = new Gerente("Carlos", 10000);
    funcs[2] = new AssistTec("Julio", 5000, 13, 10);
    funcs[3] = new AssistADM("Amanda", 7000, 20, 0);
    funcs[4] = new AssistADM("Laura", 8000, 25, 1);

    for(int x = 0; x != 5; x++) funcs[x] -> exibeDados();

    // ----- //

    system("pause");
    return 0;
}

```

Questão 02

```
#include <iostream>
#include <math.h>
#include <string.h>
#include <locale.h>

using namespace std;

//
----- //

class Animal
{
    public:

    Animal() {}

    virtual void fala() { printf("Fala genérica\n"); }
};

// ----- //

class Homem: public Animal
{
    public:

    virtual void fala() { printf("Oi\n"); }
};

// ----- //

class Cao: public Animal
{
    public:

    virtual void fala() { printf("Au au\n"); }
};

// ----- //

class Gato: public Animal
{
    public:
```

```

    virtual void fala() { printf("Miau\n"); }
};

//
-----

int main()
{
    setlocale(LC_ALL, "Portuguese");

    // -----

    Animal *zoo[10];

    zoo[0] = new Animal();

    for(int x = 1; x != 4; x++) zoo[x] = new Homem();
    for(int x = 4; x != 7; x++) zoo[x] = new Cao();
    for(int x = 7; x != 10; x++) zoo[x] = new Gato();

    for(int x = 0; x != 10; x++) zoo[x] -> fala();

    printf("\n");

    // -----

    system("pause");
    return 0;
}

```

Questão 03

```
#include <iostream>
#include <math.h>
#include <string.h>
#include <locale.h>

using namespace std;

//
----- //

class ContaCorrente
{
    float saldo;

    public:

    ContaCorrente() {}
    ContaCorrente(float c_saldo) { saldo = c_saldo; }

    float getSaldo() { return saldo; }
    void setSaldo(float newSaldo) { saldo = newSaldo; }
    void depositar(float valor) { saldo += valor; }

    virtual void sacar(float valor)
    {
        float total = valor + (valor * 0.05);

        if(getSaldo() >= total) setSaldo(getSaldo() - total);
    }
};

// ----- //

class ContaEspecial: public ContaCorrente
{
    public:

    ContaEspecial(float c_saldo) { setSaldo(c_saldo); }

    virtual void sacar(float valor)
    {
        float total = valor + (valor * 0.01);
```

```

        if(getSaldo() >= total) setSaldo(getSaldo() - total);
    }
};

//
-----
----- //

int main()
{
    setlocale(LC_ALL, "Portuguese");

    // ----- //

    ContaCorrente *contas[2];

    contas[0] = new ContaCorrente(3000);
    contas[0] -> depositar(2000);
    contas[0] -> sacar(4000);

    contas[1] = new ContaEspecial(10000);
    contas[1] -> depositar(5000);
    contas[1] -> sacar(8000);

    for(int x = 0; x != 2; x++) printf("> Saldo final (Conta %i) = R$%.2f\n",
x + 1, contas[x] -> getSaldo());

    printf("\n");

    // ----- //

    system("pause");
    return 0;
}

```


Questão 04

```
#include <iostream>
#include <math.h>
#include <string.h>
#include <locale.h>

using namespace std;

//
----- //

class Item
{
    string nome;
    float preco;

public:

    Item() {}
    Item(string c_nome, float c_preco)
    {
        nome = c_nome;
        preco = c_preco;
    }

    void setNome(string newName) { nome = newName; }
    void setPreco(float newPreco) { preco = newPreco; }

    string getNome() { return nome; }
    float getPreco() { return preco; }

    virtual void toString() { printf("[Item]\nNome: %s\nPreço: R$%.2f\n\n",
getNome().c_str(), getPreco()); }
};

// ----- //

class Livro: public Item
{
    string autor;

public:

    Livro(string c_nome, float c_preco, string c_autor)
```

```

{
    setNome(c_nome);
    setPreco(c_preco);
    setAutor(c_autor);
}

void setAutor(string newAutor) { autor = newAutor; }
string getAutor() { return autor; }

virtual void toString() { printf("[Livro]\nNome: %s\nPreço:
R$%.2f\nAutor: %s\n\n", getNome().c_str(), getPreco(), getAutor().c_str()); }
};

// ----- //

class CD: public Item
{
    int numFaixas;

public:

    CD(string c_nome, float c_preco, int c_numFaixas)
    {
        setNome(c_nome);
        setPreco(c_preco);
        setNumFaixas(c_numFaixas);
    }

    void setNumFaixas(int newNumFaixas) { numFaixas = newNumFaixas; }
    int getNumFaixas() { return numFaixas; }

    virtual void toString() { printf("[CD]\nNome: %s\nPreço: R$%.2f\nNúmero
de faixas: %i\n\n", getNome().c_str(), getPreco(), getNumFaixas()); }
};

// ----- //

class DVD: public Item
{
    int duracao;

public:

    DVD(string c_nome, float c_preco, int c_duracao)
    {

```

```

        setNome(c_nome);
        setPreco(c_preco);
        setDuracao(c_duracao);
    }

    void setDuracao(int newDuracao) { duracao = newDuracao; }
    int getDuracao() { return duracao; }

    virtual void toString() { printf("[DVD]\nNome: %s\nPreço:
R$%.2f\nDuração: %imin\n\n", getNome().c_str(), getPreco(), getDuracao()); }
};

//
----- //

int main()
{
    setlocale(LC_ALL, "Portuguese");

    // ----- //

    Item *itens[3];

    itens[0] = new Livro("Breves respostas para grandes questões", 15.50,
"Stephen Hawking");
    itens[1] = new CD("Arrochadeira Top", 5.30, 13);
    itens[2] = new DVD("Cristiano Araújo In The Cities", 29.90, 3);

    for(int x = 0; x != 3; x++) itens[x] -> toString();

    printf("\n");

    // ----- //

    system("pause");
    return 0;
}

```

Questão 05

```
#include <iostream>
```

```

#include <math.h>
#include <string.h>
#include <locale.h>

using namespace std;

//
----- //

class Veiculo
{
    string nome;

public:

    Veiculo() {}
    Veiculo(string c_nome) { setNome(c_nome); }

    void setNome(string newName) { nome = newName; }
    string getNome() { return nome; }

    virtual void listarVerificacoes() { printf("> Verificações para %s:
Ajustes, limpeza, etc\n", getNome().c_str()); }
    virtual void ajustar() { printf("> O veículo %s foi ajustado com
sucesso.\n", getNome().c_str()); }
    virtual void limpar() { printf("> O veículo %s foi limpo com sucesso.\n",
getNome().c_str()); }
};

// ----- //

class Bicicleta: public Veiculo
{
public:

    Bicicleta(string c_nome) { setNome(c_nome); }

    virtual void listarVerificacoes() { printf("> Verificações em %s:
Conferir pneus, pedais e banco\n", getNome().c_str()); }
    virtual void ajustar() { printf("> A bicicleta %s foi ajustada com
sucesso.\n", getNome().c_str()); }
    virtual void limpar() { printf("> A bicicleta %s foi limpa com
sucesso.\n", getNome().c_str()); }
};

```

```

// ----- //

class Automovel: public Veiculo
{
    float nivelOleo;

public:

    Automovel(string c_nome)
    {
        setNome(c_nome);
        setNivelOleo(0);
    }

    float getNivelOleo() { return nivelOleo; }
    virtual void setNivelOleo(float newNivel) { nivelOleo = newNivel; }

    virtual void listarVerificacoes() { printf("> Verificações em %s:
Conferir pneus, portas, motor, óleo\n", getNome().c_str()); }
    virtual void ajustar() { printf("> O automóvel %s foi ajustado com
sucesso.\n", getNome().c_str()); }
    virtual void limpar() { printf("> O automóvel %s foi limpo com
sucesso.\n", getNome().c_str()); }
};

//
----- //

int main()
{
    setlocale(LC_ALL, "Portuguese");

    // ----- //

    Bicicleta bike("BMX");
    Automovel land("Land Rover");

    bike.listarVerificacoes();
    bike.ajustar();
    bike.limpar();

    printf("\n");
}

```

```
land.listarVerificacoes();  
land.ajustar();  
land.limpar();  
land.setNivelOleo(100.0);  
  
printf("\n");  
  
// ----- //  
  
system("pause");  
return 0;  
}
```