

A01.

Classe Pessoa.h

```
#include <string>
using namespace std;

class Pessoa {
//atributos
private:
    string nome;
    int idade;
    float altura;

//métodos
public:
    //contrutor
    Pessoa(string nome, int idade, float altura);
    //sets e gets
    void setNome(string nome);
    void setIdade(int idade);
    void setAltura(float altura);
    string getNome();
    int getIdade();
    float getAltura();
    //impressao
    void imprimirDados();
};
```

Classe Pessoa.cpp

```
#include "Pessoa.h"
#include <iostream>
using namespace std;

//construtor
Pessoa::Pessoa(string nome, int idade, float altura)
    : nome(nome), idade(idade), altura(altura) {}

//sets
void Pessoa::setNome(string nom) {
    nome = nom;
}

void Pessoa::setIdade(int idade) {
```

```

        this->idade = idade;
    }

    void Pessoa::setAltura(float altura) {
        this->altura = altura;
    }

    //gets
    string Pessoa::getNome(){
        return nome;
    }

    int Pessoa::getIdade(){
        return idade;
    }

    float Pessoa::getAltura(){
        return altura;
    }

    void Pessoa::imprimirDados(){
        cout << "Nome: " << nome << "\nIdade: " << idade << "\nAltura: " << altura << " m\n";
    }

```

main.cpp

```

#include <iostream>
#include "Pessoa.h"

using namespace std;

int main() {
    //criando um tipo

    Pessoa pessoa1("João", 25, 1.75f);
    Pessoa pessoa2("Maria", 30, 1.65f);
    cout<< "Pessoa 1 "<<endl;
    pessoa1.imprimirDados();

    pessoa1.setNome("Xulambs");
    pessoa1.setIdade(23);
    pessoa1.setAltura(1.00f);

```

```
cout << "\nPessoa 1 Após modificações:\n";

cout<< "Nome: " << pessoa1.getNome() << "\n";
cout<< "Idade: " << pessoa1.getIdade() << "\n";
cout<< "Altura: " << pessoa1.getAltura() << "\n";
return 0;
}
```

A02.

Classe Cliente.h

```
#include <string>
using namespace std;

class Cliente {
private:
    string nome;
    string dataNascimento;
    string endereco;
    string telefone;

public:
    //contrutores sobrecarregados
    Cliente();
    Cliente(string nome, string dataNascimento, string endereco, string telefone);

    //sets e gets
    void setNome(string nome);
    void setDataNascimento(string dataNascimento);
    void setEndereco(string endereco);
    void setTelefone(string telefone);
    string getNome();
    string getDataNascimento();
    string getEndereco();
    string getTelefone();
    //impressao
    void imprimirDados();
};
```

Classe Cliente.cpp

```
#include <iostream>
#include <string.h>
#include "Cliente.h"

using namespace std;
Cliente:: Cliente(): nome(""), dataNascimento(""), endereco(""), telefone("") {}

Cliente::Cliente(string nome, string dataNascimento, string endereco, string telefone)
    : nome(nome), dataNascimento(dataNascimento), endereco(endereco), telefone(telefone) {}
```

```

void Cliente::setNome(string nome) {
    this->nome = nome;
}

void Cliente::setDataNascimento(string dataNascimento) {
    this->dataNascimento = dataNascimento;
}

void Cliente::setEndereco(string endereco) {
    this->endereco = endereco;
}

void Cliente::setTelefone(string telefone) {
    this->telefone = telefone;
}

string Cliente::getNome() {
    return nome;
}

string Cliente::getDataNascimento() {
    return dataNascimento;
}

string Cliente::getEndereco() {
    return endereco;
}

string Cliente::getTelefone() {
    return telefone;
}

void Cliente::imprimirDados() {
    cout << "Nome: " << nome << "\nData de Nascimento: " << dataNascimento
        << "\nEndereço: " << endereco << "\nTelefone: " << telefone << "\n";
}

```

[main.cpp](#)

```

#include <iostream>
#include "Cliente.h"

```

```

using namespace std;

```

```

int main() {
    int tamanho;
    string nome, dataNascimento, endereco, telefone;
    cout << "Digite o número de clientes que deseja cadastrar: ";
    cin >> tamanho;

    Cliente* vetclientes = new Cliente [tamanho];

    for (int i = 0; i < tamanho; i++) {
        cout << "\nCadastro do cliente " << i + 1 << ":\n";
        cout << "Nome: ";
        cin.ignore();
        getline(cin, nome);
        cout << "Data de Nascimento: ";
        getline(cin, dataNascimento);
        cout << "Endereço: ";
        getline(cin, endereco);
        cout << "Telefone: ";
        getline(cin, telefone);

        vetclientes[i] = Cliente(nome, dataNascimento, endereco, telefone);
    }

    cout << "\n--- Lista de Clientes ---\n";
    for (int i = 0; i < tamanho; i++) {
        cout << "\nCliente " << i + 1 << ":\n";
        vetclientes[i].imprimirDados();
    }
    delete[]vetclientes;
    return 0;
}

```

A03.

Classe Eletrodomestico.h

```
#include <string>
using namespace std;

class Eletrodomestico {
private:
    string loja;
    string telefone;
    float preco;

public:
    Eletrodomestico(string loja = "", string telefone = "", float preco = 0.0f);
    void setLoja(string loja);
    void setTelefone(string telefone);
    void setPreco(float preco);
    string getLoja();
    string getTelefone();
    float getPreco();
    void imprimirDados();
};
```

Classe Eletrodomestico.cpp

```
#include "Eletrodomestico.h"
#include <iostream>

Eletrodomestico::Eletrodomestico(string loja, string telefone, float preco)
    : loja(loja), telefone(telefone), preco(preco) {}

void Eletrodomestico::setLoja(string loja) {
    this->loja = loja;
}

void Eletrodomestico::setTelefone(string telefone) {
    this->telefone = telefone;
}

void Eletrodomestico::setPreco(float preco) {
    this->preco = preco;
}
```

```

string Eletrodomestico::getLoja(){
    return loja;
}

string Eletrodomestico::getTelefone(){
    return telefone;
}

float Eletrodomestico::getPreco() {
    return preco;
}

void Eletrodomestico::imprimirDados() {
    cout << "Loja: " << loja << "\nTelefone: " << telefone << "\nPreço: R$ " << preco << "\n";
}

```

[main.cpp](#)

```

#include <iostream>
#include "Eletrodomestico.h"

using namespace std;

int main() {
    int numRegistros;
    cout << "Quantos eletrodomésticos você deseja cadastrar? \n" << endl;
    cin >> numRegistros;
    //Eletrodomestico *eletrodomesticos= new Eletrodomestico[numRegistros];
    // ou
    Eletrodomestico eletrodomesticos[numRegistros];

    for (int i = 0; i < numRegistros; i++) {
        string loja, telefone;
        float preco;
        cout << "\nCadastro do Eletrodoméstico " << i + 1 << ":\n";
        cout << "Nome da Loja: ";
        cin.ignore();
        getline(cin, loja);
        cout << "Telefone da Loja: ";
        getline(cin, telefone);
        cout << "Preço: R$ ";
        cin >> preco;
    }
}

```



```

        eletrodomesticos[i] = Eletrodomestico(loja, telefone, preco);
    }

    float menorPreco = eletrodomesticos[0].getPreco();
    float maiorPreco = eletrodomesticos[0].getPreco();
    float somaPreco = 0.0f;

    for (int i = 0; i < numRegistros; i++) {
        float precoAtual = eletrodomesticos[i].getPreco();
        if (precoAtual < menorPreco) {
            menorPreco = precoAtual;
        }
        if (precoAtual > maiorPreco) {
            maiorPreco = precoAtual;
        }
        somaPreco += precoAtual;
    }

    float precoMedio = somaPreco / numRegistros;

    cout << "\n--- Estatísticas de Preços ---\n";
    cout << "Menor Preço: R$ " << menorPreco << "\n";
    cout << "Maior Preço: R$ " << maiorPreco << "\n";
    cout << "Preço Médio: R$ " << precoMedio << "\n";
    return 0;
}

```

A04.

Classe ContaCorrente.h

```
class ContaCorrente {  
private:  
    double saldo;  
  
public:  
    ContaCorrente(double saldoInicial = 0.0);  
    void depositar(double valor);  
    // Taxa de 0.5% aplicada nesta classe  
    bool sacar(double valor);  
    double getSaldo() const;  
};
```

Classe ContaCorrente.cpp

```
#include "ContaCorrente.h"  
  
ContaCorrente::ContaCorrente(double saldoInicial) : saldo(saldoInicial) {}  
  
void ContaCorrente::depositar(double valor) {  
    saldo += valor;  
}  
  
bool ContaCorrente::sacar(double valor) {  
    double taxa = valor * 0.005; // Taxa de 0.5%  
    if (saldo >= valor + taxa) {  
        saldo -= (valor + taxa);  
        return true;  
    }  
    return false;  
}  
  
double ContaCorrente::getSaldo() const {  
    return saldo;  
}
```

Classe ContaEspecial.h

```
#include "ContaCorrente.h"
```

```

class ContaEspecial : public ContaCorrente {
public:
    ContaEspecial(double saldoInicial = 0.0);
    // Taxa de 0.1% aplicada nesta classe
    bool sacarEspecial(double valor);
};

```

Classe ContaEspecial.cpp

```

#include "ContaCorrente.h"

```

```

class ContaEspecial : public ContaCorrente {
public:
    ContaEspecial(double s #include "ContaEspecial.h"

```

```

ContaEspecial::ContaEspecial(double saldoInicial) : ContaCorrente(saldoInicial) {}

```

```

bool ContaEspecial::sacarEspecial(double valor) {
    double taxa = valor * 0.001; // Taxa de 0.1%
    if (getSaldo() >= valor + taxa) {
        // Atualiza o saldo diretamente
        depositar(-(valor + taxa));
        return true;
    }
    return false;
}

double saldoInicial = 0.0);
// Taxa de 0.1% aplicada nesta classe
bool sacarEspecial(double valor);
};

```

main.cpp

```

#include "ContaEspecial.h"
#include <iostream>
#include <string>
using namespace std;

```

```

int main() {
    int tipoConta;
    int opcao;

```

```

cout << "Selecione o tipo de conta:\n";
cout << "1 - Conta Corrente Comum\n";
cout << "2 - Conta Corrente Especial\n";
cin >> tipoConta;

if (tipoConta != 1 && tipoConta != 2) {
    cout << "Tipo de conta inválido.\n";
    return 1;
}
ContaCorrente contaCorrente(0.0);
ContaEspecial contaEspecial(0.0);
do {
    cout << "\nMenu:\n";
    cout << "1 - Depositar\n";
    cout << "2 - Sacar\n";
    cout << "3 - Mostrar Saldo\n";
    cout << "4 - Sair\n";
    cout << "Escolha uma opção: ";
    cin >> opcao;

    switch (opcao) {
        case 1: {
            double valor;
            cout << "Digite o valor para depositar: ";
            cin >> valor;

            if (tipoConta == 1) {
                contaCorrente.depositar(valor);
            } else {
                contaEspecial.depositar(valor);
            }
            break;
        }
        case 2: {
            double valor;
            cout << "Digite o valor para sacar: ";
            cin >> valor;

            bool sucesso;
            if (tipoConta == 1) {
                sucesso = contaCorrente.sacar(valor);
            } else {
                sucesso = contaEspecial.sacarEspecial(valor);
            }
        }
    }
}

```

```
        if (sucesso) {
            cout << "Saque realizado com sucesso.\n";
        } else {
            cout << "Saldo insuficiente.\n";
        }
        break;
    }
    case 3:
        if (tipoConta == 1) {
            cout << "Saldo atual: R$ " << contaCorrente.getSaldo() << "\n";
        } else {
            cout << "Saldo atual: R$ " << contaEspecial.getSaldo() << "\n";
        }
        break;
    case 4:
        cout << "Saindo...\n";
        break;
    default:
        cout << "Opção inválida.\n";
        break;
}

} while (opcao != 4);

return 0;
}
```

A05.

Classe Circulo.h

```
class Circulo {  
private:  
    double raio;  
  
public:  
    Circulo(double raio = 0.0);  
    void setRaio(double raio);  
    double getRaio();  
    double calcularArea();  
    double calcularDiametro();  
    double calcularPerimetro();  
};
```

Classe Circulo.cpp

```
#include "Circulo.h"  
#include <cmath>  
  
Circulo::Circulo(double raio) : raio(raio) {}  
  
void Circulo::setRaio(double raio) {  
    this->raio = raio;  
}  
  
double Circulo::getRaio(){  
    return raio;  
}  
  
double Circulo::calcularArea(){  
    return M_PI * raio * raio;  
}  
  
double Circulo::calcularDiametro(){  
    return 2 * raio;  
}  
  
double Circulo::calcularPerimetro() {  
    return 2 * M_PI * raio;  
}
```

main.cpp

```
#include <iostream>
```

```
#include "Circulo.h"
```

```
using namespace std;
```

```
int main() {
```

```
    Circulo circulos[4] = {Circulo(3.0), Circulo(5.0), Circulo(7.5), Circulo(10.0)};
```

```
    for (int i = 0; i < 4; i++) {
```

```
        cout << "\nCírculo " << i + 1 << ":\n";
```

```
        cout << "Raio: " << circulos[i].getRaio() << endl;
```

```
        cout << "Área: " << circulos[i].calcularArea() << endl;
```

```
        cout << "Diâmetro: " << circulos[i].calcularDiametro() << endl;
```

```
        cout << "Perímetro: " << circulos[i].calcularPerimetro() << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

A06.

Classe Aluno.h

```
#include <string>
using namespace std;

class Aluno {
private:
    string nome;
    string matricula;
    double notas[10];

public:
    Aluno(string nome, string matricula, double notas[10]);
    void setNome(string nome);
    void setMatricula(string matricula);
    void setNotas(double notas[10]);
    string getNome();
    string getMatricula();
    double calcularMedia();
};
```

Classe Aluno.cpp

```
#include "Aluno.h"
#include <iostream>

Aluno::Aluno(string nome, string matricula, double notas[10])
    : nome(nome), matricula(matricula) {
    for (int i = 0; i < 10; i++) {
        this->notas[i] = notas[i];
    }
}

void Aluno::setNome(string nome) {
    this->nome = nome;
}

void Aluno::setMatricula(string matricula) {
    this->matricula = matricula;
}

void Aluno::setNotas(double notas[10]) {
```



```

        for (int i = 0; i < 10; i++) {
            this->notas[i] = notas[i];
        }
    }

    string Aluno::getNome(){
        return nome;
    }

    string Aluno::getMatricula(){
        return matricula;
    }

    double Aluno::calcularMedia(){
        double soma = 0.0;
        for (int i = 0; i < 10; i++) {
            soma += notas[i];
        }
        return soma / 10;
    }

```

[main.cpp](#)

```

#include <iostream>
#include "Aluno.h"

using namespace std;

int main() {
    double notasAluno1[10] = {8.5, 7.0, 9.0, 6.5, 8.0, 7.5, 9.5, 8.0, 7.0, 9.0};
    double notasAluno2[10] = {6.0, 5.5, 7.0, 8.0, 6.5, 7.5, 6.0, 5.5, 6.0, 7.0};

    Aluno aluno1("Xulams", "12345", notasAluno1);
    Aluno aluno2("Godofrina", "67890", notasAluno2);

    cout << "Aluno: " << aluno1.getNome() << "\nMédia: " << aluno1.calcularMedia() << endl;
    cout << "Aluno: " << aluno2.getNome() << "\nMédia: " << aluno2.calcularMedia() << endl;

    return 0;
}

```