

```
/*
 * Carro.cpp
 *
 * Created on: 21 de Ago de 2019
 * Author: escarvalho
 */

#include "Carro.h"

namespace std {

Carro::Carro() {
    // TODO Auto-generated constructor stub
}

int Carro::getCilindrada() const
{
    return cilindrada;
}

vector<Peca>::iterator Carro::getItListaPecas() const
{
    return itListaPecas;
}

vector<Peca> Carro::getListaPecas() const
{
    return listaPecas;
}
```

```
string Carro::getMarca() const
```

```
{  
    return marca;  
}
```

```
string Carro::getModelo() const
```

```
{  
    return modelo;  
}
```

```
void Carro::setCilindrada(int cilindrada)
```

```
{  
    this->cilindrada = cilindrada;  
}
```

```
void Carro::setItListaPecas(vector<Peca>::iterator itListaPecas)
```

```
{  
    this->itListaPecas = itListaPecas;  
}
```

```
void Carro::setListaPecas(vector<Peca> listaPecas)
```

```
{  
    this->listaPecas = listaPecas;  
}
```

```
void Carro::setMarca(string marca)
```

```
{  
    this->marca = marca;  
}
```

```
void Carro::setModelo(string modelo)
```

```
{
    this->modelo = modelo;}

Carro::~Carro() {
    // TODO Auto-generated destructor stub
}

}/* namespace std */

/*
 * Carro.h
 *
 * Created on: 21 de Ago de 2019
 * Author: escarvalho
 */

#ifndef CARRO_H_
#define CARRO_H_

#include <vector>

namespace std {

class Carro {
private:
    string marca;
    string modelo;
    int cilindrada;
    vector<Peca> listaPecas;
    vector<Peca>::iterator itListaPecas;
public:
    Carro();
```

```

        virtual ~Carro();
int getCilindrada() const;
vector<Peca>::iterator getItListaPecas() const;
vector<Peca> getListaPecas() const;
string getMarca() const;
string getModelo() const;
void setCilindrada(int cilindrada);
void setItListaPecas(vector<Peca>::iterator itListaPecas);
void setListaPecas(vector<Peca> listaPecas);
void setMarca(string marca);
void setModelo(string modelo);
};

} /* namespace std */
#endif /* CARRO_H_ */
/*
 * Cliente.cpp
 *
 * Created on: 21 de Ago de 2019
 * Author: escarvalho
 */

#include "Cliente.h"

namespace std {

Cliente::Cliente() {
    // TODO Auto-generated constructor stub
}

```

```
Cliente::~~Cliente() {  
    // TODO Auto-generated destructor stub  
}
```

```
int Cliente::getCc() const  
{  
    return CC;  
}
```

```
string Cliente::getMarcaCarro() const  
{  
    return marcaCarro;  
}
```

```
int Cliente::getMatricula() const  
{  
    return matricula;  
}
```

```
string Cliente::getMorada() const  
{  
    return morada;  
}
```

```
int Cliente::getNif() const  
{  
    return NIF;  
}
```

```
string Cliente::getNome() const  
{
```

```
    return nome;  
}
```

```
long int Cliente::getTelefono() const  
{  
    return telefono;  
}
```

```
void Cliente::setCc(int CC)  
{  
    this->CC = CC;  
}
```

```
void Cliente::setMarcaCarro(string marcaCarro)  
{  
    this->marcaCarro = marcaCarro;  
}
```

```
void Cliente::setMatricula(int matricula)  
{  
    this->matricula = matricula;  
}
```

```
void Cliente::setMorada(string morada)  
{  
    this->morada = morada;  
}
```

```
void Cliente::setNif(int NIF)  
{  
    this->NIF = NIF;  
}
```

```
}
```

```
void Cliente::setNome(string nome)
```

```
{
```

```
    this->nome = nome;
```

```
}
```

```
void Cliente::setTelefone(long int telefone)
```

```
{
```

```
    this->telefone = telefone;
```

```
}
```

```
float Cliente::getTotalAcumulado() const
```

```
{
```

```
    return totalAcumulado;
```

```
}
```

```
void Cliente::setTotalAcumulado(float totalAcumulado)
```

```
{
```

```
    this->totalAcumulado = totalAcumulado;
```

```
}
```

```
}
```

```
/* namespace std */
```

```
/*
```

```
* Cliente.h
```

```
*
```

```
* Created on: 21 de Ago de 2019
```

```
* Author: escarvalho
```

```
*/
```

```
#ifndef CLIENTE_H_
#define CLIENTE_H_

namespace std {

class Cliente {
    string nome;
    int CC;
    int NIF;
    long int telefone;
    string morada;
    string marcaCarro;
    int matricula;
    float totalAcumulado;
public:
    Cliente();
    virtual ~Cliente();

    int getCc() const;
    string getMarcaCarro() const;
    int getMatricula() const;
    string getMorada() const;
    int getNif() const;
    string getNome() const;
    long int getTelefone() const;
    void setCc(int CC);
    void setMarcaCarro(string marcaCarro);
    void setMatricula(int matricula);
    void setMorada(string morada);
    void setNif(int NIF);
    void setNome(string nome);
```



```
void setTelefone(long int telefone);  
float getTotalAcumulado() const;  
void setTotalAcumulado(float totalAcumulado);  
};
```

```
}/* namespace std */  
#endif /* CLIENTE_H_ */
```

```
/*  
 * Compra.cpp  
 *  
 * Created on: 21 de Ago de 2019  
 * Author: escarvalho  
 */
```

```
#include "Compra.h"
```

```
namespace std {
```

```
Compra::Compra() {  
    // TODO Auto-generated constructor stub  
}
```

```
int Compra::getCodigo() const  
{  
    return codigo;  
}
```

```
time_t Compra::getData() const  
{
```

```
    return data;
}
```

```
string Compra::getNome() const
{
    return nome;
}
```

```
float Compra::getValorPago() const
{
    return valorPago;
}
```

```
void Compra::setCodigo(int codigo)
{
    this->codigo = codigo;
}
```

```
void Compra::setData(time_t data)
{
    this->data = data;
}
```

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

```
void Compra::setValorPago(float valorPago)
{
    this->valorPago = valorPago;}
}
```

```
Compra::~Compra() {  
    // TODO Auto-generated destructor stub  
}
```

```
}/* namespace std */
```

```
/*
```

```
* Compra.h
```

```
*
```

```
* Created on: 21 de Ago de 2019
```

```
* Author: escarvalho
```

```
*/
```

```
#ifndef COMPRA_H_
```

```
#define COMPRA_H_
```

```
#include <ctime>
```

```
#include <string>
```

```
namespace std {
```

```
class Compra {
```

```
    time_t data;
```

```
    float valorPago;
```

```
    int codigo;
```

```
    string nome;
```

```
public:
```

```
    Compra();
```

```
    virtual ~Compra();
```

```
    int getCodigo() const;
```

```
    time_t getData() const;
```

```
string getNome() const;
float getValorPago() const;
void setCodigo(int codigo);
void setData(time_t data);
void setNome(string nome);
void setValorPago(float valorPago);
};

}/* namespace std */
#endif /* COMPRA_H_ */

/*
 * Fatura.cpp
 *
 * Created on: 21 de Ago de 2019
 * Author: escarvalho
 */

#include "Fatura.h"

namespace std {

Fatura::Fatura() {
    // TODO Auto-generated constructor stub
}

Fatura::~Fatura() {
    // TODO Auto-generated destructor stub
}
}
```

```
}/* namespace std */  
  
/*  
 * Fatura.h  
 *  
 * Created on: 21 de Ago de 2019  
 * Author: escarvalho  
 */
```

```
#ifndef FATURA_H_  
#define FATURA_H_
```

```
namespace std {
```

```
class Fatura {  
public:  
    Fatura();  
    virtual ~Fatura();  
};
```

```
}/* namespace std */  
#endif /* FATURA_H_ */
```

```
/*  
 * Oficina.cpp  
 *  
 * Created on: 21 de Ago de 2019  
 * Author: escarvalho  
 */
```

```
#include "Oficina.h"
```

```
namespace std {
```

```
Oficina::Oficina() {  
    // TODO Auto-generated constructor stub  
  
}  
  
vector<Carro>::iterator Oficina::getItListaCarrosPecas() const  
{  
    return itListaCarrosPecas;  
}  
  
vector<Cliente>::iterator Oficina::getItListaClientes() const  
{  
    return itListaClientes;  
}  
  
vector<Fatura>::iterator Oficina::getItListaCompras() const  
{  
    return itListaCompras;  
}  
  
vector<Fatura>::iterator Oficina::getItListaFaturas() const  
{  
    return itListaFaturas;  
}  
  
vector<Carro> Oficina::getListCarrosPecas() const  
{  
    return listaCarrosPecas;  
}
```

```
vector<Cliente> Oficina::getListaClientes() const
{
    return listaClientes;
}
```

```
vector<Fatura> Oficina::getListaCompras() const
{
    return listaCompras;
}
```

```
vector<Fatura> Oficina::getListaFaturas() const
{
    return listaFaturas;
}
```

```
void Oficina::setItListaCarrosPecas(vector<Carro>::iterator itListaCarrosPecas)
{
    this->itListaCarrosPecas = itListaCarrosPecas;
}
```

```
void Oficina::setItListaClientes(vector<Cliente>::iterator itListaClientes)
{
    this->itListaClientes = itListaClientes;
}
```

```
void Oficina::setItListaCompras(vector<Fatura>::iterator itListaCompras)
{
    this->itListaCompras = itListaCompras;
}
```

```
void Oficina::setItListaFaturas(vector<Fatura>::iterator itListaFaturas)
```

```
{
    this->itListaFaturas = itListaFaturas;
}

void Oficina::setListaCarrosPecas(vector<Carro> listaCarrosPecas)
{
    this->listaCarrosPecas = listaCarrosPecas;
}

void Oficina::setListaClientes(vector<Cliente> listaClientes)
{
    this->listaClientes = listaClientes;
}

void Oficina::setListaCompras(vector<Fatura> listaCompras)
{
    this->listaCompras = listaCompras;
}

void Oficina::setListaFaturas(vector<Fatura> listaFaturas)
{
    this->listaFaturas = listaFaturas;}

Oficina::~Oficina() {
    // TODO Auto-generated destructor stub
}

void Oficina::incluiCliente()
{
}
```



```

void Oficina::realizaCompra()
{
}

void Oficina::atualizaListaCarrosPecas()
{
}

void Oficina::geraFatura(Cliente cliente)
{}

void Oficina::listaPecasCompradas()
{
    ofstream out("arquivo.txt");
    float total = 0;

    for (itListaCompras = listaCompras.begin(); itListaCompras != listaCompras.end();
++itListaCompras)
    {
        out << itListaCompras->getNome() << " " << itListaCompras->getValorPago() <<
endl;
        cout << itListaCompras->getNome() << " " << itListaCompras->getValorPago()
<< endl;
        total = total + itListaCompras->getValorPago();
    }
    out.close();
}

void Oficina::clienteMaisGasta(vector<Cliente> listaClientes)
{
    float maiorValor = 0;

```

```

        string nome;

        for (itListaClientes = listaClientes.begin(); itListaClientes != listaClientes.end();
        ++itListaClientes)
        {
            if ((*itListaClientes).getTotalAcumulado() > maiorValor)
            {
                maiorValor = (*itListaClientes).getTotalAcumulado();
                nome = (*itListaClientes).getNome();
            }
        }

        cout << nome << " " << maiorValor << endl;
    }

} /* namespace std */

/*
 * Oficina.h
 *
 * Created on: 21 de Ago de 2019
 * Author: escarvalho
 */

#ifndef OFICINA_H_
#define OFICINA_H_

#include "Fatura.h"
#include "Cliente.h"
#include "Compra.h"
#include "Carro.h"
#include <vector>

```

```
#include <fstream>

namespace std {

class Oficina {
private:
    vector <Cliente> listaClientes;
    vector <Cliente>::iterator itListaClientes;
    vector <Fatura> listaFaturas;
    vector <Fatura>::iterator itListaFaturas;
    vector <Compra> listaCompras;
    vector <Compra>::iterator itListaCompras;
    vector <Carro> listaCarrosPecas;
    vector <Carro>::iterator itListaCarrosPecas;
public:
    Oficina();
    virtual ~Oficina();

    vector<Carro>::iterator getItListaCarrosPecas() const;
    vector<Cliente>::iterator getItListaClientes() const;
    vector<Fatura>::iterator getItListaCompras() const;
    vector<Fatura>::iterator getItListaFaturas() const;
    vector<Carro> getListaCarrosPecas() const;
    vector<Cliente> getListaClientes() const;
    vector<Fatura> getListaCompras() const;
    vector<Fatura> getListaFaturas() const;
    void setItListaCarrosPecas(vector<Carro>::iterator itListaCarrosPecas);
    void setItListaClientes(vector<Cliente>::iterator itListaClientes);
    void setItListaCompras(vector<Fatura>::iterator itListaCompras);
    void setItListaFaturas(vector<Fatura>::iterator itListaFaturas);
    void setListaCarrosPecas(vector<Carro> listaCarrosPecas);
    void setListaClientes(vector<Cliente> listaClientes);
```

```
void setListaCompras(vector<Fatura> listaCompras);  
void setListaFaturas(vector<Fatura> listaFaturas);  
void incluiCliente();  
void realizaCompra();  
void atualizaListaCarrosPecas();  
void geraFatura(Cliente cliente);  
void listaPecasCompradas();  
void clienteMaisGasta(vector <Cliente> listaClientes);  
};
```

```
}/* namespace std */  
#endif /* OFICINA_H_ */
```

```
/*  
 * Peca.cpp  
 *  
 * Created on: 21 de Ago de 2019  
 * Author: escarvalho  
 */
```

```
#include "Peca.h"
```

```
namespace std {
```

```
Peca::Peca() {  
    // TODO Auto-generated constructor stub
```

```
}
```

```
int Peca::getCodigo() const
```

```
{
```

```
        return codigo;
    }

    string Peca::getNome() const
    {
        return nome;
    }

    void Peca::setCodigo(int codigo)
    {
        this->codigo = codigo;
    }

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

Peca::~Peca() {
    // TODO Auto-generated destructor stub
}

}/* namespace std */
/*
 * Peca.h
 *
 * Created on: 21 de Ago de 2019
 * Author: escarvalho
 */

#ifndef PECA_H_
#define PECA_H_
```

```
#include <iostream>
#include <string>

namespace std {

class Peca {
private:
    string nome;
    int codigo;
public:
    Peca();
    virtual ~Peca();
    int getCodigo() const;
    string getNome() const;
    void setCodigo(int codigo);
    void setNome(string nome);
};

} /* namespace std */
#endif /* PECA_H_ */
```