

01)

```
CommissionEmployee.h
1 #ifndef COMMISSIONEMPLOYEE_H
2 #define COMMISSIONEMPLOYEE_H
3
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 //Classe que ira representar os funcionarios comissionados
8 class CommissionEmployee
9 {
10 private:
11     string firstName;
12     string lastName;
13     string socialSecurityNumber;
14     //vendas brutas totais
15     double grossSales;
16     //percentagem de comissao
17     double commissionRate;
18 public:
19     //getters declarado como const - nao podem alterar o valor de um a
20     //ComissaoEmployee
21     const string & getFirstName() const { return firstName; }
22     const string & getLastName() const { return lastName; }
23     const string & getSocialSecurityNumber() const { return socialSecurityNumber; }
24     double getGrossSales() const { return grossSales; }
25     double getCommissionRate() const { return commissionRate; }
26     void setFirstName(const string &);
27     void setLastName(const string &);
28     void setSocialSecurityNumber(const string &);
29     void setGrossSales(double);
30     void setCommissionRate(double);
31     void print() const;
32 };
33
34 //retornando o nome
35 string CommissionEmployee::getFirstName() const { return firstName; }
36 //retornando o sobrenome
37 string CommissionEmployee::getLastName() const { return lastName; }
38 //retornando o numero de comissao
39 double CommissionEmployee::getGrossSales() const { return grossSales; }
40 //retornando a taxa de comissao
41 double CommissionEmployee::getCommissionRate() const { return commissionRate; }
42 //configurando o nome
43 void CommissionEmployee::setFirstName(const string &firstName) { firstName = firstName; }
44 //configurando o sobrenome
45 void CommissionEmployee::setLastName(const string &lastName) { lastName = lastName; }
46 //configurando o numero de comissao
47 void CommissionEmployee::setGrossSales(double grossSales) { grossSales = grossSales; }
48 //configurando a taxa de comissao
49 void CommissionEmployee::setCommissionRate(double commissionRate) { commissionRate = commissionRate; }
50 void CommissionEmployee::print() const {
51     cout << "Employee information obtained by get functions: " << endl;
52     cout << "First name is: " << getFirstName() << endl;
53     cout << "Last name is: " << getLastName() << endl;
54     cout << "Social security number is: " << getSocialSecurityNumber() << endl;
55     cout << "Gross sales is: " << getGrossSales() << endl;
56     cout << "Commission rate is: " << getCommissionRate() << endl;
57 }
58
59 //instanciando objeto comissao
60 CommissionEmployee employee("Sue", "Jones", "222-22-2222", 10000
61     cout << "Fixed << setprecision(2);
62
63 //retornando os dados do empregado comissao
64 cout << "Employee information obtained by get functions: " << endl;
65 cout << "First name is: " << employee.getFirstName() << endl;
66 cout << "Last name is: " << employee.getLastName() << endl;
67 cout << "Social security number is: " << employee.getSocialSecurityNumber() << endl;
68 cout << "Gross sales is: " << employee.getGrossSales() << endl;
69 cout << "Commission rate is: " << employee.getCommissionRate() << endl;
70
71 //configurando vendas brutas
72 employee.setGrossSales(8000);
73 //configurando a taxa de comissao
74 employee.setCommissionRate(0.1);
75
76 cout << "Updated employee information output by print function
77 //retornando as novas informacoes do emprego
78 employee.print();
79
80 //retornando os rendimentos do empregado
81 cout << "Employee's earnings: " << employee.earnings() << endl;
82 }
```

02)

```
main.cpp - 02 - Visual Studio Code
1 //Classe que ira representar os funcionarios com salario base e comissao
2 #include "CommissionEmployee.h"
3 #include "BasePlusCommissionEmployee.h"
4 using namespace std;
5
6 int main()
7 {
8     //instanciando objeto comissao
9     CommissionEmployee employee("Sue", "Jones", "222-22-2222", 10000
10     cout << "Fixed << setprecision(2);
11
12     //retornando os dados do empregado comissao
13     cout << "Employee information obtained by get functions: " << endl;
14     cout << "First name is: " << employee.getFirstName() << endl;
15     cout << "Last name is: " << employee.getLastName() << endl;
16     cout << "Social security number is: " << employee.getSocialSecurityNumber() << endl;
17     cout << "Gross sales is: " << employee.getGrossSales() << endl;
18     cout << "Commission rate is: " << employee.getCommissionRate() << endl;
19
20     //configurando vendas brutas
21     employee.setGrossSales(8000);
22     //configurando a taxa de comissao
23     employee.setCommissionRate(0.1);
24
25     cout << "Updated employee information output by print function
26     //retornando as novas informacoes do emprego
27     employee.print();
28
29     //retornando os rendimentos do empregado
30     cout << "Employee's earnings: $" << employee.earnings() << endl;
31
32     //Parte 2
33     cout << "-----Parte 2-----" << endl;
34
35     //instanciando objeto comissao
36     BasePlusCommissionEmployee employee2("Bob", "Lewis", "333-33-3333
37     cout << "Fixed << setprecision(2);
38
39     //retornando os dados do empregado comissao
40     cout << "Employee information obtained by get functions: " << endl;
41     cout << "First name is: " << employee2.getFirstName() << endl;
42     cout << "Last name is: " << employee2.getLastName() << endl;
43     cout << "Social security number is: " << employee2.getSocialSecurityNumber() << endl;
44     cout << "Gross sales is: " << employee2.getGrossSales() << endl;
45     cout << "Commission rate is: " << employee2.getCommissionRate() << endl;
46     cout << "Base salary is: " << employee2.getBaseSalary() << endl;
47
48     //configurando o salario base
49     employee2.setBaseSalary(1000);
50     //configurando a taxa de comissao
51     employee2.setCommissionRate(0.04);
52
53     cout << "Updated employee information output by print function
54     //retornando as novas informacoes do emprego
55     employee2.print();
56
57     //retornando os rendimentos do empregado
58     cout << "Employee's earnings: $" << employee2.earnings() << endl;
59 }
```

03)

INCOMPLETA

```
BasePlusCommissionEmployee.cpp - 03-Incompleto - Visual Studio Code

// calculando rendimentos
22 double BasePlusCommissionEmployee::earnings() const{
23     return getBaseSalary() + CommissionEmployee::earnings();
24 }
25
26
27 //imprimindo objeto
28 void BasePlusCommissionEmployee::print() const{
29
30     cout << "base-salried ";
31
32     //chamando funcao print de CommissionEmployee
33     CommissionEmployee::print();
34     cout << "base salary: " << getBaseSalary();
35 }

//compilando
isaac@isaac-VirtualBox:~/Desktop/atividade/08-heranca/03-incompleto$ g++ -c BasePlusCommissionEmployee.cpp CommissionEmployee.cpp
isaac@isaac-VirtualBox:~/Desktop/atividade/08-heranca/03-incompleto$ g++ *.o main.cpp -o programa
/usr/bin/ld: CommissionEmployee.o: in function 'CommissionEmployee::CommissionEmployee(std::__cxx11::basic_stringchar, std::char_traits<char>, std::allocator<char> > const&, std::__cxx11::basic_stringchar, std::char_traits<char>, std::allocator<char> > const&, std::__cxx11::basic_stringchar, std::char_traits<char>, std::allocator<char> > const&, double, double)':
/usr/bin/ld: CommissionEmployee.o: undefined reference to 'CommissionEmployee::setGrossSales(double)'
/usr/bin/ld: CommissionEmployee.o: in function 'CommissionEmployee::earnings() const':
/usr/bin/ld: CommissionEmployee.o: undefined reference to 'CommissionEmployee::getCommissionRate() const'
/usr/bin/ld: CommissionEmployee.o: in function 'CommissionEmployee::print() const':
/usr/bin/ld: CommissionEmployee.o: undefined reference to 'CommissionEmployee::getFirstName(std::__cxx11::basic_stringchar const&)'
/usr/bin/ld: CommissionEmployee.o: undefined reference to 'CommissionEmployee::getLastName(std::__cxx11::basic_stringchar const&)'
/usr/bin/ld: CommissionEmployee.o: undefined reference to 'CommissionEmployee::getSocialSecurityNumber(std::__cxx11::basic_stringchar const&)'
/usr/bin/ld: CommissionEmployee.o: undefined reference to 'CommissionEmployee::getGrossSales() const'
/usr/bin/ld: CommissionEmployee.o: undefined reference to 'CommissionEmployee::getCommissionRate() const'
/usr/bin/ld: /tmp/ccN9M15.o: in function 'main':
main.cpp:(.text+0x1a): undefined reference to 'CommissionEmployee::getFirstName(std::__cxx11::basic_stringchar const&)'
main.cpp:(.text+0x24f): undefined reference to 'CommissionEmployee::getLastName(std::__cxx11::basic_stringchar const&)'
main.cpp:(.text+0x264): undefined reference to 'CommissionEmployee::getSocialSecurityNumber(std::__cxx11::basic_stringchar const&)'
main.cpp:(.text+0x28f): undefined reference to 'CommissionEmployee::getGrossSales() const'
main.cpp:(.text+0x351): undefined reference to 'CommissionEmployee::getCommissionRate() const'
collect2: error: ld returned 1 exit status
isaac@isaac-VirtualBox:~/Desktop/atividade/08-heranca/03-incompleto$
```

04)

INCOMPLETA

```
main.cpp - 04 - Visual Studio Code

1 #ifndef BASE_H
2 #define BASE_H
3 #include <bits/stdc++.h>
4
5 using namespace std;
6
7 class Base{
8     protected:
9     int protegido;
10
11     private:
12     int privado;
13
14     public:
15     int publico;
16 };
17
18 class Derivada1 : public Base{
19     private:
20     int a, b, c;
21
22     public:
23     Derivada1(){
24         a = protegido;
25         b = privado; // ERRO, nao acessivel
26         c = publico;
27     }
28 };
29
30 class Derivada2 : private Base{
31     private:
32     int a, b, c;
33
34     public:
35     Derivada2(){
36         a = protegido;
37         b = privado; // ERRO, nao acessivel
38         c = publico;
39     }
40 };
41
42 class Derivada3 : protected Base{
43     private:
44     int a, b, c;
45
46     public:
47     Derivada3(){
48         a = protegido;
49         b = privado; // ERRO, nao acessivel
50         c = publico;
51     }
52 };
53
```

05)

The image shows a Visual Studio Code editor with two C++ files open: `Imovel.cpp` and `main.cpp`. The `Imovel.cpp` file contains methods for setting and getting property details like address, neighborhood, area, and type. The `main.cpp` file contains the `main` function which creates a `galpao` object and calls its `print` method. Below the code, the terminal window shows the compilation and execution of the program, displaying the output of the `print` method for the `galpao` object.

```
#include "header/Imovel.h"
#include <bits/stdc++.h>

using namespace std;

void Imovel::setEndereco(){
    cout << "Endereco:" << endl;
    getline(cin, endereco);
}

void Imovel::setBairro(){
    cout << "Bairro:" << endl;
    getline(cin, bairro);
}

void Imovel::setAreaUtil(){
    cout << "Area util:" << endl;
}
```

```
#include "header/Venda.h"
#include <bits/stdc++.h>
using namespace std;

int main(){
    Venda galpao;
    galpao.set();
    cout << "Imovel: " << endl;
    galpao.print();
}
```

```
isaac@isaac-VirtualBox:~/Desktop/atividade/05-heranca/055 g++ -c Cadastro.cpp Imovel.cpp Tipo.cpp Venda.cpp
isaac@isaac-VirtualBox:~/Desktop/atividade/05-heranca/055 g++ Cadastro.o Imovel.o Tipo.o Venda.o main.cpp -o programa
isaac@isaac-VirtualBox:~/Desktop/atividade/05-heranca/055 ./programa
Proprietario:
Nome:
Isaac
Telefone:
40028922
Imovel:
Endereco:
rua tal com a 12
Bairro:
manguinha
Area total:
1000
Area util:
400
Tipo do imovel:
galpao
Valor de venda:
500

-----
Proprietario:
Nome:Isaac
Telefone:40028922
Imovel:
Endereco:rua tal com a 12
Bairro:manguinha
Area total:1000
Area util:400
Quartos:1591800707
Tipo do imovel:
```