

UFPB

João Pessoa

Aluno: Daniel de Queiroz Cavalcanti Matrícula:

Assunto: Resposta da Atividade de classe

QUESTÃO 2)

conta.h

```
#ifndef CONTA_H
```

```
#define CONTA_H
```

```
#include <iostream>
```

```
using namespace std;
```

```
#ifndef
```

```
class IConta
```

```
{
```

```
    public:
```

```
    //construtor
```

```
    virtual void sacar(double valor)=0;;
```

```
    virtual void depositar(double valor)=0;
```

```
    virtual ~IConta()
```

```
    {
```

```
    }
```

```
};
```

```
class conta: public IConta
```

```

{
public:
    void nomeCliente(){
        std::string getnomeCliente();
    }
    void salarioMensal(){
        std::string setsalarioMensal();
    }
    void numeroConta(){
        std::string setsalarioMensal();
    }
    void saldo(){
        std::string setsaldo();
    }
};

//destrutor
    ~conta();

    char cliente[80];
    double limite;

    virtual void sacar(double valor);
    virtual void depositar(double valor);
    void imprimirSaldo();
    void definirLimite();

```

```
        void salarioMensual::setdefinirLimite(std::string DefinirLimite){
            this->DefinirLimite = 2*salarioMensual;
        }

        private :
            double saldo;
            double salario;
    };

#endif
```

conta.cpp

```
#include "conta.h"
#include <stdio.h>
// class's header file

// class constructor
conta::conta()
{

    saldo = 0;
    // insert your code here
}

// class destructor
conta::~~conta()
{
}
```

```
void conta :: sacar(double valor) {  
    saldo = saldo - valor;  
  
}
```

```
void conta :: depositar(double valor) {  
  
    saldo = saldo + valor;  
}
```

```
void conta :: imprimirSaldo() {  
    printf("%lf", saldo);  
}
```

ContaEspecial.cpp

```
#ifndef CONTA_H  
#define CONTA_H  
  
#ifndef  
class ContaEspecial  
{  
    public:  
    //construtor  
  
    contaEspecial::conta()  
  
    definirLimite()  
public class definirLimite extends salarioMensal{  
    double salarioMensal;
```

```

double definirLimite;

public double getDefinirLimite() {
    return this.salarioMensal * 3;
}

void salarioMensal::setdefinirLimite(std::string DefinirLimite){
    this->DefinirLimite = 3*salarioMensal;
}
}

```

main.h

```

#include <cstdlib>
#include <iostream>
#include "conta.h"

using namespace std;

int main(int argc, char *argv[])
{
    conta c;
    ContaEspecial d;

    c.depositar(60);
    c.sacar(30);
    c.imprimirSaldo();
    c.definirLimite();
    d.depositar();
}

```

```
d.sacar();  
d.definirLimite();
```

```
system("PAUSE");  
return EXIT_SUCCESS;  
}
```

QUESTÃO 3)

Funcionario.cpp

```
#include <iostream>  
using namespace std;  
  
class Funcionario  
{  
public:  
    virtual String nome() = 0;  
    std::string getnome();  
    virtual int matricula() = 0;  
    std::string getmatricula();  
    int CalcularSalario();  
    // Superclasses devem ter o destrutor virtual  
    virtual ~Funcionario()  
    {  
    }  
};
```

Assalariado.cpp

```
#include <iostream>

using namespace std;

class Assalariado
{
public:
    double salario();
    virtual double CalcularSalario()=0;
}
```

Comissionado.cpp

```
#include <iostream>

using namespace std;

class Comissionado
{
public:
    double vendasSemanais();
    double PercentualComissao();
    virtual double calcularSalario()=0;
}
```

main.h

```
#include <cstdlib>
#include <iostream>
#include "conta.h"
```

```
using namespace std;

int main(int argc, char *argv[])
{
    Funcionario c;
    SistemaGerenciaFolha d;
    Assalariado e;
    Comissionado f;

    c.nomedofuncionario();
    c.matricula();
    c.salario();
    d.funcionarios();
    d.CalculaValorTotalFolha();
    d.ConsultaSalario();
    e.salariof();
    e.CalculaSalario();
    f.vendas();
    f.comissao();
    f.calculaSalario();

    system("PAUSE");
    return EXIT_SUCCESS;
}
```