UEPB	UNIVERSIDADE ESTADUAL DA PARAÍBA CENTRO DE CIÊNCIAS E TECNOLOGIA CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO	
ATIVIDADE DE ESTUDO VIII	**ANO** 2021.1	
CURSO	Ciência da Computação	
DISCIPLINA	Linguagem de Programação II	
PROFESSOR		**TITULAÇÃO**
Jucelio Soares dos Santos		Mestrado
NOME	Lucas de Lucena Siqueira	
MATRÍCULA	201080354	**CONCEITO**
DATA		

DESCRIÇÃO DA ATIVIDADE

1. O que você entende por encapsulamento? Para que serve? E como aplicar?

   R/ O encapsulamento é um pilar da orientação a objetos que permite simplificar a programação e proteger informações que sejam consideradas mais sigilosas e fiquem muito expostas. Além dessas vantagens, o encapsulamento permite que o programa fique mais legível e reutilizável. Para que o encapsulamento seja possível, os atributos declarados na classe devem estar declarados como privados e logo após os métodos setters e getters devem ser criados com seus respectivos retornos.

2. Antes de rever os conceitos, diga para que servem os modificadores de acesso:
   a. public | Esse modificador permite o acesso a partir de qualquer classe.
   b. private | Esse modificador permite o acesso apenas a partir da própria classe.
   c. Protected | O package organiza as classes evitando possíveis conflitos.
   d. default | Esse modificador é aplicado quando nenhum outro é aplicado de forma explicita.
3. Pense e responda o que faria você decidir se um método deve ser usado como privado.

   R/ Quando acredito que o método não deva ser utilizado indevidamente em certas partes do código.

4. Crie classes chamadas Usuário e Hacker. Hacker possui o método main(). A classe Usuário possui os atributos login e senha. Inicialmente, não use encapsulamento e faça com que no método main() de Hacker seja possível modificar as informações (login,senha), inicialmente definidas, de um objeto da classe Usuário que você mesmo criar. Em seguida, aplique encapsulamento e verifique que Hacker terá suas tentativas frustradas.

R/

```java
package Atividade08.Entidades;

public class Usuario {
    public String user;
    public String password;

    public Usuario(String user, String password) {
        this.user = user;
        this.password = password;
    }
}

```

```java
package Atividade08.Aplicação;

import Atividade08.Entidades.Usuario;

public class Hacker {
    public static void main(String []args) {
        Usuario user1 = new Usuario( user: "lucas",  password: "qwerty");
        //Print do nome de usuário e senha definidos
        System.out.printf("""
                Login: %s
                Senha: %s""", user1.user, user1.password);

        //Alterando os dados definidos
        user1.user = "silvio";
        user1.password = "santos";

        //Print depois do hacker ter sido hackeado
        System.out.printf("""

                Hackeado!
                Login: %s
                Senha: %s""", user1.user, user1.password);
    }
}

```

```
Run:  Hacker
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "
Login: lucas
Senha: qwerty
Hackeado!
Login: silvio
Senha: santos
Process finished with exit code 0
```

Com o encapsulamento:

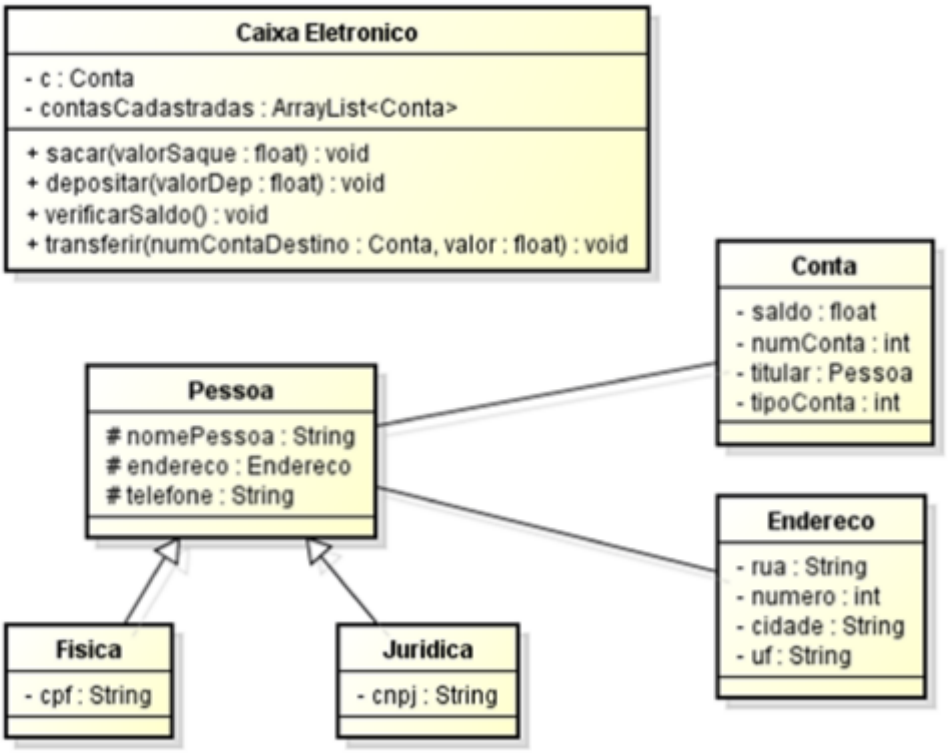
Hacker.java | Usuario.java

```java
package Atividade08.Entidades;

public class Usuario {
    private String user;
    private String password;

    public Usuario(String user, String password) {
        this.user = user;
        this.password = password;
    }

    public String getUser() {
        return user;
    }

    public void setUser(String user) {
        this.user = user;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

Hacker.java | Usuario.java

```java
package Atividade08.Aplicação;

import Atividade08.Entidades.Usuario;

public class Hacker {
    public static void main(String []args) {
        Usuario user1 = new Usuario( user: "lucas",  password: "qwerty");
        //Print do nome de usuário e senha definidos
        System.out.printf("""
                Login: %s
                Senha: %s""", user1.user, user1.password);

        //Alterando os dados definidos
        user1.user = "silvio";
        user1.password = "santos";

        //Print depois do hacker ter sido hackeado
        System.out.printf("""

                Hackeado!
                Login: %s
                Senha: %s""", user1.user, user1.password);
    }
}
```

5. Crie o projeto com as classes ilustradas na Figura abaixo separando os pacotes, assim como criando todos os métodos públicos para acesso aos atributos private e protected.



Classe Endereco:

```java
package Atividade08.Entidades;

public class Endereco {
    private String rua;
    private Integer numero;
    private String cidade;
    private String uf;

    public Endereco() {

    }

    public Endereco(String rua, Integer numero, String cidade, String uf) {
        this.rua = rua;
        this.numero = numero;
        this.cidade = cidade;
        this.uf = uf;
    }

    public String getRua() {
        return rua;
    }

    public void setRua(String rua) {
        this.rua = rua;
    }

    public Integer getNumero() {
        return numero;
    }

    public void setNumero(Integer numero) {
        this.numero = numero;
    }

    public String getCidade() {
        return cidade;
    }

    public void setCidade(String cidade) {
        this.cidade = cidade;
```

```java
    }

    public String getUf() {
        return uf;
    }

    public void setUf(String uf) {
        this.uf = uf;
    }
}

```

Classe Pessoa:

```java
package Atividade08.Entidades;

public class Pessoa {
    protected String nomePessoa;
    protected Endereco endereco;
    protected String telefone;

    public Pessoa() {

    }

    public Pessoa(String nomePessoa, Endereco endereco, String telefone) {
        this.nomePessoa = nomePessoa;
        this.endereco = endereco;
        this.telefone = telefone;
    }

    public String getNomePessoa() {
        return nomePessoa;
    }

    public void setNomePessoa(String nomePessoa) {
        this.nomePessoa = nomePessoa;
    }

    public Endereco getEndereco() {
        return endereco;
    }

    public void setEndereco(Endereco endereco) {
        this.endereco = endereco;
    }

    public String getTelefone() {
        return telefone;
    }

    public void setTelefone(String telefone) {
        this.telefone = telefone;
    }
}
```

Classe Jurídica

```java
package Atividade08.Entidades;

public class Juridica extends Pessoa{
    private String cnpj;

    public Juridica() {

    }

    public Juridica(String cnpj) {
        this.cnpj = cnpj;
    }

    public Juridica(String nomePessoa, Endereco endereco, String telefone, String cnpj) {
        super(nomePessoa, endereco, telefone);
        this.cnpj = cnpj;
    }

    public String getCnpj() {
        return cnpj;
    }

    public void setCnpj(String cnpj) {
        this.cnpj = cnpj;
    }
}
```

Classe Física

```java
package Atividade08.Entidades;

public class Fisica extends Pessoa{
    private String cpf;

    public Fisica(){

    }

    public Fisica(String cpf) {
        this.cpf = cpf;
    }

    public Fisica(String nomePessoa, Endereco endereco, String telefone, String cpf) {
        super(nomePessoa, endereco, telefone);
        this.cpf = cpf;
    }

    public String getCpf() {
        return cpf;
    }

    public void setCpf(String cpf) {
        this.cpf = cpf;
    }
}
```

Classe Conta

```java
package Atividade08.Entidades;

public class Conta {
    private Float saldo;
    private Integer numConta;
    private Pessoa titular;
    private Integer tipoConta;

    public Conta() {

    }

    public Conta(Float saldo, Integer numConta, Pessoa titular, Integer tipoConta) {
        this.saldo = saldo;
        this.numConta = numConta;
        this.titular = titular;
        this.tipoConta = tipoConta;
    }

    public Float getSaldo() {
        return saldo;
    }

    public void setSaldo(Float saldo) {
        this.saldo = saldo;
    }

    public Integer getNumConta() {
        return numConta;
    }

    public void setNumConta(Integer numConta) {
        this.numConta = numConta;
    }

    public Pessoa getTitular() {
        return titular;
    }

    public void setTitular(Pessoa titular) {
        this.titular = titular;
    }

    public Integer getTipoConta() {
        return tipoConta;
    }

    public void setTipoConta(Integer tipoConta) {
        this.tipoConta = tipoConta;
    }
}
```

Classe CaixaEletronico

```java
package Atividade08.Entidades;
import java.util.ArrayList;

public class CaixaEletronico {
    private Conta c;
    private ArrayList<Conta> contasCadastradas;
    Float saldoAtual;

    {
        assert false;
        saldoAtual = c.getSaldo();
    }

    public CaixaEletronico() {

    }

    public CaixaEletronico(Conta c, ArrayList<Conta> contasCadastradas) {
        this.c = c;
        this.contasCadastradas = contasCadastradas;
    }

    public Conta getC() {
        return c;
    }

    public void setC(Conta c) {
        this.c = c;
    }

    public void addConta(Conta c) {
        contasCadastradas.add(c);
    }

    public void removeConta(Conta c) {
        contasCadastradas.remove(c);
    }
    public ArrayList<Conta> getContasCadastradas() {
        return contasCadastradas;
    }

    public void setContasCadastradas(ArrayList<Conta> contasCadastradas) {
        this.contasCadastradas = contasCadastradas;
    }

    public void sacar(Float valor) {
        saldoAtual -= valor;
        if(saldoAtual < 0) {
            System.out.println("O seu saldo atual é insuficiente para reaizar o saque.");
        }
        else {
            c.setSaldo(saldoAtual);
        }
    }

    public void transferir(Conta contaDestino, Float valor) {
        saldoAtual -= valor;
        if(saldoAtual < 0) {
            System.out.println("O seu saldo atual é insuficiente para reaizar a transferência.");
        }
        else {
            Float saldoContaDestinoAtual = contaDestino.getSaldo();
            saldoContaDestinoAtual += valor;
            contaDestino.setSaldo(saldoContaDestinoAtual);
        }
    }

    public void verificarSaldo() {
        System.out.println("Saldo atual: " + c.getSaldo());
    }
}
```

6. Considerando que a Classe Jurídico é filha da classe Pessoa (Jurídico herda Pessoa), é possível acessar o atributo protected nomePessoa de forma direta? Por quê? Se o atributo fosse do tipo private, seria possível o acesso de forma direta?

   R/ O acesso é possível quando a declaração é protected, pois esse modificador autoriza o acesso pelo fato de ambas as classes estarem no mesmo pacote ou caso uma esteja herdando a outra. Caso a declaração fosse private o acesso não seria possível, pois esse modificador permite o acesso apenas dentro da sua própria classe.

7. Crie 3 atributos na classe Conta: limiteTransferencia, limiteSaque e taxaJuros, sendo que o limite de transferência é de R$ 3.000,00, o limite de saque é de R$ 1.000,00 e o limite de transferência é de R$ 2.500,00, e atribua os modificadores de modo que estes atributos possam ser acessados e compartilhados por todas as instâncias de conta, assim como não permita a mudança destes atributos em tempo de execução.

   R/

```java
package Atividade08.Entidades;

public class Conta {
    private Float saldo;
    private Integer numConta;
    private Pessoa titular;
    private Integer tipoConta;
    protected Double limiteTransferencia = 3000.0;
    protected Double limiteSaque = 10000.0;
    protected Double taxaJuros = 2500.0;

```

8. Dadas as classes Pessoa, PFisica e Vendedor logo abaixo. Responda: como será a saída do console quando este projeto for executado?

```
class Pessoa {
 Pessoa(){
 System.out.println("Pessoa Construct");
 }
}
```

```
class PFisica extends Pessoa{
 PFisica() {
 System.out.println("Pessoa Fisica");
 }
}
```

```
public class Vendedor extends PFisica{

 Vendedor() {
 System.out.println(“Vendedor”);
 }

 public static void main(String[] args) {
 Vendedor v = new Vendedor();
 }
}
```

R/ A saída seria “Vendedor”.