

Aula 07 - Encapsulamento

1. Especificadores de Acesso / Modificador de Acesso

Os especificadores de acesso ou modificadores de acesso, devem ser utilizados para definir a visibilidade de **classes, atributos e métodos**.

Visibilidade	public	protected	default	private
A partir da mesma classe	✓	✓	✓	✓
Qualquer classe no mesmo pacote	✓	✓	✓	✗
Qualquer classe filha no mesmo pacote	✓	✓	✓	✗
Qualquer classe filha em pacote diferente	✓	✓	✗	✗
Qualquer classe em pacote diferente	✓	✗	✗	✗

public:

Uma declaração com o modificador public pode ser acessada de qualquer lugar e por qualquer entidade que possa visualizar a classe a que ela pertence.

private:

Os membros da classe definidos como não podem ser acessados ou usados por nenhuma outra classe. Esse modificador não se aplica às classes, somente para seus métodos e atributos. Esses atributos e métodos também não podem ser visualizados pelas classes herdadas.

protected:

O modificador protected torna o membro acessível às classes do mesmo pacote ou através de herança, seus membros herdados não são acessíveis a outras classes fora do pacote em que foram declarados.

default (padrão):

A classe e/ou seus membros são acessíveis somente por classes do mesmo pacote, na sua declaração não é definido nenhum tipo de modificador, sendo este identificado pelo compilador.

Entretanto o que vamos utilizar neste momento, são:

- public
- private

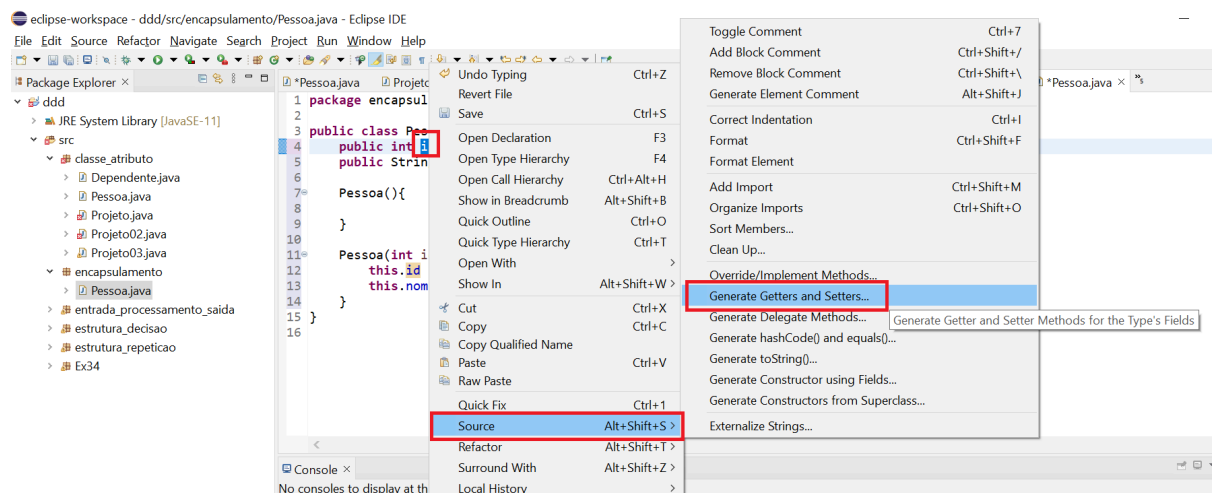
2. Encapsulamento - Getters and Setters

O Encapsulamento oferece um controle de acesso aos atributos e métodos de uma classe. É uma forma eficiente de proteger os dados manipulados dentro da classe, além de determinar onde esta classe poderá ser manipulada.

No caso abaixo, temos uma classe com seus atributos como public. Isso, tecnicamente é possível, porém não é o adequado.

```
public class Pessoa {  
    public int id;  
    public String nome;  
  
    Pessoa() {  
  
    }  
  
    Pessoa(int id, String nome) {  
        this.id = id;  
        this.nome = nome;  
    }  
}
```

O ideal é criarmos Getters e Setters para cada atributo. Para criá-los podemos clicar com botão direito em cada um desses atributos > Source > Generate Getters and Setters:



Nessa outra tela, podemos escolher se queremos criar os Getters and Setters para todos os atributos ou apenas para um só, no caso, vamos criar para todos:

Generate Getters and Setters

Select getters and setters to create:

▼

☒ id

☒ getId()

☒ setId(int)

>

☐ nome

Select All

Deselect All

Select Getters

Select Setters

☐ Allow setters for final fields (remove 'final' modifier from fields if necessary)

Insertion point:

After 'id'

Sort by:

Fields in getter/setter pairs

Access modifier

☒ public☐ protected☐ package☐ private

☐ final☐ synchronized

☐ Generate method comments

The format of the getters/setters may be configured on the [Code Templates](#) preference page.

2 of 4 selected.

?

Generate

Cancel

E, com isso, teremos o seguinte resultado:

```
public class Pessoa {  
    public int id;  
    public String nome;  
  
    public int getId() {  
        return id;  
    }  
}
```

```

    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    Pessoa() {

    }

    Pessoa(int id, String nome) {
        this.id = id;
        this.nome = nome;
    }
}

```

Mas ainda não está correto! Agora precisamos deixar os atributos como privados e deixarmos apenas os métodos Get e Set como públicos, veja:

```

public class Pessoa {
    private int id;
    private String nome;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }
}

```

```

    public void setNome(String nome) {
        this.nome = nome;
    }

    Pessoa() {

    }

    Pessoa(int id, String nome) {
        this.id = id;
        this.nome = nome;
    }
}

```

Os Getters e Setters são métodos que serão responsáveis por ler e escrever valores em atributos que não mais serão public e, sim, private.

3. Programa utilizando Encapsulamento - Getters and Setters

Agora vamos desenvolver um programa simples que irá acessar os atributos utilizando os métodos Gets e Sets criados por nós.

```

import java.util.Scanner;

public class Programa {

    public static void main(String[] args) {
        Scanner ler = new Scanner(System.in);

        // Cria um array contendo 10 posições para armazenar objetos
        Pessoa[] listaPessoas = new Pessoa[10];

        int id;
        String nome;

        for (int i=0; i<=2; i++) {

            Pessoa p = new Pessoa();

            // Obtem o id através da digitação do usuário
            System.out.printf("Digite o id da pessoa: ");
            id = ler.nextInt();

```

```

        // Popula o atributo "id" através do encapsulamento "Set"
        p.setId(id);

        // Obtem o nome através da digitação do usuário
        System.out.printf("Digite o nome da pessoa: ");
        nome = ler.next();

        // Popula o atributo "nome" através do encapsulamento "Set"
        p.setNome(nome);

        // Adiciona o objeto no array "listaPessoas"
        listaPessoas[i] = p;
    }

    for (int i=0; i<=2; i++) {
        System.out.printf("\nPessoa %d criada com sucesso (%s)!",
listaPessoas[i].getId(), listaPessoas[i].getNome());
    }
}
}

```

4. Outro exemplo utilizando get e set

Pessoa.java

```

package exemplo_aula07;

public class Pessoa {
    private int id;
    private String nome;
    private double saldo;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }
}

```

```

    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public double getSaldo() {
        return saldo;
    }

    Pessoa() {

    }

    Pessoa(int id, String nome) {
        this.id = id;
        this.nome = nome;
    }

    public void Depositar(double valor) {
        this.saldo = this.saldo + valor + 0.50;
        // gravar o histórico do depósito
        // mandar um email para o cliente
        // mandar notificação no celular
    }
}

```

Programa.java

```

package exemplo_aula07;

import java.util.Scanner;

public class Programa {

    public static void main(String[] args) {
        Scanner ler = new Scanner(System.in);

        int id;
        String nome;
        double valor;

        System.out.printf("Digite o id: ");
    }
}

```

```

        id = ler.nextInt();

        System.out.printf("Digite o nome: ");
        nome = ler.next();

        System.out.printf("Digite o valor do depósito: ");
        valor = ler.nextDouble();

        Pessoa a = new Pessoa();

        a.setId(id);
        a.setNome(nome);
        a.Depositar(valor);

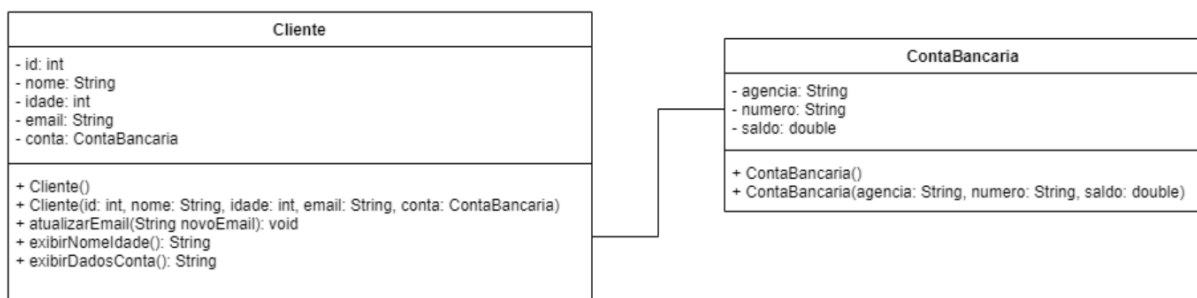
        System.out.printf("O id é: " + a.getId());
        System.out.printf("\nO nome é: " + a.getNome());
        System.out.printf("\nO saldo é: " + a.getSaldo());

        ler.close();
    }
}

```

Exercícios

31. Crie as classes conforme o Diagrama de Classe (UML) abaixo. Crie Getters e Setters para todos os atributos das classes. Crie um programa que utilize essas classes para cadastrar 5 clientes em uma lista de clientes e pergunte para cada cliente se ele tem ou não conta bancária. Caso o cliente tenha, permita ele cadastrar os dados da conta bancária. Ao final exibir todos os clientes e suas respectivas contas bancárias, se houver.



ContaBancaria.java

```

package Ex29;

public class ContaBancaria {
    private String agencia;

```



```

private String numero;
private double saldo;

public String getAgencia() {
    return agencia;
}

public void setAgencia(String agencia) {
    this.agencia = agencia;
}

public String getNumero() {
    return numero;
}

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

public double getSaldo() {
    return saldo;
}

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

public ContaBancaria() {
}

public ContaBancaria(String agencia, String numero, double saldo) {
    this.agencia = agencia;
    this.numero = numero;
    this.saldo = saldo;
}
}

```

Cliente.java

```

package Ex29;

public class Cliente {
    private int id;
    private String nome;
    private int idade;
    private String email;
    private ContaBancaria conta;

    public int getId() {

```

```
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public int getIdade() {
        return idade;
    }

    public void setIdade(int idade) {
        this.idade = idade;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public ContaBancaria getConta() {
        return conta;
    }

    public void setConta(ContaBancaria conta) {
        this.conta = conta;
    }

    Cliente(){
    }
}
```

```

    Cliente(int id, String nome, int idade, String email, ContaBancaria
conta){
        this.id = id;
        this.nome = nome;
        this.idade = idade;
        this.email = email;
        this.conta = conta;
    }

    public void atualizarEmail(String novoEmail) {
        this.email = novoEmail;
    }

    public String exibirNomeIdade() {
        return ("\nnome: " + this.nome + "\nidade: " + this.idade +
"\nemail: " + this.email);
    }

    public String exibirDadosConta() {
        if(this.conta != null){
            return "\nConta: " + this.conta.getNumero() + "\nAgência: "
+ this.conta.getAgencia() + "\nSaldo: " + this.conta.getSaldo() + "\n";
        }
        else {
            return("\nCliente sem conta cadastrada");
        }
    }
}

```

Programa.java

```

package Ex29;

import java.util.Scanner;

public class Programa {

    public static void main(String[] args) {
        int iter = 2;

        Scanner read = new Scanner(System.in);
        Cliente[] listaClientes = new Cliente[iter];
        Cliente c;
        String nome;
    }
}

```

```
int idade;
String email;

String hasAccount;

ContaBancaria conta;
String agencia;
String numero;
double saldo;

for(int i=0; i<iter; i++) {
    c = new Cliente();

    System.out.printf("nome do %do cliente: ", i+1);
    c.setNome(read.next());

    System.out.printf("idade do %do cliente: ", i+1);
    c.setIdade(read.nextInt());

    System.out.printf("email do %do cliente: ", i+1);
    c.setEmail(read.next());

    System.out.printf("O cliente tem conta bancária? [S/N]");
    hasAccount = read.next();

    if(hasAccount.equals("S")) {
        conta = new ContaBancaria();

        System.out.printf("Informe os dados da conta:\n");

        System.out.printf("agencia: ");
        conta.setAgencia(read.next());

        System.out.printf("núemro da conta: ");
        conta.setNumero(read.next());

        System.out.printf("saldo: ");
        conta.setSaldo(read.nextDouble());

        c.setConta(conta);
    } else {
        c.setConta(null);
    }
}
```

```

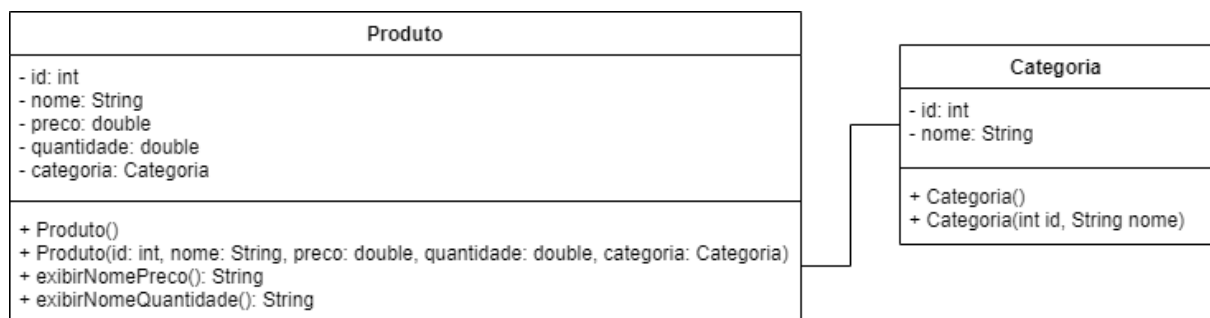
        listaClientes[i] = c;
    }

    for(int i=0; i<iter; i++) {
        System.out.printf(listaClientes[i].exibirNomeIdade());
        System.out.printf(listaClientes[i].exibirDadosConta());
    }

    read.close();
}
}

```

30. Crie as classes conforme o Diagrama de Classe (UML) abaixo. Crie Getters e Setters para todos os atributos das classes. Crie um programa que utilize essas classes para cadastrar 5 produtos em uma lista de produtos e pergunte para cada produto se ele tem ou não uma categoria. Caso o produto tenha, permita ele cadastrar os dados da categoria. Ao final, exibir todos os produtos e suas respectivas categorias, se houver.



Categoria.java

```

package Ex30;

public class Categoria {
    private int id;
    private String nome;

    public Categoria() {

    }

    public Categoria(int id, String nome){
        this.id = id;
        this.nome = nome;
    }
}

```

```

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return nome;
    }

    public void setName(String nome) {
        this.nome = nome;
    }
}

```

Produto.java

```

package Ex30;

public class Produto {
    private int id;
    private String nome;
    private double preco;
    private double quantidade;
    private Categoria categoria;

    public Produto() {

    }

    public Produto(int id, String nome, double preco, double
quantidade, Categoria categoria) {
        this.id = id;
        this.nome = nome;
        this.preco = preco;
        this.quantidade = quantidade;
        this.categoria = categoria;
    }

    public String exibirNomePreco() {
        return this.nome+" "+this.preco;
    }
}

```

```
public String exibirNomeQuantidade() {  
    return this.nome+" "+this.quantidade;  
}  
  
public int getId() {  
    return id;  
}  
  
public void setId(int id) {  
    this.id = id;  
}  
  
public String getNome() {  
    return nome;  
}  
  
public void setNome(String nome) {  
    this.nome = nome;  
}  
  
public double getPreco() {  
    return preco;  
}  
  
public void setPreco(double preco) {  
    this.preco = preco;  
}  
  
public double getQuantidade() {  
    return quantidade;  
}  
  
public void setQuantidade(double quantidade) {  
    this.quantidade = quantidade;  
}  
  
public Categoria getCategoria() {  
    return categoria;  
}  
  
public void setCategoria(Categoria categoria) {
```

```
        this.categoria = categoria;
    }
}
```

Programa.java

```
package Ex30;

import java.util.Scanner;

public class Programa {
    public static void main(String[] args) {
        Produto produto;
        Categoria categoria = null;
        Produto[] listaProdutos = new Produto[5];
        Scanner ler = new Scanner(System.in);
        System.out.println("Bem vindo ao cadastro de produtos!");

        for(int i=0; i<=4; i++){
            categoria = null;
            System.out.println("Qual o nome do produto?");
            String nProduto = ler.nextLine();
            System.out.println("Qual o preço?");
            double preco = ler.nextDouble();
            System.out.println("Quantidade? ");
            int quantidade = ler.nextInt();

            ler.nextLine();

            System.out.println("O produto possui alguma categoria? Se
sim Digite S, se não, digite N");
            String userResp = ler.nextLine();

            if(userResp.equalsIgnoreCase("S")){
                System.out.println("Qual o nome da categoria??");
                String categoriaNome = ler.nextLine();
                categoria = new Categoria(i, categoriaNome);
            }

            produto = new Produto(i, nProduto, preco, quantidade,
categoria);
            listaProdutos[i] = produto;
        }
    }
}
```



```
        for(int i=0; i<=4; i++){
            System.out.println("----Produto----");
            System.out.println("Nome: "+ listaProdutos[i].getNome());
            System.out.println("Preco: "+ listaProdutos[i].getPreco());
            System.out.println("Quantidade: "+
listaProdutos[i].getQuantidade());

            if(listaProdutos[i].getCategoria() != null)
                System.out.println("Categoria: "+
listaProdutos[i].getCategoria().getNome());
            else
                System.out.println("Produto não possui categoria!");
        }
    }
}
```