

## Agenda

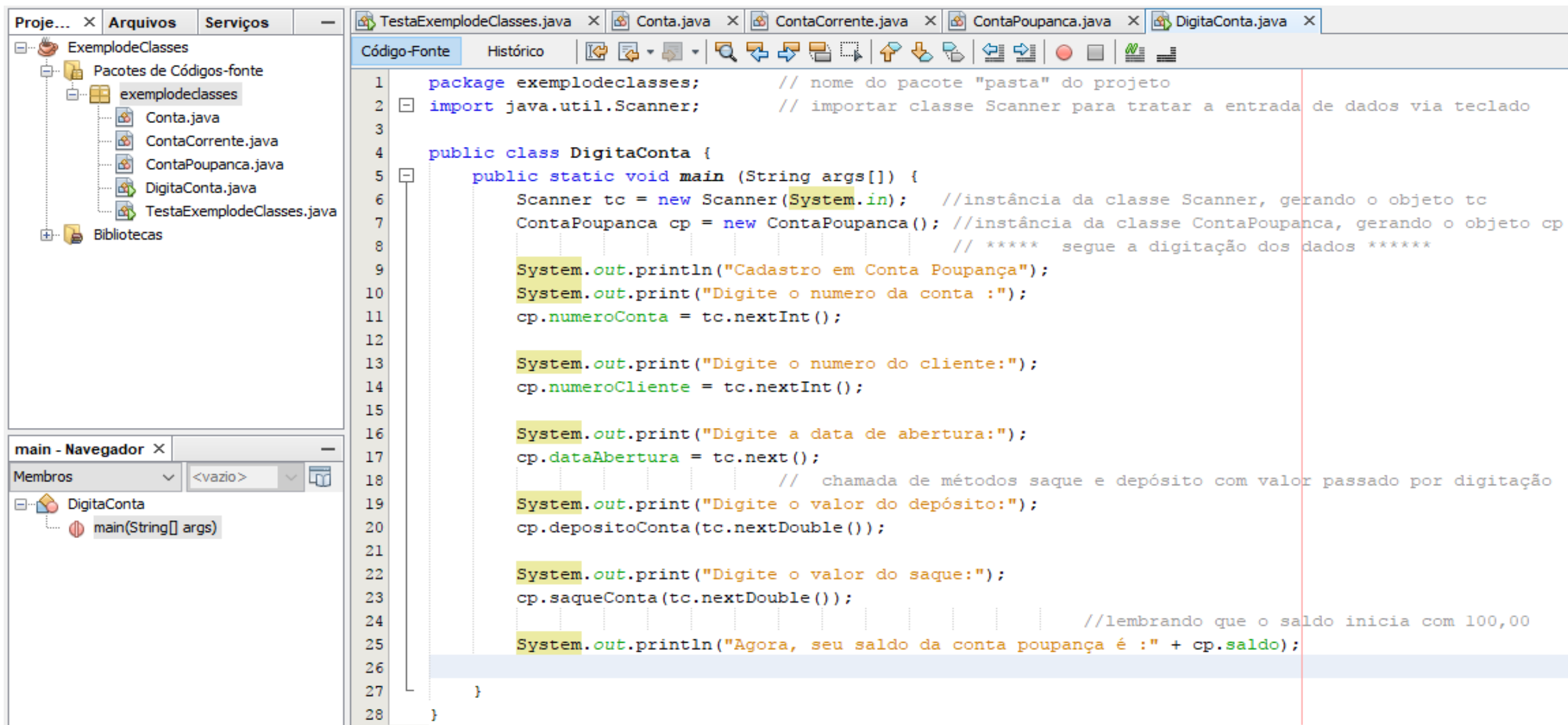
- Conclusão do projeto ExplodeClasses e do programa DigitaConta
- Melhorias no programa DigitaConta
- Gerar outro programa no mesmo projeto com o nome de MenuDigitaDados
- Manipulando objetos múltiplos no mesmo programa (classes ContaPoupança e ContaCorrente)
- Adicionando loopings para consistências e reiniciar o programa (**while** ou **do while**)

Como ficará o Projeto **ExplodeClasses** :

Classes concretas: **Conta, ContaCorrente, ContaPoupanca**

Classes “programas”: **TestaExplodeClasses, DigitaConta e MenuDigitaDados**

# Resolução da atividade anterior- Criar um programa **DigitaConta** no mesmo projeto para entrar com os dados via teclado



The screenshot shows an IDE with the following components:

- Project Explorer (left):** Shows the project structure with a package named `exemplodeclasses` containing files `Conta.java`, `ContaCorrente.java`, `ContaPoupanca.java`, `DigitaConta.java`, and `TestaExemplodeClasses.java`.
- Editor (center):** Displays the source code for `DigitaConta.java`. The code is as follows:

```
1 package exemplodeclasses;           // nome do pacote "pasta" do projeto
2 import java.util.Scanner;           // importar classe Scanner para tratar a entrada de dados via teclado
3
4 public class DigitaConta {
5     public static void main (String args[]) {
6         Scanner tc = new Scanner(System.in); //instância da classe Scanner, gerando o objeto tc
7         ContaPoupanca cp = new ContaPoupanca(); //instância da classe ContaPoupanca, gerando o objeto cp
8         // ***** segue a digitação dos dados *****
9         System.out.println("Cadastro em Conta Poupança");
10        System.out.print("Digite o numero da conta :");
11        cp.numeroConta = tc.nextInt();
12
13        System.out.print("Digite o numero do cliente:");
14        cp.numeroCliente = tc.nextInt();
15
16        System.out.print("Digite a data de abertura:");
17        cp.dataAbertura = tc.next();
18        // chamada de métodos saque e depósito com valor passado por digitação
19        System.out.print("Digite o valor do depósito:");
20        cp.depositoConta(tc.nextDouble());
21
22        System.out.print("Digite o valor do saque:");
23        cp.saqueConta(tc.nextDouble());
24        //lembrando que o saldo inicia com 100,00
25        System.out.println("Agora, seu saldo da conta poupança é : " + cp.saldo);
26
27    }
28 }
```
- Debugger (bottom left):** Shows the `main` method of the `DigitaConta` class with the signature `main(String[] args)`.

**\*\* Perceba que falta agora instanciar a classe `ContaCorrente` e digitar os dados. Faça isso a partir da linha 26 \*\***

# Para executar a classe DigitaConta, clique no menu Executar / Executar Arquivo

ExemplodeClasses - NetBeans IDE 8.1

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Proje... x Arquivos Serviços

ExemplodeClasses

- Pacotes de Códigos-fonte
  - exemplodeclasses
    - Conta.java
    - ContaCorrente.java
    - ContaPoupanca.java
    - DigitaConta.java
    - TestaExemplodeClasses.java
  - Bibliotecas

Código-Fonte

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15

main - Navegador x

Membros

DigitaConta

- main(String[] args)

exemplodeclasses.DigitaConta > main >

Saída - ExemplodeClasses (run) #4 x

run:  
Cadastro em Conta Poupança  
Digite o numero da conta :123456  
Digite o numero do cliente:50023  
Digite a data de abertura:02/09/2019  
Digite o valor do depósito:700,00  
Digite o valor do saque:300,00  
Agora, seu saldo da conta poupança é :500.0  
CONSTRUÍDO COM SUCESSO (tempo total: 38 segundos)

Resultado

# Atividade 1 – melhorias no programa DigitaConta

**\*\* Veja a execução já com a parte referente a conta corrente implementada \*\***

**Resultado quando o programa DigitaConta estiver também com a instância da classe ContaCorrente**

```
Saída - ExemplodeClasses (run) #12 X
run:
Cadastro em Conta Poupança
Digite o numero da conta :500123
Digite o numero do cliente:2020
Digite a data de abertura:25/08/2019
Digite o valor do depósito:2500,00
Digite o valor do saque:200,00
Agora, seu saldo em conta poupança é :2400.0

Cadastro em Conta Corrente
Digite o numero da conta :900321
Digite o numero do cliente:3030
Digite a data de abertura:02/09/2019
Digite o valor do depósito:1000,00
Digite o valor do saque:300,00
Agora, seu saldo em conta corrente é :800.0
CONSTRUÍDO COM SUCESSO (tempo total: 2 minutos 5 segundos)
```

**Lembrando que o saldo inicia em 100,00 no construtor da classe**

## Atividade 2:

- No mesmo projeto **ExemplodeClasses** da aula passada, crie um outro programa de nome **MenuDigitaDados.java** para receber dados via teclado (classe Scanner) em todos os atributos e métodos das classes **ContaPoupanca** e **ContaCorrente** mas individualmente, dependendo de uma seleção do usuário em um menu.
- No início do programa, haverá um menu para perguntar qual tipo de conta será utilizada.
- Para o atributo **saldo**, não digitar diretamente nesse atributo. Use os métodos **saqueConta()** e **depositoConta()** para receber e ajustar o valor do saldo. Lembrando que o saldos já iniciam com 100,00.
- O programa encerrará apenas quando o tipo de conta digitado for 0 (use o **do while** ou **while**).

### Layout de entrada de dados

#### Cadastro da Conta

Selecione o Tipo de Conta : <1> para Corrente <2> para Poupança <0> para sair : \_\_

**Numero da Conta** : \_\_

**Numero do Cliente** : \_\_

**Data de Abertura** : \_\_

...

...

...

**Valor do depósito** : \_\_

**Valor do saque** : \_\_

adicionar os outros atributos, dependendo do tipo de conta

## Exemplo de loopings com **while** ou **do while**, coloque onde desejar repetições ou consistências (trechos dentro do main())

```
System.out.println("Cadastro em Conta Poupança");
```

```
while (cp.numeroConta == 0 ) {  
    System.out.print("Digite o numero da conta :");  
    cp.numeroConta = tc.nextInt();  
}
```

```
do{  
    System.out.print("Digite o numero do cliente:");  
    cp.numeroCliente = tc.nextInt();  
}while (cp.numeroCliente == 0);
```

```
System.out.print("Digite a data de abertura:");  
cp.dataAbertura = tc.next();
```

```
System.out.print("Digite o valor do depósito:");  
cp.depositoConta(tc.nextDouble());
```

```
System.out.print("Digite o valor do saque:");  
cp.saqueConta(tc.nextDouble());
```

```
System.out.println("Agora, seu saldo em conta poupança é : " + cp.saldo);  
}
```

Exemplos de consistências.

### Atividade:

Experimente agora adicionar um outro looping para iniciar o programa até que algo seja digitado para encerrar. Exemplo:

Digite 1 para Conta Corrente, 2 para Poupança ou 0 para encerrar



Durante o curso:

- Leitura do artigo indicado (Recurso 1)
- Leitura do livro indicado (Recurso 2)

### **Recurso 1**

Artigo Devmedia: “Principais conceitos da Programação Orientada a Objetos” Disponível em:

<https://www.devmedia.com.br/principais-conceitos-da-programacao-orientada-a-objetos/32285>

### **Recurso 2**

Livro: FÉLIX, R. Programação Orientada a Objetos. São Paulo: Pearson Education do Brasil, 2016. 164p.  
[Biblioteca Virtual Universitária] pp 1-4.

# Indicações

- Tiexpert
- Devmedia
- GUJ
- Video-aulas youtube



## Referências

Livro: FURGERI, S. Java 8 - Ensino Didático - Desenvolvimento e Implementação de Aplicações. São Paulo: Érica, 2015. 320p. [Minha Biblioteca]. Capítulos 1 e 2.

**Te espero na próxima aula!**

