FMU
CENTRO UNIVERSITÁRIO

Prof. Me. Renato Alves Ferreira
email: renato.ferreira@fmu.br

Disciplina:

Programação Orientada a Objetos

Agenda da aula teórica/prática

- Correção ou revisão dos exercícios da aula anterior
- Mais detalhes sobre a linguagem Java
- Exemplo completo para manipulação de classes, objetos, etc
- Atividades em laboratório
- Pesquisas / Atividades extra aula

Programa da aula anterior (para efeito de comparação com o desafio)

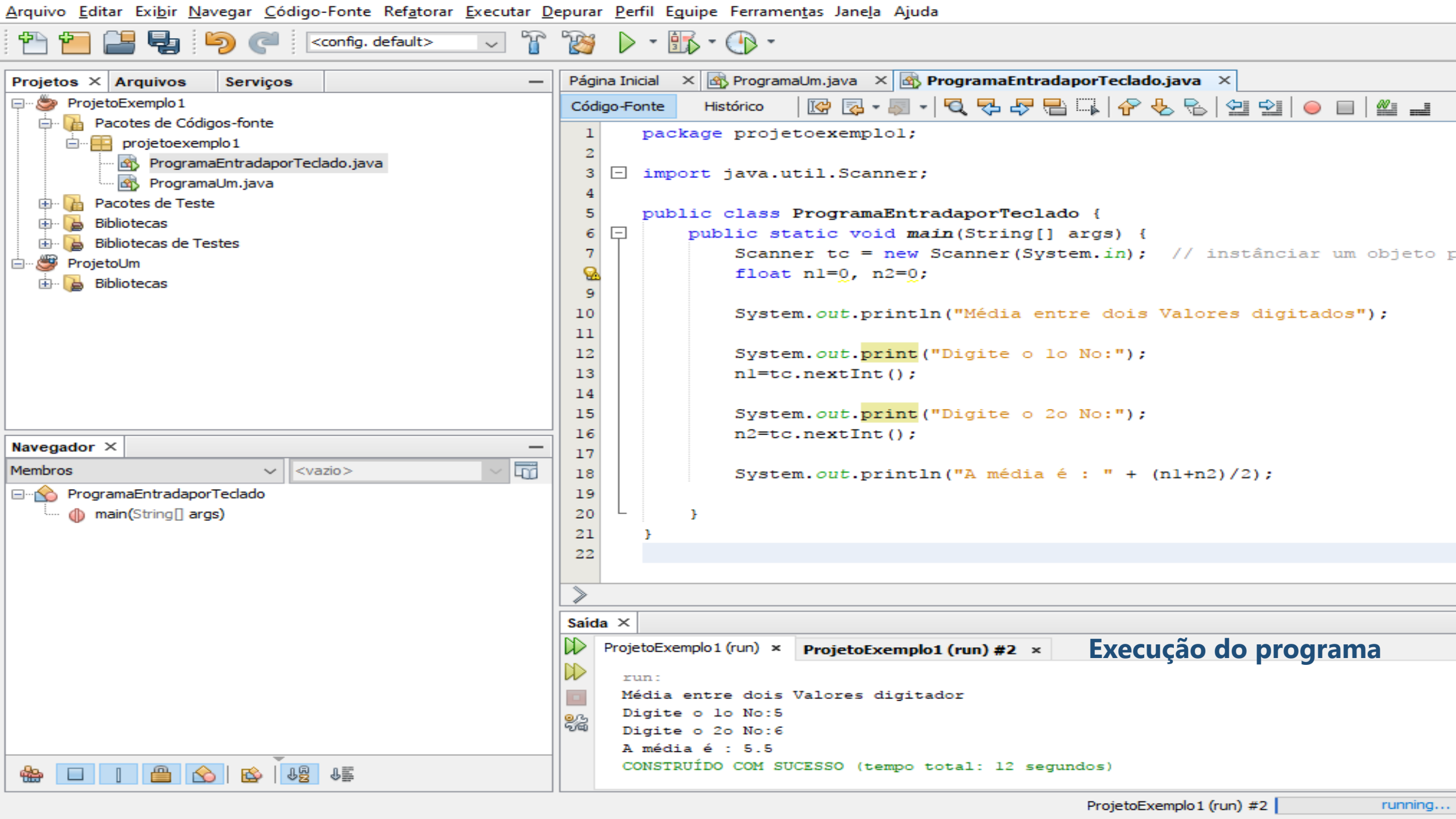
(**Sem** entrada de dados via teclado)

```
1  ...5 linhas
6
7  package projetoexemplo;
8
9  import java.lang.*;
10
11 /**...4 linhas */
15 public class ProgramaUm {
16     /**...3 linhas */
19     public static void main(String[] args) {
20         // TODO code application logic here
21         System.out.println("Programa Exemplo");
22         float n1=7, n2=8, resultado=0;
23         System.out.println("O valor do primeiro numero é :" + n1);
24         System.out.println("O valor do segundo numero é :" + n2);
25         resultado = (n1+n2) / 2;
26         System.out.println("A média é " + resultado);
27     }
28 }
```

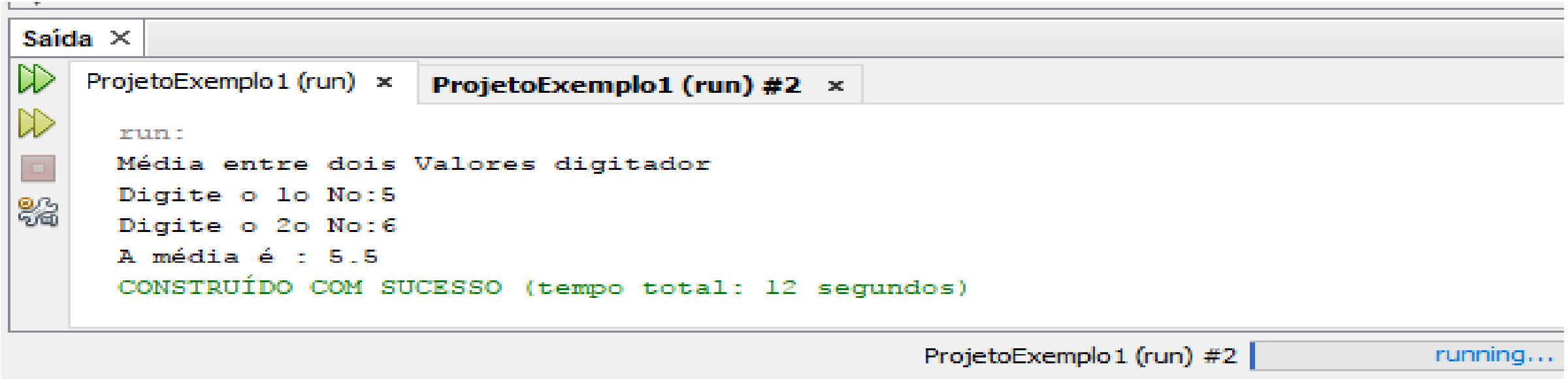
Correção do desafio (pesquisa sobre a classe Scanner)

(**Com** entrada de dados via teclado)

```java
package projetoexemplo1;

import java.util.Scanner;

public class ProgramaEntradaporTeclado {
    public static void main(String[] args) {
        Scanner tc = new Scanner(System.in);  // instânciar um objeto para leitura de teclado
        float n1=0, n2=0;

        System.out.println("Média entre dois Valores digitados");

        System.out.print("Digite o 1o No:");
        n1=tc.nextInt();

        System.out.print("Digite o 2o No:");
        n2=tc.nextInt();

        System.out.println("A média é : " + (n1+n2)/2);

    }
}
```


Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda
<config. default>
Projetos
Arquivos
Serviços
ProjetoExemplo1
Pacotes de Códigos-fonte
projetoexemplo1
ProgramaEntradaporTeclado.java
ProgramaUm.java
Pacotes de Teste
Bibliotecas
Bibliotecas de Testes
ProjetoUm
Bibliotecas
Navegador
Membros
<vazio>
ProgramaEntradaporTeclado
main(String[] args)
Página Inicial
ProgramaUm.java
ProgramaEntradaporTeclado.java
Código-Fonte
Histórico
1 package projetoexemplo1;
2
3 import java.util.Scanner;
4
5 public class ProgramaEntradaporTeclado {
6 public static void main(String[] args) {
7 Scanner tc = new Scanner(System.in); // instânciar um objeto p
8 float n1=0, n2=0;
9
10 System.out.println("Média entre dois Valores digitados");
11
12 System.out.print("Digite o 1o No:");
13 n1=tc.nextInt();
14
15 System.out.print("Digite o 2o No:");
16 n2=tc.nextInt();
17
18 System.out.println("A média é : " + (n1+n2)/2);
19
20 }
21 }
22
Saída
ProjetoExemplo1 (run)
ProjetoExemplo1 (run) #2
Execução do programa
run:
Média entre dois Valores digitador
Digite o 1o No:5
Digite o 2o No:6
A média é : 5.5
CONSTRUÍDO COM SUCESSO (tempo total: 12 segundos)
ProjetoExemplo1 (run) #2
running...

Em destaque a Área de execução do Netbeans



Atividade para praticar

(15 minutos. Concluir no final da aula)

Codifique um programa em Java que **receba três números via teclado** e mostre o resultado das 4 operações matemáticas entre eles.

Layout de vídeo

Digite o 1o No: _ _

Digite o 2o No: _ _

Digite o 3o No: _ _

A soma é _____
A subtração é _____
A divisão é _____
A multiplicação é _____

Alguns detalhes sobre a linguagem Java

JRE - Java Runtime Environment - Cntém as bibliotecas que são responsáveis pela execução das aplicações na JVM.

JVM - Java Virtual Machine - Ambiente de computação virtualizado que executa e gerencia os processos Java. Controla a a alocação de memória e recursos de CPU.

JDK - Java Development Kit - É o Kit de Desenvolvimento Java (compilador).

JSE - Java Standard Edition - Edição essencial do Java e já contém o JDK, JVM e JRE.

JEE - Java Enterprise Edition – Edição avançada do Java. Bibliotecas para desenvolvimento de aplicações mais avançadas e sofisticadas, como JSP e JSF.

JME - Java Micro Edition - Edição do Java que permitem desenvolvimento para sistemas embarcados, como carros, eletrodomésticos, celulares, etc. Não confundir com desenvolvimento para Android ou iOS, pois estes sistemas possuem ferramentas de desenvolvimento específicas de cada fabricante, mas que também usam o Java.

Variáveis e Tipos Primitivos do Java

Tipo	Tamanho	Exemplos de uso
char	2 bytes	char sexo= ‘F’;
Inteiros:		
byte	1	byte idade=49;
short	2	short x=1234;
int	4	int i= 554434;
long	8	long ra=1756453;
Reais:		
float	4	float pi=3.1415f;
double	8	double val=34.56;
boolean	1	boolean ativo=true;
Especiais:		
String	n	String curso=“Aula de Java”;
Date	n	Date dtCompra=“20-08-2019”;
Time	n	Time entrada=“19:15:00”;

Criando e manipulando Classes, Objetos, Atributos e Métodos

As classes estão para os objetos, assim como as plantas arquitetônicas estão para as casas. Você não pode fazer refeições em uma cozinha de uma planta; isso só é possível em uma cozinha real (DEITEL, 2005, p.15).

- Uma **classe** é um modelo para **objetos**.
- O **objeto** é uma instância da **classe**.
- **Objetos** têm **identidade** (nome), **características** (atributos), **estado** (conteúdos) e **comportamento** (métodos).

*Os dados contidos nos atributos do objeto definem seu **estado** e suas características.*

Exemplos de Classes Concretas, Heranças e Polimorfismo(override)

Conta.java - SuperClasse

```
public class Conta{
        int numeroConta;
        int numeroCliente;
        double saldo;
        String dataAbertura;

        void saqueConta(double valor){
            saldo = saldo - valor;
        }

        void depositoConta(double valor) {
            saldo = saldo + valor;
        }

        Conta(){
            saldo=100.00;
        }
}
```

atributos

métodos

construtor

Herança

ContaCorrente.java - SubClasse

```
public class ContaCorrente extends Conta{
        double limite;
        int gerente;
        double taxaMensal;
}
```

Herança

ContaPopança.java - SubClasse

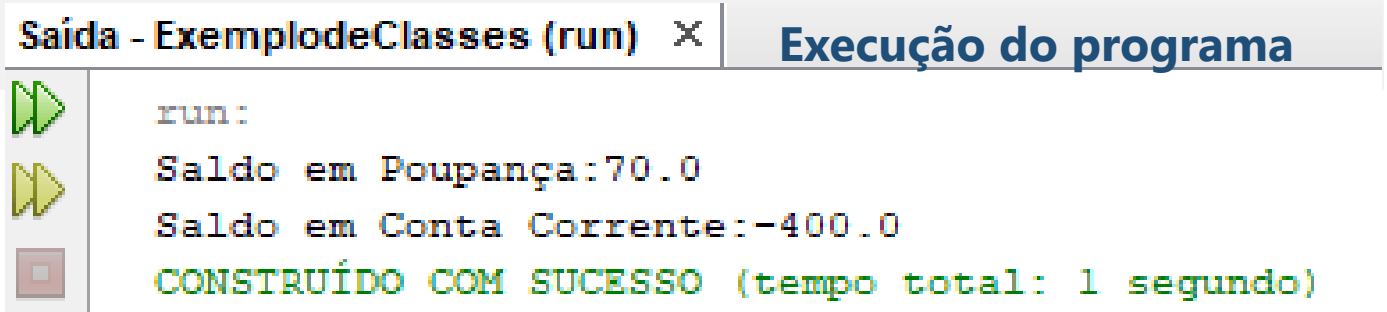
```
public class ContaPoupança extends Conta{
    String datasAniversários;

    void saqueConta(double valor){        //polimorfismo
        if((saldo - valor) < 0)           //reescrita do método
            System.out.println(" ERRO: Saldo ficará
                                    abaixo de 0.00 ");
        else saldo=saldo - valor;
    }
}
```

Programa exemplo para testar as classes criadas

TestaExemplodeClasses.java - Classe "executável"

```
// Classe Programa: TestaExemplodeClasses.java
// Aula de POO - prof. Renato Alves
//Grupo: <<nome completo dos integrantes do grupo>>

package exemplodeclasses;

public class TestaExemplodeClasses {
    public static void main(String[] args) {                // método main permite  que a classe seja executada
        ContaPoupanca cp = new ContaPoupanca();       //Objeto cp para conta poupança
        cp.saqueConta(30.00);
        System.out.println("Saldo em Poupança:" + cp.saldo);

        ContaCorrente cr = new ContaCorrente();       //Objeto cr para conta corrente
        cr.saqueConta(500.00);
        System.out.println("Saldo em Conta Corrente:" + cr.saldo);
    }
}
```

Execução do programa



Atividades da aula – Crie um novo projeto no Netbeans chamado: **ExemplodeClasses**

1- Digitar e testar os seguintes códigos em java:

- ✓ **Conta.java**
- ✓ **ContaCorrente.java**
- ✓ **ContaPoupanca.java**
- ✓ **TestaExemplodeClasses.java**

(seu projeto no Netbeans ficará semelhante as imagens dos dois slides seguintes)

ExemplodeClasses - NetBeans IDE 8.1

Janela do projeto depois de pronto

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

<config. default>

Projetos | **Arquivos** | **Serviços**

- ExemplodeClasses
  - Pacotes de Códigos-fonte
    - exemplodeclasses
      - Conta.java
      - ContaCorrente.java
      - ContaPoupanca.java
      - TestaExemplodeClasses.java
  - Pacotes de Teste
  - Bibliotecas
  - Bibliotecas de Testes

Navegador

Membros <vazio>

- TestaExemplodeClasses
  - main(String[] args)

TestaExemplodeClasses.java | Conta.java | ContaCorrente.java | ContaPoupanca.java

Código-Fonte | Histórico

```
1  // Classe Programa: TestaExemplodeClasses.java
2  // Aula de POO - prof. Renato Alves
3
4  package exemplodeclasses;
5
6  public class TestaExemplodeClasses {
7      public static void main(String[] args) {
8          ContaPoupanca cp = new ContaPoupanca();          //Objeto cp para conta poupnça
9          cp.saqueConta(30.00);
10         System.out.println("Saldo em Poupança:" + cp.saldo);
11
12         ContaCorrente cr = new ContaCorrente();          //Objeto cr para conta corrente
13         cr.saqueConta(500.00);
14         System.out.println("Saldo em Conta Corrente:" + cr.saldo);
15     }
16 }
17
```

Saída - ExemplodeClasses (run)

```
run:
Saldo em Poupança:70.0
Saldo em Conta Corrente:-400.0
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```


17:1 INS

TestaExemplodeClasses.java | **Conta.java** | ContaCorrer

Código-Fonte | Histórico

```java
// Classe: Conta.java
// Aula de POO - prof. Renato Alves

package exemplodeclasses;

public abstract class Conta {
    int numeroConta;
    int numeroCliente;
    double saldo;
    String dataAbertura;

    void saqueConta(double valor){
        saldo = saldo - valor;
    }

    void depositoConta(double valor) {
        saldo = saldo + valor;

    }

    Conta(){
        saldo=100.00;
    }
}
```

Conta.java

TestaExemplodeClasses.java | **Conta.java** | ContaCorrente.java

Código-Fonte | Histórico

```java
// Classe: ContaCorrente.java
// Aula de POO - prof. Renato Alves

package exemplodeclasses;

public class ContaCorrente extends Conta{
    double limite;
    int gerente;
    double taxaMensal;
}
```

ContaCorrente.java

TestaExemplodeClasses.java | **Conta.java** | ContaCorrente.java | ContaPoupanca.java

Código-Fonte | Histórico

```java
// Classe: ContaPoupanca.java
// Aula de POO - prof. Renato Alves

package exemplodeclasses;

public class ContaPoupanca extends Conta{
    String datasAniversários;

    @Override
    void saqueConta(double valor){
        if((saldo - valor) < 0)
          System.out.println(" ERRO: Saldo ficará abaixo de 0.00 ");
        else saldo=saldo - valor;
    }
}
```

ContaPoupanca.java

TestaExemplodeClasses.java | **Conta.java** | ContaCorrente.java | ContaPoupanca.java

Código-Fonte | Histórico

```java
// Classe Programa: TestaExemplodeClasses.java
// Aula de POO - prof. Renato Alves

package exemplodeclasses;

public class TestaExemplodeClasses {
    public static void main(String[] args) {
        ContaPoupanca cp = new ContaPoupanca();          //Objeto cp para
        cp.saqueConta(30.00);
        System.out.println("Saldo em Poupança:" + cp.saldo);

        ContaCorrente cr = new ContaCorrente();          //Objeto cr pa
        cr.saqueConta(500.00);
        System.out.println("Saldo em Conta Corrente:" + cr.saldo);
    }
}
```

TestaExemplodeClasses.java

Atividade de classes, objetos, atributos, métodos e programas

-No mesmo projeto **ExemplodeClasses** da aula passada, crie um outro programa de nome **DigitaConta.java** para receber dados via teclado (classe Scanner) em todos os atributos e métodos das classes **ContaPoupanca** e em seguida **ContaCorrente**.

-Acrescente os atributos dependendo do tipo da conta selecionada (poupança ou corrente)*.

-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 em Conta Poupança **DigitaConta.java**

Numero da Conta : __
Numero do Cliente : __
Data de Abertura : __
...
...
...
Valor do depósito : __
Valor do saque : __

*adicionar os outros atributos, dependendo do tipo de conta

(repita o mesmo layout para conta corrente)

3 – **Atividade extra** de Aprendizagem e pesquisa– extra aula pesquisa

(portfólio do aluno)

(na sala, fazer apenas se ou quando terminar as atividades do dia)

- Elabore um programa para converter a temperatura de graus Celsius para graus Farenheit e vice-versa.

- Elabore um programa para imprimir a tabuada de um número N.

- Elabore um programa para calcular o MDC de dois números inteiros.

(Essa atividade é importante, mas não será corrigida em sala)

ATIVIDADE EXTRACLASSE

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.

REFERÊNCIAS

BIBLIOGRAFIA BÁSICA

- ✓ FURGERI, S. Programação orientada a objetos: Conceitos e Técnicas. São Paulo: Érica, 2016. 168p.
- ✓ MANZANO, J. A. G.; COSTA JR., R. Programação de Computadores com Java. Érica, 2014. 127p. [Minha Biblioteca]
- ✓ MANZANO, J. A. G. Programação de Computadores com C/C++. Érica, 06/2014. 120p. [Minha Biblioteca].

BIBLIOGRAFIA COMPLEMENTAR

- ✓ BARNES, D. J.; KOLLING, M. Programação Orientada a Objetos com Java: uma introdução prática usando o BlueJ - 4ª edição. São Paulo: Pearson Prentice Hall, 2009. 480p. [Biblioteca Virtual Universitária].
- ✓ MEILIR, P. Fundamentos do Desenho Orientado a Objeto com UML. São Paulo: Makron Books, 2001. 462p. [Biblioteca Virtual Universitária].
- ✓ FÉLIX, R. Programação Orientada a Objetos. São Paulo: Pearson Education do Brasil, 2016. 164p. [Biblioteca Virtual Universitária].
- ✓ KOFFMAN, E. B., WOLFGANG, P. T. Objetos, Abstração, Estrutura de Dados e Projeto Usando C++. Rio de Janeiro: LTC, 2008. 455p. [Minha Biblioteca].
- ✓ FURGERI, S. Java 8 - Ensino Didático - Desenvolvimento e Implementação de Aplicações. São Paulo: Érica, 2015. 320p. [Minha Biblioteca].

Te espero na próxima aula!

FMU
CENTRO UNIVERSITÁRIO