



Buscar no fórum

Prezado(a) estudante,

Período de Provas EAD

Estamos em período de provas entre os dias **17/09 e 28/09**.

Para quem realizou o agendamento, a senha da prova será enviada um dia antes via E-mail e SMS.

Para quem não agendou, por favor aguarde o contato da Católica.

Fique atento ao seu Plano de Ensino. Caso possua disciplinas sem provas como Estágio, Práticas e TCC, por favor desconsidere este aviso.

Boa prova!

[Administração ▶](#)

[Acessibilidade ▶](#)



[Meus cursos ▶](#)

[Mensagem para meu professor ▶](#)

[Acesso rápido ▶](#)

PROGRAMAÇÃO ORIENTADA A OBJETOS

Atividade Avaliativa 3

Configurações ▾

[Mostrar respostas aninhadas](#)

A data limite para postagem neste fórum foi atingida, portanto, você não poderá mais postar nela.

Atividade Avaliativa 3

quarta, 15 jul 2020, 10:48

Neste fórum você vai criar um projeto Java e as classes necessárias para resolver **UMA atividade** do Item na prática e **comentar com sugestões construtivas** uma postagem realizada por um de seus colegas.

Orientações:

- 1) Leia o conteúdo das aulas de 5 a 8
- 2) Selecione pelo menos uma questão (Item Na Prática) e **poste seu código no fórum. (5 pontos)**
- 3) **Corrija/comente uma resposta postada por um de seus colegas**, objetiva-se que todos tenham sua resposta comentada. **(5 pontos)**

[Link direto](#) [Marcar como não lida](#)

Re: Atividade Avaliativa 3
por [GUILHERME MIRANDA DE CARVALHO](#) - domingo, 23 ago 2020, 11:15

Unidade 3 - Aula 5 - Atividade 1

```
import java.util.Scanner;

public class Programa {

    public static void main(String[] args) {
        char profissao;
        double ImpRenda;

        Professor prof = new Professor();
        Administrativo admin = new Administrativo();

        do {
            System.out.println("Administrativo - A/a ");
            System.out.println("Professor - P/p ");
            System.out.printf("Qual a sua profissão? ");
            profissao = new Scanner(System.in).next().charAt(0);

            if(profissao != 'A' && profissao != 'P' && profissao != 'a' &&
            profissao != 'p') {
                System.out.println("-----");
                System.out.println("Caractere Inválido! Digite Novamente!");
            }
        }while (profissao != 'A' && profissao != 'P' && profissao != 'a' &&
        profissao != 'p');

        System.out.println("-----");

        if (profissao == 'P' || profissao == 'p') {
            System.out.printf("Qual o seu nome? ");
            prof.nome = new Scanner(System.in).nextLine();

            System.out.printf("Qual o seu cpf? ");
            prof.cpf = new Scanner(System.in).nextLine();
        }
    }
}
```



```

System.out.printf("Qual o seu Salário Bruto? ");
prof.salario = new Scanner(System.in).nextDouble();

System.out.printf("Qual a sua titulação? ");
prof.titulacao = new Scanner(System.in).nextLine();

System.out.println("-----");

System.out.println("Confirmar Dados : ");
System.out.println("Nome: "+prof.nome);
System.out.println("CPF: "+prof.cpf);

System.out.println("Salário Líquido: "+(prof.salario -
getImpostoRenda(prof.salario)));

System.out.println("Titulação: "+prof.titulacao);

}else {

System.out.printf("Qual o seu nome? ");
admin.nome = new Scanner(System.in).nextLine();

System.out.printf("Qual o seu cpf? ");
admin.cpf = new Scanner(System.in).nextLine();

System.out.printf("Qual o seu Salário Bruto? ");
admin.salario = new Scanner(System.in).nextDouble();

System.out.printf("Possui quantos dependentes? ");
admin.dependentes = new Scanner(System.in).nextInt();

System.out.println("-----");

System.out.println("Confirmar Dados : \n");
System.out.println("Nome: "+admin.nome);
System.out.println("CPF: "+admin.cpf);

System.out.println("Salário Líquido: "+(admin.salario -
getContribuicaoSindical(admin.salario)));

System.out.println("Dependentes: "+admin.dependentes);

}

public static double getImpostoRenda(double salario) {
    return(0.23*salario);
}

```



```
}
```

```
public static double getContribuicaoSindical(double salario) {  
    return(0.01*salario);  
}
```

```
}
```

```
class Funcionario{
```

```
    String nome;  
    String cpf;  
    double salario;
```

```
}
```

```
class Professor extends Funcionario{
```

```
    String titulacao;  
}
```

```
class Administrativo extends Funcionario{
```

```
    int dependentes;  
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ANTONIO VINICIUS XIMENES DA ROSA](#) - domingo, 23 ago 2020, 12:05

Ótimo código Guilherme, só um detalhe, a variável Double ImpRenda não está sendo utilizada, já que o responsável por calcular o imposto de renda e retornar o valor fazendo uso do salario está sendo o método abaixo:

```
public static double getImpostoRenda(double salario) {  
  
    return(0.23*salario);  
  
}  
Se retirar essa a variável o código funciona normalmente, tirando esse detalhe  
Parabéns !!!
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [RAIMUNDO FAGNER MONTEIRO](#) - sexta, 28 ago 2020, 23:33

Discordo com o colega ANTONIO VINICIUS, ai digitar
public static double getImpostoRenda(double salario) {

```
    return(0.23*salario);
```

}

Há necessidade de acrescentar o sinal } sim, pois ao não acrescenta esse simbolo essa mensagem sugira

Syntax error, insert ")" to complete MethodBody , tendo que corrigir o erro.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [BRUNO ALVES BECO](#) - sexta, 11 set 2020, 23:19

Olá Raimundo, concordo com o Antônio, no caso ele se referiu a variável que está sendo declarada no início a chave nesse caso não pode ser retirada. Mas se fosse uma estrutura condição (if por exemplo) ou uma estrutura de repetição (for por exemplo) e tivesse apenas uma instrução dentro dessa estrutura as chaves poderiam ser omitidas.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [MARTON PAULO DOS SANTOS SILVA](#) - domingo, 23 ago 2020, 20:15

Muito bom, Guilherme!

Antonio já comentou sobre a variável `ImpRenda`, vou complementar com mais uma dica: para que exista um padrão nos códigos escritos em Java, quando algo começa com letra maiúscula geralmente é um objeto, variáveis começam com letra minúscula.

Acredito que você tenha escrito essa variável no início do código, mudou a estrutura e depois esqueceu de removê-la. Mas no final de tudo, não atrapalhou o resultado do programa. Parabéns.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [LUCIANO ARMANDO VILETE](#) - segunda, 24 ago 2020, 16:27

Muito bom Guilherme! Conseguiu desenvolver muito bem o código proposto na questão.

Me ajuntando aos amigos que abaixo já deram contribuições valiosas ao código, gostaria de fazer apenas uma breve menção a questão dos espaços entre as linhas.

É uma boa prática deixar o código o máximo possível, enxuto e legível. O excesso de espaços nesse caso, dificulta bastante com as duas propostas. Mas como dito anteriormente, o código foi feito com maestria! Parabéns!

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [JONATHAN REBOUÇAS BESSA](#) - terça, 25 ago 2020, 16:41

É interessante vc criar métodos para evitar o if. Sei que no inicio é meio complicado, mas é uma boa prática evitar.

Veja a forma que fiz abaixo.

O código fica mais fácil de entender

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [RODRIGO RIBEIRO CAMPOS](#) - quinta, 27 ago 2020, 20:02

Olá, Guilherme!

Gostei da forma como você respondeu ao exercício, pois me ajudou muito com a lógica do algoritmo para o [trabalho final](#). Programação orientada a objetos é complexo, a princípio, porém, depois, facilita demais a nossa vida.

Abraços!

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [CELSO PEQUENO CERQUEIRA](#) - domingo, 30 ago 2020, 18:17

Prezado Guilherme,

A título de contribuição, salvo esteja eu equivocado, creio que os métodos abaixo que, no meu entender, estão dentro da sua classe Programa deveriam está, respectivamente, nas classes Professor e Administrativo, já que essas ações dependem do atributo salário de ambas:

```
public static double getImpostoRenda(double salario) {  
  
    return(0.23*salario);  
  
}
```

```
public static double getContribuicaoSindical(double salario) {  
  
    return(0.01*salario);  
  
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [RODOLFO CAVALCANTE DE OLIVEIRA](#) - segunda, 31 ago 2020, 16:33

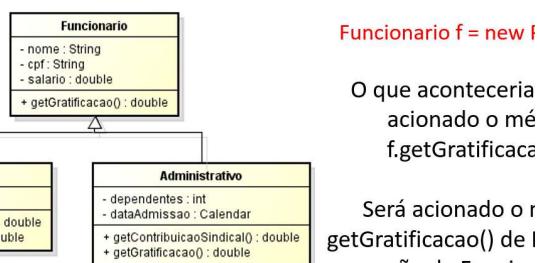


Olá Celso,

Acredito que nesse caso podemos aplicar o que vimos em aula e sobreescriver o método.



Polimorfismo



[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [EDEN IVO BEZERRA](#) - quarta, 2 set 2020, 00:55

Parabéns pelo código, GUILHERME MIRANDA DE CARVALHO!!!

Minha sugestão, como contribuição, é que o sistema escrito por você seja escalável, ou seja, **possível de expansão quanto à opção dada (Professor ou Administrativo)**.

Traduzindo, se amanhã surgir uma (ou mais) nova categoria (cargo), com suas peculiaridades, como ficaria esse trecho do seu código "while (profissao != 'A' && profissao != 'P' && profissao != 'a' && profissao != 'p');"??

Nesse sentido, **permita ao usuário "errar" a opção dada**, mas informando

ao final que a opção "ainda não está disponível", como no trecho de código abaixo:

```
if(opcao == 'P' || opcao == 'p') {  
    // Conteúdo da opção professor  
} else if(opcao == 'A' || opcao == 'a') {  
    // Conteúdo da opção administrativo  
} else {  
    System.out.println("\n-----");  
    System.out.println("Ops! O(A) Senhor(a) escolheu uma opção não  
disponível");  
}
```

Outra boa opção de testagem de opção seria por meio do controle de fluxo SWITCH, conforme código abaixo:

```
switch(opcao){  
    case 'P': System.out.print("Professor");break;  
    case 'A': System.out.print("Administrativo");break;  
    case 'X': System.out.print("Cargo X");break;  
    case 'Y': System.out.print("Cargo Y");break;  
    case 'Z': System.out.print("Cargo Z");break;  
  
    default:System.out.print("Acorda para Jesus!!!");break;  
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [LARA PATTY RODRIGUES BARBOSA](#) - quarta, 2 set 2020, 20:51

Guilherme, seu raciocínio está ótimo! Sugiro, para otimizar a leitura do código, que você crie uma classe para o professor.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ALLAN CARLOS BARROS MORAES](#) - quarta, 2 set 2020, 22:44

Amigo, você poderia ter definidos os métodos dentro da Classe Funcionário. Desse modo, estaria aplicando os três primeiros pilares da POO: Abstração, Encapsulamento e Herança.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ALEXSSANDRO DE SOUSA FEITOSA](#) - domingo, 6 set 2020, 06:40

Código perfeito com exceção de um erro mínimo de semântica nessa parte "System.out.printf("Qual o seu nome? ");". esqueceu do "In" para completar o printf. Mas fora isso, esta muito bom.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [LUCAS LUAN DA SILVA LACERDA](#) - sexta, 11 set 2020, 11:04

Parabéns pelo código Guilherme, deixo duas sugestões para você: O primeiro é para deixar getImpostoRenda() dentro da classe professor, e getContribuiçãoSindical() dentro de Administrativo, a segunda já emenda com a primeira, fazendo encapsulamento deixando as informações private e fazendo o uso de getters e setter.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [VIVIANE CRISTINA BUGE BRASIL](#) - sexta, 11 set 2020, 19:59

Boa Noite,

gostei do código, talvez fosse melhor apenas criar mais métodos, para as leituras, o que deixaria ele mais claro, acredito eu.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [MARTON PAULO DOS SANTOS SILVA](#) - domingo, 23 ago 2020, 20:02

Saudações, professora e colegas!

Aula 05 - Atividade 02

Um matemático trabalha com dois tipos de figuras: os triângulos retângulos e os retângulos. Crie um programa Java (classes) que represente a modelagem para abstrair este problema. Ambos têm dois dados comuns: Base e Altura. Cada uma tem um cálculo de forma distinta de área. O triângulo retângulo é base X altura / 2, já do retângulo é apenas base X altura. Um cálculo diferente somente para o triângulo retângulo deve ser realizado, que é a hipotenusa. Crie um programa de teste, que instancie objetos, atribua valores e execute para apresentação.

Resposta

```
import java.lang.*; // importação para usar raiz quadrada

public class Programa {
    public static void main (String args[]) {

        TrianguloRetangulo triangulo = new
        TrianguloRetangulo();
        Retangulo retangulo = new Retangulo();

        triangulo.setBase(6.0);
        triangulo.setAltura(8.0);
        System.out.println("Área: " + triangulo.getArea());
        System.out.println("Hipotenusa: " +
        triangulo.getHipotenusa());

        retangulo.setBase(4.0);
        retangulo.setAltura(5.0);
        System.out.println("Área: " + retangulo.getArea());
    }
}

class Figura {
    private double base;
    private double altura;

    public void setBase (double base) {
        this.base = base;
    }
    public double getBase() {
        return base;
    }
    public void setAltura (double altura) {
        this.altura = altura;
    }
    public double getAltura() {
        return altura;
    }
    public double getArea() { // método que será
    sobrescrito
        return 0.0;
    }
}
```



```

        }
    }

    class TrianguloRetangulo extends Figura {
        public double getArea() {
            return (this.getBase() * this.getAltura()) / 2.0;
        }
        public double getHipotenusa() {
            return Math.sqrt(this.getBase() * this.getBase() +
            this.getAltura() * this.getAltura());
        }
    }

    class Retangulo extends Figura {
        public double getArea() {
            return (this.getBase() * this.getAltura());
        }
    }
}

```

Deixo também o código .java em anexo.

 [Programa.java](#)

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [EVALDO JOSE DE ARRUDA LIMA BRASIL](#) - domingo, 23 ago 2020, 21:43

Excelente código, limpo didático.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [NELSON SILVA LOPES](#) - sexta, 28 ago 2020, 14:55

Boa tarde Professora Rosana e colegas de curso!

Esperando contribuir com sua solução Marton, eu creio que a palavra-chave "this" nos métodos de cálculo de área e da hipotenusa não é necessária. Penso que não há ambiguidade aí. Apenas no método assessor "set". Ainda, uma alternativa ao cálculo da hipotenusa é o método hypot na API Math. Segue sua descrição:

```
static double hypot(double x, double y)
Returns sqrt(x2 +y2) without intermediate overflow or underflow.
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [JONATHAN ALBERTO RODRIGUES DAS NEVES](#) - quinta, 10 set 2020, 15:34

ótimo, parabéns

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [GABRIEL DOREA DE MELO](#) - sexta, 11 set 2020, 18:45

boa tarde, otimo projeto. objetivo e funcional

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [EVALDO JOSE DE ARRUDA LIMA BRASIL](#) - domingo, 23 ago 2020, 21:40

HERANÇA:

(NA PRÁTICA 02)

CLASSE FIGURA:

=====

```
public class Figura {  
    private double base;  
    private double altura;  
  
    public double getBase() {  
        return base;  
    }  
  
    public void setBase(double base) {  
        this.base = base;  
    }  
  
    public double getAltura() {  
        return altura;  
    }  
  
    public void setAltura(double altura) {  
        this.altura = altura;  
    }  
  
    public void cadTriangulo(double base, double altura) {  
  
        Triangulo tri = new Triangulo();  
  
        tri.setBase(base);  
        tri.setAltura(altura);  
  
        System.out.println("Base: " + tri.getBase());  
        System.out.println("Altura: " + tri.getAltura());  
        System.out.println("Área: " + tri.getArea());  
        System.out.println("Hipotenusa: " + tri.getHipotenusa());  
  
    }  
  
    public void cadRetangulo(double base1, double altura1) {  
  
        Retangulo ret = new Retangulo();  
  
        ret.setBase(base1);  
        ret.setAltura(altura1);  
  
        System.out.println("Base: " + ret.getBase());  
        System.out.println("Altura: " + ret.getAltura());  
        System.out.println("Área: " + ret.getArea());  
  
    }  
}
```

=====

CLASSE TRIÂNGULO:

=====

```
public class Triangulo extends Figura {  
    public double getArea() {  
        return (getBase() * getAltura()) / 2;  
    }  
  
    public double getHipotenusa() {  
        return (getBase() * getBase() + getAltura() * getAltura());  
    }  
}
```



=====

CLASSE RETÂNGULO:

=====

```
public class Retangulo extends Figura {  
    public double getArea() {  
        return getBase() * getAltura();  
    }  
}
```

=====

CLASSE MAIN:

=====

```
import java.util.Scanner;  
  
public class Main extends Figura {  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
        int opcao = 0;  
  
        System.out.println("===== Atividade 02 =====");  
        System.out.println("Escolha uma opção: ");  
        System.out.println("1 - Para Triângulo:");  
        System.out.println("2 - Para Retângulo:");  
        opcao = Integer.parseInt(sc.nextLine());  
  
        switch (opcao) {  
  
            case 1:  
  
                Figura tri = new Triangulo();  
  
                System.out.println("===== Dados do Triângulo =====");  
  
                System.out.println("Base: ");  
                double base = Integer.parseInt(sc.nextLine());  
  
                System.out.println("Altura: ");  
                double altura = Integer.parseInt(sc.nextLine());  
  
                tri.cadTriangulo(base, altura);  
  
                break;  
  
            case 2:  
                Figura ret = new Retangulo();  
  
                System.out.println("===== Dados do Retângulo =====");  
  
                System.out.println("Base: ");  
                double base1 = Integer.parseInt(sc.nextLine());  
  
                System.out.println("Altura: ");  
                double altura1 = Integer.parseInt(sc.nextLine());  
  
                ret.cadRetangulo(base1, altura1);  
  
                break;  
        }  
    }  
}
```



**Re: Atividade Avaliativa 3**por [GUSTAVO LESSA BENEDET](#) - segunda, 24 ago 2020, 17:53

EM RELAÇÃO AO TRECHO:

```
Scanner sc = new Scanner(System.in);
int opcao = 0;

System.out.println("===== Atividade 02 =====");
System.out.println("Escolha uma opção: ");
System.out.println("1 - Para Triângulo:");
System.out.println("2 - Para Retângulo:");
opcao = Integer.parseInt(sc.nextLine());
```

GOSTEI DO USO DA CLASSE INTEGER COM MÉTODO PARSEINT(STRING);
PARA CONVERSÃO DE UMA STRING EM INTEIRO, MAS NÃO SERIA MAIS
FÁCIL SIMPLESMENTE USAR :

OPCAO = SC.NEXTINT();

?

ASSIM POSTERIORMENTE PODERIAS ADICIONAR UM LIMITADOR (CRIAR
RANGE) COM MAIS FACILIDADE.

**Re: Atividade Avaliativa 3**por [LUCIANO ARMANDO VILETE](#) - segunda, 24 ago 2020, 16:25

Aula 5, atividade 1

Tive problemas com a parte final do exercício (a questão de perguntar ao usuário e retornar os dados, mas fiz todo o código do UML proposto.

```
package TerceiraSemana;

import java.util.Scanner;

public class Funcionario {

    public static void main(String[] args) {
        System.out.println("Qual o seu cargo na empresa? [P/p para professor] ou [A/a para administrativo]");
        String pergunta = new Scanner(System.in).toString().toUpperCase();

    }

    String nome;
    int CPF;
    double salario, imposto, contribuicao;
    public double getContribuicao() {
        return contribuicao;
    }
    public void setContribuicao(double contribuicao) {
        this.contribuicao = contribuicao;
    }
    public double getImposto() {
        return imposto;
    }
    public void setImposto(double imposto) {
```

```

this.imposto = imposto;
}
public String getNome() {
return nome;
}
public void setNome(String nome) {
this.nome = nome;
}
public int getCPF() {
return CPF;
}
public void setCPF(int cPF) {
CPF = cPF;
}
public double getSalario() {
return salario;
}
public void setSalario(double salario) {
this.salario = salario;
}

}

package TerceiraSemana;

public class Administrativo extends Funcionario{
int dependentes;

double getContribuicaoSindical(double salario) {
return contribuicao = salario*0.1;
}
Administrativo a = new Administrativo();
nome = "Luciano";
CPF = 11111111;
salario = 1975.00;

getContribuicaoSindical(salario);
System.out.println(contribuicao);
}
}

```

```

package TerceiraSemana;
public class Professor extends Funcionario {
String titulacao;

double getImpostoRenda(double salario){
return imposto = salario*23/100;
}

}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [FERNANDO SANTOS FERREIRA](#) - quinta, 10 set 2020, 21:42

Olá, Luciano. Parabéns pelo código!

Gostaria de deixar uma sugestão, considere declarar o atributo cpf como String, pois, além do [int não suportar os dígitos de um cpf](#), declarando um atributo como esse usando o tipo int, se o cpf inserido começar com um "zero", esse zero é desconsiderado. E assim, dependendo das condições pelas quais esse atributo passará no decorrer do código, você poderá se deparar com erros inesperados. Por exemplo:

//suponhamos que o valor do atributo cpf seja 07005048651

```
int tamanhoCPF = String.valueOf(getCPF).length();
```

```
if (tamanhoCPF < 11){
```



```

        System.out.println("CPF Inválido. Este cpf contém " + tamanhoCPF + "
caracteres);

/*
a execução deste código retornará que o cpf é inválido, pois ele possui menos
de 11 caracteres em razão do 0 inicial que é removido na tratativa int

*/
}

```

Soma das avaliações: 10 (1)

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [GUSTAVO LESSA BENEDET](#) - segunda, 24 ago 2020, 17:35

AULA 06

ATIVIDADE 01:

```

1 import java.util.Scanner;
2
3 public class Leitores {
4     /* LER STRING RECEBENDO MSG */
5     public String lerString(String msg) {
6
7         System.out.println(msg);
8
9         return lerString();
10    }
11    /* LER STRING SEM PARAMETROS */
12    @SuppressWarnings("resource")
13    public String lerString() {
14
15        return new Scanner(System.in).next();
16    }
17    /* LER DOUBLE RECEBENDO MSG */
18    public double lerDouble(String msg) {
19
20        System.out.println(msg);
21        return lerDouble();
22    }
23    /* LER DOUBLE SEM PARAMETROS */
24    @SuppressWarnings("resource")
25    public double lerDouble() {
26
27        return new Scanner(System.in).nextDouble();
28    }
29    /* LER DOUBLE RECEBENDO MSG, ERRO, MÍNIMO E MÁXIMO POSSÍVEL */
30    public double lerDouble(String msg, String erro, int min, int max) {
31        boolean error;
32        double numero;
33
34        do {
35
36            numero = lerDouble(msg);
37
38            if ((numero >= max) || (numero <= min)) {
39                error = true;
40                System.out.println(erro);
41
42            } else
43                error = false;
44
45        } while (error == true);
46
47        return numero;
48    }
49
50
51 package br.edu.utfpr.atividade01;
52
53     /* Método de Verificação da Situação Financeira */
54
55     public class Pessoa {
56
57         private String nome;
58         private String codigo;
59         private double salario;
60
61         /* Construtor com parâmetros */
62         public Pessoa(String nome, String codigo, double salario) {
63             super();
64             this.nome = nome;
65             this.codigo = codigo;
66             this.salario = salario;
67         }
68
69         /* Método Gets e Sets */
70         public String getNome() {
71             return nome;
72         }
73         public void setNome(String nome) {
74             this.nome = nome;
75         }
76         public String getCodigo() {
77             return codigo;
78         }
79         public void setCodigo(String codigo) {
80             this.codigo = codigo;
81         }
82         public double getSalario() {
83             return salario;
84         }
85         public void setSalario(double salario) {
86             this.salario = salario;
87         }
88
89         /* Método getClass() de Pessoa */
90
91         @Override
92         public String toString() {
93             return "===== Pessoa =====\n" +
94                 "Nome: " + nome + "\n" +
95                 "Codigo: " + codigo + "\n" +
96                 "Salario: " + salario + "\n" +
97             "===== ";
98     }
99
100 }

```



```

1 package br.aula06.atividade01;
2
3
4 public class Main {
5
6     public static void main(String[] args) {
7
8         Leitor leitor = new Leitor();
9
10        Pessoa p = new Pessoa();
11
12        (leitor.lerString("Digite o nome da Pessoa: ")),
13        (leitor.lerString("Digite o Código da pessoa: ")) ,
14        (leitor.lerDouble("Digite o Salário: ", "Número Inválido", 0, 10000)) ;
15
16        System.out.println(p = p.classeFinanceira());
17
18    }
19
20 }
21

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [GUILHERME MIRANDA DE CARVALHO](#) - segunda, 24 ago 2020, 23:17

Interessante essa forma de resolução do exercício através do uso de pacotes, uma vez que organizam as linhas de código e deixam o documento final mais fácil de compreender e fazer a manutenção (quando necessário) por outros desenvolvedores. Bem como, o uso desse recurso dinamiza o processo ao reduzir o tempo gasto na elaboração e gestão de projetos que exijam uma construção maior.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [JONATHAN REBOUÇAS BESSA](#) - terça, 25 ago 2020, 16:49

Exercício 1 Aula 5

Vou postar o Código da Classe Principal.Pelos Print's ficaria muito pequeno.



```

import java.util.Scanner;

public class Programa {

    public static void main(String args[]) {

        String tipo = VerificarTipoDeFuncionario();

        switch (tipo) {
            case "A":
                CriarFuncionarioAdministrativo();
                break;
            case "P":
                CriarProfessor();
                break;
            default:
                VerificarTipoDeFuncionario();
        }

    }

    public static String VerificarTipoDeFuncionario() {

        System.out.print("Informe o tipo de Funcionário : (P) Professor, A
(Administrativo) :");
        String tipoFuncionario = new Scanner(System.in).next();
        return tipoFuncionario;
    }

    public static void CriarFuncionarioAdministrativo() {

        Administrativo novoFuncionario = new Administrativo();
        double contribSindical;

        System.out.print("Informe o nome do Funcionário : ");
        novoFuncionario.nome = new Scanner(System.in).next();
    }
}

```

```

System.out.print("Informe o nome do CPF : ");
novoFuncionario.cpf = new Scanner(System.in).next();

System.out.print("Informe o salário do Funcionário : ");
novoFuncionario.salario = new Scanner(System.in).nextDouble();

System.out.print("Informe o número de dependentes : ");
novoFuncionario.dependentes = new Scanner(System.in).nextInt();

contribSindical =
novoFuncionario.getContribuicaoSindical(novoFuncionario.salario);

System.out.println("Nome do Funcionário :" + novoFuncionario.nome);
System.out.println("CPF :" + novoFuncionario.cpf);
System.out.println("Salário :" + novoFuncionario.salario);
System.out.println("Dependentes:" + novoFuncionario.dependentes);
System.out.println("Contribuição Sindical:" + contribSindical);

}

public static void CriarProfessor(){
Professor novoFuncionario = new Professor();
double impostoRenda;

System.out.print("Informe o nome do Funcionário : ");
novoFuncionario.nome = new Scanner(System.in).next();

System.out.print("Informe o nome do CPF : ");
novoFuncionario.cpf = new Scanner(System.in).next();

System.out.print("Informe o salário do Funcionário : ");
novoFuncionario.salario = new Scanner(System.in).nextDouble();

System.out.print("Informe a titulação : ");
novoFuncionario.titulacao = new Scanner(System.in).next();

impostoRenda = novoFuncionario.getImpostoRenada(novoFuncionario.salario);

System.out.println("Nome do Funcionário :" + novoFuncionario.nome);
System.out.println("CPF :" + novoFuncionario.cpf);
System.out.println("Salário :" + novoFuncionario.salario);
System.out.println("Titulacão:" + novoFuncionario.titulacao);
System.out.println("Imposto de Renda:" + impostoRenda);

}
}

```



```

Funcionario.java ✘
1
2 public class Funcionario {
3     String nome;
4     String cpf;
5     double salario;
6 }
7

Administrativo.java ✘
1
2 public class Administrativo extends Funcionario{
3
4     int dependentes;
5
6     public double getContribuicaoSindical(double salario){
7         double contribSindical;
8         return contribSindical = (salario * 1) / 100;
9     }
10 }
11

```

```
Professor.java
1
2 public class Professor extends Funcionario{
3
4     String titulacao;
5
6     public double getImpostoRenada(double salario){
7         double imposto;
8         return imposto = (salario * 23) /100;
9     }
10}
11
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [JOADSON ELIÉZER CARVALHO DE ARAÚJO](#) - quinta, 27 ago 2020, 09:57

Muito legal, Jonathan! Nos getters, acredito não ser obrigatório a declaração da variável, visto que o método já retorna o tipo que você informou, no caso, double.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [DIEGO CEZAR GOMES MAGALHAES](#) - domingo, 6 set 2020, 14:10



Ótimo código!

Poderia ser melhorado com a utilização do princípio do encapsulamento. Por exemplo, nas classes professor, administrativo e funcionário, poderia colocar os atributos em private e criar getters e setters para acessá-los nas outras classes.

Poderia haver também classes separadas de leitura e visão de dados, de forma a deixar o código mais limpo na classe principal.

Uma estrutura de repetição do-while no início também é interessante, porque permitiria o usuário repetir o experimento quantas vezes quisesse.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [ROSANA CAVALCANTE DE OLIVEIRA](#) - quarta, 26 ago 2020, 14:30

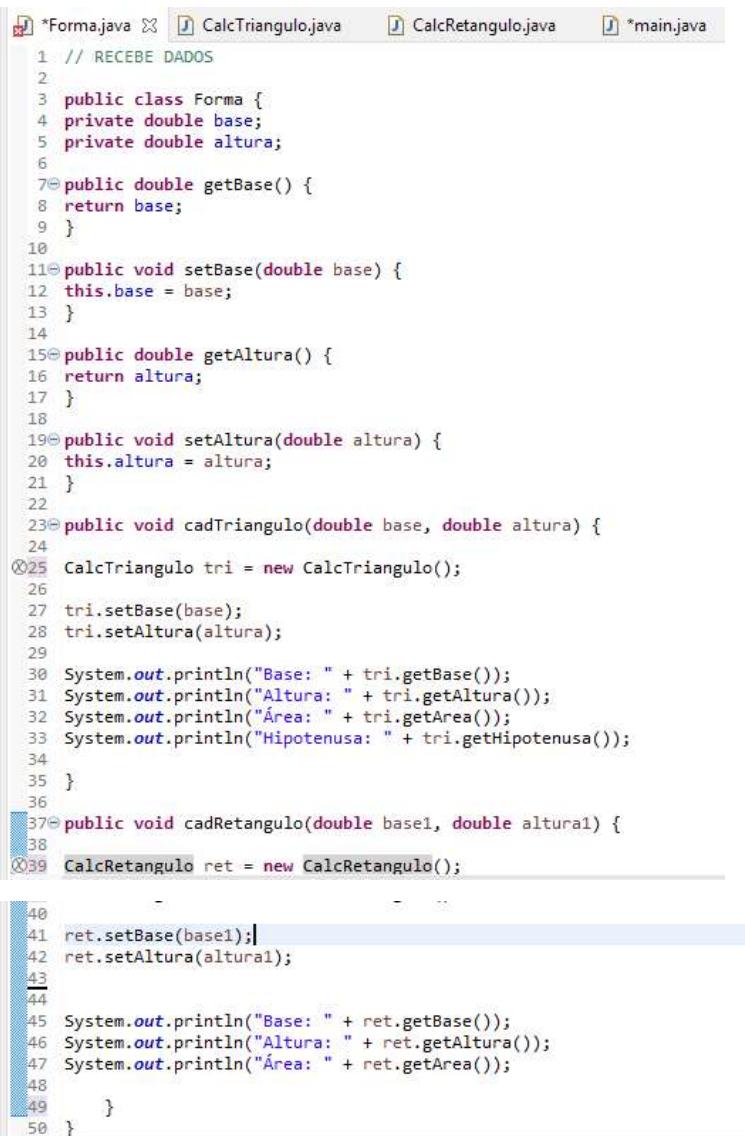
Pessoal adicionei a aula em pdf da Unidade III em [Material de Apoio 3](#)

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

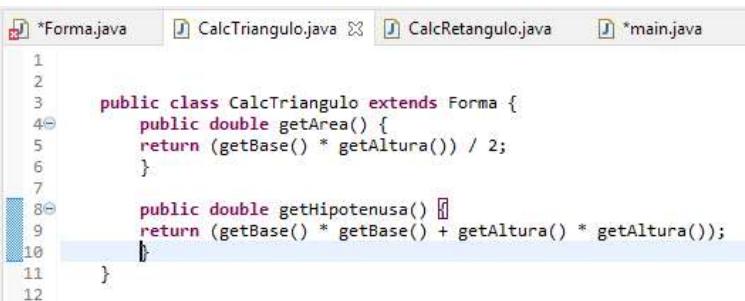
**Re: Atividade Avaliativa 3**

por [RODRIGO RIBEIRO CAMPOS](#) - quinta, 27 ago 2020, 19:59

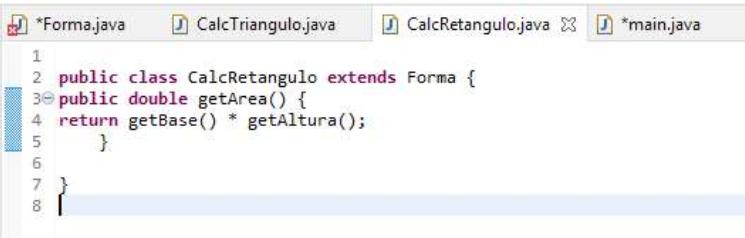
Boa noite, atividade nr. 02, aula 05



```
1 // RECEBE DADOS
2
3 public class Forma {
4     private double base;
5     private double altura;
6
7     public double getBase() {
8         return base;
9     }
10
11    public void setBase(double base) {
12        this.base = base;
13    }
14
15    public double getAltura() {
16        return altura;
17    }
18
19    public void setAltura(double altura) {
20        this.altura = altura;
21    }
22
23    public void cadTriangulo(double base, double altura) {
24
25        CalcTriangulo tri = new CalcTriangulo();
26
27        tri.setBase(base);
28        tri.setAltura(altura);
29
30        System.out.println("Base: " + tri.getBase());
31        System.out.println("Altura: " + tri.getAltura());
32        System.out.println("Área: " + tri.getArea());
33        System.out.println("Hipotenusa: " + tri.getHipotenusa());
34
35    }
36
37    public void cadRetangulo(double base1, double altural1) {
38
39        CalcRetangulo ret = new CalcRetangulo();
40
41        ret.setBase(base1);
42        ret.setAltura(altural1);
43
44
45        System.out.println("Base: " + ret.getBase());
46        System.out.println("Altura: " + ret.getAltura());
47        System.out.println("Área: " + ret.getArea());
48
49    }
50 }
```



```
1
2
3     public class CalcTriangulo extends Forma {
4         public double getArea() {
5             return (getBase() * getAltura()) / 2;
6         }
7
8         public double getHipotenusa() {
9             return (getBase() * getBase() + getAltura() * getAltura());
10        }
11    }
12
```



```
1
2     public class CalcRetangulo extends Forma {
3         public double getArea() {
4             return getBase() * getAltura();
5         }
6
7     }
8 }
```

```
*Forma.java  CalcTriangulo.java  CalcRetangulo.java  *main.java
1 import java.util.Scanner;
2
3 public class main extends Forma {
4     public static void main(String[] args) {
5
6         Scanner sc = new Scanner(System.in);
7         int opcao = 0;
8
9         System.out.println("===== Atividade 02 =====");
10        System.out.println("Escolha uma opcao:");
11        System.out.println("1 - Para Triângulo:");
12        System.out.println("2 - Para Retângulo:");
13        opcao = Integer.parseInt(sc.nextLine());
14
15        switch (opcao) {
16
17            case 1:
18
19                Forma tri = new CalcTriangulo();
20
21                System.out.println("===== Dados do Triângulo =====");
22
23                System.out.println("Base: ");
24                double base = Integer.parseInt(sc.nextLine());
25
26                System.out.println("Altura: ");
27                double altura = Integer.parseInt(sc.nextLine());
28
29                tri.cadTriangulo(base, altura);
30
31            break;
32
33            case 2:
34                Forma ret = new CalcRetangulo();
35
36                System.out.println("===== Dados do Retângulo =====");
37
38                System.out.println("Base: ");
39                double base1 = Integer.parseInt(sc.nextLine());
40
41                System.out.println("Altura: ");
42                double altura1 = Integer.parseInt(sc.nextLine());
43
44                ret.cadRetangulo(base1, altura1);
45
46            break;
47        }
48
49        return;
50    }
51 }
52 }
53 }
54
55
```

```
40
41 System.out.println("Altura: ");
42 double altura1 = Integer.parseInt(sc.nextLine());
43
44 ret.cadRetangulo(base1, altura1);
45
46 break;
47 }
48
49 return;
50 }
51 }
52 }
53 }
54
55
```

```
<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (27/08/2020 1
===== Atividade 02 =====
Escolha uma opcao:
1 - Para Triângulo:
2 - Para Retângulo:
1
===== Dados do Triângulo =====
Base:
23
Altura:
34
Base: 23.0
Altura: 34.0
Área: 391.0
Hipotenusa: 1685.0
```

```
<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (27/08/2020)
===== Atividade 02 =====
Escolha uma opcao:
1 - Para Triângulo:
2 - Para Retângulo:
2
===== Dados do Retângulo =====
Base:
12
Altura:
4
Base: 12.0
Altura: 4.0
Área: 48.0

Link direto Marcar como não lida Mostrar principal
```



Re: Atividade Avaliativa 3
por [MATHEUS JACOBINA DE JESUS](#) - sexta, 28 ago 2020, 09:17

Atividade 01 - Aula 08

```
package forum;

public class Cliente {

    private String nome;
    private String endereco;
    private EnumFPagamento formaDePagamento;

    public Cliente() {
    }

    public Cliente(String nome, String endereco, EnumFPagamento formaDeP
        this.nome = nome;
        this.endereco = endereco;
        this.formaDePagamento = formaDePagamento;
    }

    public String getNome() {
        return this.nome;
    }

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

    public String getEndereco() {
        return this.endereco;
    }

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

    public EnumFPagamento getFormaDePagamento() {
        return this.formaDePagamento;
    }

    public void setFormaDePagamento(EnumFPagamento formaDePagamento) {
        this.formaDePagamento = formaDePagamento;
    }

    public Cliente nome(String nome) {
        this.nome = nome;
```



```

        return this;
    }

    public Cliente endereço(String endereço) {
        this.endereço = endereço;
        return this;
    }

    public Cliente formaDePagamento(EnumFPagamento formaDePagamento) {
        this.formaDePagamento = formaDePagamento;
        return this;
    }

}

package forum;

public enum EnumFPagamento {
    VISA("Visa"), MASTER("Master"), BOLETO("Boleto");
    private String nome;

    EnumFPagamento(String nome){
        this.nome = nome;
    }

    public String nome() {
        return this.nome;
    }
}

```



```

package forum;

import java.util.Scanner;

public class App
{
    public static void main( String[] args )
    {
        Cliente c = new Cliente();
        Scanner scan = new Scanner(System.in);
        String[] s;

        boolean cadastrado = false;
        do {
            System.out.println(" Digite nome: ");
            c.setNome(scan.nextLine());

            System.out.println(" Digite endereço: ");
            c.setEndereço(scan.nextLine());

            System.out.println(" Digite a forma de pagamento: 1 para VISA | 2 para ");
            s = scan.nextLine().trim().split(" ");

            int n = Integer.parseInt(s[0]);
            switch (n) {
                case 1:
                    c.setFormaDePagamento(EnumFPagamento.VISA);
                    cadastrado = true;
                    break;
                case 2:
                    c.setFormaDePagamento(EnumFPagamento.MASTER);
            }
        } while (!cadastrado);
    }
}

```

```

        cadastrado = true;
        break;
    case 3:
        c.setFormaDePagamento(EnumFPagamento.BOLETO);
        cadastrado = true;
        break;
    default:
        System.out.println("\n Erro - Tente Novamente!\n");
        break;
    }
} while(!cadastrado);

System.out.println(
    "\n Cliente " +
    "\n Nome: " + c.getNome() +
    "\n Endereço: " + c.getEndereco() +
    "\n Forma de Pagamento: " + c.getFormaDePagamento().nome()
);
}
}
}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [RAIMUNDO FAGNER MONTEIRO](#) - sexta, 28 ago 2020, 23:41

Unidade 3 - Aula 5 - Atividade 1

```

import java.util.Scanner;

public class Programa {

    public static void main(String[] args) {

        char profissao;

        double ImpRenda;

        Professor prof = new Professor();

        Administrativo admin = new Administrativo();

        do {

            System.out.println("Administrativo - A/a ");

            System.out.println("Professor - P/p ");

            System.out.printf("Qual a sua profissão? ");

            profissao = new Scanner(System.in).next().charAt(0);

```

```
if(profissao != 'A' && profissao != 'P' && profissao != 'a' && profissao != 'p') {  
  
    System.out.println("-----");  
  
    System.out.println("Caractere Inválido! Digite Novamente!");  
  
    System.out.println("-----");  
  
}  
  
}  
  
}while (profissao != 'A' && profissao != 'P' && profissao != 'a' && profissao  
!= 'p');
```

```
System.out.println("-----");
```

```
if (profissao == 'P' || profissao == 'p') {
```

```
    System.out.printf("Qual o seu nome? ");  
  
    prof.nome = new Scanner(System.in).nextLine();
```

```
    System.out.printf("Qual o seu cpf? ");  
  
    prof.cpf = new Scanner(System.in).nextLine();
```

```
    System.out.printf("Qual o seu Salário Bruto? ");  
  
    prof.salario = new Scanner(System.in).nextDouble();
```

```
    System.out.printf("Qual a sua titulação? ");  
  
    prof.titulacao = new Scanner(System.in).nextLine();
```

```
System.out.println("-----");
```

```
System.out.println("Confirmar Dados : ");  
  
System.out.println("Nome: "+prof.nome);  
  
System.out.println("CPF: "+prof.cpf);  
  
System.out.println("Salário Líquido: "+(prof.salario -  
getImpostoRenda(prof.salario)));  
  
System.out.println("Titulação: "+prof.titulacao);
```



```
else {

System.out.printf("Qual o seu nome? ");

admin.nome = new Scanner(System.in).nextLine();

System.out.printf("Qual o seu cpf? ");

admin.cpf = new Scanner(System.in).nextLine();

System.out.printf("Qual o seu Salário Bruto? ");

admin.salario = new Scanner(System.in).nextDouble();

System.out.printf("Possui quantos dependentes? ");

admin.dependentes = new Scanner(System.in).nextInt();

System.out.println("-----");

System.out.println("Confirmar Dados : \n");

System.out.println("Nome: "+admin.nome);

System.out.println("CPF: "+admin.cpf);

System.out.println("Salário Líquido: "+(admin.salario -
getContribuicaoSindical(admin.salario)));

System.out.println("Dependentes: "+admin.dependentes);

}

}

public static double getImpostoRenda(double salario) {

return(0.23*salario);

}

public static double getContribuicaoSindical(double salario) {

return(0.01*salario);

}

}
```



```
class Funcionario{  
    String nome;  
    String cpf;  
    double salario;  
  
}  
  
class Professor extends Funcionario{  
    String titulacao;  
}  
  
class Administrativo extends Funcionario{  
    int dependentes;  
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [RHENAN MARTINS CARNEIRO DE SOUZA](#) - domingo, 30 ago 2020, 19:01



Boa noite,

Bom código (totalmente executável), porém existe muito espaçamento que não interferiria na execução do código. Com isso, torna o mesmo muito extenso e um pouco difícil a leitura, podendo eliminar e enxutar o espaçoamento e apresentar um código organizado e limpo.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ROSANA CAVALCANTE DE OLIVEIRA](#) - segunda, 31 ago 2020, 16:38

Olá Rhenan, além disso, sugiro a todos adicionarem nos comentários a descrição do seu código. Facilita a leitura e a compreensão.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [JOADSON ELIÉZER CARVALHO DE ARAÚJO](#) - sábado, 29 ago 2020, 12:38

Polimorfismo - Atividade 4, Itens 1 e 2

```
import java.util.Scanner;  
import java.util.ArrayList;  
public class teste {  
  
    public static void main(String[] args) {  
  
        // ArrayList<Terrorista> homens = new ArrayList<Terrorista>();  
  
        ArrayList<Suicidas> suicidas = new ArrayList<Suicidas>();  
        ArrayList<Naosuicidas> naosuicidas = new ArrayList<Naosuicidas>();  
        Scanner input = new Scanner(System.in);  
        boolean informar = true;  
        String nome;
```



```
int tipo;
int qtd;
String religiao;
String paisOrigem;
do {

    System.out.println("Informe o Terrorista: 1-Suicida, 2-Não Suicida");
    tipo = input.nextInt();
    if(tipo == 1) {
        clearBuffer(input);
        System.out.println("Informe o nome:");
        nome = input.nextLine();

        System.out.println("Informe a qtd de explosivos:");
        qtd = input.nextInt();

        clearBuffer(input);
        System.out.println("Informe a religião:");
        religiao = input.nextLine();
        suicidas.add(new Suicidas(nome, qtd, religiao));

    }else {
        clearBuffer(input);
        System.out.println("Informe o nome:");
        nome = input.nextLine();
        System.out.println("Informe a qtd de explosivos:");
        qtd = input.nextInt();
        clearBuffer(input);
        System.out.println("Informe o pais de origem:");
        paisOrigem = input.nextLine();
        naosuicidas.add(new Naosuicidas(nome, qtd, paisOrigem));
    }

}

System.out.println("Deseja catalogar outro Terrorista? s- SIM / n - NAO");
if (input.next().charAt(0) == 'n') {
informar = false;

}
}while(informar);

System.out.println("SUICIDAS");
for (Suicidas i : suicidas) {
System.out.println(" Nome: " + i.getNome() + " Qdt Explosivos: " +
i.getQtdExplosivos()
+ "Religião: " + i.getReligiao());
}

System.out.println("NÃO SUICIDAS");
for (Naosuicidas i : naosuicidas) {
System.out.println(" Nome: " + i.getNome() + " Qtd Explosivos: " +
i.getQtdExplosivos()
+ " País de Origem: " + i.getPaisOrigem());
}

System.out.println("SUICIDAS - PERICULOSIDADE > 50");
for (Suicidas i : suicidas) {
if(i.grauPericulosidade() > 50) {
System.out.println("Nome: " + i.getNome() + " Qtd Explosivos: " +
i.getQtdExplosivos()
+ " Grau Periculosidade: " + i.grauPericulosidade());
}else {
System.out.println("Não há.");
}
}

System.out.println("NÃO SUICIDAS - PERICULOSIDADE > 50");
```

```

for (Naosuicidas i : naosuicidas) {
    if(i.grauPericulosidade() > 50) {
        System.out.println(" Nome: " + i.getNome() + " Qtd Explosivos: " +
        i.getQtdExplosivos() +
        " Grau Periculosidade: " + i.grauPericulosidade());
    }else {
        System.out.println("Não há.");
    }
}
}
}

```

private static void clearBuffer(Scanner scanner) {

```

if (scanner.hasNextLine()) {
    scanner.nextLine();
}
}
}
}
```

classe Suicidas

```

public class Suicidas extends Terrorista {

private String religiao;

public Suicidas(String nome, int qdtExplosivos, String religiao) {
super(nome, qdtExplosivos);
setReligiao(religiao);
}

public String getReligiao() {
return religiao;
}

public void setReligiao(String religiao) {
this.religiao = religiao;
}

public int grauPericulosidade () {
return (super.getQtdExplosivos() * 2) * constanteReligiao();
}

public int constanteReligiao() {
if(this.getReligiao().equals("islamismo")) {
return 5;
}
return 2;
}
}
```



Classe Naosuicidas

```

public class Naosuicidas extends Terrorista {

private String paisOrigem;

public Naosuicidas(String nome, int qdtExplosivos, String paisOrigem)
{
super(nome, qdtExplosivos);
setPaisOrigem(paisOrigem);
}

public String getPaisOrigem() {
return paisOrigem;
}
}
```

```
public void setPaisOrigem(String paisOrigem) {  
    this.paisOrigem = paisOrigem;  
}  
  
public double grauPericulosidade () {  
    return (super.getQtdExplosivos() * 3) + constantePais();  
}  
  
public int constantePais() {  
    if(getPaisOrigem().equals("arabia")) {  
        return 7;  
    }  
    return 5;  
}  
  
}
```

classe mae - Terrorista

```
public class Terrorista {  
  
    private String nome;  
    private int qtdExplosivos;  
    private static int total = 0;  
  
    public Terrorista(String nome, int qdtExplosivos) {  
        setNome(nome);  
        setQtdExplosivos(qdtExplosivos);  
        Terrorista.total++;  
    }  
  
    public int getTotal() {  
        System.out.println("total de terroristas: ");  
        return Terrorista.total;  
    }  
    public String getNome() {  
        return nome;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
    public int getQtdExplosivos() {  
        return qtdExplosivos;  
    }  
    public void setQtdExplosivos(int qtdExplosivos) {  
        this.qtdExplosivos = qtdExplosivos;  
    }  
  
}
```



[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [NELSON SILVA LOPES](#) - sábado, 29 ago 2020, 18:10

Unidade 3 - Aula 6 - Atividade 1

```
import java.util.Scanner;
@SuppressWarnings("all")
public class Programa {

    public static void main(String[] args) {
        System.out.println("Informe o nome da Pessoa:");
        String nome = new Scanner(System.in).nextLine();
        System.out.println("Informe o código da Pessoa:");
        int codigo = new Scanner(System.in).nextInt();
        System.out.println("Informe o salário da Pessoa:");
        double salario = new Scanner(System.in).nextDouble();
        Pessoa pessoa = new Pessoa(nome,codigo,salario);
        String texto = pessoa.toString();
        System.out.println(texto);
    }
}

//*****
```



```
public class Pessoa {
    private String nome;
    private int codigo;
    private double salario;

    @Override //sobrescrita do método toString
    public String toString() {
        String texto = "\nNome: "+getNome()+"\nCódigo: "+getCodigo()+"\nSalário:
        "+getSalario()
        +"\nClasse: "+getClasse();
        return texto;
    }

    //métodos construtores
    public Pessoa() {}
    public Pessoa(String nome, int codigo, double salario) {
        setNome(nome);
        setCodigo(codigo);
        setSalario(salario);
    }

    //determina a classe em função do salário
    public String getClasse() {
        String classe = "A1";
        if(getSalario()<=400) classe = "Indefinida";
        else if(getSalario()<=950) classe = "F";
        else if(getSalario()<=1400) classe = "E";
        else if(getSalario()<=2300) classe = "D";
        else if(getSalario()<=4600) classe = "C";
        else if(getSalario()<=8100) classe = "B";
        else if(getSalario()<=14400) classe = "A2";

        return classe;
    }

    //métodos assessores
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public int getCodigo() {
        return codigo;
    }
    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }
    public double getSalario() {
```



```
        return salario;
    }
    public void setSalario(double salario) {
        this.salario = salario;
    }
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [CELSO PEQUENO CERQUEIRA](#) - domingo, 30 ago 2020, 18:08

ATIVIDADE 2 DA AULA 5

```
public class Main {
    public static void main(String[] args) {
        TrianguloRetangulo tr = new TrianguloRetangulo();
        Retangulo r = new Retangulo();

        tr.setAltura(7);
        tr.setBase(4);

        r.setAltura(8);
        r.setBase(6);

        System.out.println("Altura do triangulo: "+tr.getAltura());
        System.out.println("Base do triangulo: "+tr.getBase());
        System.out.println("Valor da área: "+tr.getCalculoArea());
        System.out.println("Valor da Hipotenusa: "+tr.getHipotenusa());
        System.out.println();
        System.out.println("Altura do retângulo: "+r.getAltura());
        System.out.println("Base do retângulo: "+r.getBase());
        System.out.println("Valor da área: "+r.getCalculoArea());
    }
}
```



```
public class FiguraGeometrica {
    double base;
    double altura;

    public double getCalculoArea() {
        return 0;
    }

    public double getBase() {
        return base;
    }
    public void setBase(double base) {
        this.base = base;
    }
    public double getAltura() {
        return altura;
    }
    public void setAltura(double altura) {
        this.altura = altura;
    }
}

public class TrianguloRetangulo extends FiguraGeometrica {

    public double getCalculoArea() {
        return (getBase() * getAltura())/2;
    }
}
```

```
}
```

```
public double getHipotenusas() {
    return (getBase() * getBase()) + (getAltura() * getAltura());
}
```

```
}
```

```
public class Retangulo extends FiguraGeometrica {
    public double getCalculoArea() {
        return getBase() * getAltura();
    }
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [ALESSANDRO SANTOS MEDEIROS](#) - segunda, 31 ago 2020, 22:56

Esta atividade era um pouco complicada, mas ainda assim vc fez um bom código.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [RHENAN MARTINS CARNEIRO DE SOUZA](#) - domingo, 30 ago 2020, 18:54

Unidade III - Aula 5 - Exercício 1

```
import java.util.Scanner;
```

```
public class Programa{
    public static void main(String args[]){
        Professor prof = new Professor();
        prof.setNome("Rhenan Martins");
        prof.setCpf("123.456.789-10");
        prof.setSalario(5000);
        prof.setTitulacao("Prof.");
        double ir = prof.getImpostoRenda();
        System.out.println("Sr(a) "+prof.getNome()+" irá pagar de imposto o valor: R$ "+prof.getImpostoRenda());
    }
}

class Funcionario {
    private String nome;
    private String cpf;
    private double salario;
    public void setNome(String nome){
        this.nome = nome;
    }
    public String getNome(){
        return nome;
    }
    public void setCpf(String cpf){
        this.cpf = cpf;
    }
    public String getCpf(){
        return cpf;
    }
    public void setSalario(double salario){
        this.salario = salario;
    }
    public double getSalario(){
        return salario;
    }
}

class Professor extends Funcionario{
    private String titulacao;
```



```

public void setTitulacao(String titulacao){
    this.titulacao = titulacao;
}
public String getTitulacao(){
    return titulacao;
}
public double getImpostoRenda(){
    return getSalario() * 0.23; //taxa fixa de 15% do salário
}
}
class Administrativo extends Funcionario{
    private String dependentes;
    public void setDependentes(String dependentes){
        this.dependentes = dependentes;
    }
    public String getDependentes(){
        return dependentes;
    }
    public double getContribuicaoSindical(){
        return getSalario() * 0.01; //taxa fixa de 1% do salario
    }
}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [WISTERLEYS DA SILVA SOUSA](#) - domingo, 30 ago 2020, 20:58

RHENAN MARTINS CARNEIRO DE SOUZA

Boa noite.

Percebi que você usou o encapsulamento e herança! Parabéns!!! Ficou muito bom seu programa. Só uma sugestão, tente aumentar a complexidade colocando o usuário para preencher os dados e aproveita para criar classes de serviços utilizando sobrecarga. Tirando esses detalhes que comentei seu programa ficou show de bola!

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [WISTERLEYS DA SILVA SOUSA](#) - domingo, 30 ago 2020, 21:08

Boa noite pessoal!

Realizando o exercício 1 da aula 5 - UNIDADE III

```

/*
* Aula 5 Atividade 1
*
*/
public class Main {
    public static void main(String args[]) {
        char info=Ler.charac("Que tipo de funcionário deseja informar? P - Professor ou
A - Administrativo");
        if(info=='P' || info=='p') {
            impProf(new Professor());
        }
        else if(info=='A' || info=='a') {
            impAdm(new Administrativo());
        }
        else {
            Imp.imp("Desculpe! Valor que voce informou é invalido.");
        }
    }
}

```

```

public static void impAdm(Administrativo adm) {
    adm.setNome(Ler.str("Digite o nome: "));
    adm.setCpf(Ler.str("Informe o CPF: "));
    adm.setDepedentes(Ler.numI("Informe a quantidade de depedentes: "));
    adm.setSalario(Ler.numD("Informe o salario: "));
    Imp.imp("");Imp.imp("");Imp.imp("");Imp.imp("=====");
    Imp.imp("Nome: "+adm.getNome());
    Imp.imp("CPF: "+adm.getCpf());
    Imp.imp("Salario: "+adm.getDepedentes());
    Imp.imp("Salario: "+adm.getSalario());
    Imp.imp("Contribuição Sindical: "+adm.getContribuicaoSindical());
    Imp.imp("=====");
}

public static void impProf(Professor prof) {
    prof.setNome(Ler.str("Digite o nome: "));
    prof.setCpf(Ler.str("Informe o CPF: "));
    prof.setTitulacao(Ler.str("Informe a Titulacao: "));
    prof.setSalario(Ler.numD("Informe o salario: "));
    Imp.imp("");Imp.imp("");Imp.imp("");Imp.imp("=====");
    Imp.imp("Nome: "+prof.getNome());
    Imp.imp("CPF: "+prof.getCpf());
    Imp.imp("Salario: "+prof.getSalario());
    Imp.imp("Titulacao: "+prof.getTitulacao());
    Imp.imp("Imposto de renda: "+prof.getImpostoRenda());
    Imp.imp("=====");
}
}

```

Classes externa

```

public class Funcionario {
    private String nome;
    private String cpf;
    private double salario;

    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getCpf() {
        return cpf;
    }
    public void setCpf(String cpf) {
        this.cpf = cpf;
    }
    public double getSalario() {
        return salario;
    }
    public void setSalario(double salario) {
        this.salario = salario;
    }
}

*****

```

```

public class Professor extends Funcionario{
    private String titulacao;

    public double getImpostoRenda() {
        return getSalario()*0.23;
    }
    public String getTitulacao() {
        return titulacao;
    }
}

```



```

public void setTitulacao(String titulacao) {
    this.titulacao = titulacao;
}

*****
```

public class Administrativo extends Funcionario{
 private int dependentes;

```

    public double getContribuicaoSindical() {
        return getSalario() * 0.01;
    }
    public int getDependentes() {
        return dependentes;
    }

    public void setDependentes(int dependentes) {
        this.dependentes = dependentes;
    }
}
```

```
//=====
```

Criei algumas classes de serviços

```

import java.util.Scanner;
public class Ler{
    public static String str(String msg){
        System.out.print(msg);
        return str();
    }
    public static String str(){
        return new Scanner(System.in).nextLine();
    }
    public static char charac(String msg) {
        System.out.println(msg);
        return charac();
    }
    public static char charac() {
        return new Scanner(System.in).next().charAt(0);
    }
    public static int numI( String msg){
        System.out.print(msg);
        return numI();
    }
    public static int numI(){
        return new Scanner(System.in).nextInt();
    }
    public static double numD( String msg){
        System.out.print(msg);
        return numD();
    }
    public static double numD(){
        return new Scanner(System.in).nextDouble();
    }
}
```

```
*****
```

```

import javax.swing.JOptionPane;
public class Imp{
    public static void imp(String msg){
        System.out.println(msg);
    }
    public static void imp(int msg){
```



```
System.out.println(msg);
}
public static void impPane(String msg) {
JOptionPane.showMessageDialog(null, msg);
}
public static void impPane(Double msg) {
JOptionPane.showMessageDialog(null, msg);
}
public static void imp(Double msg){
System.out.println(msg);
}
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [ALESSANDRO SANTOS MEDEIROS](#) - segunda, 31 ago 2020, 22:54

Vc usou a estrutura condicional "If", mas também poderia ter usado a Switch Case, de todo modo ficou muito bom.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [YGOR YAN ALVES DE SOUZA FERREIRA](#) - terça, 1 set 2020, 23:12

Ótimo código Wisterleys, confesso que tive dificuldade nesse exercício na parte da escolha do usuário em ser professor ou administrador porque esqueci como era feito essa função e com o seu código pude relembrar como era usando o "if" e o "else" , ta de parabéns codigo bem estruturado e me fez relembrar do uso de forma correta, obrigado!

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [GUSTAVO LESSA BENEDET](#) - segunda, 31 ago 2020, 21:11

...

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [ALESSANDRO SANTOS MEDEIROS](#) - segunda, 31 ago 2020, 22:46

Aula 5 Atividade 1:

Crie um programa Java que implemente o projeto da figura acima. Crie uma classe chamada Funcionario com os atributos (nome, cpf e salário). Crie uma classe Professor, que é um Funcionário e, além dos dados de funcionário, tem a titulação e o cálculo do seu imposto de renda por intermédio do método getImpostoRenda() que é 23 % do salário. Crie também a classe Administrativo, que também é Funcionário e tem um dado que representa a quantidade de dependentes. Não se esqueça de criar o método getContribuicaoSindical(), que retornará 1% do salário.

Seu programa deve perguntar qual tipo de Funcionário deseja informar P - Professor e A - Administrativo. Quando o usuário escolher a opção receba os dados do respectivo objeto e apresente todas as suas informações.

```
import java.util.Scanner;

public class Exercicio1 {

    public static void main(String[] args) {
        Professor p = new Professor();
        p.setNome("Eurico Miranda");
    }
}
```

```

p.setCpf("123.456.789-10");
p.setSalario(23000);

Administrativo adm = new Administrativo();
adm.setNome("Eurico Miranda");
adm.setDependentes(3);
adm.setCpf("321.123.456-90");
adm.setSalario(5000);
adm.setDependentes(3);
adm.getContribuicaoSindical();;

Scanner entrada = new Scanner(System.in);
System.out.println("Entre com P - Professor ou A - Administrativo:");
String nome = entrada.nextLine();

switch(nome){
case "P":
case "p":
System.out.println("O nome do professor é: "+p.getNome());
System.out.println("O CPF do professor acima é: "+p.getCpf());
System.out.println("O salário do professor acima é: R$"+p.getSalario());
System.out.println("Imposto de renda: R$"+p.getImpostoRenda());
break;

case "A":
case "a":
System.out.println("O nome do funcionário administrativo é: "+adm.getNome());
System.out.println("O CPF do funcionário administrativo é: "+adm.getCpf());
System.out.println("O salário do funcionário administrativo acima é de
R$"+adm.getSalario());
System.out.println("Ele possui "+adm.getDependentes()+" dependentes.");
System.out.println("E a contribuição sindical deste funcionário administrativo foi
de R$"+adm.getContribuicaoSindical());
break;
}
}
}

public class Funcionario {
private String nome;
private String cpf;
private double salario;

public String getNome() {
return nome;
}
public void setNome(String nome) {
this.nome = nome;
}
public String getCpf() {
return cpf;
}
public void setCpf(String cpf) {
this.cpf = cpf;
}
public double getSalario() {
return salario;
}
public void setSalario(double salario) {
this.salario = salario;
}
}

public class Professor extends Funcionario{

public double getImpostoRenda() {

```



```

        return getSalario()*0.23;
    }
}

public class Administrativo extends Funcionario{
private int dependentes;

public double getContribuicaoSindical() {
return getSalario()*0.01;
}

public int getDependentes() {
return dependentes;
}

public void setDependentes(int dependentes) {
this.dependentes = dependentes;
}
}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [CESAR AUGUSTO DOS SANTOS ALMEIDA JUNIOR](#) - terça, 1 set 2020, 18:00

Excelente código, ficaria melhor se os objetos fossem carregador pela interação com o usuário, ficaria mais dinâmico.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ALESSANDRO SANTOS MEDEIROS](#) - terça, 1 set 2020, 19:37

Ainda não tô nesse nível, mas tenho estudado arduamente pra evoluir meu aprendizado em desenvolvimento pq é o coração deste curso.

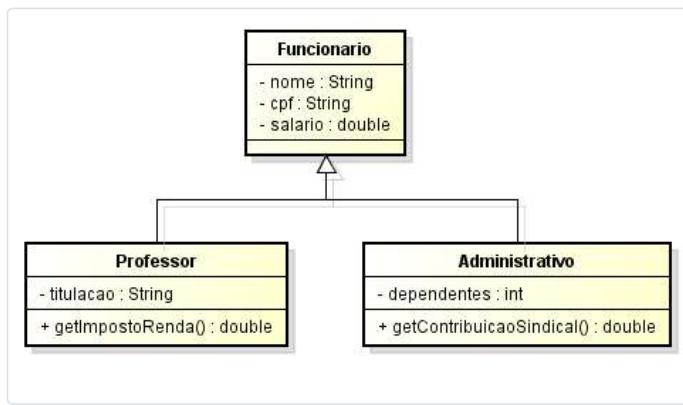
[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [CESAR AUGUSTO DOS SANTOS ALMEIDA JUNIOR](#) - terça, 1 set 2020, 17:53

Em uma universidade, há dois tipos de funcionários: professor e administrativo. Conforme a figura a seguir.



Crie um programa Java que implemente o projeto da figura acima. Crie uma classe chamada Funcionario com os atributos (nome, cpf e salário). Crie uma classe Professor, que é um Funcionário e, além dos dados de funcionário, tem a titulação e o cálculo do seu imposto de renda por intermédio do método getImpostoRenda() que é 23 % do salário. Crie também a classe Administrativo, que também é Funcionário e tem um dado que representa a quantidade de dependentes. Não se esqueça de criar o método getContribuicaoSindical(), que retornará 1% do salário.

Seu programa deve perguntar qual tipo de Funcionário deseja informar P - Professor e A - Administrativo. Quando o usuário escolher a opção receba os dados do respectivo objeto e apresente todas as suas informações.

```
import java.util.Scanner;

public class Aula05 {
    public static void main(String args[]){
        boolean validacao;
        do{
            System.out.println("Qual tipo de funcionário deseja informar?\nP - Professor \nA - Administrativo");
            String res = new Scanner(System.in).nextLine();

            if(res.equals("P")){
                Professor prof = new Professor();
                System.out.println("Digite o nome:");
                prof.setNome(new Scanner(System.in).nextLine());

                System.out.println("Digite o CPF:");
                prof.setCpf(new Scanner(System.in).nextLine());

                System.out.println("Digite o salário:");
                prof.setSalario(new Scanner(System.in).nextDouble());

                System.out.println("Digite o título:");
                prof.setTitulacao(new Scanner(System.in).nextLine());

                System.out.println("===== RESULTADO =====");
                System.out.println(""\nNome: " + prof.getNome() +
                "\nCPF: " + prof.getCpf() +
                "\nSalário: " + prof.getSalario() +
                "\nTítulo: " + prof.getTitulacao() +
                "\nImposto de renda: " + prof.getImpostoRenda(prof.getSalario()) +
                "\n");

                validacao = true;
            }else if(res.equals("A")){
                Administrativo adm = new Administrativo();
                System.out.println("Digite o nome:");
                adm.setNome(new Scanner(System.in).nextLine());

                System.out.println("Digite o CPF:");
                adm.setCpf(new Scanner(System.in).nextLine());

                System.out.println("Digite o salário:");
                adm.setSalario(new Scanner(System.in).nextDouble());

                System.out.println("Digite a quantidade de dependentes:");
                adm.setDependentes(new Scanner(System.in).nextInt());

                System.out.println("===== RESULTADO =====");
                System.out.println(""\nNome: " + adm.getNome() +
                "\nCPF: " + adm.getCpf() +
                "\nSalário: " + adm.getSalario() +
                "\nDependentes: " + adm.getDependentes() +
                "\nContribuição Sindical: " + adm.getContribuicaoSindical(adm.getSalario()) +
                "");

                validacao = true;
            }else{
                System.out.println("ERRO - Você informou o parâmetro de entrada errado!");
                validacao = false;
            }
        } while(!validacao);
    }
}
```



```
public class Funcionario {  
  
    private String nome;  
    private String cpf;  
    private double salario;  
  
    public String getNome() {  
        return nome;  
    }  
  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
  
    public String getCpf() {  
        return cpf;  
    }  
  
    public void setCpf(String cpf) {  
        this.cpf = cpf;  
    }  
  
    public double getSalario() {  
        return salario;  
    }  
  
    public void setSalario(double salario) {  
        this.salario = salario;  
    }  
}
```



```
public class Professor extends Funcionario {  
    private String titulacao;  
  
    public double getImpostoRenda(double salario){  
        return salario * 0.23;  
    }  
  
    public void setTitulacao(String titulacao) {  
        this.titulacao = titulacao;  
    }  
  
    public String getTitulacao() {  
        return titulacao;  
    }  
}
```

```
public class Administrativo extends Funcionario {  
  
    private int dependentes;  
  
    public double getContribuicaoSindical(double salario){  
        return salario * 0.1;  
    }  
  
    public void setDependentes(int dependentes) {  
        this.dependentes = dependentes;  
    }  
  
    public int getDependentes() {  
        return dependentes;  
    }  
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Excelente, código pode ser aproveitado no [trabalho final](#) da disciplina!

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

Re: Atividade Avaliativa 3

por [FRANCISCO WALTER MOREIRA](#) - terça, 1 set 2020, 21:31

AULA 5 - ATIVIDADE 2

```
public class Programa {  
    public static void main (String args[]) {  
  
        Triangulo triangulo = new Triangulo();  
        //Cria objeto triângulo  
        Retangulo retangulo = new Retangulo();  
        //Cria objeto retângulo  
  
        triangulo.setBase(3);  
        triangulo.setAltura(4);  
        System.out.println("O valor da hipotenusa  
        do triângulo é: " +  
        triangulo.getValorHipotenusa());  
        System.out.println("A área do triângulo é: "  
        + triangulo.getValorArea());
```



```
        retangulo.setBase(6);  
        retangulo.setAltura(2);  
        System.out.println("A área do retângulo é:  
        " + retangulo.getValorArea());  
    }  
}
```

```
class Figura { //Possui atributos comuns ao  
Triângulo e ao Retângulo  
private double base;  
private double altura;
```

```
    public void setBase (double base) {  
        this.base = base;  
    }  
    public double getBase() {  
        return base;  
    }
```

```
public void setAltura (double altura) {  
    this.altura = altura;  
}  
public double getAltura() {  
    return altura;  
}  
public double getArea() {  
    return 0.0;  
}  
}
```

class Triangulo extends Figura { //classe
Triângulo tem vínculo de herança com a
classe Figura

```
public double getValorArea() {  
    return (this.getBase() * this.getAltura()) /  
    2.0; // cálculo da área do triângulo  
}  
public double getValorHipotenusa() {  
    return Math.sqrt(this.getBase() *  
    this.getBase() + this.getAltura() *  
    this.getAltura()); //cálculo da hipotenusa  
}  
}
```



class Retangulo extends Figura { //classe
Retângulo tem vínculo de herança com a
classe Figura

```
public double getValorArea() {  
    return (this.getBase() * this.getAltura());  
//cálculo da área do retângulo  
}  
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [WANDERLEY FERNANDES RABELO FILHO](#) - sábado, 5 set 2020, 00:11

O seu código ficou muito bem escrito e comentado, achei bem fluido o entendimento. O único comentário que eu tenho a respeito, na tentativa de utilizar mais os recursos do java, seria na multiplicação da hipotenusa do triângulo retângulo, utilizar o método Math.pow(), que consegue realizar o

cálculo de potenciação. No mais, acho que foi um dos melhores códigos que acompanhei por aqui, no que tange a leitura e clareza. Parabéns!

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [PEDRO VINÍCIUS BARBOSA DO VALLE FURTADO](#) - sexta, 11 set 2020, 17:05

Parabéns pelo código! Gostei particularmente do uso de comentários, assim fica de mais fácil entendimento.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)

Re: Atividade Avaliativa 3

por [ANTONIO VINICIUS XIMENES DA ROSA](#) - terça, 1 set 2020, 22:02

```
//classe Main
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        String tipo;
        String resp;
        do {
            do{
                tipo = dados("Qual tipo de funcionario deseja informar? (P - Professor | A - Administrador)");
                if(tipo.equals("P")){
                    Professor prof = new Professor();
                    prof.setNome(dados("Nome: "));
                    prof.setCpf(dadosInt("Cpf: "));
                    prof.setSalario(dadosDouble("Salario: "));
                    prof.setTitulacao(dados("Titulação: "));
                    prof.getImpostoRenda();
                    System.out.println(prof.toString());
                    tipo = "";
                }else if(tipo.equals("A")){
                    Administrativo admin = new Administrativo();
                    admin.setNome(dados("Nome: "));
                    admin.setCpf(dadosInt("Cpf: "));
                    admin.setSalario(dadosDouble("Salario: "));
                    admin.setDependentes(dadosInt("Dependentes: "));
                    admin.getContribuicaoSindical();
                    System.out.println(admin.toString());
                    tipo = "";
                }else{
                    System.out.println("Tipo Invalido!!!\"");
                }
            }while( tipo.equals("A") || tipo.equals("P"));

            resp = dados("deseja Continuar?(S|N)");
        } while (resp.equals("S"));
    }
}
```

```
public static String dados(String mensagem){
    System.out.println(mensagem);
    return dados();
}
public static String dados(){
    return new Scanner(System.in).next();
}
public static int dadosInt(String mensagem){
    System.out.println(mensagem);
    return dadosInt();
}
```



```
public static int dadosInt(){
    return new Scanner(System.in).nextInt();
}

public static double dadosDouble(String mensagem){
    System.out.println(mensagem);
    return dadosDouble();
}

public static double dadosDouble(){
    return new Scanner(System.in).nextInt();
}

@Override
public String toString() {

    return "";
}

}

//classe Funcionario

import java.util.Scanner;

public class Funcionario {
    protected String nome;
    protected int cpf;
    protected double salario;

    public String getNome() {
        return nome;
    }

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

    public int getCpf() {
        return cpf;
    }

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

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salario) {
        this.salario = salario;
    }

}

//classe Professor

public class Professor extends Funcionario {
    private String titulacao;

    public double getImpostoRenda(){

        return 0.23 * salario;
    }

    public String getTitulacao() {
        return titulacao;
    }

    public void setTitulacao(String titulacao) {
```



```

this.titulacao = titulacao;
}

@Override
public String toString() {

return "----Professor----" + "\nNome: " + nome + "\nCpf: " + cpf + "\nTitulação:
"+titulacao+"\nSalario: "+ salario + "\nImposto de Renda: " +
getImpostoRenda();
}
}

//classe Administrativo

public class Administrativo extends Funcionario{

private int dependentes;

public double getContribuicaoSindical(){
return 0.01 * salario;

}

public int getDependentes() {
return dependentes;
}

public void setDependentes(int dependentes) {
this.dependentes = dependentes;
}

}


```



```

@Override
public String toString() {

return "----Administrativo----" + "\nNome: " + nome + "\nCpf: " + cpf + "
\nDependentes: "+dependentes+"\nSalario: "+ salario + "\nContribuição
Sindical: " + getContribuicaoSindical();
}
}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [ANTONIO VINICIUS XIMENES DA ROSA](#) - quarta, 2 set 2020, 01:31

AULA 05 ATV 01

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [YGOR YAN ALVES DE SOUZA FERREIRA](#) - terça, 1 set 2020, 23:16

5.2-Um matemático trabalha com dois tipos de figuras: os triângulos retângulos e os retângulos. Crie um programa Java (classes) que represente a modelagem para abstrair este problema. Ambos têm dois dados comuns: Base e Altura. Cada uma tem um cálculo de forma distinta de área. O triângulo retângulo é base X altura / 2, já do retângulo é apenas base X altura. Um cálculo diferente somente para o triângulo retângulo deve ser realizado, que é a hipotenusa. Crie um programa de teste, que instancie objetos, atribua valores e execute para apresentação.

```

import java.lang.*;
public class Programa {
public static void main (String args[]) {

```

```

TrianguloRetangulo triangulo = new TrianguloRetangulo();
Retangulo retangulo = new Retangulo();
triangulo.setBase(2.0);
triangulo.setAltura(4.0);
System.out.println("Área: " + triangulo.getArea());
System.out.println("Hipotenusa: " + triangulo.getHipotenusa());
retangulo.setBase(8.0);
retangulo.setAltura(5.0);
System.out.println("Área: " + retangulo.getArea());
}
}

class Figura {
private double base;
private double altura;
public void setBase (double base) {
this.base = base;
}
public double getBase() {
return base;
}
public void setAltura (double altura) {
this.altura = altura;
}
public double getAltura() {
return altura;
}
public double getArea() {
return 0.0;
}
}

class TrianguloRetangulo extends Figura {
public double getArea() {
return (this.getBase() * this.getAltura() / 2.0);
}
public double getHipotenusa() {
return Math.sqrt(this.getBase() * this.getBase() + this.getAltura() *
this.getAltura());
}
}

class Retangulo extends Figura {
public double getArea() {
return (this.getBase() * this.getAltura());
}
}

```



Att;

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [RAFAEL DOMINGOS BENTO](#) - sexta, 4 set 2020, 20:30

Boa noite YGOR YAN, tudo bem?

Achei muito bom o seu código, consegui fazer a leitura e execução facilmente, isso é muito importante para nós programadores que precisam dar manutenções futuras.

Grato pela oportunidade,
Rafael Bento

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [EDEN IVO BEZERRA](#) - quarta, 2 set 2020, 00:33

Unidade 3 - Aula 5 - Atividade 1

```

/*
* Síntese:
* Crie uma classe chamada Funcionario com os atributos (nome, cpf e salário).
*/
public class Funcionario {
    private String nome;
    private String cpf;
    private double salario;

    // Gets and Sets
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getCpf() {
        return cpf;
    }
    public void setCpf(String cpf) {
        this.cpf = cpf;
    }
    public double getSalario() {
        return salario;
    }
    public void setSalario(double salario) {
        this.salario = salario;
    }
}

*****
/* 
* Síntese:
* Crie uma classe Professor, que é um Funcionário e, além dos dados de
funcionário,
* tem a titulação e o cálculo do seu imposto de renda por intermédio do método
getImpostoRenda() que é 23 % do salário.
*/
public class Professor extends Funcionario {
    private String titulacao;

    // Função Definida pelo Usuário
    public double getImpostoRenda() {
        return getSalario() * 0.23;
    }

    // Gets and Sets
    public String getTitulacao() {
        return titulacao;
    }
    public void setTitulacao(String titulacao) {
        this.titulacao = titulacao;
    }
}

*****
/*
* Síntese:
* Crie também a classe Administrativo, que também é Funcionário e tem um
 dado que representa a quantidade de dependentes.
* Não se esqueça de criar o método getContribuicaoSindical(), que retornará 1%
 do salário.
*/
public class Administrativo extends Funcionario {
    private int qtdeDependentes;

```



```

// Função Definida pelo Usuário
public double getContribuicaoSindical() {
    return getSalario() * 0.01;
}

// Gets and Sets
public int getQtdeDependentes() {
    return qtdeDependentes;
}
public void setQtdeDependentes(int qtdeDependentes) {
    this.qtdeDependentes = qtdeDependentes;
}
}

*****
*/

/*
* Síntese:
* Seu programa deve perguntar qual tipo de Funcionário deseja informar P -
Professor e A - Administrativo.
* Quando o usuário escolher a opção receba os dados do respectivo objeto e
apresente todas as suas informações.
*/
public class Principal {
    public static void main(String[] args) {
        char opcao = ScannerUCB.lerChar("Qual tipo de Funcionário deseja informar? (P
- Professor e A - Administrativo)");
        if(opcao == 'P' || opcao == 'p') {
            Professor prof = new Professor();
            prof.setNome(ScannerUCB.lerString("Informe o nome do professor: ")); //
Método acessor herdado
            prof.setCpf(ScannerUCB.lerString("Informe o CPF (sem os pontos e o traço) do
professor [" +prof.getNome()+ "]: ", 0, 11)); // Método acessor herdado
            prof.setSalario(ScannerUCB.lerDouble("Informe o salário (de R$ 0,00 a R$
10.000,00) do professor [" +prof.getNome()+ "]: ", 0, 10000)); // Método acessor
herdado
            prof.setTitulacao(ScannerUCB.lerString("Informe a titulação (Doutor, Mestre,
Graduado) do professor [" +prof.getNome()+ "]: ")); // Método acessor de
professor

            // Apresentando todas as informações
            System.out.println("\n-----");
            System.out.println("Apresentando todas as informações (PROFESSOR)");
            System.out.println("-----\n");
            System.out.println("O(A) Sr(a) "+prof.getNome()+" , professor com titulação de
["+prof.getTitulacao+
"], portador do CPF nº ["+prof.getCpf()+
"], irá pagar de imposto de renda o valor de [R$ "+prof.getImpostoRenda()+
"], referente a 23% do seu salário de [R$ "+prof.getSalario()+"].");
        } else if(opcao == 'A' || opcao == 'a') {
            Administrativo adm = new Administrativo();
            adm.setNome(ScannerUCB.lerString("Informe o nome do administrativo: ")); //
Método acessor herdado
            adm.setCpf(ScannerUCB.lerString("Informe o CPF (sem os pontos e o traço) do
administrativo [" +adm.getNome()+ "]: ", 0, 11)); // Método acessor herdado
            adm.setSalario(ScannerUCB.lerDouble("Informe o salário (de R$ 0,00 a R$
10.000,00) do administrativo [" +adm.getNome()+ "]: ", 0, 10000)); // Método
acessor herdado
            adm.setQtdeDependentes(ScannerUCB.lerInt("Informe a quantidade de
dependentes (de 0 a 10) do administrativo [" +adm.getNome()+ "]: ", 0, 10)); //
Método acessor de professor

            // Apresentando todas as informações
            System.out.println("\n-----");
            System.out.println("Apresentando todas as informações (ADMINISTRATIVO)");
            System.out.println("-----\n");
            System.out.println("O(A) Sr(a) "+adm.getNome()+" , funcionário administrativo
com [" +adm.getQtdeDependentes()+ "] dependentes registrados"+
", portador do CPF nº ["+adm.getCpf()+
"]

```



```
"], irá pagar de contribuição sindical o valor de [R$  
"+adm.getContribuicaoSindical()  
"], referente a 1% do seu salário de [R$ "+adm.getSalario()+"].");  
} else {  
System.out.println("\n-----");  
System.out.println("Ops! O(A) Senhor(a) escolheu uma opção não disponível");  
}  
  
}  
}  
  
*****  
  
/*  
* Síntese:  
* Esta classe reúne todas as entradas para o usuário  
*/  
import java.util.Scanner;  
public class ScannerUCB {  
// Entrada simples para integer  
public static int lerInt() {  
return new Scanner(System.in).nextInt();  
}  
// Entrada com msg para integer  
public static int lerInt(String msg) {  
System.out.println(msg);  
return lerInt();  
}  
// Entrada com msg e com limites (mínimo e máximo) para integer  
public static int lerInt(String msg, int min, int max) {  
int valor;  
boolean range;  
do {  
valor = lerInt(msg);  
range = (valor < min || valor > max);  
if(range) {  
System.out.println("\n-----");  
System.out.println("Ops! número informado está fora do limite [ "+min+" a  
"+max+" ]");  
System.out.println("-----\n");  
}  
}while(range);  
return valor;  
}  
// *****  
// Entrada simples para double  
public static double lerDouble() {  
return new Scanner(System.in).nextDouble();  
}  
// Entrada com msg para double  
public static double lerDouble(String msg) {  
System.out.println(msg);  
return lerDouble();  
}  
// Entrada com msg e com limites (mínimo e máximo) para double  
public static double lerDouble(String msg, double min, double max) {  
double valor;  
boolean range;  
do {  
valor = lerDouble(msg);  
range = (valor < min || valor > max);  
if(range) {  
System.out.println("\n-----");  
System.out.println("Ops! número informado está fora do limite [ "+min+" a  
"+max+" ]");  
System.out.println("-----\n");  
}  
}while(range);  
return valor;
```



```
}

// ****
// Entrada simples para string
public static String lerString() {
    return new Scanner(System.in).nextLine();
}

// Entrada com msg para string
public static String lerString(String msg) {
    System.out.println(msg);
    return lerString();
}

// Entrada com msg e com limites (mínimo e máximo) para string
public static String lerString(String msg, int min, int max) {
    String texto;
    boolean range;
    do {
        texto = lerString(msg);
        range = (texto.length() < min || texto.length() > max);
        if(range) {
            System.out.println("\n-----\n");
            System.out.println("Ops! número de caracteres informado está fora do limite [");
            System.out.println("+" + min + " a " + max + " ]");
            System.out.println("-----\n");
        }
    }while(range);
    return texto;
}

// ****
// Entrada simples para char
public static char lerChar() {
    return new Scanner(System.in).next().charAt(0);
}

// Entrada com msg para char
public static char lerChar(String msg) {
    System.out.println(msg);
    return lerChar();
}
```



[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [NANDESSON CAPUCHINHO DE FARIA](#) - sexta, 4 set 2020, 09:44

Muito Bom seu código ficou realmente bem estruturado e complexo você conseguiu preencher os dados e aproveitar as classes de serviços. Parabéns.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ALEX DA SILVA](#) - quarta, 9 set 2020, 10:02

Bom dia Eden.

Rapaz, muito bom seu código. A interatividade com o usuário ficou bem definida, e os métodos limpos para manutenções futuras ou melhorias. Parabéns



Re: Atividade Avaliativa 3
por [LARA PATTY RODRIGUES BARBOSA](#) - quarta, 2 set 2020, 20:53

```
//CLASSE PRINCIPAL

package unidade3;

import java.util.Scanner;

public class Aula6 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        String name;
        float salary;

        System.out.println("Qual seu nome?");
        name = sc.nextLine();
        System.out.println("Qual seu salário?");
        salary = sc.nextFloat();

        Pessoa pessoa1 = new Pessoa();

        pessoa1.setName(name);
        pessoa1.setSalary(salary);
        pessoa1.makeClass(salary);

        System.out.println(pessoa1.toString());
    }

}

//CLASSE PESSOA

package unidade3;

public class Pessoa {

    private String name;
    private float salary;
    private String codigo;

    public Pessoa() {
    }

    public void makeClass(float salary) {
        if (salary > 400 && salary < 950) {
            this.codigo = "F";
        } else if (salary > 950 && salary < 1400) {
            this.codigo = "E";
        } else if (salary > 1400 && salary < 2300) {
            this.codigo = "D";
        } else if (salary > 2300 && salary < 4600) {
            this.codigo = "C";
        } else if (salary > 4600 && salary < 8100) {
            this.codigo = "B";
        } else if (salary > 8100 && salary < 14400) {
            this.codigo = "A2";
        } else if (salary > 14400) {
            this.codigo = "A1";
        } else {
            System.out.println("Classe não encontrada");
        }
    }
}
```



```

}

public String getName() {
    return name;
}

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

public float getSalary() {
    return salary;
}

public void setSalary(float salary) {
    this.salary = salary;
}

public String getCodigo() {
    return codigo;
}

public void setCodigo(String codigo) {
    this.codigo = codigo;
}

@Override
public String toString() {
    return "Pessoa [Nome =" + name + ", Salário = " + salary + ", Código=" +
        codigo + "]";
}
}

```



[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

Re: Atividade Avaliativa 3
por [ALLAN CARLOS BARROS MORAES](#) - quarta, 2 set 2020, 22:35

```

package ead.catolica;

import java.util.Calendar;

public class Programa {
    public static void main(String[] args) {
        Professor prof = new Professor();
        prof.setNome("Allan");
        prof.setCpf("123.456.789-00");
        prof.setSalario(11300);
        prof.setTitulacao("Especialista");

        Calendar dtAdmissao = Calendar.getInstance();
        dtAdmissao.set(Calendar.DAY_OF_MONTH, 1);
        dtAdmissao.set(Calendar.MONTH, Calendar.MARCH);
        dtAdmissao.set(Calendar.YEAR, 2000);

        Administrativo adm = new Administrativo();
        adm.setNome("Allan");
        adm.setDependentes(2);
        adm.setCpf("123.456.789-01");
        adm.setSalario(1356);
        adm.setDtAdmissao(dtAdmissao);

        Funcionario f;
        f = prof;
        System.out.println("Funcionário(a) "+f.getNome()+" tem direito de: R$ "
            "+f.getGratificacao());
        f = adm;
    }
}

```

```
System.out.println("Funcionário(a) "+f.getNome()+" tem direito de: R$  
"+f.getGratificacao());  
}  
}  
  
package ead.catolica;  
  
import java.util.Calendar;  
  
public class Administrativo extends Funcionario {  
    private int dependentes;  
    private Calendar dtAdmissao;  
  
    public double getContribuicaoSindical() {  
        return getSalario() * 0.01;  
    }  
    public int getDiasTrabalhados() {  
        Calendar dtAtual = Calendar.getInstance();  
        int MILLIS_IN_DAY = 86400000;  
        return (int)((dtAtual.getTimeInMillis() -  
        getDtAdmissao().getTimeInMillis()) / MILLIS_IN_DAY);  
    }  
    public double getGratificacao() {  
        return getDiasTrabalhados() * 0.3;  
    }  
    public Calendar getDtAdmissao() {  
        return dtAdmissao;  
    }  
    public void setDtAdmissao(Calendar dtAdmissao) {  
        this.dtAdmissao = dtAdmissao;  
    }  
    public int getDependentes() {  
        return dependentes;  
    }  
    public void setDependentes(int dependentes) {  
        this.dependentes = dependentes;  
    }  
}  
  
package ead.catolica;  
  
public abstract class Funcionario {  
    private String nome;  
    private String cpf;  
    private double salario;  
  
    public double getGratificacao() {  
        return 0;  
    }  
    public String getNome() {  
        return nome;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
    public String getCpf() {  
        return cpf;  
    }  
    public void setCpf(String cpf) {  
        this.cpf = cpf;  
    }  
    public double getSalario() {  
        return salario;  
    }  
    public void setSalario(double salario) {  
        this.salario = salario;  
    }  
}
```



```
}
```

```
package ead.catolica;
```

```
public class Professor extends Funcionario {  
    private String titulacao;  
  
    public double getGratificacao() {  
        if(getTitulacao().equalsIgnoreCase("especialista")) {  
            return getSalario() * 0.10;  
        } else if (getTitulacao().equalsIgnoreCase("mestre")) {  
            return getSalario() * 0.12;  
        } else if(getTitulacao().equalsIgnoreCase("doutor")){  
            return getSalario() * 0.15;  
        }  
        return 0;  
    }  
    public double getImpostoRenda() {  
        return getSalario() * 0.15;  
    }  
    public String getTitulacao() {  
        return titulacao;  
    }  
}
```

```
public void setTitulacao(String titulacao) {  
    this.titulacao = titulacao;  
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [NANDESSON CAPUCHINHO DE FARIA](#) - sexta, 4 set 2020, 09:37

Bom Dia Professora e colegas!

Realizei a Atividade 1 da Aula 5 - Unidade 3

```
import java.util.Scanner;
```

```
public class Atividade{
```

```
    public static void main(String args[]){  
        Professor prof = new Professor();  
        prof.setNome("Andre Sousa");  
        prof.setCpf("123.456.789-10");  
        prof.setSalario(2000);  
        prof.setTitulacao("Prof.");  
        double ir = prof.getImpostosRen();  
        System.out.println("Sr(a) "+prof.getNome()+" devera pagar os impostos de  
Renda no valor: R$ "+prof.getImpostosRen());  
    }  
}
```

```
class Funcionario {  
    private String nome;  
    private String cpf;  
    private double salario;  
    public void setNome(String nome){  
        this.nome = nome;  
    }
```

```
    public String getNome(){  
        return nome;
```

```

}

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

public String getCpf(){
return cpf;
}

}

public void setSalario(double salario){
this.salario = salario;
}

public double getSalario(){
return salario;
}

}

class Professor extends Funcionario{
private String titulacao;
public void setTitulacao(String titulacao){
this.titulacao = titulacao;
}

public String getTitulacao(){
return titulacao;
}

public double getImpostosRen(){
return getSalario() * 0.23; //taxa fixa de 15% do salário
}

}

class Administrativo extends Funcionario{
private String dependentes;
public void setDependentes(String dependentes){
this.dependentes = dependentes;
}

public String getDependentes(){
return dependentes;
}

public double getContribuicaoSindical(){
return getSalario() * 0.01; //taxa fixa de 1% do salario
}
}

```



[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [BRUNA SOUSA CARVALHO ALVES](#) - domingo, 6 set 2020, 16:34

Oi Nandesson!

Executei seu código aqui e senti falta da opção de selecionar se o funcionário é um professor ou um administrativo (para que sejam feitos os cálculos)

pertinentes conforme o cargo do funcionário).

Também senti falta de solicitar os dados do funcionário (nome, CPF, salário, titulação, dependentes) depois de dada a opção se o funcionário é professor ou administrativo!

Por fim, não achei função para a variável "double ir = prof.getImpostosRen();", uma vez que quando você coloca "System.out.println("Sr(a)" + prof.getNome() + " deverá pagar os impostos de Renda no valor: R\$ " + prof.getImpostosRen());" ele pega do dado da classe que você colocou abaixo e não dessa variável que você declarou!

Bom, são essas observações que tem, ao meu ver!

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [RAFAEL DOMINGOS BENTO](#) - sexta, 4 set 2020, 20:47

Boa noite caros colegas e professora,

Aula 05 - Atividade 01.

Segue o código digitado;

```
#####
package unidade_03;

import java.util.Scanner; // INVOCANDO A CLASSE SCANNER

public class Programa {
    public static void main(String[] args) {
        boolean opcao; // VERDADEIRO OU FALSO
        do { // AQUI COMEÇA O LAÇO DE REPETIÇÃO DO WHILE
            System.out.println("Escolha um funcionário da lista; \n 'A' para
Administrador \n 'P' para Professor");
            String resposta = new Scanner(System.in).nextLine();

            if (resposta.equals("P")) { // AQUI COMEÇA "SE" A OPÇÃO FOR "P" DE
PROFESSOR
                Professor prof = new Professor();
                System.out.println("Digite o NOME Funcionário: ");
                prof.setNome(new Scanner(System.in).nextLine());

                System.out.println("Digite a TITULAÇÃO deste Funcionário: ");
                prof.setTitulacao(new Scanner(System.in).nextLine());

                System.out.println("Digite o CPF: ");
                prof.setCpf(new Scanner(System.in).nextLine());

                System.out.println("Digite o SALÁRIO :");
                prof.setSalario(new Scanner(System.in).nextDouble());

                System.out.println("SEGUE DADOS DIGITADOS ABAIXO");
                System.out.println("Nome: " + prof.getNome());
                System.out.println("Titulação: " + prof.getTitulacao());
                System.out.println("CPF: " + prof.getCpf());
                System.out.println("Salário: " + prof.getSalario());
                System.out.println("Este funcionário tem 23% de importo de renda
sobre seu salário. O valor é: R$ " + prof.getImpostodeRenda());

                opcao = true; // VERDADEIRO OU FALSO
            } else if (resposta.equals("A")) { // AQUI COMEÇA "SE" A OPÇÃO FOR
"A" DE ADMINISTRADOR
                Administrativo adm = new Administrativo();
                System.out.println("Digite o NOME Funcionário: ");
                adm.setNome(new Scanner(System.in).nextLine());

                System.out.println("Digite a quantidade de dependentes: ");
                adm.setDependentes(new Scanner(System.in).nextInt());
            }
        }
    }
}
```



```

        System.out.println("Digite o CPF:");
        adm.setCpf(new Scanner(System.in).nextLine());

        System.out.println("Digite o SALÁRIO :");
        adm.setSalario(new Scanner(System.in).nextDouble());

        System.out.println("SEGUE DADOS DIGITADOS ABAIXO");
        System.out.println("Nome: " + adm.getNome());
        System.out.println("Número de Dependentes: " +
        adm.getDependentes());
        System.out.println("CPF: " + adm.getCpf());
        System.out.println("Salário: " + adm.getSalario());
        System.out.println("Este funcionário colabora com 1% de
Contribuição Sindical sobre seu salário. O valor é: R$ " +
adm.getContribuicaoSindical());

        opcao = true; // VERDADEIRO OU FALSO

    } else {
        System.out.println("Informar 'A' para ADMINISTRADOR ou 'P' para
PROFESSOR");
        opcao = false; // VERDADEIRO OU FALSO
    }

} while (opcao == false); // AQUI TERMINA O LAÇO DE REPETIÇÃO
DO WHILE
}

#####
package unidade_03;

public class Funcionario {

    private String nome;
    private String cpf;
    private double salario;

    // COMEÇO GET/SET
    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getNome() {
        return nome;
    }
    // FIM GET/SET

    // COMEÇO GET/SET
    public void setCpf(String cpf) {
        this.cpf = cpf;
    }

    public String getCpf() {
        return cpf;
    }
    // FIM GET/SET

    // COMEÇO GET/SET
    public void setSalario(double salario) {
        this.salario = salario;
    }

    public double getSalario() {
        return salario;
    }
    // FIM GET/SET

}
#####


```



```
package unidade_03;

public class Professor extends Funcionario {

    private String titulacao;
    //COMEÇO GET/SET
    public void setTitulacao(String titulacao) {
        this.titulacao = titulacao;
    }

    public String getTitulacao() {
        return titulacao;
    }

    public double getImpostodeRenda() {
        return getSalario() * 0.23;
    } //FIM GET/SET
}

#####
package unidade_03;

public class Administrativo extends Funcionario {

    private int dependentes;
    //COMEÇO GET/SET
    public void setDependentes(int dependentes) {
        this.dependentes = dependentes;
    }

    public int getDependentes() {
        return dependentes;
    }

    public double getContribuicaoSindical() {
        return getSalario() * 0.01;
    } //FIM GET/SET
}

#####
```

Segue o mesmo código no Eclipse;



The screenshot shows the Eclipse IDE interface with a Java file named 'Programajava.java' open. The code implements a menu-driven application for managing employee data. It uses a Scanner class to read input from the user and a series of if-else statements to determine the type of employee (Professor or Administrativo) based on the user's choice. For each employee type, it prompts for name, title, CPF, and salary, and then calculates additional values like titulacao and sindical. The code includes comments explaining the logic and variable names.

```
1 package unidade_03;
2
3 import java.util.Scanner; // INVOCANDO A CLASSE SCANNER
4
5 public class Programa {
6     public static void main(String[] args) {
7         boolean opcao; // VERDADEIRO OU FALSO
8         do { // AQUI COMEÇA O LAÇO DE REPETIÇÃO DO WHILE
9             System.out.println("Escolha um Funcionário da lista: \n 'A' para Administrador \n 'P' para Professor");
10            String resposta = new Scanner(System.in).nextLine();
11
12            if (resposta.equals("P")) { // AQUI COMEÇA "SE" A OPÇÃO PARA "P" DE PROFESSOR
13                Professor prof = new Professor();
14                System.out.println("Digite o NOME Funcionário: ");
15                prof.setName(new Scanner(System.in).nextLine());
16
17                System.out.println("Digite a TITULAÇÃO deste Funcionário: ");
18                prof.setTitulacao(new Scanner(System.in).nextLine());
19
20                System.out.println("Digite o CPF: ");
21                prof.setCpf(new Scanner(System.in).nextLine());
22
23                System.out.println("Digite o SALÁRIO :");
24                prof.setSalario(new Scanner(System.in).nextDouble());
25
26                System.out.println("SEGUINTE DADOS DIGITADOS ABAIXO");
27                System.out.println("Nome: " + prof.getName());
28                System.out.println("Titulação: " + prof.getTitulacao());
29                System.out.println("CPF: " + prof.getCpf());
30                System.out.println("Salário: " + prof.getSalario());
31                System.out.println("Este funcionário tem 23% de imposto de renda sobre seu salário. O valor é: R$ " + prof.getImposto());
32
33                opcao = true; // VERDADEIRO OU FALSO
34
35            } else if (resposta.equals("A")) { // AQUI COMEÇA "SE" A OPÇÃO PARA "A" DE ADMINISTRADOR
36                Administrativo adm = new Administrativo();
37                System.out.println("Digite o NOME Funcionário: ");
38                adm.setName(new Scanner(System.in).nextLine());
39
40                System.out.println("Digite a quantidade de dependentes: ");
41                adm.setDependentes(new Scanner(System.in).nextInt());
42
43                System.out.println("Digite o CPF: ");
44                adm.setCpf(new Scanner(System.in).nextLine());
45
46                System.out.println("Digite o SALÁRIO :");
47                adm.setSalario(new Scanner(System.in).nextDouble());
48
49                System.out.println("SEGUINTE DADOS DIGITADOS ABAIXO");
50                System.out.println("Nome: " + adm.getName());
51                System.out.println("Número de Dependentes: " + adm.getDependentes());
52                System.out.println("CPF: " + adm.getCpf());
53                System.out.println("Salário: " + adm.getSalario());
54                System.out.println("Este funcionário colabora com 1% de Contribuição Sindical sobre seu salário. O valor é: R$ " + adm.getContribuicaoSindical());
55
56                opcao = true; // VERDADEIRO OU FALSO
57
58            } else {
59                System.out.println("Informar 'A' para ADMINISTRADOR ou 'P' para PROFESSOR");
60                opcao = false; // VERDADEIRO OU FALSO
61            }
62
63        } while (opcao == false); // AQUI TERMINA O LAÇO DE REPETIÇÃO DO WHILE
64    }
65 }
66
```

Grato pela oportunidade;

Rafael Bento

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [LAISSON RODRIGUES DE BESSA](#) - domingo, 6 set 2020, 14:17

Olá Rafael, muito bom seu código, organizado e comentado. Só uma sugestão , vc pode ir usar as classes de Visão para a implementar a interação com o usuário, assim como a classe Leitor/Scanner para fazer a leitura dos dados inseridos, no mais seu programa está ótimo.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [WANDERLEY FERNANDES RABELO FILHO](#) - sexta, 4 set 2020, 23:58

Aula 5 - Atividade 02

Um matemático trabalha com dois tipos de figuras: os triângulos retângulos e os retângulos. Crie um programa Java (classes) que represente a modelagem para abstrair este problema. Ambos têm dois dados comuns: Base e Altura. Cada uma tem um cálculo de forma distinta de área. O triângulo retângulo é base X altura / 2, já do retângulo é apenas base X altura. Um cálculo diferente somente para o triângulo retângulo deve ser realizado, que é a hipotenusa. Crie um programa de teste, que instancie objetos, atribua valores e execute para apresentação.

```
///////////
// Figura.java //
///////////
package a502;

class Figura {

    private double base;
    private double altura;

    double getBase() {
        return base;
    }

    void setBase(double base) {
        this.base = base;
    }

    double getAltura() {
        return altura;
    }

    void setAltura(double altura) {
        this.altura = altura;
    }

}

///////////
// Retangulo.java //
/////////
package a502;

class Retangulo extends Figura {

    double getArea() {
        return (getBase() * getAltura());
    }

}
```



```

///////////
// TrianguloRetangulo.java //
///////////

package a502;

class TrianguloRetangulo extends Figura {

    double getArea() {
        return (getBase() * getAltura() / 2);
    }

    double getHipotenus() {
        return (Math.sqrt(Math.pow(getBase(), 2) +
        Math.pow(getAltura(), 2)));
    }

}

/////////
// Programa.java //
/////////

package a502;

class Programa {

    public static void main(String[] args) {

        Retangulo retangulo = new Retangulo();
        TrianguloRetangulo trianguloRetangulo = new
        TrianguloRetangulo();

        // inserindo valores arbitrários para o
        retângulo
        retangulo.setBase(4.0);
        retangulo.setAltura(5.0);

        // inserindo valores arbitrários para o
        triângulo retângulo
        trianguloRetangulo.setBase(3.0);
        trianguloRetangulo.setAltura(4.0);

        // mostrando resultados no console
        System.out.println("\nRETANGULO");
        System.out.println("Area: " +
        retangulo.getArea());
        System.out.println("\nTRIANGULO RETANGULO");
        System.out.println("Area: " +
        trianguloRetangulo.getArea());
        System.out.println("Hipotenusa: " +
        trianguloRetangulo.getHipotenus());

    }

}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [FRANCISCO FROTA DE AGUIAR](#) - quarta, 9 set 2020, 12:13

Muito bom Wanderley, seu código ficou ótimo, vi que usou a classe Math com métodos estáticos. É isso aí devemos sempre aproveitar a API do Java e não sofrer tentando criar ferramentas que já existem.
Gostei também do comentário bem destacado.
Só uma consideração que você poderia fazer o método area() na superclasse para sobreescrita nas subclasses, já que todas as figuras tem uma área.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [JAREDE SOUZA LIMA](#) - sábado, 5 set 2020, 10:16

```
import java.util.Scanner;
public class Main {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        String funcao;
        Professor prof = new Professor();
        Administrativo admin = new Administrativo();

        System.out.println("Informe a função do colaborador 'A' para Administrativo e 'P' para professor ");
        funcao = scan.next();

        if (funcao.equalsIgnoreCase("P")) {
            System.out.println("Informe a titulação do colaborador: ");
            prof.setTitulacao(scan.next());
            System.out.println("Informe o nome do colaborador: ");
            prof.setNome(scan.next());
            System.out.println("Informe o cpf do colaborador: ");
            prof.setCpf(scan.next());
            System.out.println("Informe o salário do colaborador: ");
            prof.setSalario(scan.nextDouble());

            System.out.println("-----");
            System.out.println("O funcionário é um professor");
            System.out.println("Titulação do funcionário: " + prof.getTitulacao());
            System.out.println("Nome do funcionário: " + prof.getNome());
            System.out.println("CPF do funcionário: " + prof.getCpf());
            System.out.println("Salário do funcionário: " + prof.getSalario());
            System.out.println("O funcionário deverá pagar : " + prof.getImpostoRenda() + " de IR");

        } else if (funcao.equalsIgnoreCase("A")) {
            System.out.println("Informe o nome do colaborador: ");
            admin.setNome(scan.next());
            System.out.println("Informe o cpf do colaborador: ");
            admin.setCpf(scan.next());
            System.out.println("Informe o salário do colaborador: ");
            admin.setSalario(scan.nextDouble());
            System.out.println("Informe a quantidade de dependentes do colaborador: ");
            admin.setDependentes(scan.nextInt());

            System.out.println("-----");
            System.out.println("O funcionário tem um cargo Administrativo");
            System.out.println("Nome do funcionário: " + admin.getNome());
            System.out.println("CPF do funcionário: " + admin.getCpf());
            System.out.println("Salário do funcionário: " + admin.getSalario());
            System.out.println("O funcionário tem : " + admin.getDependentes() + " dependentes");
            System.out.println("O funcionário deverá pagar : " + admin.getContribuicaoSindical() + " de Contribuição Sindical");
        } else {
            System.out.println("Função informada inválida");
        }
    }
}
```



```
}
```

```
class Funcionario {  
  
    private String nome;  
    private String cpf;  
    private double salario;  
  
    public String getNome() {  
        return nome;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
    public String getCpf() {  
        return cpf;  
    }  
    public void setCpf(String cpf) {  
        this.cpf = cpf;  
    }  
    public double getSalario() {  
        return salario;  
    }  
    public void setSalario(double salario) {  
        this.salario = salario;  
    }  
  
}  
  
class Professor extends Funcionario{  
  
    private String titulacao;  
  
    public double getImpostoRenda() {  
        return getSalario() * 0.23;  
    }  
    public String getTitulacao() {  
        return titulacao;  
    }  
  
    public void setTitulacao(String titulacao) {  
        this.titulacao = titulacao;  
    }  
  
}  
  
class Administrativo extends Funcionario{  
  
    private int dependentes;  
  
    public double getContribuicaoSindical() {  
        return getSalario() * 0.01;  
    }  
  
    public int getDependentes() {  
        return dependentes;  
    }  
    public void setDependentes(int dependentes) {  
        this.dependentes = dependentes;  
    }  
  
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [PATRICIA ALMEIDA DE SOUZA](#) - domingo, 6 set 2020, 22:44



Olá, Jarede.

Seu código me ajudou ainda mais no entendimento da função

equalsIgnoreCase. Obrigada!

Como sugestão e visando o reuso das funcionalidades você poderia criar uma classe com todos os métodos de leitura, por exemplo, através de métodos para ler Double, ler Int, Carectere e etc.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ROSANA CAVALCANTE DE OLIVEIRA](#) - sábado, 5 set 2020, 15:02

Excelente pessoal! Vocês podem aproveitar alguns exemplos aqui postados e elaborar o [trabalho final](#) da disciplina!

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ALEXSSANDRO DE SOUSA FEITOSA](#) - domingo, 6 set 2020, 00:08

```
import java.lang.*;  
  
public class Teste {  
    public static void main (String args[]) {  
  
        TrianguloRetangulo triangulo = new TrianguloRetangulo();  
        Retangulo retangulo = new Retangulo();  
  
        triangulo.setBase(4.0);  
        triangulo.setAltura(7.0);  
        System.out.println("Área: " + triangulo.getArea());  
        System.out.println("Hipotenusa: " + triangulo.getHipotenus());  
  
        retangulo.setBase(9.0);  
        retangulo.setAltura(2.0);  
        System.out.println("Área: " + retangulo.getArea());  
    }  
}  
  
class FigurasGeometricas {  
    private double base;  
    private double altura;  
  
    public void setBase (double base) {  
        this.base = base;  
    }  
    public double getBase() {  
        return base;  
    }  
    public void setAltura (double altura) {  
        this.altura = altura;  
    }  
    public double getAltura() {  
        return altura;  
    }  
    public double getArea() {  
        return 0.0;  
    }  
}  
  
class TrianguloRetangulo extends FigurasGeometricas {  
    public double getArea() {  
        return (this.getBase() * this.getAltura() / 2.0);  
    }  
    public double getHipotenus() {  
        return Math.sqrt(this.getBase() * this.getBase() + this.getAltura() *  
        this.getAltura());  
    }  
}
```



```
this.getAltura());  
}  
}  
  
class Retangulo extends FigurasGeometricas {  
    public double getArea() {  
        return (this.getBase() * this.getAltura());  
    }  
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [MIRIÃ COSTA BARROS DE AGUIAR](#) - segunda, 7 set 2020, 13:10

Olá, Alexssandro.
Tudo bem?

Também realizei o mesmo exercício proposto e gostaria de deixar uma sugestão bacana quanto aos modificadores de acesso.
Vi que utilizou na class FigurasGeometricas o acesso private, bem como alguns outros colegas que fizeram o mesmo exercício. Porém, ao pesquisar sobre o uso deste, pude perceber que o Private é muito restritivo e os dados são acessíveis apenas pela classe que os declara, isto é, os dados não podem ser herdados.
Logo, se torna mais interessante utilizar o Protected, pois permite às subclasses o acesso a seus atributos.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [ALEXSSANDRO DE SOUSA FEITOSA](#) - quinta, 10 set 2020, 06:22

Sugestão aceita, MIRIÃ

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [ALEXSSANDRO DE SOUSA FEITOSA](#) - sexta, 11 set 2020, 12:53

Uma classe abstrata não pode ser instanciada, ou seja, não pode ser chamada pelos seus construtores.
Então se eu precisar instanciar não vou poder, por isso, é melhor continuar usando private, outra qualquer que me permita instânciá-la.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [HELOISA MARTINS CAMBOIM DE SA](#) - quinta, 10 set 2020, 09:07

Oi, Alexssandro!
Gostaria de dar uma dica: não há necessidade de importar o "java.lang"
porque ele é o único pacote que já é automaticamente importado.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [ALEXSSANDRO DE SOUSA FEITOSA](#) - sexta, 11 set 2020, 12:51

Ok. vivendo e aprendendo. Obrigado!

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [LAISSON RODRIGUES DE BESSA](#) - domingo, 6 set 2020, 15:20

Aula 6 – Atividade 1

```

public class Programa {

    public static void main(String[] args) {

        Pessoa p = Visao.lerPessoa();
        Visao.imprimir(p);
    }
}

*=====

```



```

public class Pessoa {

    private String nome;
    private int cod;
    private double salario;

    public String getNome() {
        return nome;
    }

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

    public int getCod() {
        return cod;
    }

    public void setCod(int cod) {
        this.cod = cod;
    }

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salario) {
        this.salario = salario;
    }

    public String getClasseSocial() {
        String classeSocial = "A1";
        if(getSalario()<=400) classeSocial = "Não atribuída Classe Social";
        else if(getSalario()<=950) classeSocial = "F";
        else if(getSalario()<=1400) classeSocial = "E";
        else if(getSalario()<=2300) classeSocial = "D";
        else if(getSalario()<=4600) classeSocial = "C";
        else if(getSalario()<=8100) classeSocial = "B";
        else if(getSalario()<=14400) classeSocial = "A2";
    }
}

```



```

return classeSocial;
}

@Override //sobrescrita do método toString()

public String toString() {

    String texto = "\nNome:      "+getNome()+"\nCódigo:
"+getCod()+"\nSalário:     "+getSalario()

        +"\nClasse Social: "+getClasseSocial();

    return texto;
}

}

*=====
public class Visao {

public static Pessoa lerPessoa() {

    Pessoa p = new Pessoa();

    p.setNome(ScannerUCB.lerString("Informe o nome da pessoa: "));

    p.setCod(ScannerUCB.lerInt("Informe o código da pessoa"));

    p.setSalario(ScannerUCB.lerDouble("Informe o salario da pessoa: ",
"OPS!, Salário inválido", 0, Double.MAX_VALUE));

    return p;
}

public static void imprimir(Pessoa p) {

    System.out.println(p.toString());
}

}

*=====

import java.util.Scanner;

@SuppressWarnings("all")

public class ScannerUCB {

// String

public static String lerString(String msg) {

    System.out.println(msg);

    return lerString();
}

public static String lerString() {

    return new Scanner(System.in).nextLine();
}
}

```



```
}

// char

public static char lerCaractere(String msg) {

    System.out.println(msg);

    return lerCaractere();
}

public static char lerCaractere() {

    return new Scanner(System.in).next().charAt(0);
}

// int

public static int lerInt(String msg, String msgErro, int min, int max) {

    int valor;

    boolean errado;

    do {

        valor = lerInt(msg);

        errado = valor < min || valor > max;

        if (errado) {

            System.out.println(msgErro);

        }

    } while (errado);

    return valor;
}

public static int lerInt(String msg) {

    System.out.println(msg);

    return lerInt();
}

public static int lerInt() {

    return new Scanner(System.in).nextInt();
}

// double

public static double lerDouble(String msg, String msgErro, double min,
double max) {

    double valor;

    boolean errado;
```



```

do {

    valor = lerDouble(msg);

    errado = valor < min || valor > max;

    if (errado) {

        System.out.println(msgErro);

    }

} while (errado);

return valor;
}

public static double lerDouble(String msg) {

    System.out.println(msg);

    return lerDouble();
}

public static double lerDouble() {
    return new Scanner(System.in).nextDouble();
}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



 **Re: Atividade Avaliativa 3**
por [BRUNA SOUSA CARVALHO ALVES](#) - domingo, 6 set 2020, 16:02

Boa tarde!

Vou postar meu código da aula 5, atividade 1:

"Crie um programa Java que implemente o projeto da figura acima. Crie uma classe chamada Funcionario com os atributos (nome, cpf e salário). Crie uma classe Professor, que é um Funcionário e, além dos dados de funcionário, tem a titulação e o cálculo do seu imposto de renda por intermédio do método getImpostoRenda() que é 23 % do salário. Crie também a classe Administrativo, que também é Funcionário e tem um dado que representa a quantidade de dependentes. Não se esqueça de criar o método getContribuicaoSindical(), que retornará 1% do salário.

Seu programa deve perguntar qual tipo de Funcionário deseja informar P - Professor e A - Administrativo. Quando o usuário escolher a opção receba os dados do respectivo objeto e apresente todas as suas informações."

```

import java.util.Scanner;

@SuppressWarnings("all")

```

```
public class Aula4Exercicio1 {  
    public static void main(String args[]) {  
  
        String tipoFuncionario;  
        System.out.  
        .print("Digite o tipo de funcionário: \n Para professor, digite P \n Para  
        administrativo digite A \n ");  
        tipoFuncionario = new Scanner(System.in).nextLine();  
  
        if (tipoFuncionario.equals("P")) {  
            Professor prof = new Professor();  
  
            System.out.print("Digite o nome do professor \n ");  
            String nome = new Scanner(System.in).nextLine();  
            prof.setNome(nome);  
  
            System.out.print("Digite CPF do professor \n ");  
            String Cpf = new Scanner(System.in).nextLine();  
            prof.setCpf(Cpf);  
  
            System.out.print("Digite o salário do professor \n ");  
            double salario = new Scanner(System.in).nextDouble();  
            prof.setSalario(salario);  
  
            System.out.print("Digite a titulação do professor \n ");  
            String titulacao = new Scanner(System.in).nextLine();  
            prof.setTitulacao(titulacao);  
  
            System.out.println("O professor (a)" + prof.getNome() + ", CPF " +  
                prof.getCpf() + ", Titulação: "  
                + prof.getTitulacao() + ", irá pagar R$ " + prof.getImpostoRenda() + " de  
                imposto de renda");  
        } else if (tipoFuncionario.equals("A")) {  
            Administrativo adm = new Administrativo();  
  
            System.out.print("Digite o nome do funcionário \n ");  
            String nome = new Scanner(System.in).nextLine();  
            adm.setNome(nome);  
  
            System.out.print("Digite CPF do funcionário \n ");  
            String Cpf = new Scanner(System.in).nextLine();  
            adm.setCpf(Cpf);  
  
            System.out.print("Digite o salário do funcionário \n ");  
            double salario = new Scanner(System.in).nextDouble();  
            adm.setSalario(salario);  
  
            System.out.print("Digite quantos dependentes o funcionário tem \n ");  
            int dependentes = new Scanner(System.in).nextInt();  
            adm.setDependentes(dependentes);  
  
            System.out.println("O funcionário " + adm.getNome() + ", CPF " +  
                adm.getCpf() + ", com "  
                + adm.getDependentes() + " dependentes irá pagar R$ " +  
                adm.getContribuicaoSindical()  
                + " de contribuição sindical");  
        } else {  
            System.out.print("Dados inválidos \n ");  
        }  
    }  
}  
  
class Funcionario {  
    private String nome;  
    private String cpf;  
    private double salario;
```



```

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

public String getNome() {
    return nome;
}

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

public String getCpf() {
    return cpf;
}

public void setSalario(double salario) {
    this.salario = salario;
}

public double getSalario() {
    return salario;
}

class Professor extends Funcionario {
    private String titulacao;

    public void setTitulacao(String titulacao) {
        this.titulacao = titulacao;
    }

    public String getTitulacao() {
        return titulacao;
    }

    public double getImpostoRenda() {
        return getSalario() * 0.23;
    }
}

class Administrativo extends Funcionario {
    private int dependentes;

    public void setDependentes(int dependentes) {
        this.dependentes = dependentes;
    }

    public int getDependentes() {
        return dependentes;
    }

    public double getContribuicaoSindical() {
        return getSalario() * 0.01;
    }
}

```



Os prints das telas simulando as três execuções possíveis:

1) Professor

```

@ Javadoc Declaration Console ✎
<terminated> Aula4Exercicio1 [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (6
Digite o tipo de funcionário:
Para professor, digite P
Para administrativo digite A
P
Digite o nome do professor
Ana
Digite CPF do professor
123456789
Digite o salário do professor
12425
Digite a titulação do professor
Mestre
O professor (a)Ana, CPF 123456789, Titulação: Mestre, irá pagar R$ 2857.75

```

2) Administrativo

```

@ Javadoc Declaration Console ✎
<terminated> Aula4Exercicio1 [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (1
Digite o tipo de funcionário:
Para professor, digite P
Para administrativo digite A
A
Digite o nome do funcionário
Maria
Digite CPF do funcionário
789456123
Digite o salário do funcionário
4255
Digite quantos dependentes o funcionário tem
3
O funcionário Maria, CPF 789456123, com 3 dependentes irá pagar R$ 42.55000

```

3) Qualquer outra entrada

```

@ Javadoc Declaration Console ✎
<terminated> Aula4Exercicio1 [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (6
Digite o tipo de funcionário:
Para professor, digite P
Para administrativo digite A
G
Dado inválido
|
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [GUILHERME RODRIGUES ROSA](#) - segunda, 7 set 2020, 21:10

Boa noite.
Gostei da validação dos dados com equals. Parabéns pelos resultados.
Fiz com switch/case:

```

char q = lo.lerChar("P - professor, A - administrador");
switch (q) {
case 'p': {
Professor p = new Professor();
p.setNome(lo.lerString("Digite o nome do professor: "));
p.setCpf(lo.lerString("Digite o CPF do(a) "+p.getNome()));
p.setTitulacao(lo.lerString("Digite a Titulação: "));
p.setSalario(lo.lerDouble("Digite o valor do salário: "));
System.out.println("nome: "+p.getNome());
System.out.println("CPF: "+p.getCpf());
}
}

```

```
System.out.println("Titulação: "+p.getTitulacao());
System.out.println("Salário: "+p.getSalario());
System.out.println("Valor do Imposto de Renda: "+p.getImpostoDeRenda());
break;
}
case 'a': {
Administrativo p = new Administrativo();
p.setNome(lo.lerString("Digite o nome do administrador: "));
p.setCpf(lo.lerString("Digite o Cpf do(a) "+p.getNome()));
p.setDependentes(lo.lerInt("Digite o numero de dependentes: "));
p.setSalario(lo.lerDouble("Digite o valor do salário: "));
System.out.println("nome: "+p.getNome());
System.out.println("CPF: "+p.getCpf());
System.out.println("numero de dependentes: "+p.getDependentes());
System.out.println("Salário: "+p.getSalario());
System.out.println("Valor da contribuição Sindical:
"+p.getContribuicaoSindical());
break;
}
default:
System.out.println("Valor Inválido");
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

Re: Atividade Avaliativa 3
por GUSTAVO COSTA RIBEIRO SOARES - domingo, 6 set 2020, 18:22

Aula 6

Atividade 1

```
import java.util.Scanner;  
public class Classes {
```

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    String nome;  
    double salario;  
    System.out.println("Nome do usuário:");  
    nome = sc.nextLine();  
    System.out.println("Salário:");  
    salario = sc.nextDouble();  
    Pessoa pessoa1 = new Pessoa();  
    pessoa1.setNome(nome);  
    pessoa1.setSalario(salario);  
    pessoa1.makeClass(salario);  
    System.out.println(pessoa1.toString());  
}
```

```
public class Pessoa {
```

```
private String nome;  
private double salario;  
private String cod;
```

```
public Pessoa() {  
}
```

```
public void makeClass(double salario) {  
    if (salario > 400 && salario < 950) {  
        this.cod = "F";  
    } else if (salario > 950 && salario < 1400) {  
        this.cod = "E";  
    } else if (salario > 1400 && salario < 2300) {  
        this.cod = "D";  
    } else if (salario > 2300 && salario < 4600) {  
        this.cod = "C";  
    } else {  
        this.cod = "A";  
    }  
}
```



```

} else if (salario > 4600 && salario < 8100) {
this.cod = "B";
} else if (salario > 8100 && salario < 14400) {
this.cod = "A2";
} else if (salario > 14400) {
this.cod = "A1";
} else {
System.out.println("Classe não encontrada");
}
}

public String getNome() {
return nome;
}
public void setNome(String nome) {
this.nome = nome;
}
public double getSalario() {
return salario;
}
public void setSalario(double salario) {
this.salario = salario;
}
public String getCod() {
return cod;
}
public void setCod(String cod) {
this.cod = cod;
}
@Override
public String toString() {
return "Pessoa [Nome =" + nome + ", Salário = " + salario + ", Código=" + cod
+ "]";
}
}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [PATRICIA ALMEIDA DE SOUZA](#) - domingo, 6 set 2020, 22:19

Boa noite, professor Rosana e colegas.

Segue minha contribuição. Resolução do Exercício 1 da Aula 6:

```

/*
 * Síntese
 * Objetivo: Criar uma classe pessoa e definir a classe social
conforme os parametros informados
 * Entrada: Nome, codigo, salario
 * Saída: Nome, codigo, salario e classe social através do método
toString
*/
package Aula6;

public class Atividade1 {

public static void main (String args[]) {

Pessoa p = new Pessoa();
p.setNome(Leitor.lerString("Informe nome: "));
p.setCodigo(Leitor.lerInt("Informe código: "));
p.setSalario(Leitor.lerDouble("Informe salário: "));
System.out.println(p.toString());
}

}

package Aula6;

```

```
public class Pessoa {  
  
    private String nome;  
    private int codigo;  
    private double salario;  
  
    public Pessoa() {  
    }  
  
    public Pessoa(String nome, int codigo, double salario) {  
        this.nome = nome;  
        this.codigo = codigo;  
        this.salario = salario;  
    }  
  
    public String getNome() {  
        return nome;  
    }  
  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
  
    public int getCodigo() {  
        return codigo;  
    }  
  
    public void setCodigo(int codigo) {  
        this.codigo = codigo;  
    }  
  
    public double getSalario() {  
        return salario;  
    }  
  
    public void setSalario(double salario) {  
        this.salario = salario;  
    }  
  
    public String getClasseSocial() {  
  
        String classeSocial = "A";  
  
        if(getSalario()>14400) classeSocial = "A1";  
        else if(getSalario()<=14400 && getSalario()>8100) classeSocial =  
        "A2";  
        else if(getSalario()<=8100 && getSalario()>4600)classeSocial =  
        "B";  
        else if(getSalario()<=4600 && getSalario()>2300)classeSocial =  
        "C";  
        else if(getSalario()<=2300 && getSalario()>1400)classeSocial =  
        "D";  
        else if(getSalario()<=1400 && getSalario()>950)classeSocial = "E";  
        else if(getSalario()<=950 && getSalario()>400)classeSocial = "F";  
        else if(getSalario()<=400) classeSocial = "Classe não definida";  
        return classeSocial;  
    }  
  
    //sobreescrita do Método toString()  
    public String toString() {  
        String retorno = "\nNome: "+getNome()+"\nCódigo:  
        "+getCodigo()+"\nSalário: "+getSalario()+"\nClasse Social:  
        "+getClasseSocial();  
        return retorno;  
    }  
}
```



```

package Aula6;

import java.util.Scanner;

@SuppressWarnings("all")

public class Leitor{

    public static String lerString(String msg) {
        System.out.println(msg);
        return lerString();
    }

    public static String lerString() {
        return new Scanner(System.in).nextLine();
    }

    public static int lerInt(String msg){
        System.out.println(msg);
        return lerInt();
    }

    public static int lerInt(){
        return new Scanner(System.in).nextInt();
    }

    public static char lerCaractere(String msg){
        System.out.println(msg);
        return lerCaractere();
    }

    public static char lerCaractere(){
        return new Scanner(System.in).next().charAt(0);
    }

    public static double lerDouble(String msg){
        System.out.println(msg);
        return lerDouble();
    }

    public static double lerDouble(){
        return new Scanner(System.in).nextDouble();
    }
}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [DIEGO CEZAR GOMES MAGALHAES](#) - segunda, 7 set 2020, 09:11

```
/*
Aula 05 - Atividade 02
```

Um matemático trabalha com dois tipos de figuras: os triângulos retângulos e os retângulos.
Crie um programa Java (classes) que represente a modelagem para abstrair este problema.
Ambos têm dois dados comuns: Base e Altura. Cada uma tem um cálculo de forma distinta de área.
O triângulo retângulo é base X altura / 2, já do retângulo é apenas base X altura.
Um cálculo diferente somente para o triângulo retângulo deve ser realizado, que é a hipotenusa.
Crie um programa de teste, que instancie objetos, atribua valores e execute para apresentação.

```
*/
```

```

public class Programa {
    public static void main (String[] args) {

        char opcao;

        do{

            TrianguloRetangulo t = Visao.lerTrianguloRetangulo();
            Retangulo r = Visao.lerRetangulo();

            Visao.imprimir(t);
            Visao.imprimir(r);

            opcao = Leitor.lerChar("Deseja continuar? S - Sim e N - Não");

            } while (opcao == 's' || opcao == 'S');

        }
    }
}

```

```

public class Visao {

    public static TrianguloRetangulo lerTrianguloRetangulo() {
        double base = Leitor.lerDouble("Informe a medida da base do Triângulo
Retângulo: ");
        double altura = Leitor.lerDouble("Informe a medida da altura do
Triângulo Retângulo: ");
        TrianguloRetangulo t = new TrianguloRetangulo();
        t.setBase(base);
        t.setAltura(altura);
        return t;
    }

    public static Retangulo lerRetangulo() {
        double base = Leitor.lerDouble("Informe a medida da base do Retângulo:
");
        double altura = Leitor.lerDouble("Informe a medida da altura do
Retângulo: ");
        Retangulo r = new Retangulo();
        r.setBase(base);
        r.setAltura(altura);
        return r;
    }

    public static void imprimir(TrianguloRetangulo t) {
        System.out.println("*****");
        System.out.println("Medidas do Triângulo Retângulo:");
        System.out.println("Base: "+t.getBase());
        System.out.println("Altura: "+t.getAltura());
        System.out.println("Área: "+t.getArea());
        System.out.println("Hipotenusa: "+t.getHipotenusa());
        System.out.println("*****");
    }

    public static void imprimir(Retangulo r) {
        System.out.println("*****");
        System.out.println("Medidas do Retângulo:");
        System.out.println("Base: "+r.getBase());
        System.out.println("Altura: "+r.getAltura());
        System.out.println("Área: "+r.getArea());
        System.out.println("*****");
    }
}

```



```
}

import java.util.Scanner;

public class Leitor {

    // Double

    public static double lerDouble(String mensagem) {
        System.out.println(mensagem);
        return lerDouble();
    }

    public static double lerDouble() {
        return new Scanner(System.in).nextDouble();
    }

    // Char

    public static char lerChar(String mensagem) {
        System.out.println(mensagem);
        return lerChar();
    }

    public static char lerChar() {
        return new Scanner(System.in).next().charAt(0);
    }

}
```



```
class Figura {

    private double base;
    private double altura;

    public double getArea() {
        return 0.0;
    }

    // setters e getters

    public void setBase (double base) {
        this.base = base;
    }
    public double getBase() {
        return base;
    }
    public void setAltura (double altura) {
        this.altura = altura;
    }
    public double getAltura() {
        return altura;
    }

}
```

```
class TrianguloRetangulo extends Figura {

    public double getArea() {
        return (getBase() * getAltura() / 2.0);
    }
```

```

        public double getHipotenusa() {
            return Math.sqrt(getBase() * getBase() + getAltura() * getAltura());
        }
    }

    class Retangulo extends Figura {

        public double getArea() {
            return (getBase() * getAltura());
        }
    }

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [MIRIÁ COSTA BARROS DE AGUIAR](#) - segunda, 7 set 2020, 12:47

Aula 05 - Exercício 02:

Um matemático trabalha com dois tipos de figuras: os triângulos retângulos e os retângulos. Crie um programa Java (classes) que represente a modelagem para abstrair este problema. Ambos têm dois dados comuns: Base e Altura. Cada uma tem um cálculo de forma distinta de área. O triângulo retângulo é base X altura / 2, já do retângulo é apenas base X altura. Um cálculo diferente somente para o triângulo retângulo deve ser realizado, que é a hipotenusa. Crie um programa de teste, que instancie objetos, atribua valores e execute para apresentação.

Resolução:

```

package entidades;

public abstract class Figura { //Numa classe abstrata não é necessário
    implementar um método, apenas descrevê-lo.

    protected double base;
    protected double altura;

    public abstract double area();
}

package entidades;

public class Retangulo extends Figura{ //extends se refere a ligação de herança
    com a classe figura

    @Override //polimorfismo de método
    public double area() {
        return base*altura;
    }

    public double getBase() {
        return base;
    }
    public void setBase(double base) {
        this.base = base;
    }
    public double getAltura() {
        return altura;
    }
    public void setAltura(double altura) {
        this.altura = altura;
    }
}

package entidades;

```



```

public class TrianguloRetangulo extends Figura{

    @Override
    public double area() {
        return (base*altura)/2;
    }

    public double hipotenusa() {
        return Math.sqrt(base*base + altura*altura);
    }

    public double getBase() {
        return base;
    }
    public void setBase(double base) {
        this.base = base;
    }
    public double getAltura() {
        return altura;
    }
    public void setAltura(double altura) {
        this.altura = altura;
    }
}

package application;
import entidades.Retangulo;
import entidades.TrianguloRetangulo;

public class Program {

    public static void main(String[] args) {//método main executará o ponto de
    entrada da aplicação

        //Calculo da área do retangulo
        Retangulo r1 = new Retangulo();
        r1.setBase(10.0);
        r1.setAltura(5.0);
        System.out.println(r1.area());

        //Calculo da área do triangulo retangulo
        TrianguloRetangulo tr1 = new TrianguloRetangulo();
        tr1.setBase(15.0);
        tr1.setAltura(20.0);
        System.out.println(tr1.area());

        //Calculo da hipotenusa
        System.out.println(tr1.hipotenusa());
    }
}

```



Saída Console:

50.0
150.0
25.0

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [GRAZIELE ADEISA MONTEIRO DE CASTRO](#) - segunda, 7 set 2020, 15:11

Aula 5 atividade 2

```

/*Um matemático trabalha com dois tipos de figuras: os triângulos retângulos e
os retângulos.
// Crie um programa Java (classes) que represente a modelagem para abstrair
este problema.
// Ambos têm dois dados comuns: Base e Altura.
// Cada uma tem um cálculo de forma distinta de área.
// O triângulo retângulo é base X altura / 2, já do retângulo é apenas base X
altura.
// Um cálculo diferente somente para o triângulo retângulo deve ser realizado,
que é a hipotenusa.
Crie um programa de teste, que instancie objetos, atribua valores e execute para
apresentação.*/

```

```

package a5a2;

import java.util.Scanner;

public class A5a2 {
    public static void main(String[] args) {
        // Triangulo Retângulo
        trianguloRetangulo a = new trianguloRetangulo();

        System.out.print("Informe o valor da base do triângulo retângulo: ");
        a.setBase(new Scanner(System.in).nextDouble());
        System.out.print("Informe o valor da altura do triângulo retângulo: ");
        a.setAltura(new Scanner(System.in).nextDouble());

        System.out.println("O calculo da área do triângulo retângulo é : " +
a.getCalcArea() + " metros2");
        System.out.println("O valor da hipotenusa é: " + a.getHipotenus());
        System.out.println(" ");
        // Retângulo
        Retangulo b = new Retangulo();

        System.out.print("Informe o valor da base do retângulo: ");
        b.setBase(new Scanner(System.in).nextDouble());
        System.out.print("Informe o valor da altura do retângulo: ");
        b.setAltura(new Scanner(System.in).nextDouble());

        System.out.println("O calculo da área to retângulo é : " + b.getCalcArea()+
metros2);
    }
}
=====CLASSE FIGURA

```

```

package a5a2;
public class Figura {
    double base;
    double altura;

    public double getBase() {
        return base;
    }

    public void setBase(double base) {
        this.base = base;
    }

    public double getAltura() {
        return altura;
    }

    public void setAltura(double altura) {
        this.altura = altura;
    }

    public double getCalcArea() {
        return 0;
    }
}

```



```
}

=====CLASSE TRIÂNGULO RETÂNGULO

package a5a2;
public class trianguloRetangulo extends Figura {

@Override
public double getCalcArea() {
return (this.getBase() * this.getAltura() / 2.0);
}
public double getHipotenusa() {
return (getBase() * getBase() + getAltura() * getAltura());
}
}
```

=====CLASSE RETÂNGULO

```
package a5a2;  
public class Retangulo extends Figura {  
    @Override  
    public double getCalcArea() {  
        return (this.getBase() * this.getAltura());  
    }  
}
```

The screenshot shows the BlueJ IDE interface with several tabs open at the top: Página Inicial, A5a2.java, Figura.java, triânguloRetângulo.java, and Retângulo.java. The A5a2.java tab is active, displaying Java code for calculating the area of a triangle and a rectangle based on user input. The Saida - A5a2 (run) tab at the bottom shows the console output for the run.

```
13 public static void main(String[] args) {
14     // Triângulo Retângulo
15     trianguloRetângulo a = new trianguloRetângulo();
16
17     System.out.print("Informe o valor da base do triângulo retângulo: ");
18     a.setBase(new Scanner(System.in).nextDouble());
19     System.out.print("Informe o valor da altura do triângulo retângulo: ");
20     a.setAltura(new Scanner(System.in).nextDouble());
21
22     System.out.println("O cálculo da área do triângulo retângulo é : " + a.getCalcArea() + " metros");
23     System.out.println("O valor da hipotenusa é : " + a.getHipotenusa());
24     System.out.println(" ");
25     // Retângulo
26     Retângulo b = new Retângulo();
27
28     System.out.print("Informe o valor da base do retângulo: ");
29     b.setBase(new Scanner(System.in).nextDouble());
30     System.out.print("Informe o valor da altura do retângulo: ");
31     b.setAltura(new Scanner(System.in).nextDouble());
32
33     System.out.println("O cálculo da área do retângulo é : " + b.getCalcArea() + " metros");
34
35 }
36
37
38 }
```

Saida - A5a2 (run) :

```
RUN:
1 Informe o valor da base do triângulo retângulo: 50
1 Informe o valor da altura do triângulo retângulo: 100
1 O cálculo da área do triângulo retângulo é : 2500.0 metros
1 O valor da hipotenusa é: 12500.0

2 Informe o valor da base do retângulo 50
2 Informe o valor da altura do retângulo 100
2 O cálculo da área do retângulo é : 5000.0 metros
CONSTRUIDO COM SUCESSO (tempo total: 1 minuto 23 segundos)
```

[Link direto](#) | [Marcar como não lida](#) | [Mostrar principal](#)



Re: Atividade Avaliativa 3

Re: Atividade Avaliativa 3
por PEDRO HENRIQUE RODRIGUES DE OLIVEIRA - segunda, 7 set 2020, 18:45

Aula 6 Atividade 1

Crie uma classe pessoa com nome, código inteiro e salário. A classe social de uma pessoa é calculada conforme o seguinte:

Classe

A1: inclui as famílias com renda mensal maior que R\$ 14.400

Classe A2: maior que R\$ 8.100

Classe B: maior que R\$ 4.600

Classe C: maior que R\$ 2.300

Classe D: maior que R\$ 1.400

Classe E: maior que R\$ 950

Classe F: maior que R\$ 400

Receba do usuário os dados de uma pessoa e apresente todas as informações da pessoa, inclusive a classe por meio do método `toString`.

Classe Pessoa

```
package model;

import java.text.DecimalFormat;

public class Pessoa {

    private String nome;
    private String codigo;
    private double salario;
    private String classeSocial;

    public String getNome() {
        return nome;
    }

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

    public String getCodigo() {
        return codigo;
    }

    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salariao) {
        this.salario = salariao;
    }

    public static String formatarSalario(double salario) {
        DecimalFormat df = new DecimalFormat("R$ 0,000.00");
        return df.format(salario);
    }

    public String getClasseSocial() {

        if (getSalario() > 400 && getSalario() <= 950 ) {
            classeSocial = "F";
        }else if(getSalario() > 950 && getSalario() <= 1400) {
            classeSocial = "E";
        }else if(getSalario() > 1400 && getSalario() <= 2300) {
            classeSocial = "D";
        }else if(getSalario() > 2300 && getSalario() <= 4600) {
            classeSocial = "C";
        }else if(getSalario() > 4600 && getSalario() <= 8100) {
            classeSocial = "B";
        }else if(getSalario() > 8100 && getSalario() <= 14400) {
            classeSocial = "A2";
        }else if(getSalario() > 14400) {
            classeSocial = "A1";
        }
    }
}
```



```

        return classeSocial;
    }

    public String toString() {
        String output = "Nome: " + getNome() + "\n"
        + "Código: " + getCodigo() + "\n"
        + "Salário: " + formatarSalario(getSalario()) + "\n"
        + "Classe social: " + getClasseSocial();

        return output;
    }
}

```

Principal

```

package exercicios;

import model.Pessoa;

/*
 * Objetivo: Criar uma classe Pessoa com nome, código inteiro e salário
 * Entrada: Informações da pessoa
 * Saída: Dados do objeto pessoa fazendo o uso do toString.
 */

public class ForumDois {

    public static void main(String[] args) {

        Pessoa pessoa1 = new Pessoa();
        pessoa1.setNome("Pedro");
        pessoa1.setCodigo("PS001");
        pessoa1.setSalario(50067);

        System.out.println(pessoa1.toString());
    }
}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ÉRICO TOSCANO DE OLIVEIRA](#) - terça, 8 set 2020, 23:27

Excelente o seu trabalho, Pedro! Como sugestão, acrescentaria um texto de instruções para o usuário inserir os dados solicitados.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [NATANAEL BATISTA GREGORIO DOS SANTOS](#) - quarta, 9 set 2020, 13:19

Olá Pedro , seu código funciona porém o exercício pede que o usuário insira os dados e você apenas os passou. Aproveito para deixar uma dica. Veja sobre JOptionPane, é uma interface gráfica de fácil implementação. Bons estudos!

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [PEDRO HENRIQUE RODRIGUES DE OLIVEIRA](#) - quarta, 9 set 2020, 19:19

Boa observação, Natanael! Acabei por deixar passar por despercebido. JOptionPane é uma boa ideia de fato tanto que fiz uso no [trabalho final](#)

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [GUILHERME RODRIGUES ROSA](#) - segunda, 7 set 2020, 20:43

Aula 05 Atividade 02

Um matemático trabalha com dois tipos de figuras: os triângulos retângulos e os retângulos. Crie um programa Java (classes) que represente a modelagem para abstrair este problema. Ambos têm dois dados comuns: Base e Altura. Cada uma tem um cálculo de forma distinta de área. O triângulo retângulo é base X altura / 2, já do retângulo é apenas base X altura. Um cálculo diferente somente para o triângulo retângulo deve ser realizado, que é a hipotenusa. Crie um programa de teste, que instancie objetos, atribua valores e execute para apresentação.

=====

Main

```
public class Main {  
    public static void main(String[] args) {  
  
        trianguloRetangulo tr = new trianguloRetangulo();  
        tr.setAltura(lo.lerDouble("altura: "));  
        tr.setBase(lo.lerDouble("Base: "));  
        System.out.println("Area: "+tr.getArea());  
        System.out.println("hipotenusa: "+tr.getHipotenus());  
  
        Retangulo r = new Retangulo();  
        r.setAltura(lo.lerDouble("altura: "));  
        r.setBase(lo.lerDouble("Base: "));  
        System.out.println("Area: "+r.getArea());  
    }  
}
```

=====

Classe pai Atributos

```
public class Atributos {  
    private double altura;  
    private double base;  
  
    public double getArea() {  
        return 0;  
    }  
    public double getAltura() {  
        return altura;  
    }  
    public void setAltura(double altura) {  
        this.altura = altura;  
    }  
    public double getBase() {  
        return base;  
    }  
    public void setBase(double base) {  
        this.base = base;  
    }  
}
```

=====

Class retangulo

```
public class Retangulo extends Atributos{  
  
    public double getArea() {  
        return getBase() * getAltura();  
    }  
}
```

=====



```
class trianguloRetangulo

public class trianguloRetangulo extends Atributos{
public double getArea() {
return (getBase() * getAltura()) / 2;
}
public double getHipotenusa() {
double hipotenusa = (Math.pow(getBase(), 2)) + (Math.pow(getAltura(), 2));
return Math.sqrt(hipotenusa);
}
}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ÉRICO TOSCANO DE OLIVEIRA](#) - terça, 8 set 2020, 20:42

Boa noite a todos!

Selecionei a Atividade 1 da Aula 5, Unidade 3.

Código da classe Funcionario:

```
public class Funcionario {

private String nome;
private String cpf;
private double salario;

public String getNome() {
return nome;
}
public void setNome(String nome) {
this.nome = nome;
}
public String getCpf() {
return cpf;
}
public void setCpf(String cpf) {
this.cpf = cpf;
}
public double getSalario() {
return salario;
}
public void setSalario(double salario) {
this.salario = salario;
}

}
```

Código da classe Professor:

```
public class Professor extends Funcionario {

private String titulacao;

public double getImpostoRenda() {
return getSalario()*0.23;
}

public String getTitulacao() {
return titulacao;
}

public void setTitulacao(String titulacao) {
this.titulacao = titulacao;
}
```



```
}
```

Código da classe Administrativo:

```
public class Administrativo extends Funcionario {  
  
    private int dependentes;  
  
    public int getDependentes() {  
        return dependentes;  
    }  
  
    public void setDependentes(int dependentes) {  
        this.dependentes = dependentes;  
    }  
  
    public double getContribuicaoSindical() {  
        return getSalario()*0.01;  
    }  
}
```

Código da classe Programa:

```
import java.util.Scanner;  
  
public class Programa {  
    public static void main(String[] args) {  
  
        char tipo;  
  
        Professor prof = new Professor();  
        Administrativo adm = new Administrativo();  
  
        System.out.print("Informe a classe de funcionário (P - Professor ou A - Administrativo ");  
        tipo = new Scanner(System.in).next().charAt(0);  
  
        if(tipo == 'P') {  
  
            System.out.print("Informe o nome do professor: ");  
            prof.setNome(new Scanner(System.in).nextLine());  
  
            System.out.print("Informe o cpf do professor: ");  
            prof.setCpf(new Scanner(System.in).nextLine());  
  
            System.out.print("Informe o salário do professor, em reais: ");  
            prof.setSalario(new Scanner(System.in).nextDouble());  
  
            System.out.printf("Informe a titulação do professor: ");  
            prof.setTitulacao(new Scanner(System.in).nextLine());  
  
            System.out.println("Dados informados: ");  
            System.out.println("Nome do professor: "+prof.getNome());  
            System.out.println("Cpf do professor: "+prof.getCpf());  
            System.out.println("Salário do professor, em reais: "+prof.getSalario());  
            System.out.println("Titulação do professor: "+prof.getTitulacao());  
            System.out.println("Imposto de renda pago pelo professor, em reais:  
            "+prof.getImpostoRenda());  
  
        }  
  
        else {  
  
            System.out.print("Informe o nome do funcionário administrativo: ");  
            adm.setNome(new Scanner(System.in).nextLine());  
  
            System.out.print("Informe o cpf do funcionário administrativo: ");  
            adm.setCpf(new Scanner(System.in).nextLine());
```



```
System.out.print("Informe o salário do funcionário administrativo, em reais: ");
adm.setSalario(new Scanner(System.in).nextDouble());

System.out.printf("Informe o número de dependentes do funcionário
administrativo: ");
adm.setDependentes(new Scanner(System.in).nextInt());

System.out.println("Dados informados: ");
System.out.println("Nome do funcionário administrativo: "+adm.getNome());
System.out.println("Cpf do funcionário administrativo: "+adm.getCpf());
System.out.println("Salário do funcionário administrativo, em reais:
"+adm.getSalario());
System.out.println("Número de dependentes do funcionário administrativo:
"+adm.getDependentes());
System.out.println("Contribuição sindical paga pelo funcionário administrativo,
em reais: "+adm.getContribuicaoSindical());

}
```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [DENISE COSTA RAMALHO DE SOUZA](#) - quarta, 9 set 2020, 04:54

Érico, excelente trabalho! Também fiz esse exercício e, por sinal, o desenvolvimento ficou muito parecido com o seu. Deixo como sugestão acrescentar comentários ao seu código, pois facilita muito o entendimento.

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**

por [DENISE COSTA RAMALHO DE SOUZA](#) - quarta, 9 set 2020, 04:49

```
/* Criação da entidade genérica funcionário que tem: nome, CPF e
salário*/
```

```
public class Funcionario {
```

```
    private String nome;
    private String cpf;
    private double salario;
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getCpf() {
        return cpf;
    }
    public void setCpf(String cpf) {
        this.cpf = cpf;
    }
    public double getSalario() {
        return salario;
    }
    public void setSalario(double salario) {
        this.salario = salario;
    }
}
```

```
/* Criação da entidade professor que é um tipo de funcionário e a
titulação é exclusiva dele */
```

```

public class Professor extends Funcionario{ //Professor é um tipo de
funcionário, por isso usa-se a palavra reservada extends
private String titulacao;

public double getImpostoRenda() {
return getSalario() * 0.23; //IR é 23% do salário do professor
}

public String getTitulacao() {
return titulacao;
}

public void setTitulacao(String titulacao) {
this.titulacao = titulacao;
}

}
-----
```

**/* Criação da entidade Administrativo que é um tipo de funcionário e a
característica "dependentes" é exclusiva dele */**

**public class Administrativo extends Funcionario { //Administrativo é um tipo
de funcionário, por isso usa-se a palavra reservada extends**

```

private int dependentes;

public double getContribuicaoSindical () {
return getSalario () * 0.01;

}

public int getDependentes() {
return dependentes;
}

public void setDependentes(int dependentes) {
this.dependentes = dependentes;
}

}
-----
```

**/*Seu programa deve perguntar qual tipo de Funcionário deseja informar
P - Professor e A - Administrativo.
* Quando o usuário escolher a opção receba os dados do respectivo
objeto e apresente todas as suas informações.**

***/**

```

import java.util.Scanner;

public class Main {
public static void main(String[] args {

char opcao;
System.out.println("Que tipo de funcionário deseja informar: P - Professor ou
A - Administrativo");
opcao = new Scanner(System.in).next().charAt (0);
if(opcao == 'P' || opcao == 'p'){
Professor p = new Professor();
System.out.print("Informe o nome do professor: ");
p.setNome(new Scanner(System.in).nextLine());
System.out.print("Informe o CPF do professor: ");
p.setCpf(new Scanner(System.in).nextLine());
System.out.print("Informe o salário do professor: ");
p.setSalario(new Scanner(System.in).nextDouble());
System.out.print("Informe a titulação do professor: ");
p.setTitulacao(new Scanner(System.in).nextLine());
```



```

System.out.println("\n.....");
System.out.println("Nome do professor: "+p.getNome());
System.out.println("Cpf do professor: "+p.getCpf());
System.out.println("Salário do professor: "+p.getSalario());
System.out.println("Titulação do professor: "+p.getTitulacao());
System.out.println("Imposto de renda do professor: "+p.getImpostoRenda());

} else if(opcao == 'A' || opcao == 'a') {
Administrativo adm = new Administrativo ();
System.out.print("Informe o nome do administrativo: ");
adm.setNome(new Scanner(System.in).nextLine());
System.out.print("Informe o CPF do administrativo: ");
adm.setCpf(new Scanner(System.in).nextLine());
System.out.print("Informe o salário do administrativo: ");
adm.setSalario(new Scanner(System.in).nextDouble());
System.out.print("Informe nº de dependentes do administrativo: ");
adm.setDependentes(new Scanner (System.in).nextInt ());

System.out.println("\n.....");
System.out.println("Nome do administrativo: "+adm.getNome());
System.out.println("Cpf do administrativo: "+adm.getCpf());
System.out.println("Salário do administrativo: "+adm.getSalario());
System.out.println("Número de dependentes do administrativo:
"+adm.getDependentes());
System.out.println("Contribuição sindical do administrativo:
"+adm.getContribuicaoSindical());
}
}
}

```



[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [ALEX DA SILVA](#) - quarta, 9 set 2020, 09:55

Atividade 01 - Aula 05

Classe Principal Main:

```

import java.util.Calendar;
import java.util.Scanner;

@SuppressWarnings("all")
public class Main {
public static void main(String[] args) {
char opcao = 'N';
do{
System.out.println("Qual Tipo de Funcionario Deseja informar? P - Professor / A -
Administrativo");
opcao = new Scanner(System.in).next().charAt(0);
if (opcao == 'P' || opcao == 'p'){
Professor();
}
else if (opcao == 'A' || opcao == 'a'){
Administrativo();
}
else
System.out.println("Informar somente? P - Professor / A - Administrativo");
}while(opcao !='S' || opcao != 's' || opcao != 'N' || opcao != 'n' );

}

public static void Professor(){
Professor p = new Professor();
p.setNome("Alex Silva");
}

```

```

p.setCpf("123.456.789-00");
p.setSalario(10000);
p.setTitulacao("MESTRE");
System.out.println("Imposto de Renda: " + p.getImpostoRenda());
System.out.println("Gratificação: " + p.getGratificacao());
}

public static void Administrativo(){
Calendar dtAdmissao = Calendar.getInstance();
dtAdmissao.set(Calendar.DAY_OF_MONTH, 1);
dtAdmissao.set(Calendar.MONTH, Calendar.MARCH);
dtAdmissao.set(Calendar.YEAR, 2016);

Administrativo adm = new Administrativo();
adm.setNome("Alex da Silva");
adm.setDependentes(2);
adm.setCpf('123.456.789-00');
adm.setSalario(15000);
adm.setDtAdmissao(dtAdmissao);
System.out.println("Sindicato: " + adm.getContribuicaoSindical());
System.out.println("Dias Trabalhados: " + adm.getDiasTrabalhados());
System.out.println("Gratificação: " + adm.getGratificacao());
}
}

```

Classe Administrativo

```

import java.util.Calendar;

public class Administrativo extends Funcionario{
private int dependentes;
private Calendar dtAdmissao;

public double getContribuicaoSindical(){
return getSalario() * 0.01;
}
public int getDependentes() {
return dependentes;
}
public void setDependentes(int dependentes) {
this.dependentes = dependentes;
}
public Calendar getDtAdmissao() {
return dtAdmissao;
}
public void setDtAdmissao(Calendar dtAdmissao) {
this.dtAdmissao = dtAdmissao;
}

public int getDiasTrabalhados(){
Calendar dtAtual = Calendar.getInstance();
int MILLIS_IN_DAY = 86400000;
return (int) ((dtAtual.getTimeInMillis() - getDtAdmissao().getTimeInMillis())
/MILLIS_IN_DAY );
}

public double getGratificacao(){
return getDiasTrabalhados() * .30;
}
}

```

Classe Funcionário

```

public class Funcionario {
private String nome;
private String cpf;

```



```

private double salario;

public double getGratificacao(){
    return 0;
}

public String getNome() {
    return nome;
}

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

public String getCpf() {
    return cpf;
}

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

public double getSalario() {
    return salario;
}

public void setSalario(double salario) {
    this.salario = salario;
}
*****
Classe Professor

```



```

public class Professor extends Funcionario {
    private String titulacao;

    public double getImpostoRenda(){
        return getSalario() * 0.23;
    }

    public String getTitulacao() {
        return titulacao;
    }

    public void setTitulacao(String titulacao) {
        this.titulacao = titulacao;
    }

    public double getGratificacao(){
        if (getTitulacao().equalsIgnoreCase("ESPECIALISTA"))
            return getSalario() * 0.01;
        else if (getTitulacao().equalsIgnoreCase("MESTRE"))
            return getSalario() * 0.12;
        else if (getTitulacao().equalsIgnoreCase("DOUTOR"))
            return getSalario() * 0.15;
        else
            return 0;
    }
}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [FRANCISCO FROTA DE AGUIAR](#) - quarta, 9 set 2020, 12:03

Unidade 4 - Aula 7 - Atividade 4

package entidades;

```

import java.util.ArrayList;
import java.util.List;

public class Terrorista {

    static List t = new ArrayList<>();

    protected String nome;
    protected int qtExplosivos;

    public int grauDePericulosidade() {
        return 0;
    }

    // Apresente todas as informações de todos os terroristas informados pelo
    // usuário.
    public static void informacoes() {
        for (Terrorista terrorista : t) {
            if (terrorista instanceof Suicida) {
                System.out.println(terrorista);
            } else {
                System.out.println(terrorista);
            }
        }
    }

    //Mostre todos os terroristas que possuem MOHAMED no nome, em qualquer
    //formato.
    public static void buscaPorNome() {
        for (Terrorista terrorista : t) {
            if (terrorista.nome.equalsIgnoreCase("MOHAMED")) {
                System.out.println(terrorista);
            }
        }
    }

    //A quantidade de terroristas encontrados com mais de 10 explosivos.
    public static int buscaPorExplosivos() {
        int cont = 0;
        for (Terrorista terrorista : t) {
            if(terrorista.qtExplosivos > 10) {
                cont++;
            }
        }

        return cont;
    }

    public String getNome() {
        return nome;
    }

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

    public int getQtExplosivos() {
        return qtExplosivos;
    }

    public void setQtExplosivos(int qtExplosivos) {
        this.qtExplosivos = qtExplosivos;
    }

}

//-----
package entidades;

```



```

public class Suicida extends Terrorista {

    private String RELIGIAO;

    @Override
    public int grauDePericulosidade() {
        int conReligiao;
        if (RELIGIAO.equals("Islamismo")) {
            conReligiao = 5;
        } else {
            conReligiao = 2;
        }

        return qtExplosivos * 2 * conReligiao;
    }

    // Apresente todos os terroristas que possuem grau de periculosidade maior que
    // 50.
    public static void maisPerigosos() {
        for (Terrorista terrorista : t) {
            if(terrorista instanceof Suicida) {
                if (terrorista.grauDePericulosidade() > 50) {
                    System.out.println(terrorista);
                }
            }
        }
    }

    public void adcTerrorista(Suicida obj) {
        t.add(obj);
    }

    public String getRELIGIAO() {
        return RELIGIAO;
    }

    public void setRELIGIAO(String rELIGIAO) {
        RELIGIAO = rELIGIAO;
    }

    @Override
    public String toString() {
        return "Suicida [RELIGIAO=" + RELIGIAO + ", nome=" + nome +",
               qtExplosivos=" + qtExplosivos + "]";
    }
}

//-----
package entidades;

public class NaoSuicida extends Terrorista {

    private String PAIS_DE_ORIGEM;

    @Override
    public int grauDePericulosidade() {
        int conPais;
        if (PAIS_DE_ORIGEM.equals("Arábia")) {
            conPais = 7;
        } else {
            conPais = 5;
        }

        return qtExplosivos * 3 * conPais;
    }

    public void adcTerrorista(NaoSuicida obj) {
        t.add(obj);
    }
}

```



```

}

// Apresente todos os terroristas que possuem grau de periculosidade maior que
50.
public static void maisPerigosos() {
for (Terrorista terrorista : t) {
if(terrorista instanceof NaoSuicida) {
if (terrorista.grauDePericulosidade() > 50) {
System.out.println(terrorista);
}
}
}
}

// O percentual de terroristas que não são do 'Egito'.
public static int percentualNaoEgito() {
int cont = 0;
for (Terrorista terrorista : t) {
if (terrorista instanceof NaoSuicida) {
if (!(((NaoSuicida) terrorista).PAIS_DE_ORIGEM.equals("Egito"))) {
cont++;
}
}
}
return cont / t.size() * 100;
}

public String getPAIS_DE_ORIGEM() {
return PAIS_DE_ORIGEM;
}

public void setPAIS_DE_ORIGEM(String pAIS_DE_ORIGEM) {
PAIS_DE_ORIGEM = pAIS_DE_ORIGEM;
}

@Override
public String toString() {
return "NaoSuicida [PAIS_DE_ORIGEM=" + PAIS_DE_ORIGEM + ", nome=" +
nome + ", qtExplosivos=" + qtExplosivos
+ "]";
}
}

//-----
package programa;

import java.util.Scanner;

import entidades.NaoSuicida;
import entidades.Suicida;
import entidades.Terrorista;

public class aplicacao {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("CADASTRO DE TERRORISTA");

char op = 'y';
int opcao;

while (op == 'y') {
System.out.println("Digite (1) para Suicida e (2) para Não Suicida");
opcao = sc.nextInt();
if (opcao == 1) {
Terrorista t = new Suicida();
System.out.println("Nome:");
}
}
}
}

```



```

t.setNome(sc.next());
System.out.println("Quantidade de Explosivos:");
t.setQtExplosivos(sc.nextInt());
System.out.println("Religiao");
((Suicida) t).setRELIGIAO(sc.next());
((Suicida)t).adcTerrorista((Suicida)t);

} else if (opcao == 2) {
Terrorista t = new NaoSuicida();
System.out.println("Nome:");
t.setNome(sc.next());
System.out.println("Quantidade de Explosivos:");
t.setQtExplosivos(sc.nextInt());
System.out.println("País de Origem");
((NaoSuicida) t).setPAIS_DE_ORIGEM(sc.next());
((NaoSuicida)t).adcTerrorista((NaoSuicida)t);
} else {
System.out.println("Opção Inválida");
}
System.out.println("Deseja continua a cadastrar? (y)sim");
op = sc.next().charAt(0);
}

System.out.println("GERAL");
Terrorista.informacoes();
System.out.println();

System.out.println("MAIS PERIGOSOS:");
Suicida.maisPerigosos();
NaoSuicida.maisPerigosos();
System.out.println();

System.out.println("NOME: MOHAMED");
Terrorista.buscaPorNome();
System.out.println();

System.out.println("TERRORISTAS COM MAIS DE 10 EXPLOSIVOS:");
System.out.println(Terrorista.buscaPorExplosivos());
System.out.println();

System.out.println("PERCENTUAL FORA DO EGITO");
System.out.println(NaoSuicida.percentualNaoEgitto() + "%");

sc.close();
}
}

```

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)



 **Re: Atividade Avaliativa 3**
por [NATANIEL BATISTA GREGORIO DOS SANTOS](#) - quarta, 9 set 2020, 13:11

Aula 6 atividade 1

```

package EAD;

import javax.swing.JOptionPane;

/**
 *
 * @author natanael
 */
public class Programa {
public static void main(String[] args) {

String nome1 = JOptionPane.showInputDialog("Digite o nome da pessoa: ");
String cod = JOptionPane.showInputDialog("Digite o codigo da pessoa: ");
int cod1 = Integer.parseInt(cod);

```

```
String salario = JOptionPane.showInputDialog("Digite o salario da pessoa: ");
double salario1 = Double.parseDouble(salario);
Pessoa p1 = new Pessoa(nome1, cod1, salario1);
```

```
JOptionPane.showMessageDialog(null,p1.toString());
}
}
```

```
package EAD;
```

```
/*
*
* @author natanael
*/
class Pessoa {
private String nome;
private int codigo;
private double salario;
private String classe;
```

```
public String getClasse() {
if (salario >= 14400 ) {
classe = "A1";
}else if (salario >= 8100 && salario < 14400) {
classe ="A2";
}else if (salario >= 4600 && salario < 8100) {
classe = "B";
}else if (salario >= 2300 && salario < 4600) {
classe = "C";
}else if (salario >= 1400 && salario < 2300) {
classe = "D";
}else if (salario >= 950 && salario < 1400) {
classe ="E";
}else{
classe = "F";
}
return classe;
}
```

```
@Override
public String toString(){
String infos = "\nNome: "+getNome()+"\n"
+"nCódigo: "+getCodigo()+"\n"
+"nSalário: R$"+getSalario()+"\n"
+"nClasse Social : "+getClasse()+"\n";
return infos;
```

```
}
```

```
public Pessoa() {}
public Pessoa(String nome, int codigo, double salario) {
setNome(nome);
setCodigo(codigo);
setSalario(salario);
}
```

```
public String getNome() {
return nome;
}
```

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

```
public int getCodigo() {
return codigo;
}
```



```
public void setCodigo(int codigo) {  
    this.codigo = codigo;  
}  
  
public double getSalario() {  
    return salario;  
}  
  
public void setSalario(double salario) {  
    this.salario = salario;  
}  
}
```

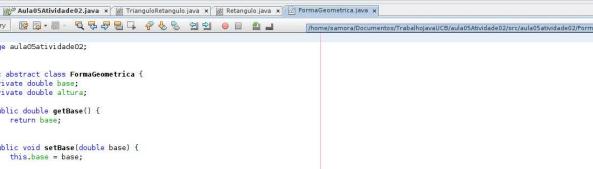
[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

A portrait of a man with dark hair and glasses, wearing a maroon t-shirt. He is looking directly at the camera.

Re: Atividade Avaliativa 3

por [JOÃO VITOR SAMORA LOPES](#) - quarta, 9 set 2020, 18:33

Aula 05 Atividade 02



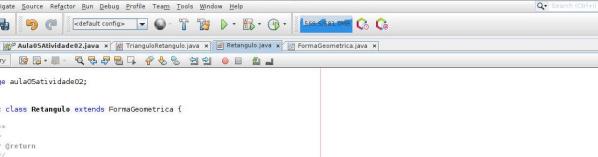
The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** aula05Atividade02 - Apache NetBeans IDE 12.0
- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Search Bar:** search (CTRL+F)
- Project Explorer (Left):** Shows a single project named "aula05Atividade02".
- Code Editor (Center):** Displays the Java code for the `FormaGeometrica` class. The code defines a base class with methods for setting and getting base and height values.
- Output Bar (Bottom):** Shows the path to the file: /home/lemoni/Documentos/trabalhoJava/aula05Atividade02/aula05Atividade02/FormaGeometrica.java

```
1 package aula05Atividade02;
2
3 public abstract class FormaGeometrica {
4     private double base;
5     private double altura;
6
7     public double getBase() {
8         return base;
9     }
10
11    public void setBase(double base) {
12        this.base = base;
13    }
14
15    public double getAltura() {
16        return altura;
17    }
18
19    public void setAltura(double altura) {
20        this.altura = altura;
21    }
22
23 }
24
25
26
27 }
```

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** aula05Atividade02 - Apache NetBeans IDE 12.0
- Toolbar:** Standard NetBeans toolbar with icons for file operations, search, and help.
- Code Editor:** The main window displays Java code for a class named `TrianguloRetangulo`. The code includes imports for `java.lang.Math.sqrt`, a package declaration for `aula05Atividade02`, and two methods: `getCalculoTriangle()` and `getCalculoHipot()`.
- Project Explorer:** Shows a project structure with files like `Aula05Atividade02.java`, `TrianguloRetangulo.java`, `Retangulo.java`, and `FormaGeometrica.java`.
- Task List:** A sidebar on the left lists tasks such as `Start Page`, `aula05Atividade02.java`, `TrianguloRetangulo.java`, `Retangulo.java`, and `FormaGeometrica.java`.

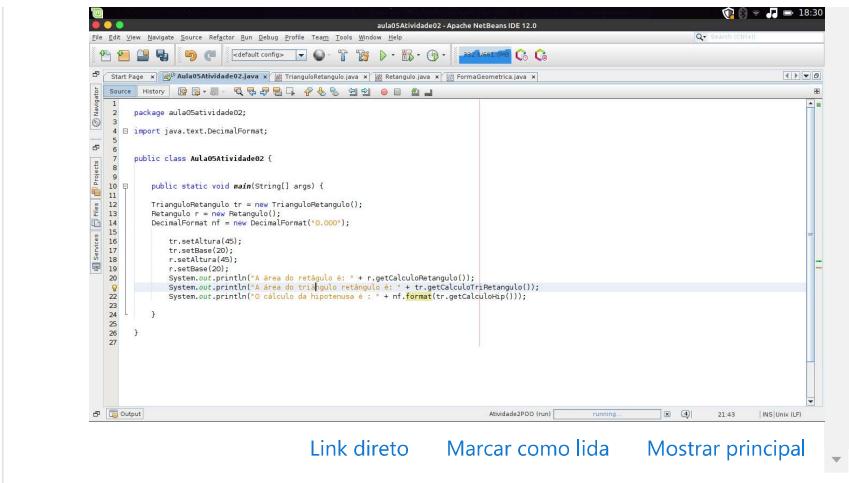


The screenshot shows the NetBeans IDE interface with the title bar "aula05Atividade02 - Apache NetBeans IDE 12.0". The code editor displays the file "Retangulo.java" containing the following code:

```
package aula05Atividade02;

public class Retangulo extends FormaGeometrica {

    /**
     * @return
     */
    public double getCalculoRetangulo(){
        return getBase() * getAltura();
    }
}
```



```

1 package aula05Atividade02;
2 import java.text.DecimalFormat;
3
4 public class Aula05Atividade02 {
5
6     public static void main(String[] args) {
7
8         TrianguloRetangulo tr = new TrianguloRetangulo();
9         Retangulo r = new Retangulo();
10        DecimalFormat nf = new DecimalFormat("0.000");
11
12        tr.setAltura(40);
13        tr.setBase(20);
14        r.setAltura(40);
15        r.setBase(20);
16
17        System.out.println("A área do retângulo é: " + r.getCalculoArea());
18        System.out.println("O cálculo da hipotenusa é: " + nf.format(tr.getHipotenusa()));
19        System.out.println("A área do triângulo retângulo é: " + tr.getCalculoTriRetangulo());
20
21    }
22
23}
24
25
26
27

```

Output: Atividade2POO (run) running 21:43 [INS] Unix (LF)

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [DIEGO VALADARES BAHIA DUARTE](#) - quarta, 9 set 2020, 19:41

ATIVIDADE 2 - AULA 5

```

public class programa {

    public static void main(String[] args) {
        TrianguloRetangulo tr = new TrianguloRetangulo();
        Retangulo r = new Retangulo();

        tr.setAltura(7);
        tr.setBase(4);

        r.setAltura(8);
        r.setBase(6);

        System.out.println("ALTURA DO TRIANGULO: " + tr.getAltura());
        System.out.println("BASE DO TRIANGULO: " + tr.getBase());
        System.out.println("VALOR DA AREA: " + tr.getCalculoArea());
        System.out.println("VALOR DA HIPOTENUSA: " + tr.getHipotenusa());
        System.out.println();
        System.out.println("ALTURA DO RETANGULO: " + r.getAltura());
        System.out.println("BASE DO RETANGULO: " + r.getBase());
        System.out.println("VALOR DA AREA: " + r.getCalculoArea());
    }
}

public class Figura {
    double base;
    double altura;

    public double getCalcArea() {
        return 0;
    }

    public double getBase() {
        return base;
    }

    public void setBase(double base) {
        this.base = base;
    }

    public double getAltura() {
        return altura;
    }

    public void setAltura(double altura) {
        this.altura = altura;
    }
}

public class TrianguloRetangulo extends Figura {

    public double getCalcArea() {

```



```

        return (getBase() * getAltura()) / 2;
    }

    public double getHipotenusa() {
        return (getBase() * getBase()) + (getAltura() * getAltura());
    }
}

public class Retangulo extends Figura {
    public double getCalcArea() {
        return getBase() * getAltura();
    }
}

```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [HELOISA MARTINS CAMBOIM DE SA](#) - quinta, 10 set 2020, 08:43

Aula 05 – Atividade 02

Main.java

```

/*
 * Um matemático trabalha com dois tipos de figuras: os triângulos retângulos
 * e os retângulos.

 * Crie um programa Java (classes) que represente a modelagem para abstrair
 este problema.

 * Ambos têm dois dados comuns: Base e Altura.

 * Cada uma tem um cálculo de forma distinta de área.

 * O triângulo retângulo é base X altura / 2, já do retângulo é apenas base X
altura.

 * Um cálculo diferente somente para o triângulo retângulo deve ser realizado,
que é a hipotenusa.

 * Crie um programa de teste, que instancie objetos, atribua valores e execute
para apresentação.

 */

```



```

public class Main {

    public static void main(String[] args) {
        Retangulo ret = new Retangulo();
        TrianguloRetangulo triRet = new TrianguloRetangulo();
    }
}

```

```

        ret.setBase(10.0);
        ret.setAltura(4.0);

        triRet.setBase(7.0);
        triRet.setAltura(5.0);

        System.out.println("A área do retângulo é: " + ret.getArea());
        System.out.println("A área do triângulo retângulo é: " +
triRet.getArea());
    }
}

```

```
        System.out.println("A hipotenusa do triângulo retângulo é: " +  
triRet.getHipotenusa());  
  
    }  
  
}
```

Figura.java

```
public class Figura {  
  
    private double base;  
  
    private double altura;  
  
    private double area;  
  
  
    public double getBase() {  
  
        return base;  
  
    }  
  
    public void setBase(double base) {  
  
        this.base = base;  
  
    }  
  
    public double getAltura() {  
  
        return altura;  
  
    }  
  
    public void setAltura(double altura) {  
  
        this.altura = altura;  
  
    }  
  
    public double getArea() {  
  
        return area;  
  
    }  
  
    public void setArea(double area) {  
  
        this.area = area;  
  
    }  
  
}
```



Retangulo.java

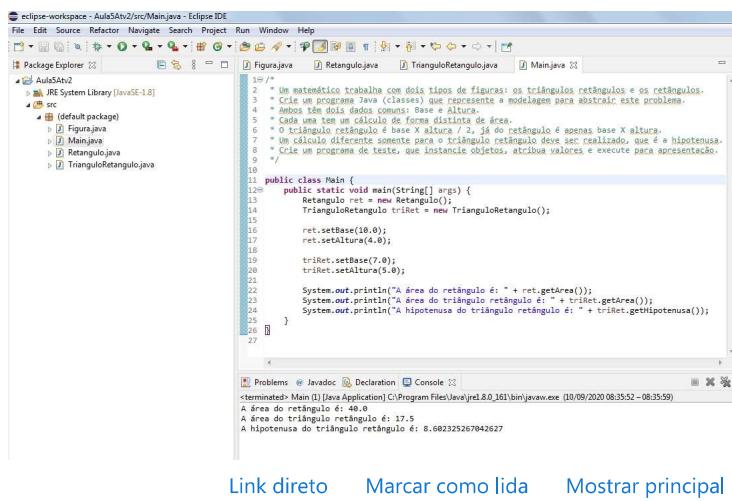
```
public class Retangulo extends Figura {  
  
  
    public double getArea() {  
  
        return getBase() * getAltura();  
  
    }  
  
}
```

TrianguloRetangulo.java

```
public class TrianguloRetangulo extends Figura{

    public double getArea() {
        return getBase() * getAltura() /2;
    }

    public double getHipotenusa() {
        return Math.sqrt(this.getBase() * this.getBase() + this.getBase() * this.getAltura());
    }
}
```



```
1/* 
2 * Um estatístico trabalha com dois tipos de figuras: os triângulos retângulos e os retângulos.
3 * Cria um programa Java (classe) que representa a modelagem para abstrair este problema.
4 * Ambos têm dois dados comuns: Base e Altura.
5 * Cada uma tem o cálculo de forma distinta de área.
6 * Um triângulo retângulo é base X altura / 2, já do retângulo é apenas base X altura.
7 * Um cálculo é aceite somente para o triângulo retângulo devido seu resultado, que é a hipotenusa.
8 * Cria um programa de teste, que instancia objetos, atribui valores e execute para apresentação.
9 */
10
11 public class Main {
12     public static void main(String[] args) {
13         Retangulo ret = new Retangulo();
14         TrianguloRetangulo triRet = new TrianguloRetangulo();
15
16         ret.setBase(10.0);
17         ret.setAltura(4.0);
18
19         triRet.setBase(7.0);
20         triRet.setAltura(3.0);
21
22         System.out.println("A área do retângulo é: " + ret.getArea());
23         System.out.println("A área do triângulo retângulo é: " + triRet.getArea());
24         System.out.println("A hipotenusa do triângulo retângulo é: " + triRet.getHipotenusa());
25     }
26 }
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [PAULLO HENRIQUE DE SOUZA LANDIM](#) - quinta, 10 set 2020, 23:58

Parabéns, cumpriu com o solicitado no exercício.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [MAURICIO APRIGO ANTERO JUNIOR](#) - quinta, 10 set 2020, 09:20

Bom dia, Professora e Colegas!

A atividade que escolhi foi a Atividade 1 do material que trata de Herança.
Meu código ficou assim:

```
import java.util.Scanner;

public class Universidade {
    public static void main (String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Informe o tipo de funcionários.");
        System.out.println("Digite 'P' para 'Professor' e 'A' para 'Administrativo'");
        String profissao = in.nextLine();
```

```

// AQUI USEI O MÉTODO "equals()" PORQUE É A MELHOR MANEIRA DE SE
COMPARAR STRINGS
if (profissao.equals("P")) {
    Professor professor = new Professor();
    String nome;
    String cpf;
    String titulacao;
    double salario;

    // LÊ DADOS DO USUÁRIO
    System.out.println("");
    System.out.println("Informe os seguintes dados:");
    System.out.println("Nome:");
    nome = in.next();
    System.out.println("CPF:");
    cpf = in.next();
    System.out.println("Salário:");
    salario = in.nextDouble();
    System.out.println("Titulação:");
    titulacao = in.next();

    // INSERE DADOS NO OBJETO
    professor.setNome(nome);
    professor.setCpf(cpf);
    professor.setSalario(salario);
    professor.setTitulacao(titulacao);

    // RETORNA VALORES INSERIDOS AO USUÁRIO
    System.out.println("");
    System.out.println("Os dados informados foram:");
    System.out.println("Nome: " + professor.getNome());
    System.out.println("CPF: " + professor.getCpf());
    System.out.println("Profissão: Professor(a)");
    System.out.println("Salário: " + professor.getSalario());
    System.out.println("Titulação: " + professor.getTitulacao());
    System.out.println("Imposto de renda: " + professor.getImpostoRenda());
}
else {
    Administrativo adm = new Administrativo();
    String nome;
    String cpf;
    int dependentes;
    double salario;

    // LÊ DADOS DO USUÁRIO
    System.out.println("");
    System.out.println("Informe os seguintes dados:");
    System.out.println("Nome:");
    nome = in.next();
    System.out.println("CPF:");
    cpf = in.next();
    System.out.println("Salário:");
    salario = in.nextDouble();
    System.out.println("Dependentes:");
    dependentes = in.nextInt();

    // INSERE DADOS NO OBJETO
    adm.setNome(nome);
    adm.setCpf(cpf);
    adm.setSalario(salario);
    adm.setDependentes(dependentes);

    // RETORNA VALORES INSERIDOS AO USUÁRIO
    System.out.println("");
    System.out.println("Os dados informados foram:");
    System.out.println("Nome: " + adm.getNome());
    System.out.println("CPF: " + adm.getCpf());
    System.out.println("Profissão: Administrativo(a)");
}

```



```
System.out.println("Salário: " + adm.getSalario());
System.out.println("Titulação: " + adm.getDependentes());
System.out.println("Contribuição Sindical: " + adm.getContribuicaoSindical());
}

}

}

class Funcionario {
private String nome;
private String cpf;
private double salario;

public String getNome() {
return this.nome;
}

public String getCpf() {
return this.cpf;
}

public double getSalario() {
return this.salario;
}

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

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

public void setSalario(double salario) {
this.salario = salario;
}

}

class Professor extends Funcionario {
private String titulacao;

public double getImpostoRenda() {
return getSalario() * 0.23;
}

public String getTitulacao() {
return this.titulacao;
}

public void setTitulacao(String titulacao) {
this.titulacao = titulacao;
}

}

class Administrativo extends Funcionario {
private int dependentes;

public double getContribuicaoSindical() {
return getSalario() * 0.01;
}

public int getDependentes() {
return this.dependentes;
}

public void setDependentes(int dependentes) {
this.dependentes = dependentes;
}

}
```



É isso aí, galera. Espero que dê pra entender direitinho.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [GRAZIELE ADEISA MONTEIRO DE CASTRO](#) - quinta, 10 set 2020, 15:02

Olá Maurício! Gostei do seu código. achei simples e me ajudou bastante no entendimento dessa questão. Parabéns!

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [TATIANE BORGES RODRIGUES ORSO](#) - quinta, 10 set 2020, 09:42

Bom dia!

Aula 05 - Atividade 02

Um matemático trabalha com dois tipos de figuras: os triângulos retângulos e os retângulos. Crie um programa Java (classes) que represente a modelagem para abstrair este problema. Ambos têm dois dados comuns: Base e Altura. Cada uma tem um cálculo de forma distinta de área. O triângulo retângulo é base X altura / 2, já do retângulo é apenas base X altura. Um cálculo diferente somente para o triângulo retângulo deve ser realizado, que é a hipotenusa. Crie um programa de teste, que instancie objetos, atribua valores e execute para apresentação.



```
1. import java.lang.*; // importação para usar raiz quadrada
2.
3. public class Programa {
4.     public static void main (String args[]) {
5.
6.         Retangulo retangulo = new Retangulo();
7.         TrianguloRetangulo triangulo = new TrianguloRetangulo();
8.
9.         triangulo.setAltura(120.0);
10.        triangulo.setBase(90.0);
11.        System.out.println("Área: " + triangulo.getArea());
12.        System.out.println("Hipotenusa: " + triangulo.getHipotenusa());
13.
14.        retangulo.setAltura(30.0);
15.        retangulo.setBase(20.0);
16.        System.out.println("Área: " + retangulo.getArea());
17.    }
18. }
19.
20. class Figura {
21.     private double altura;
22.     private double base;
23.
24.     public void setBase (double base) {
25.         this.base = base;
26.     }
27.     public double getBase() {
28.         return base;
29.     }
30.     public void setAltura (double altura) {
31.         this.altura = altura;
32.     }
33.     public double getAltura() {
34.         return altura;
35.     }
```

```

36.     public double getArea() { // método que será sobreescrito
37.         return 0.0;
38.     }
39. }
40.
41. class TrianguloRetangulo extends Figura {
42.     public double getArea() {
43.         return (this.getAltura() * this.getBase()) / 2.0;
44.     }
45.     public double getHipotenusa() {
46.         return Math.sqrt(this.getBase() * this.getBase() + this.getAltura() * this.getAltura());
47.     }
48. }
49.
50. class Retangulo extends Figura {
51.     public double getArea() {
52.         return (this.getBase() * this.getAltura());
53.     }
54. }
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [JOSÉ ABELARDO VICENTE FILHO](#) - quinta, 10 set 2020, 12:33

Unidade 3, aula 5, atividade 1:

```

import java.util.Scanner;

public class Atividate01 {

    public static void main(String args[]) {
        String letra;

        System.out.println("Iniciar o cadastro.");
        System.out.println("Para cadastra um Professor digite P, para um
Administrador digite A.");
        letra = new Scanner(System.in).next();

        switch (letra) {
            case "P":
            case "p":
                professor pf = new professor();
                System.out.println("Digite o nome do Professor: ");
                pf.SetNome(new Scanner(System.in).next());
                System.out.println("Digite o CPF: ");
                pf.SetCpf(new Scanner(System.in).next());
                System.out.println("Digite o Salário: ");
                pf.SetSalario(new Scanner(System.in).nextDouble());
                System.out.println("O Valor do imposto de renda é: " + pf.getImpostoRenda());
                break;

            case "A":
            case "a":
                administrativo ad = new administrativo();
                System.out.println("Digite o nome do Administrador: ");
                ad.SetNome(new Scanner(System.in).next());
                System.out.println("Digite o CPF: ");
                ad.SetCpf(new Scanner(System.in).next());
                System.out.println("Digite o Salário: ");
                ad.SetSalario(new Scanner(System.in).nextDouble());
                System.out.println("O Valor da contribuição sindical é: " +
ad.GetcontribuicaoSindical() );
                break;
        }
    }
}
```



```
}

}

class funcionario {
    private String nome;
    private String cpf;
    private double salario;
    //Set
    public void SetNome(String nome) {
        this.nome = nome;
    }
    public void SetCpf(String cpf) {
        this.cpf = cpf;
    }
    public void SetSalario(double salario) {
        this.salario = salario;
    }
    //Get
    public String GetNome() {
        return nome;
    }
    public String GetCpf() {
        return cpf;
    }
    public double GetSalario() {
        return salario;
    }
}

class professor extends funcionario {
    private String titulacao;

    //Get
    public String getTitulacao(){
        return titulacao;
    }

    public double getImpostoRenda(){
        return GetSalario() * 0.23;
    }
}

class administrativo extends funcionario{
    private int dependentes;

    //get
    public int GetDependentes() {
        return dependentes;
    }
    public double GetcontribuicaoSindical() {
        return GetSalario() * 0.01;
    }
}
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [JONATHAN ALBERTO RODRIGUES DAS NEVES](#) - quinta, 10 set 2020, 15:33

```
1  public class Funcionario {  
2  
3      private String nome;  
4      private String cpf;  
5      private double salario;  
6  
7      public String getNome() {  
8          return nome;  
9      }  
10     public void setNome(String nome) {  
11         this.nome = nome;  
12     }  
13     public String getCpf() {  
14         return cpf;  
15     }  
16     public void setCpf(String cpf) {  
17         this.cpf = cpf;  
18     }  
19     public double getSalario() {  
20         return salario;  
21     }  
22     public void setSalario(double salario) {  
23         this.salario = salario;  
24     }  
25 }  
26  
27 Código da classe Professor:  
28  
29     public class Professor extends Funcionario {  
30  
31         private String titulacao;  
32  
33         public double getImpostoRenda() {  
34             return getSalario()*0.23;  
35         }  
36  
37         public String getTitulacao() {  
38             return titulacao;  
39         }  
40  
41         public void setTitulacao(String titulacao) {  
42             this.titulacao = titulacao;  
43         }  
44     }  
45 }  
46  
47 Código da classe Administrativo:  
48  
49     public class Administrativo extends Funcionario {
```



```

51  private int dependentes;
52
53  public int getDependentes() {
54      return dependentes;
55  }
56
57  public void setDependentes(int dependentes) {
58      this.dependentes = dependentes;
59  }
60
61  public double getContribuicaoSindical() {
62      return getSalario()*0.01;
63  }
64
65
66  Código da classe Programa:
67
68  import java.util.Scanner;
69
70  public class Programa {
71      public static void main(String[] args) {
72
73          char tipo;
74
75          Professor prof = new Professor();
76          Administrativo adm = new Administrativo();
77
78          System.out.print("Informe a classe de funcionário (P - Professor ou A - Administrativo): ");
79          tipo = new Scanner(System.in).next().charAt(0);
80
81          if(tipo == 'P') {
82
83              System.out.print("Informe o nome do professor: ");
84              prof.setNome(new Scanner(System.in).nextLine());
85
86              System.out.print("Informe o cpf do professor: ");
87              prof.setCpf(new Scanner(System.in).nextLine());
88
89              System.out.print("Informe o salário do professor, em reais: ");
90              prof.setSalario(new Scanner(System.in).nextDouble());
91
92              System.out.printf("Informe a titulação do professor: ");
93              prof.setTitulacao(new Scanner(System.in).nextLine());
94
95              System.out.println("Dados informados: ");
96              System.out.println("Nome do professor: "+prof.getNome());
97              System.out.println("Cpf do professor: "+prof.getCpf());
98              System.out.println("Salário do professor, em reais: "+prof.getSalario());
99              System.out.println("Titulação do professor: "+prof.getTitulacao());
100             System.out.println("Imposto de renda pago pelo professor, em reais: "+prof.getContribuicaoSindical());
101
102         }
103
104         else {
105
106             System.out.print("Informe o nome do funcionário administrativo: ");
107             adm.setNome(new Scanner(System.in).nextLine());
108
109             System.out.print("Informe o cpf do funcionário administrativo: ");
110             adm.setCpf(new Scanner(System.in).nextLine());
111
112             System.out.print("Informe o salário do funcionário administrativo: ");
113             adm.setSalario(new Scanner(System.in).nextDouble());
114
115             System.out.printf("Informe o número de dependentes do funcionário administrativo: ");
116             adm.setDependentes(new Scanner(System.in).nextInt());
117
118             System.out.println("Dados informados: ");
119             System.out.println("Nome do funcionário administrativo: "+adm.getNome());
120             System.out.println("Cpf do funcionário administrativo: "+adm.getCpf());
121             System.out.println("Salário do funcionário administrativo, em reais: "+adm.getSalario());
122             System.out.println("Número de dependentes do funcionário administrativo: "+adm.getDependentes());
123             System.out.println("Contribuição sindical paga pelo funcionário administrativo, em reais: "+adm.getContribuicaoSindical());
124
125         }
126     }
127 }

```



**Re: Atividade Avaliativa 3**por [RAYANE MACIEL RIBEIRO](#) - sexta, 11 set 2020, 22:37

Boa noite,

Muito interessante a forma como você fez a classe de teste/impressão para o usuário, fiz a mesma questão porém a minha tela ficou bem diferente, muito interessante conhecer uma nova forma de implementação.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**por [EDEGELSON PEREIRA DE NOVAES](#) - quinta, 10 set 2020, 19:47

Unidade 3 - Aula 5 - Atividade 1

```
public class Funcionario {  
    private String nome;  
    private String cpf;  
    private double salario;  
  
    public String getNome() {  
        return nome;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
    public String getCpf() {  
        return cpf;  
    }  
    public void setCpf(String cpf) {  
        this.cpf = cpf;  
    }  
    public double getSalario() {  
        return salario;  
    }  
    public void setSalario(double salario) {  
        this.salario = salario;  
    }  
}
```

```
-----  
-  
-----  
public class Professor extends Funcionario {  
    private String titulacao;
```

```
public double getImpostoRenda() {
    return getSalario() * 0.23;
}

public String getTitulacao() {
    return titulacao;
}

public void setTitulacao(String titulacao) {
    this.titulacao = titulacao;
}
```

```
-
```

```
public class Administrativo extends Funcionario {
    private int dependentes;

    public double getContribuicaoSindical() {
        return getSalario() * 0.01;
    }

    public int getDependentes() {
        return dependentes;
    }

    public void setDependentes(int dependentes) {
        this.dependentes = dependentes;
    }
}
```

```
-
```

```
import java.util.Scanner;

@SuppressWarnings("all")
public class Ler {
    public static String lerString() {
        return new Scanner(System.in).nextLine();
    }

    public static String lerString(String msg) {
        System.out.println(msg);
        return lerString();
    }

    public static double lerDouble() {
        return new Scanner(System.in).nextDouble();
    }

    public static double lerDouble(String msg) {
        System.out.println(msg);
        return lerDouble();
    }

    public static char lerChar() {
        return new Scanner(System.in).next().charAt(0);
    }

    public static char lerChar(String msg) {
        System.out.println(msg);
        return lerChar();
    }

    public static int lerInt() {
        return new Scanner(System.in).nextInt();
    }
}
```



```
}

public static int lerInt(String msg) {
    System.out.println(msg);
    return lerInt();
}
}

-----
-
import java.lang.Character;
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        char opcao = ' ';
        ArrayList proList = new ArrayList();
        ArrayList admList = new ArrayList();

        do {
            opcao = Ler.lerChar("Para começar o cadastro digite P - Professor ou A - Administrativo.");
            switch (Character.toLowerCase(opcao)) {
                case 'p':
                    proList.add(setProfessor());
                    break;
                case 'a':
                    admList.add(setAdministrativo());
                    break;
                default:
                    System.out.println("Opção inválida");
            }
        } while(Character.toUpperCase(opcao) != 'N');

        for(Professor prof : proList) {
            professor(prof);
        }

        for/Administrativo adm : admList) {
            administrativo(adm);
        }
    }

    public static Professor setProfessor() {
        Professor pro = new Professor();

        pro.setNome(Ler.lerString("Digite o nome do Professor: "));
        pro.setCpf(Ler.lerString("Digite o CPF: "));
        pro.setSalario(Ler.lerDouble("Digite o salário: "));
        pro.setTitulacao(Ler.lerString("Digite a Titulação: "));

        return pro;
    }

    public static void professor(Professor prof) {
        System.out.println("*****");
        System.out.println("");
        System.out.println("O nome do professor é: " + prof.getNome());
        System.out.println("O seu CPF é: " + prof.getCpf());
        System.out.println("O salário é: " + prof.getSalario());
        System.out.println("A titulação é: " + prof.getTitulacao());
        System.out.println("O imposto de renda pago é: " + prof.getImpostoRenda());
    }
}
```



```

System.out.println("");
System.out.println("*****");
System.out.println("");
}

public static Administrativo setAdministrativo() {
Administrativo adm = new Administrativo();

adm.setNome(Ler.lerString("Digite o nome: "));
adm.setCpf(Ler.lerString("Digite o CPF: "));
adm.setDependentes(Ler.lerInt("Digite o numero de dependentes: "));
adm.setSalario(Ler.lerDouble("Digite o salario: "));

return adm;
}

public static void administrativo(Administrativo adm) {
System.out.println("*****");
System.out.println("");
System.out.println("O nome do administrativo é: " + adm.getNome());
System.out.println("O seu CPF é: " + adm.getCpf());
System.out.println("O salário é: " + adm.getSalario());
System.out.println("O Numero de dependentes é: " + adm.getDependentes());
System.out.println("O valor da contribuição sindical é: " +
adm.getContribuicaoSindical());
System.out.println("");
System.out.println("*****");
System.out.println("");
}
}

```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [PATRICIA PEDROSA ALVES BRAGA](#) - quinta, 10 set 2020, 22:46

Olá, Edeelson! fiz essa mesma atividade.

Achei interessante a forma como você resolveu. Deixo como sugestão utilizar o método `toString()` para exibir as informações do objeto, bem como criar construtores secundários (sobrecarga de construtor) para instanciar os objetos.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [EDEGELSON PEREIRA DE NOVAES](#) - sexta, 11 set 2020, 15:59

Eu acho que vi rapidamente sobre o `toString()`, eu não consegui encachar a utilização do construtor, vou utilizar a sua sugestão e vou tentar aplicar nesse código, obrigado pela sugestão.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [GISELE SODRÉ DE SOUSA](#) - quinta, 10 set 2020, 21:17

Aula 2 _ Atividade 5

```
// Importa biblioteca que habilita o uso de raízes quadradas
import java.lang.*;
```

```
public class Programa {
public static void main (String[] args) {
```

```
    Retangulo retangulo = new Retangulo();
    TrianguloRetangulo triangulo = new TrianguloRetangulo();
```

```
    triangulo.setAltura(50.0);
```

```

triangulo.setBase(25.0);
System.out.println("A área do triangulo é de " + triangulo.getArea());
System.out.println("A hipotenusa do triangulo é de " +
triangulo.getHipotenusa());

retangulo.setAltura(50.0);
retangulo.setBase(20.0);
System.out.println("A área do retângulo é de " + retangulo.getArea());
}

}

class Figura {
private double altura;
private double base;

public void setBase (double base) {
this.base = base;
}
public double getBase() {
return base;
}
public void setAltura (double altura) {
this.altura = altura;
}
public double getAltura() {
return altura;
}
public double getArea() {
return 0.0;
}
}

class TrianguloRetangulo extends Figura {
public double getArea() {
return (this.getAltura() * this.getBase() / 2.0);
}
public double getHipotenusa() {
return Math.sqrt(this.getBase() * this.getBase() + this.getAltura() *
this.getAltura());
}
}

class Retangulo extends Figura {
public double getArea() {
return (this.getBase() * this.getAltura());
}
}

```



[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [DANIEL SOARES SEABRA](#) - sexta, 11 set 2020, 22:22

Boa noite colega.

Código suscinto, claro e elegante.

Só tenho pequenos adendos a tomar quanto a ele:

1) Indentação

O uso da indentação sempre deixa o código mais claro e mais "confortável" de se ler. Ajuda bastante na manutenção de qualquer programa que vc vier a fazer.

2)Comentários

Percebi que seu código não possui quase nenhum comentário.
Comentários são muito importantes para que outros colegas desenvolvedores entendam seu código.

No mais é isso.

[Link direto](#)

[Marcar como lida](#)

[Mostrar principal](#)



Re: Atividade Avaliativa 3

por [FERNANDO SANTOS FERREIRA](#) - quinta, 10 set 2020, 21:46

Unidade III, aula 6, atividade 1

Main

```
import java.util.Scanner;
//unidade III, aula 6, atividade 1
public class main{
    public static void main(String [] args){
        Scanner scan = new Scanner(System.in);
        Pessoa pessoa = new Pessoa();

        do{
            System.out.println("Olá! Que tal se cadastrar para verificar
sua classe social? \nPor favor, digite o seu nome: ");
            pessoa.setName(scan.nextLine());
        } while(pessoa.getName().equals(""));

        System.out.println("\nInsira algum código para que possamos te
identificar. \nFavor inserir somente números: ");
        pessoa.setCode(scan.nextInt());

        do{
            System.out.println("\n" + pessoa.getName() + ", qual o seu
salário? ");
            pessoa.setPayment(scan.nextDouble());
            if(pessoa.getPayment() < 400){
                System.out.println("Hm... O valor mínimo para verificação
é de R$400,00");
            }
        } while(pessoa.getPayment() < 400);

        System.out.println(pessoa.toString());
    }
}
```



Classe Pessoa

```

public class Pessoa{
    private String name;
    private int code;
    private double payment;

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

    public void setCode(int code){
        this.code = code;
    }

    public void setPayment(double payment){
        this.payment = payment;
    }
    //GET
    public String getName(){
        return name;
    }

    public int getCode(){
        return code;
    }

    public double getPayment(){
        return payment;
    }

    public String getClasse(){
        String classe = "";
        if(this.getPayment() > 400 && this.getPayment() <= 950){
            classe = "F";
        } else if(this.getPayment() > 950 && this.getPayment() <= 1400){
            classe = "E";
        } else if(this.getPayment() > 1400 && this.getPayment() <= 2300){
            classe = "D";
        } else if(this.getPayment() > 2300 && this.getPayment() <= 4600){
            classe = "C";
        } else if(this.getPayment() > 4600 && this.getPayment() <= 8100)
        {
            classe = "B";
        } else if(this.getPayment() > 8100 && this.getPayment() <= 14400)
        {
            classe = "A2";
        } else{
            classe = "A1";
        }
        return classe;
    }

    @Override
    public String toString() {
        return ("\nNome: "+this.getName()+
            "\nCódigo de identificação: "+ this.getCode() +
            "\nSalário: R$" + this.getPayment() +
            "\nClasse social: " + this.getClasse()
            );
    }
}

```



Soma das avaliações: 10 (1)

[Link direto](#) [Marcar como não lida](#) [Mostrar principal](#)

**Re: Atividade Avaliativa 3**por [JOÃO IGOR ALVES DA SILVA](#) - quinta, 10 set 2020, 22:45

Aula 5 Exercício 1

```
import java.util.Scanner;

public class Escolha {
    public static void main(String args[]){
        char opcao;
        Professor prof = new Professor();
        Administrativo admin = new Administrativo();

        System.out.print("Informe P para professor ou A para Administrador:");
        opcao = new Scanner(System.in).next().charAt(0);

        switch (opcao) {
            case 'P': System.out.println("Informe o nome do Professor:");
                        prof.setNome(new Scanner(System.in).nextLine());
                        System.out.println("Informe o CPF: ");
                        prof.setCpf(new Scanner(System.in).nextLine());
                        System.out.println("Informe o salario: ");
                        prof.setSalario(new Scanner(System.in).nextDouble());
                        System.out.println("Informe a titulação: ");
                        prof.setTitulacao(new Scanner(System.in).nextLine());

                        System.out.println();
                        System.out.println();

                        System.out.println("Seguem os dados:");
                        System.out.println();

                        System.out.println("Nome: "+prof.getNome());
                        System.out.println("CPF: "+prof.getCpf());
                        System.out.println("Salário: R$"+prof.getSalario());
                        System.out.println("Titulação: "+prof.getTitulacao());
                        System.out.println("Valor do imposto de renda: R$"+prof.getImpostoRenda());

                        System.out.println();
                        System.out.println();
                        break;

            case 'A': System.out.println("Informe o nome do Administrador: ");
                        admin.setNome(new Scanner(System.in).nextLine());
                        System.out.println("Informe o CPF: ");
                        admin.setCpf(new Scanner(System.in).nextLine());
                        System.out.println("Informe o salario: ");
                        admin.setSalario(new Scanner(System.in).nextDouble());
                        System.out.println("Informe a quantidade de dependentes: ");
                        admin.setDependentes(new Scanner(System.in).nextInt());

                        System.out.println();
                        System.out.println("Seguem os dados:");
                        System.out.println();
```



```

        System.out.println("Nome: " + admin.getNome());
        System.out.println("CPF: " + admin.getCpf());
        System.out.println("Salário: R$" + admin.getSalario());
        System.out.println("Quantidade de dependentes: " + admin.getDependentes());
        System.out.println("Valor da contribuição sindical:");
        R$" + admin.getControbuicaoSindical());
        System.out.println();
        System.out.println();

        break;

    default: System.out.println("Repita a operação, informação inválida");

}
}
}

```

```

public class Funcionario {
private String nome;
private String cpf;
private double salario;

public double getGratificacao(){
return 0;
}

public String getNome() {
return nome;
}

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

public String getCpf() {
return cpf;
}

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

public double getSalario() {
return salario;
}

public void setSalario(double salario) {
this.salario = salario;
}

}

```

```

public class Administrativo extends Funcionario {
private int dependentes;

public double getControbuicaoSindical(){
return getSalario()*0.01;
}

public int getDependentes() {
return dependentes;
}

public void setDependentes(int dependentes) {
this.dependentes = dependentes;
}

```



```
}
```

```
public class Professor extends Funcionario{
    private String titulacao;

    public double getImpostoRenda(){
        return getSalario()*0.23;
    }
    public String getTitulacao() {
        return titulacao;
    }

    public void setTitulacao(String titulacao) {
        this.titulacao = titulacao;
    }
```

```
}
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [FRANCISCO DE ASSIS M DE SOUZA M REZENDE](#) - quinta, 10 set 2020, 22:56



Boa noite profa. e colegas
Aula 05, atividade 1
/*
Crie um programa Java que implemente o projeto da figura acima. Crie
uma classe chamada Funcionario com os atributos (nome, cpf e salário).
Crie uma classe Professor, que é um Funcionário e, além dos dados de
funcionário, tem a titulação e o cálculo do seu imposto de renda por
intermédio
do método getImpostoRenda() que é 23 % do salário. Crie também a
classe Administrativo, que também é Funcionário e tem um dado que
representa a quantidade de dependentes.
Não se esqueça de criar o método getContribuicaoSindical(), que
retornará 1% do salário. Seu programa deve perguntar qual tipo de
Funcionário deseja informar P - Professor e
A - Administrativo. Quando o usuário escolher a opção receba os dados
do respectivo objeto e apresente todas as suas informações.
*/
import java.util.Scanner;
import javax.swing.JButton;
public class Aula5_ativ1 {

 static char P, p;
 static char A,a;
 static int opcao;
 static char pesquisa;
 static char resp;
 public static void main(String[] args) {
 System.out.printf("Qual o funcionário pretende pesquisar?\n" + "\nDigite
P - Professor ou A - Administrativo\n");
 pesquisa = lerChar();
 System.out.println("A opção digitada foi: " + lerCondicao());
 }
 public static char lerChar() {
 return new Scanner(System.in).next().charAt(0);
 }
 public static int opcao() {
 return lerCondicao();
 }

```
}

public static String lerString() {
    return new Scanner(System.in).nextLine();
}

public static double lerDouble() {
    return new Scanner(System.in).nextDouble();
}

public static int lerInt() {

    return new Scanner(System.in).nextInt();
}

public static int lerCondicao() {
    do
    {
        if (pesquisa == 'P' || pesquisa == 'p') {

            Professor professor = new Professor();

            System.out.printf("\nDigite o nome do professor: ");
            professor.getNome = lerString();
            System.out.println("O nome deste (a) professor é: " +
                professor.getNome);
            System.out.println("Digite o salário do professor: ");
            professor.getSalario = lerDouble();
            System.out.println("O salário do professor é: " + professor.getSalario);
            System.out.println("O desconto no Imposto de renda do professor é: " +
                professor.getImpostoRenda());
            System.out.println("Digite a titulação do professor: ");
            professor.getTitulacao = lerString();
            System.out.println("A titulação do professor é: " +
                professor.getTitulacao);
            System.out.printf("\n\nDeseja incluir outro professor?\nDigite 1 - Sim ou
2 - não ");
            resp = lerChar();
            break;

        }else if(pesquisa == 'A' || pesquisa == 'a'){

            Administrativo a = new Administrativo();

            System.out.println("Digite o nome do funcionário administrativo: ");

            a.getNome = lerString();
            System.out.println("O nome deste (a) funcionário (a) é : " + a.getNome);

            System.out.printf("\nDigite a quantidade de dependentes tem "
                +a.getNome +": ");
            a.getGetQntDependente= lerDouble();
            System.out.println(a.getNome + " tem " + a.getGetQntDependente + "
dependentes.");
            System.out.printf("\nDigite o salário do funcionário administrativo: ");
            a.getSalario = lerDouble();
            System.out.printf("\nO salário do funcionário administrativo é: "+
                a.getSalario + " reais, " + " e sua contribuição sindical é: R$ " +
                a.getGetContribuicaoSindical() + " reais.");
            System.out.printf("\n\nDeseja incluir outro funcionário administrativo?
\nDigite 1 - Sim ou 2 - não ");
            resp = lerChar();
            break;
        }else {

            System.out.println("Dados Inválidos!");
        }
    }while (resp == 1); }

    System.out.println("fim");
}

return lerCondicao();
```



```
}

}



---



```
import java.util.Scanner;

public class Professor extends Funcionario {
 private String titulacao;
 public String getNome;
 public double getSalario;
 public String getTitulacao;
 public double getImpostoRenda(){
 return getSalario * 0.23;
 }
 public String getTitulacao() {
 return titulacao;
 }
 public void setTitulacao(String titulacao) {
 this.titulacao = titulacao;
 }
}
```


```



```
public class Administrativo extends Funcionario {

    public String getNome;
    public double getSalario;
    private double qntDependente;
    public double getQntDependente;
    public String getContribuicaoSindical;

    public double getGetContribuicaoSindical() {
        return getSalario * 0.01;
    }

    public double getGetQntDependente() {
        return qntDependente;
    }

    public void setGetQntDependente(double qntDependente) {
        this.qntDependente = qntDependente;
    }
}
```

```
public class Funcionario {

    private double contribuicaoSindical;
    public String nome;
    public double salario;
    private double qntDependente;
    private String titulacao;

    public double getContribuicaoSindical() {
        return contribuicaoSindical;
    }

    public void setContribuicaoSindical(double contribuicaoSindical) {
        this.contribuicaoSindical = contribuicaoSindical;
    }

    public String getNome() {
        return nome;
    }

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

```
public double getSalario() {  
    return salario;  
}  
public void setSalario(double salario) {  
    this.salario = salario;  
}  
public double getQntDependente() {  
    return qntDependente;  
}  
public void setQntDependente(double qntDependente) {  
    this.qntDependente = qntDependente;  
}  
public String getTitulacao() {  
    return titulacao;  
}  
public void setTitulacao(String titulacao) {  
    this.titulacao = titulacao;  
}
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#) ▾



Re: Atividade Avaliativa 3

por [THIAGO DA SILVA SENNA](#) - quinta, 10 set 2020, 23:41

Ótimo código, fácil de entender apesar de ter ficado gigante.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [PATRICIA PEDROSA ALVES BRAGA](#) - quinta, 10 set 2020, 22:52

Olá, segue minha resposta:

Unidade III - Aula 05 - Atividade 01

----- Classe Principal -----

```
public class Principal {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.println("Qual o tipo de funcionário que deseja cadastrar?\n <P>  
professor | <A> Administrador");  
        String tipoDeFuncionario = scanner.nextLine();  
  
        if(tipoDeFuncionario.equalsIgnoreCase("P")) {  
  
            System.out.println("Insira o nome: ");  
            String nome = scanner.nextLine();  
            System.out.println("Insira o CPF: ");  
            String cpf = scanner.nextLine();  
            System.out.println("Insira a titulação: ");  
            String titulacao = scanner.nextLine();  
            System.out.println("Insira o salário: ");  
            double salario = scanner.nextDouble();  
  
            Professor prof = new Professor(nome, cpf, salario, titulacao);  
  
            exibirDados(prof);  
  
        }else if (tipoDeFuncionario.equalsIgnoreCase("A")) {  
  
            System.out.println("Insira o nome: ");  
            String nome = scanner.nextLine();  
            System.out.println("Insira o CPF: ");  
        }  
    }  
}
```

```

String cpf = scanner.nextLine();
System.out.println("Insira o número de dependentes:");
int dependentes = scanner.nextInt();
System.out.println("Insira o salário:");
double salario = scanner.nextDouble();

Administrativo adm = new Administrativo(nome, cpf, dependentes, salario);

exibirDados(adm);

} else {
System.out.println("Opção Inválida!");
}

scanner.close();
}

```

private static void exibirDados(Funcionario funcionario) {
 System.out.println(funcionario.toString());
}

}

----- Classe Funcionário -----

```

public class Funcionario {

private String nome;
private String cpf;
private double salario;

public String getNome() {
return nome;
}

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

public String getCpf() {
return cpf;
}

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

public double getSalario() {
return salario;
}

public void setSalario(double salario) {
this.salario = salario;
}
}

```

----- Classe Professor -----

```

public class Professor extends Funcionario {

private String titulacao;

```



```

public Professor(String nome, String cpf, double salario, String titulacao) {
    this.setNome(nome);
    this.setCpf(cpf);
    this.setSalario(salario);
    this.setTitulacao(titulacao);
}

public String getTitulacao() {
    return titulacao;
}

public void setTitulacao(String titulacao) {
    this.titulacao = titulacao;
}

public double getImpostoRenda() {
    return getSalario() * 0.23;
}

@Override
public String toString() {
    return "[Professor]" + "\nTitulação = " + getTitulacao() + "\nNome = " +
        getName() + "\nCPF = " + getCpf() + "\nSalário: "
        + getSalario() + "\nImposto de renda = " + getImpostoRenda();
}
}

```

----- Classe Administrativo -----



```

public class Administrativo extends Funcionario {

    private int dependentes;

    public Administrativo(String nome, String cpf, int dependentes, double salario) {
        this.setNome(nome);
        this.setCpf(cpf);
        this.setSalario(salario);
        this.setDependentes(dependentes);
    }

    public int getDependentes() {
        return dependentes;
    }

    public void setDependentes(int dependentes) {
        this.dependentes = dependentes;
    }

    public double getContribuicaoSindical() {
        return getSalario() * 0.01;
    }

    @Override
    public String toString() {
        return "[Administrativo]" + "\nDependentes = " + getDependentes() + "\nNome
= " + getName() + "\nCPF = " + getCpf()
        + "\nSalário = " + getSalario() + "\nContribuição Sindical: " +
        getContribuicaoSindical();
    }
}

```

**Re: Atividade Avaliativa 3**por [LEIDIANA SILVA DE OLIVEIRA GONÇALVES](#) - sexta, 11 set 2020, 10:38

Bom dia Patricia,muito bom seu código. Ficou bem interativo e bem definida, ajudou bastante no entendimento da questão.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)**Re: Atividade Avaliativa 3**por [THIAGO DA SILVA SENNA](#) - quinta, 10 set 2020, 23:39**Aula 05 - Atividade 02**

```
import java.lang.*; // importação para usar raiz quadrada

public class Programa {

    public static void main (String args[]) {

        Retangulo retangulo = new Retangulo();

        TrianguloRetangulo triangulo = new TrianguloRetangulo();

        triangulo.setAltura(120.0);

        triangulo.setBase(90.0);

        System.out.println("Área: " + triangulo.getArea());

        System.out.println("Hipotenusa: " + triangulo.getHipotenusa());

        retangulo.setAltura(30.0);

        retangulo.setBase(20.0);

        System.out.println("Área: " + retangulo.getArea());

    }

}

class Figura {

    private double altura;

    private double base;

    public void setBase (double base) {

        this.base = base;

    }

    public double getBase() {

        return base;

    }

    public void setAltura (double altura) {

        this.altura = altura;

    }

    public double getAltura() {

        return altura;

    }

    public double getArea() { // método que será sobreescrito

        return 0.0;

    }

}
```



```
}

class TrianguloRetangulo extends Figura {

    public double getArea() {

        return (this.getAltura() * this.getBase() / 2.0);

    }

    public double getHipotenus() {

        return Math.sqrt(this.getBase() * this.getBase() + this.getAltura() * this.getAltura());

    }

}

class Retangulo extends Figura {

    public double getArea() {

        return (this.getBase() * this.getAltura());

    }

}

import java.lang.*; // importação para usar raiz quadrada

public class Programa {

    public static void main (String args[]) {

        Retangulo retangulo = new Retangulo();

        TrianguloRetangulo triangulo = new TrianguloRetangulo();

        triangulo.setAltura(120.0);

        triangulo.setBase(90.0);

        System.out.println("Área: " + triangulo.getArea());

        System.out.println("Hipotenusa: " + triangulo.getHipotenus());

        retangulo.setAltura(30.0);

        retangulo.setBase(20.0);

        System.out.println("Área: " + retangulo.getArea());

    }

}

class Figura {

    private double altura;

    private double base;

    public void setBase (double base) {

        this.base = base;

    }

    public double getBase() {

        return base;

    }

    public void setAltura (double altura) {

        this.altura = altura;

    }

}
```



```

public double getAltura() {
    return altura;
}

public double getArea() { // método que será sobreescrito
    return 0.0;
}

}

class TrianguloRetangulo extends Figura {

    public double getArea() {
        return (this.getAltura() * this.getBase() / 2.0);
    }

    public double getHipotenusa() {
        return Math.sqrt(this.getBase() * this.getBase() + this.getAltura() * this.getAltura());
    }
}

class Retangulo extends Figura {

    public double getArea() {
        return (this.getBase() * this.getAltura());
    }
}

```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [PAULLO HENRIQUE DE SOUZA LANDIM](#) - quinta, 10 set 2020, 23:56

Exercício 1 aula 5

```

import java.util.Scanner;

public class ProgramaJava {

    public static void main(String args[]) {

        String tipo = TipoDeFuncionario();

        switch (tipo) {
            case "A":
                CriacaoFuncionarioAdministrativo();
                break;
            case "P":
                CriacaoProfessor();
                break;
            default:
                TipoDeFuncionario();
        }
    }

    public static String TipoDeFuncionario() {
        System.out.print("Favor informar o tipo de Funcionario : (P) Professor, A
(Administrativo ):");
    }
}

```

```
String tipoFuncionario = new Scanner(System.in).next();
return tipoFuncionario;
}

public static void CriacaoFuncionarioAdministrativo() {

Administrativo novoFuncionario = new Administrativo();
double contSindical;

System.out.print("Favor informar o nome do Funcionario : ");
novoFuncionario.nome = new Scanner(System.in).next();

System.out.print("Favor Informar o nome do CPF : ");
novoFuncionario.cpf = new Scanner(System.in).next();

System.out.print("Favor informar o numero de dependentes : ");
novoFuncionario.dependentes = new Scanner(System.in).nextInt();

System.out.print("Favor informar o salario do Funcionario : ");
novoFuncionario.salario = new Scanner(System.in).nextDouble();

contSindical =
novoFuncionario.getContribuicaoSindical(novoFuncionario.salario);

System.out.println("Nome do Funcionario :" + novoFuncionario.nome);
System.out.println("CPF :" + novoFuncionario.cpf);
System.out.println("Dependentes:" + novoFuncionario.dependentes);
System.out.println("Salario :" + novoFuncionario.salario);
System.out.println("Contribuicao Sindical:" + contSindical);

}

public static void CriacaoProfessor(){
Professor novoFuncionario = new Professor();
double impostoRenda;

System.out.print("Favor informar o nome do Funcionario : ");
novoFuncionario.nome = new Scanner(System.in).next();

System.out.print("Favor informar o numero do CPF : ");
novoFuncionario.cpf = new Scanner(System.in).next();

System.out.print("Favor informar o salario do Funcionario : ");
novoFuncionario.salario = new Scanner(System.in).nextDouble();

System.out.print("Favor informar a Titulacao : ");
novoFuncionario.titulacao = new Scanner(System.in).next();

impostoRenda =
novoFuncionario.getImpostoRenada(novoFuncionario.salario);

System.out.println("Nome do Funcionario :" + novoFuncionario.nome);
System.out.println("CPF :" + novoFuncionario.cpf);
System.out.println("Salario :" + novoFuncionario.salario);
System.out.println("Titulacao:" + novoFuncionario.titulacao);
System.out.println("Imposto de Renda:" + impostoRenda);

}

public class Funcionario{
String nome;
String cpf;
String salario;
}

public class Professor extends Funcionario{
```



```
String titulacao;

public double getImpostoRenda (double salario){
    double imposto;
    return imposto = (salario * 23)/100;
}

public class Administrativo extends Funcionario{

    int dependentes;

    public double getContribuicaoSindical(double salario){
        double contSindical;
        return contSindical = (salario * 1)/100;
    }
}
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [LEIDIANA SILVA DE OLIVEIRA GONÇALVES](#) - sexta, 11 set 2020, 10:31

Olá professora e colegas!

Atividade 2 - Aula 5

```
public class programa {

    public static void main(String[] args) {
        TrianguloRetangulo tr = new TrianguloRetangulo();
        Retangulo r = new Retangulo();

        tr.setAltura(7);
        tr.setBase(4);

        r.setAltura(8);
        r.setBase(6);

        System.out.println("ALTURA DO TRIANGULO: "+tr.getAltura());
        System.out.println("BASE DO TRIANGULO: "+tr.getBase());
        System.out.println("VALOR DA AREA: "+tr.getCalculoArea());
        System.out.println("VALOR DA HIPOTENUSA: "+tr.getHipotenusa());
        System.out.println();
        System.out.println("ALTURA DO RETANGULO: "+r.getAltura());
        System.out.println("BASE DO RETANGULO: "+r.getBase());
        System.out.println("VALOR DA AREA: "+r.getCalculoArea());
    }
}

public class Figura {
    double base;
    double altura;

    public double getCalcArea() {
        return 0;
    }
    public double getBase() {
        return base;
    }
    public void setBase(double base) {
        this.base = base;
    }
    public double getAltura() {
        return altura;
    }
    public void setAltura(double altura) {
        this.altura = altura;
    }
}
```



```

        }
    }

public class TrianguloRetangulo extends Figura {

    public double getCalcArea() {
        return (getBase() * getAltura())/2;
    }

    public double getHipotenusas() {
        return (getBase() * getBase()) + (getAltura() * getAltura());
    }
}

public class Retangulo extends Figura {
    public double getCalcArea() {
        return getBase() * getAltura();
    }
}

```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [RUBEM NERE COUTINHO COELHO](#) - sexta, 11 set 2020, 10:18

Aula 5 atividade 1

package aula05;

```

import javax.swing.JOptionPane;

public class atividade01 {
    public static void main(String args[]){
        Professor prof = new Professor();
        Administrativo adm = new Administrativo();
        String opcao = "";
        //Garante a escolha correta até o usuário digitar para cancelar
        while ((opcao.equals("A")||opcao.equals("a")) == false ||
        (opcao.equals("P")||opcao.equals("p")) == false ) {

            opcao = JOptionPane.showInputDialog("informe\n P - Professor e A -
            Administrativo\n Ou clique em cancelar para sair");

            // caso a escolha seja de um funcionário da ADM
            if (opcao.equals("A")||opcao.equals("a")) {

                adm.setNome(JOptionPane.showInputDialog("Digite o nome do Funcionário"));
                adm.setCpf(JOptionPane.showInputDialog("Digite o CPF"));
                adm.setSalario(Double.parseDouble(JOptionPane.showInputDialog("Digite o
                salário do Funcionário")));
                adm.setDependentes(Integer.parseInt(JOptionPane.showInputDialog("Digite a
                quantidade de dependentes")));
                JOptionPane.showMessageDialog(null,"Sr(a) "+adm.getNome()+"\n CPF:
                "+adm.getCpf()+"\nCom : "+adm.getDependentes()+" Dependentes \n e Salário
                de: "
                +adm.getSalario()+"\n irá pagar a contribuição sindical de : R$
                "+adm.getContribuicaoSindical());

                //caso a escolha seja de um Professor
            }else if (opcao.equals("P")||opcao.equals("p")){
                prof.setNome(JOptionPane.showInputDialog("Digite o nome do Professor"));
                prof.setCpf(JOptionPane.showInputDialog("Digite o CPF"));
                prof.setSalario(Double.parseDouble(JOptionPane.showInputDialog("Digite o
                salário do Professor")));
                prof.setTitulacao(JOptionPane.showInputDialog("Digite a titulação do
                Professor"));
                JOptionPane.showMessageDialog(null,"Sr(a) "+prof.getNome()+" Titulação:
                "+prof.getTitulacao()+"\nCPF: "+prof.getCpf()+
                "\nSalário de: "+prof.getSalario()+" irá pagar de imposto o valor: R$"
            }
        }
    }
}

```



```

"+prof.getImpostoRenda());

}else {
//caso o usuário escolha uma opção não válida
JOptionPane.showMessageDialog(null, "Atenção, "+opcao+" Não é uma escolha
válida");
}

}

}

}

class Funcionario {
private String nome;
private String cpf;
private double salario;
public void setNome(String nome){
this.nome = nome;
}
public String getNome(){
return nome;
}
public void setCpf(String cpf){
this.cpf = cpf;
}
public String getCpf(){
return cpf;
}
public void setSalario(double salario){
this.salario = salario;
}
public double getSalario(){
return salario;
}
}

class Professor extends Funcionario{//vinculo de herança
private String titulacao;
public void setTitulacao(String titulacao){
this.titulacao = titulacao;
}
public String getTitulacao(){
return titulacao;
}
public double getImpostoRenda(){
return getSalario() * 0.23; //taxa fixa de 23% do salário
}
}

class Administrativo extends Funcionario{//vínculo de herança
private int dependentes;
public void setDependentes(int dependentes){
this.dependentes = dependentes;
}
public int getDependentes(){
return dependentes;
}
public double getContribuicaoSindical(){
return getSalario() * 0.01; //taxa fixa de 1% do salario
}
}

```



[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [CLEITON ANTÔNIO GONÇALVES](#) - sexta, 11 set 2020, 19:50

Muito bom o código e está bem comentado também...



Re: Atividade Avaliativa 3
por [LUCAS LUAN DA SILVA LACERDA](#) - sexta, 11 set 2020, 11:01

```
import java.util.Scanner;

public class Programa {
    @SuppressWarnings("resource")
    public static void main(String[] args) {

        char escolhaProfissao;

        Professor prof = new Professor();

        Administrativo admin = new Administrativo();

        do {

            System.out.println("Administrativo - A/a ");

            System.out.println("Professor - P/p ");

            System.out.printf("Qual a sua profissão? ");

            escolhaProfissao = new Scanner(System.in).next().charAt(0);

            if (escolhaProfissao == 'P' || escolhaProfissao == 'p') {

                System.out.println("-----");

                System.out.println("PROFESSOR");

            } else if (escolhaProfissao == 'A' || escolhaProfissao == 'a') {

                System.out.println("-----");

                System.out.println("ADMINISTRATIVO");

            } else {

                System.out.println("-----");

                System.out.println("OPS! Profissão não cadastrada");

            }

        } while (escolhaProfissao != 'A' && escolhaProfissao != 'P' && escolhaProfissao != 'a' && escolhaProfissao != 'p');

        System.out.println("-----");

        if (escolhaProfissao == 'P' || escolhaProfissao == 'p') {

            System.out.printf("Qual o seu nome? ");

            prof.setNome(new Scanner(System.in).nextLine());

            ;

            System.out.printf("Qual o seu cpf? ");

            prof.setCpf(new Scanner(System.in).nextLine());

            System.out.printf("Qual o seu Salário Bruto? ");

            prof.setSalario(new Scanner(System.in).nextDouble());

            System.out.printf("Qual a sua titulação? ");

            prof.setTitulacao(new Scanner(System.in).nextLine());

        }
```



```

;

System.out.println("-----");

System.out.println("Confirmar Dados : ");

System.out.println("Nome: " + prof.getNome());

System.out.println("CPF: " + prof.getCpf());

System.out.println("Salário Líquido: ");

System.out.println("Imposto de renda: " + prof.getImpostoRenda());

System.out.println("Titulação: " + prof.getTitulacao());

} else {

System.out.printf("Qual o seu nome? ");

admin.setNome(new Scanner(System.in).nextLine());

System.out.printf("Qual o seu cpf? ");

admin.setCpf(new Scanner(System.in).nextLine());

System.out.printf("Qual o seu Salário Bruto? ");

admin.setSalario(new Scanner(System.in).nextDouble());

System.out.printf("Possui quantos dependentes? ");

admin.setDependentes(new Scanner(System.in).nextInt());

System.out.println("-----");

System.out.println("Confirmar Dados : \n");

System.out.println("Nome: " + admin.getNome());

System.out.println("CPF: " + admin.getCpf());

System.out.println("Salário Líquido: " + admin.getSalario());

System.out.println("Contribuição sindical: " + admin.getContribuicaoSindical());

System.out.println("Dependentes: " + admin.getDependentes());
}

}

class Funcionario {

private String nome;

private String cpf;

private double salario;

public String getNome() {
return nome;
}

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

```



```
public String getCpf() {
    return cpf;
}

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

public double getSalario() {
    return salario;
}

public void setSalario(double salario) {
    this.salario = salario;
}

}

class Professor extends Funcionario {

    private String titulacao;

    public String getTitulacao() {
        return titulacao;
    }

    public void setTitulacao(String titulacao) {
        this.titulacao = titulacao;
    }

    public double getImpostoRenda() {
        return getSalario() * 0.23;
    }
}

class Administrativo extends Funcionario {

    private int dependentes;

    public int getDependentes() {
        return dependentes;
    }

    public void setDependentes(int dependentes) {
        this.dependentes = dependentes;
    }

    public double getContribuicaoSindical() {
        return getSalario() * .01;
    }
}
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [RICARDO NUNES PEREIRA](#) - sexta, 11 set 2020, 14:58

Crie uma classe pessoa com nome, código inteiro e salário. A classe social de uma pessoa é calculada conforme o seguinte:

Classe

A1: inclui as famílias com renda mensal maior que R\$ 14.400

Classe A2: maior que R\$ 8.100

Classe B: maior que R\$ 4.600

Classe C: maior que R\$ 2.300

Classe D: maior que R\$ 1.400

Classe E: maior que R\$ 950

Classe F: maior que R\$ 400

Receba do usuário os dados de uma pessoa e apresente todas as informações da pessoa, inclusive a classe por meio do método `toString`. Atenção! O método `toString` não deve fazer o cálculo de classe da pessoa.

```
public class Programa {  
    public static void main(String[] args) {  
        Pessoa p = Visao.lerPessoa();  
        Visao.imprimir(p);  
    }  
}  
public class Pessoa {  
    private String nome;  
    private int cod;  
    private double salario;  
    public String getNome() {  
        return nome;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
    public int getCod() {  
        return cod;  
    }  
    public void setCod(int cod) {  
        this.cod = cod;  
    }  
    public double getSalario() {  
        return salario;
```



```

}

public void setSalario(double salario) {
    this.salario = salario;
}

public String getClasseSocial() {
    String classeSocial = "A1";
    if(getSalario()<=400) classeSocial = "Não atribuída Classe Social";
    else if(getSalario()<=950) classeSocial = "F";
    else if(getSalario()<=1400) classeSocial = "E";
    else if(getSalario()<=2300) classeSocial = "D";
    else if(getSalario()<=4600) classeSocial = "C";
    else if(getSalario()<=8100) classeSocial = "B";
    else if(getSalario()<=14400) classeSocial = "A2";
    return classeSocial;
}

@Override //sobrescrita do método toString()

public String toString() {
    String texto = "\nNome: "+getNome()+"\nCódigo: "+getCod()+"\nSalário: "+getSalario()
        +" \nClasse Social: "+getClasseSocial();
    return texto;
}

import java.util.Scanner;

@SuppressWarnings("all")

public class ScannerUCB {

    // String

    public static String lerString(String msg) {
        System.out.println(msg);
        return lerString();
    }

    public static String lerString() {

```



```
return new Scanner(System.in).nextLine();

}

// char

public static char lerCaractere(String msg) {
    System.out.println(msg);
    return lerCaractere();
}

public static char lerCaractere() {
    return new Scanner(System.in).next().charAt(0);
}

// int

public static int lerInt(String msg, String msgErro, int min, int max) {
    int valor;
    boolean errado;
    do {
        valor = lerInt(msg);
        errado = valor < min || valor > max;
        if (errado) {
            System.out.println(msgErro);
        }
    } while (errado);
    return valor;
}

public static int lerInt(String msg) {
    System.out.println(msg);
    return lerInt();
}

public static int lerInt() {
    return new Scanner(System.in).nextInt();
}
```



```
}
```



```
// double
```



```
public static double lerDouble(String msg, String msgErro, double min, double max) {
```



```
    double valor;
```



```
    boolean errado;
```



```
    do {
```



```
        valor = lerDouble(msg);
```



```
        errado = valor < min || valor > max;
```



```
        if (errado) {
```



```
            System.out.println(msgErro);
```



```
        }
```



```
    } while (errado);
```



```
    return valor;
```



```
}
```

```
public static double lerDouble(String msg) {
```



```
    System.out.println(msg);
```



```
    return lerDouble();
```



```
}
```

```
public static double lerDouble() {
```



```
    return new Scanner(System.in).nextDouble();
```



```
}
```



```
}
```



```
public class Visao {
```

```
public static Pessoa lerPessoa() {
```



```
    Pessoa p = new Pessoa();
```



```
    p.setNome(ScannerUCB.lerString("Informe o nome da pessoa: "));
```



```
    p.setCod(ScannerUCB.lerInt("Informe o código da pessoa"));
```



```
    p.setSalario(ScannerUCB.lerDouble("Informe o salario da pessoa: ", "OPS!, Salário inválido", 0, Double.MAX_VALUE));
```



```
return p;  
}  
  
}  
  
public static void imprimir(Pessoa p) {  
    System.out.println(p.toString());  
}
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [PEDRO VINÍCIUS BARBOSA DO VALLE FURTADO](#) - sexta, 11 set 2020, 16:58

AULA 05 EXERCÍCIO 01

```
import java.util.Scanner;  
  
public class Programa {  
    public static void main(String[] args) {  
        char profissao;  
  
        professor prof = new professor();  
        administrativo admin = new administrativo();  
  
        do {  
            System.out.printf("Qual tipo de funcionário você é? Escreva A para administrativo  
e P para Professor. ");  
            profissao = new Scanner(System.in).next().charAt(0);  
  
            if(profissao != 'A' && profissao != 'P') {  
                System.out.println("Profissão inválida! Digite uma profissão válida!");  
            }  
        }while (profissao != 'A' && profissao != 'P');  
  
        if (profissao == 'P') {  
  
            System.out.printf("Informe o seu nome. ");  
            prof.nome = new Scanner(System.in).nextLine();  
            System.out.printf("Insira o seu CPF. ");  
            prof.cpf = new Scanner(System.in).nextLine();  
            System.out.printf("Informe o seu salário. ");  
            prof.salario = new Scanner(System.in).nextDouble();  
            System.out.printf("Qual a sua titulação? ");  
            prof.titulacao = new Scanner(System.in).nextLine();  
  
            System.out.println("Seus dados: ");  
            System.out.println("Nome: "+prof.nome);  
            System.out.println("CPF: "+prof.cpf);  
            System.out.println("Salário Líquido: "+(prof.salario -  
getImpostoRenda(prof.salario)));  
            System.out.println("Titulação: "+prof.titulacao);  
  
        }else {  
  
            System.out.printf("Informe o seu nome. ");  
            admin.nome = new Scanner(System.in).nextLine();  
            System.out.printf("Insira o seu CPF. ");  
            admin.cpf = new Scanner(System.in).nextLine();  
            System.out.printf("Informe o seu salário. ");  
            admin.salario = new Scanner(System.in).nextDouble();  
            System.out.printf("Possui quantos dependentes? ");  
        }  
    }  
}
```



```

admin.dependentes = new Scanner(System.in).nextInt();

System.out.println("Seus dados: ");
System.out.println("Nome: "+admin.nome);
System.out.println("CPF: "+admin.cpf);
System.out.println("Salário Líquido: "+(admin.salario -
getContribuicaoSindical(admin.salario)));
System.out.println("Dependentes: "+admin.dependentes);
}

}

public static double getImpostoRenda(double salario) {
return(0.23*salario);
}
public static double getContribuicaoSindical(double salario) {
return(0.01*salario);
}
}
class funcionario{
String nome;
String cpf;
double salario;
}
class professor extends funcionario{
String titulacao;
}
class administrativo extends funcionario{
int dependentes;
}

```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [AVIVALDO RIBEIRO DOS SANTOS JUNIOR](#) - sexta, 11 set 2020, 17:35

```

import java.lang.*;

public class Teste {
public static void main (String args[]) {

    TrianguloRetangulo triangulo = new TrianguloRetangulo();
    Retangulo retangulo = new Retangulo();

    triangulo.setBase(4.0);
    triangulo.setAltura(7.0);
    System.out.println("Área: " + triangulo.getArea());
    System.out.println("Hipotenusa: " + triangulo.getHipotenusa());

    retangulo.setBase(9.0);
    retangulo.setAltura(2.0);
    System.out.println("Área: " + retangulo.getArea());
}

}

class FigurasGeometricas {
private double base;
private double altura;

public void setBase (double base) {
this.base = base;
}
public double getBase() {
return base;
}
public void setAltura (double altura) {
this.altura = altura;
}
public double getAltura() {

```

```

        return altura;
    }
    public double getArea() {
        return 0.0;
    }

    class TrianguloRetangulo extends FigurasGeometricas {
        public double getArea() {
            return (this.getBase() * this.getAltura()) / 2.0;
        }
        public double getHipotenusa() {
            return Math.sqrt(this.getBase() * this.getBase() + this.getAltura() *
            this.getAltura());
        }
    }

    class Retangulo extends FigurasGeometricas {
        public double getArea() {
            return (this.getBase() * this.getAltura());
        }
    }

```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [RUBEM NERE COUTINHO COELHO](#) - sexta, 11 set 2020, 18:18

Prezado boa tarde excelente código, eu só implementaria uma forma do usuário digitar os valores e a linha import java.lang.* é opcional pois de forma implícita todas as classes do lang são importadas.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [MARÍLIA BOVO LOPES](#) - sexta, 11 set 2020, 18:22

Abaixo consta a resolução para o exercício 1 da aula 5:

```

import java.util.Scanner;

public class A5E1 {
    public static void main(String args[]){
        char escolha;
        Professor profe = new Professor();
        Administrativo adm = new Administrativo();
    }
}

```

```

System.out.print("Informe P para professor ou A para Administrador:");
escolha = new Scanner(System.in).next().charAt(0);

```

```

switch (escolha) {
    case 'P': System.out.println("Informe o nome do Professor:");
        profe.setNome(new Scanner(System.in).nextLine());
        System.out.println("Informe o CPF:");
        profe.setCpf(new Scanner(System.in).nextLine());
        System.out.println("Informe o salário:");
        profe.setSalario(new Scanner(System.in).nextDouble());
        System.out.println("Informe o título:");
        profe.setTitulacao(new Scanner(System.in).nextLine());
}

```

```

System.out.println();
System.out.println();

```

```

System.out.println("Seguem os dados:");
System.out.println();

```

```

System.out.println("Nome: "+profe.getNome());
System.out.println("Título: "+profe.getTitulo());

```

```

System.out.println("CPF: "+profe.getCpf());
System.out.println("Salário: "+profe.getSalario() + "reais");

System.out.println("Valor do imposto de renda é
de:"+profe.getImpostoRenda() + "reais");

System.out.println();

break;

case 'A': System.out.println("Informe o nome do Administrador: ");
adm.setNome(new Scanner(System.in).nextLine());
System.out.println("Informe o número CPF: ");
adm.setCpf(new Scanner(System.in).nextLine());
System.out.println("Informe o salário em reais: ");
adm.setSalario(new Scanner(System.in).nextDouble());
System.out.println("Informe a quantidade de dependentes: ");
adm.setDependentes(new Scanner(System.in).nextInt());

System.out.println();

System.out.println("A seguir os dados informados:");
System.out.println();

System.out.println("Nome: "+adm.getNome());
System.out.println("CPF: "+adm.getCpf());
System.out.println("Salário:" +adm.getSalario() + "reais");
System.out.println("Quantidade de dependentes: "+adm.getDependentes());
System.out.println("Valor da contribuição sindical:
"+adm.getControbuicaoSindical() + "reais");

System.out.println();

System.out.println();

break;

default: System.out.println("Repita a operação, pois esta é inválida");

}

}

}

public class Funcionario {

private String nome;
private String cpf;
private double salario;

public double getGratificacao(){
return 0;
}

public String getNome() {
return nome;
}

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

public String getCpf() {
return cpf;
}

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

```



```
public double getSalario() {  
    return salario;  
}  
  
public void setSalario(double salario) {  
    this.salario = salario;  
}  
  
}  
  
public class Administrativo extends Funcionario {  
    private int dependentes;  
  
    public double getControbuicaoSindical(){  
        return getSalario()*0.01;  
    }  
  
    public int getDependentes() {  
        return dependentes;  
    }  
    public void setDependentes(int dependentes) {  
        this.dependentes = dependentes;  
    }  
}
```

```
public class Professor extends Funcionario{  
    private String titulo;  
  
    public double getImpostoRenda(){  
        return getSalario()*0.23;  
    }  
    public String getTitulacao() {  
        return titulo;  
    }  
  
    public void setTitulacao(String titulacao) {  
        this.titulo = titulo;  
    }  
}
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#) ▾



Re: Atividade Avaliativa 3
por [CLEITON ANTÔNIO GONÇALVES](#) - sexta, 11 set 2020, 19:48

Atividade 1 Aula 6
=====

Main.java

```
public class Main {  
  
    public static void main(String[] args){  
  
        Pessoa p = Visao.lerPessoa();  
        Visao.imprimir(p);  
    }  
  
}
```

=====

Pessoa.java

```
public class Pessoa {  
  
    public Pessoa(){
```

```
}

public Pessoa(String nomeAluno, int codigo, double salario) {
    setNome(nome);
    setCodigo(codigo);
    setSalario(salario);
}

@Override
public String toString(){
    return "\n Nome : "+getNome()+"\t Código :" + getCodigo() + "\t Salário
    :" + getSalario() + "\t Classe Social : " + getClasseSocial();
}

private String nome;
private int codigo;
private double salario;

// Classe
// A1: inclui as famílias com renda mensal maior que R$ 14.400
// Classe A2: maior que R$ 8.100
// Classe B: maior que R$ 4.600
// Classe C: maior que R$ 2.300
// Classe D: maior que R$ 1.400
// Classe E: maior que R$ 950
// Classe F: maior que R$ 400

public String getClasseSocial(){
    if(getSalario() > 14400){
        String tipo = "A1";
        return tipo;
    }else if(getSalario() > 8100){
        String tipo = "A2";
        return tipo;
    }else if(getSalario() > 4600){
        String tipo = "B";
        return tipo;
    }else if(getSalario() > 2300){
        String tipo = "C";
        return tipo;
    }else if(getSalario() > 1400){
        String tipo = "D";
        return tipo;
    }else if(getSalario() > 950){
        String tipo = "E";
        return tipo;
    }
    String tipo = "F";
    return tipo;
}

public String getNome() {
    return nome;
}

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

public int getCodigo() {
    return codigo;
}

public void setCodigo(int codigo) {
    this.codigo = codigo;
}
```



```
}

public double getSalario() {
    return salario;
}

public void setSalario(double salario) {
    this.salario = salario;
}
=====
Imprimir.java

import java.util.Scanner;

@SuppressWarnings("all")
public class Imprimir {

    public static int lerInt(){
        return new Scanner(System.in).nextInt();
    }

    public static int lerInt(String msg){
        System.out.println(msg);
        return lerInt();
    }

    public static int lerInt(String msg, String erro, int min, int max){
        int valor;
        boolean errado;
        do{
            valor = lerInt(msg);
            errado = valor < min || valor > max;
            if(errado){
                System.out.println(erro);
            }
        }while(errado);
        return valor;
    }

    public static double lerDouble(){
        return new Scanner(System.in).nextDouble();
    }

    public static double lerDouble(String msg){
        System.out.println(msg);
        return lerDouble();
    }

    public static double lerDouble(String msg, String erro, double min, double max){
        double valor;
        boolean errado;
        do{
            valor = lerDouble(msg);
            errado = valor < min || valor > max;
            if(errado){
                System.out.println(erro);
            }
        }while(errado);
        return valor;
    }

    public static String lerString(){
        return new Scanner(System.in).nextLine();
    }

    public static String lerString(String msg){
        System.out.println(msg);
        return lerString();
    }
```



```

}

public static char lerChar(){
    return lerString().charAt(0);
}

public static char lerChar(String msg){
    System.out.println(msg);
    return lerChar();
}

}

=====
Visao.java
public class Visao {

    public static Pessoa lerPessoa(){
        String nome = Imprimir.lerString("Informe o Nome : ");
        int codigo = Imprimir.lerInt("Informe o Código : ");
        double salario = Imprimir.lerDouble("Informe o Salário : ", "Ops Salário Inválida",
            0, Double.MAX_VALUE);
        Pessoa p = new Pessoa();
        p.setNome(nome);
        p.setCodigo(codigo);
        p.setSalario(salario);
        return p;
    }

    public static void imprimir(Pessoa p){
        System.out.println("##### Informações da Pessoa");
        System.out.println("#####");
        System.out.println(p.toString());
    }
}

```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [JOÃO GABRIEL MOREIRA LACERDA](#) - sexta, 11 set 2020, 20:08

Parabéns.

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [JOÃO GABRIEL MOREIRA LACERDA](#) - sexta, 11 set 2020, 20:07

Atividade 2,Aula 5

```

class Figura {

    private double base;
    private double altura;

    double getBase() {
        return base;
    }

    void setBase(double base) {
        this.base = base;
    }

    double getAltura() {
        return altura;
    }
}

```

```

}

void setAltura(double altura) {
this.altura = altura;
}

}

// Retangulo.java
package a502;

class Retangulo extends Figura {

double getArea() {
return (getBase() * getAltura());
}

}

// TrianguloRetangulo.java
package a502;

class TrianguloRetangulo extends Figura {

double getArea() {
return (getBase() * getAltura() / 2);
}

double getHipotenus() {
return (Math.sqrt(Math.pow(getBase(), 2) + Math.pow(getAltura(), 2)));
}

}

//Programa.java
package a502;

class Programa {

public static void main(String[] args) {

Retangulo retangulo = new Retangulo();
TrianguloRetangulo trianguloRetangulo = new TrianguloRetangulo();

// inserindo valores arbitrários para o retângulo
retangulo.setBase(4.0);
retangulo.setAltura(5.0);

// inserindo valores arbitrários para o triângulo retângulo
trianguloRetangulo.setBase(3.0);
trianguloRetangulo.setAltura(4.0);

// mostrando resultados no console
System.out.println("\nRETANGULO");
System.out.println("Area: " + retangulo.getArea());
System.out.println("\nTRIANGULO RETANGULO");
System.out.println("Area: " + trianguloRetangulo.getArea());
System.out.println("Hipotenusa: " + trianguloRetangulo.getHipotenus());

}
}

```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [DANIEL SOARES SEABRA](#) - sexta, 11 set 2020, 22:13



Atividade : Um matemático trabalha com dois tipos de figuras: os triângulos retângulos e os retângulos. Crie um programa Java (classes) que represente a modelagem para abstrair este problema. Ambos têm dois dados comuns: Base e Altura. Cada uma tem um cálculo de forma distinta de área. O triângulo retângulo é base X altura / 2, já do retângulo é apenas base X altura. Um cálculo diferente somente para o triângulo retângulo deve ser realizado, que é a hipotenusa. Crie um programa de teste, que instancie objetos, atribua valores e execute para apresentação

```
package Aula05Exerc02;

public abstract class Figura_Geometrica {

    protected double h; // h = altura
    protected double b; // b = base
    public abstract double area();
}

package Aula05Exerc02;

public class retangulo extends Figura_Geometrica{

    @Override
    public double area() {

        return h*b;
    }

    public double getBase() {
        return b;
    }

    public void setBase(double b) {
        this.b = b;
    }

    public double getAltura() {
        return h;
    }

    public void setAltura(double altura) {
        this.h = h;
    }
}

package Aula05Exerc02;

public class triangulo_Retangulo extends Figura{
```



```

@Override
public double area() {
    return ((h*b)/2);
}

public double hipotenus() {
    return Math.sqrt((b*b) + (h*h));
}

public double getBase() {
    return b;
}

public void setBase(double base) {
    this.b = b;
}

public double getAltura() {
    return h;
}

public void setAltura(double altura) {
    this.h = h;
}

package application;
import entidades.retangulo;
import entidades.triangulo_Retangulo;

public class Program {
    public static void main(String[] args) {
        triangulo_Retangulo tr1 = new triangulo_Retangulo();
        tr1.setBase(8.0);
        tr1.setAltura(12.0);
        System.out.print(tr1.area());

        retangulo r1 = new retangulo();
        r1.setBase(2.0);
        r1.setAltura(6.0);
        System.out.print(r1.area());

        System.out.print(tr1.hipotenus());
    }
}

```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [JOAO BENICIO STRAEHL DE SOUSA](#) - sexta, 11 set 2020, 22:14

Atividade 2 - Aula 5

GEOMETRIA DO TRIANGULO

```
import java.util.Scanner;
public class Area{
public static void main(String [] args){

double a,b,c ,Area,perimetro,semip, ;

Scanner entrada=new Scanner(System.in);

System.out.println("Digite a");
a=entrada.nextDouble();
System.out.println("Digite b");
b=entrada.nextDouble();
System.out.println("Digite c");
c=entrada.nextDouble();
if( (a
perimetro=a+b+c;

semip=perimetro/2;
Area=Math.sqrt(semip*(semip-a)(semip-b)(semip-c));
System.out.println(" perimetro= "+perimetro);
System.out.println("Area= "+Area);
}
else{
System.out.println("Medidas tornam o triangulo inexistente");
}

}
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3

por [GABRIEL DOREA DE MELO](#) - sexta, 11 set 2020, 22:24

atividade 1 da aula 5

```
public class Programa {
public static void main (String args[]) {

TrianguloRetangulo trianguloRet = new TrianguloRetangulo();
Retangulo retangulo = new Retangulo();

trianguloRet.setBase(3.0);
trianguloRet.setAltura(4.0);
System.out.println("Área Triângulo Retângulo: " + trianguloRet.getArea());
System.out.println("Hipotenusa: " + trianguloRet.getHipotenusa());

retangulo.setBase(4.0);
retangulo.setAltura(5.0);
System.out.println("Área Retângulo: " + retangulo.getArea());
}

}

class Figura {
private double base;
private double altura;

public void setBase (double base) {
this.base = base;
}
public double getBase() {
return base;
}
public void setAltura (double altura) {
```

```
this.altura = altura;
}
public double getAltura() {
return altura;
}
public double getArea() {
return 0.0;
}

class Triangulo extends Figura {
public double getArea() {
return (this.getBase() * this.getAltura() / 2.0);
}
}

class TrianguloRetangulo extends Triangulo{
public double getHipotenusa() {
return Math.sqrt(this.getBase() * this.getBase() + this.getAltura() *
this.getAltura());
}
}

class Retangulo extends Figura {
public double getArea() {
return (this.getBase() * this.getAltura());
}
}
```

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



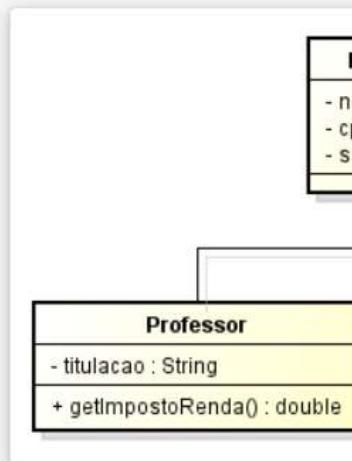
Re: Atividade Avaliativa 3

por [RAYANE MACIEL RIBEIRO](#) - sexta, 11 set 2020, 22:35

Boa noite, vou responder a questão abaixo:

Atividade 01

Em uma universidade, há dois tipos de funcionários: professor e administrativo.



Crie um programa Java que implemente o projeto da figura acima (nome, cpf e salário). Crie uma classe Professor, que é herdeira da classe Administrativo, que também é Funcionário e tem o método getSalario(). Crie também a classe Administrativo, que retorna o cálculo do seu imposto de renda por intermédio do método getImpostoRenda(), que retornará o valor de 0,05 de imposto de renda.

Seu programa deve perguntar qual tipo de Funcionário deseja ser criado. Dependendo da opção escolhida, o programa deve receber os dados do respectivo objeto e exibir o resultado.



Super classe funcionario:

```
package atividade01;

public class Funcionario {

    private String nome;
    private String cpf;
    private double salario;

    public Funcionario() {

    }

    public Funcionario(String nome, String cpf, double salario) {
        super();
        this.nome = nome;
        this.cpf = cpf;
        this.salario = salario;
    }
}
```

```
public String getNome() {
    return nome;
}
public void setNome(String nome) {
    this.nome = nome;
}
public String getCpf() {
    return cpf;
}
public void setCpf(String cpf) {
    this.cpf = cpf;
}
public double getSalario() {
    return salario;
}
public void setSalario(double salario) {
    this.salario = salario;
}

}
```



Classe Professor:

```
package atividade01;

public class Professor extends Funcionario {

    private String titulacao;

    public String getTitulacao() {
        return titulacao;
    }

    public void setTitulacao(String titulacao) {
        this.titulacao = titulacao;
    }

    public double getImpostoRenda(double salario) {
        return salario * 0.23;
    }

    @Override
    public String toString() {
        return "Professor [titulacao=" + titulacao + ", getTitulacao()=" + getTitulacao()
            + ", getNome()=" + getNome()
            + ", getCpf()=" + getCpf() + ", getSalario()=" + getSalario() + ", getClass()=" +
            getClass()
            + ", hashCode()=" + hashCode() + ", toString()=" + super.toString() + "]";
    }
}
```

Classe Administrativo:

```

package atividade01;

public class Administrativo extends Funcionario{

private int dependentes;

public int getDependentes() {
return dependentes;
}

public void setDependentes(int dependentes) {
this.dependentes = dependentes;
}

public double getContribuicaoSindical(double salario) {
return salario * 0.1;
}

@Override
public String toString() {
return "Administrativo [dependentes=" + dependentes + ",
getDependentes()=" + getDependentes() + ", getNome()="
+ getNome() + ", getCpf()=" + getCpf() + ", getSalario()=" + getSalario() + ",
getClass()=" + getClass()
+ ", hashCode()=" + hashCode() + ", toString()=" + super.toString() + "]";
}
}

```



Classe de teste:

```

-----
-----
-----



package atividade01;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class teste {

public static void main(String[] args) {

int opcao; //variável de controle do menu de seleção

List<Professor> professores = new ArrayList<Professor>(); //lista de
Professores
List<Administrativo> administrativos = new ArrayList<Administrativo>();
//lista do administrativo

Scanner ler = new Scanner(System.in);

do {

    /**
     * Menu de seleção*
     */
    System.out.println("\n \nQual tipo de funcionário deseja informar?\nP -
    Professor \nA - Administrativo"
    + "\n3 - Para Sair digite 3 a qualquer momento");

```

```
opcao = Integer.parseInt(ler.nextLine());

switch(opcao){

    case 1:
        Professor professor = new Professor();

        System.out.println("Informe o Nome do Professor:");
        professor.setNome(new Scanner(System.in).nextLine());

        System.out.println("Informe a CPF:");
        professor.setCpf(new Scanner(System.in).nextLine());

        System.out.println("Informe o Salario:");
        professor.setSalario(new Scanner(System.in).nextDouble());

        System.out.println("Informe a titulação do professor:");
        professor.setTitulacao(new Scanner(System.in).nextLine());

        professor.getImpostoRenda(professor.getSalario());

        professores.add(professor);

        System.out.println(professor);

        break;

    case 2:
        Administrativo administrativo = new Administrativo();

        System.out.println("Informe o Nome do Administrador:");
        administrativo.setNome(new Scanner(System.in).nextLine());

        System.out.println("Informe a CPF:");
        administrativo.setCpf(new Scanner(System.in).nextLine());

        System.out.println("Informe o Salario:");
        administrativo.setSalario(new Scanner(System.in).nextDouble());

        System.out.println("Informe a quantidade de dependentes:");
        administrativo.setDependentes(new Scanner(System.in).nextInt());

        administrativo.getContribuicaoSindical(administrativo.getSalario());

        administrativos.add(administrativo);

        break;

    case 3:
        System.out.println("Finalizando cadastro, os dados serão impressos a seguir");
        break;

    default:
        System.out.println("ERRO - Você informou uma entrada incorreta, informe os
dados novamente");

}

}while (opcao!=3);

new Thread();
try {
    Thread.sleep(5000);
} catch (InterruptedException e) {
    e.printStackTrace();
}

System.out.println("\nLista de professores cadastrados: \n"+ professores);
```



```
System.out.println("\n-----\n");
System.out.println("\nLista de Administradores cadastrados: "+  
administrativos);
}  
  
}
```

Qual tipo de funcionário deseja informar?
P - Professor
A - Administrativo
3 - Para Sair digite 3 a qualquer momento
1

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)



Re: Atividade Avaliativa 3
por [ROSANA CAVALCANTE DE OLIVEIRA](#) - sábado, 12 set 2020, 22:45

Olá Pessoal, as participações foram analisadas e as notas lançadas, caso tenham dúvidas me mandem uma mensagem ou postem no [fórum de dúvidas](#). Bons Estudos!

[Link direto](#) [Marcar como lida](#) [Mostrar principal](#)

◀ Chat 3

Seguir para...

Material de Apoio 3 ►



Você acessou como FERNANDO SANTOS FERREIRA (Sair)
Resumo de retenção de dados
Obter o aplicativo para dispositivos móveis

