

Ex1. Faça um programa que receba como argumentos pela linha de comando dois números inteiros e calcule o máximo divisor comum entre estes dois números.

Neste primeiro exercício (apenas neste exercício), o programa todo ficará contido no método *main()*. Trate as exceções de entrada e conversão (mesmo que não sejam verificadas). Critique os dados de entrada.

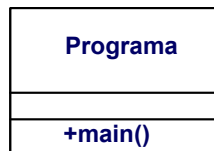


Figura1. Diagrama de classe

Dica: Use o algoritmo de Euclides.

Exemplo:

```
>java Divisores 20 25  
MDC (20,25) = 5
```

Ex2. Crie uma classe *Angulo* que deverá ter o seguintes **métodos de classe**:

- `converteAngulo` que recebe como parâmetro um valor do tipo *double* que é a medida em graus de um ângulo e retorna um valor do tipo *double* que é a medida deste ângulo em radianos.
- `funcaoSeno` que recebe como parâmetro um ângulo e retorna um valor do tipo *double* que é o seno deste ângulo.
- `funcaoCoseno` que recebe como parâmetro um ângulo e retorna um valor do tipo *double* que é o coseno deste ângulo.
- `funcaoTangente` que recebe como parâmetro um ângulo e retorna um valor do tipo *double* que é a tangente deste ângulo.
- `funcaoCotangente` que recebe como parâmetro um ângulo e retorna um valor do tipo *double* que é a cotangente deste ângulo.

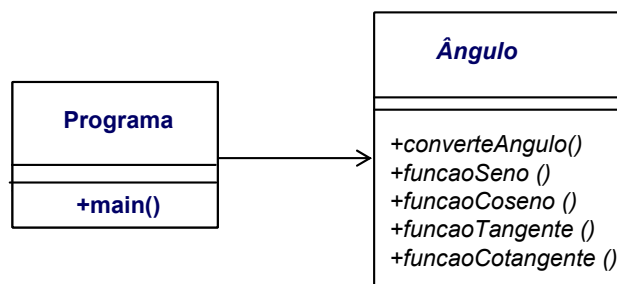


Figura2. Diagrama de classe

Crie uma classe para o programa principal, com o método **main()**, que leia através de um *fluxo de entrada* (a entrada de uma *String* vazia encerra a leitura de valores) a medida em graus de um ângulo, e utiliza os métodos da classe *Angulo* para convertê-lo para radianos e calcular o valor de suas funções trigonométricas, imprimindo estes valores. Trate as exceções de entrada (exceções de E/S e passagem de argumentos inválidos).

Ex3. Crie a classe *AnguloObj*, que tem papel semelhante a da classe *Angulo* do exercício anterior com as seguintes modificações (o objetivo é comparar os dois estilos de arquitetura):

- A classe possui o campo protegido (encapsulado) *arcoRad* que é a medida em radianos de um ângulo.
- A classe deverá ter um construtor que recebe um valor do tipo *double*, que é a medida de um ângulo em graus, e o converte para radianos.
- Seus métodos (os mesmos listados para a classe *Angulo*) agora devem ser **métodos de instância**, e não recebem parâmetros (obs: não recebem parâmetros neste exercício – “não receber parâmetros” não caracteriza métodos de instância).
- A classe *anguloObj* também implementa o método *toString()* que retorna uma instância da classe *String* na seguinte forma:

Arco: <medida em radianos do ângulo> rad

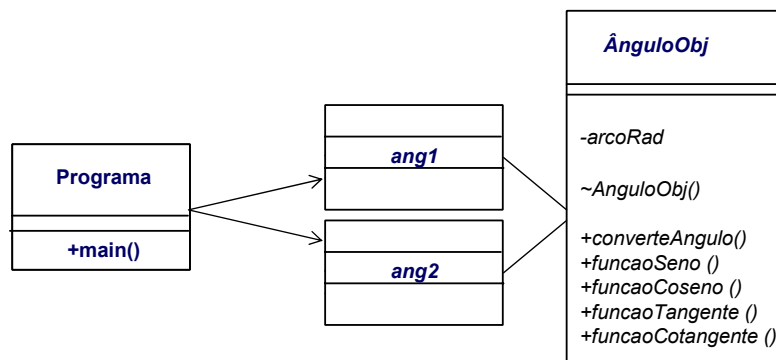


Figura3. Diagrama de classe

O programa principal deve criar duas instâncias da classe *AnguloObj*, lendo do teclado via *stream* a medida dos ângulos, calcular suas funções trigonométricas e exibir o resultado. As exceções de entrada devem ser tratadas convenientemente.

Exemplo:

```

java AnguloObjMain
Digite a medida em graus do primeiro Ângulo:
90
Digite a medida em graus do segundo Ângulo:
60
Arco: 1.5707963267948966
Seno: 1.0
Coseno: 6.123233995736766E-17
Tangente: 1.633123935319537E16
Cotangente: 6.123233995736766E-17

Arco: 1.0471975511965976
  
```

Seno: 0.8660254037844386 Coseno: 0.5000000000000001 Tangente: 1.7320508075688767 Cotangente: 0.577350269189626

Ex4. Crie a classe *Pessoa* com os campos protegidos (encapsulados), *nome* e *dataNascimento*, objetos da classe *String*, que vão representar o nome e data de nascimento. A classe *Pessoa* deve conter:

- Um construtor que recebe como parâmetros duas *strings* e inicializa os campos *nome* e *dataNascimento*.
- O método *toString*, que não recebe parâmetros e retorna um objeto da classe *String* na seguinte forma:

Nome: <nome da pessoa>
Data de Nascimento: <data de nascimento da pessoa>

Crie a classe abstrata *PessoaIMC* que herde da classe *Pessoa* e contenha tenha os campos protegidos *peso* e *altura*, ambos do tipo *double*. O construtor desta classe deve receber como parâmetros duas *strings* e dois valores do tipo *double* e inicializar os campos *nome*, *dataNascimento*, *peso* e *altura*. A classe *PessoaIMC* deve conter os seguintes métodos:

- *public double getPeso()* que retorna o peso;
- *public double getAltura()* que retorna a altura;
- *calculaIMC()* que recebe como parâmetros dois valores do tipo *double* que são a altura e o peso e retorna um valor do tipo *double* correspondente ao IMC (Índice de Massa Corporal = peso / altura ao quadrado) calculado.
- o método abstrato *resultIMC()* que não recebe parâmetros e retorna uma instância da classe *String*. (o método não é implementado nesta classe - **abstrato**)
- O método *toString()* desta classe deve retornar uma string da seguinte forma:

Nome: <nome da pessoa>
Data de Nascimento: <sua data de nascimento>
Peso: <seu peso>
Altura: <sua altura>

Crie as classes *Homem* e *Mulher*, herdeiras de *PessoaIMC*. Cada uma deve implementar o método abstrato *resultIMC()* para realiza o calculo do IMC e exibe uma mensagem de resultado acordo com o valor obtido.

Para Homem: IMC < 20.7 : Abaixo do peso ideal 20.7 < IMC < 26.4: Peso ideal IMC > 26.4 : Acima do peso ideal	Para Mulher: IMC < 19 : Abaixo do peso ideal 19 < IMC < 25.8: Peso ideal IMC > 25.8 : Acima do peso ideal
---	--

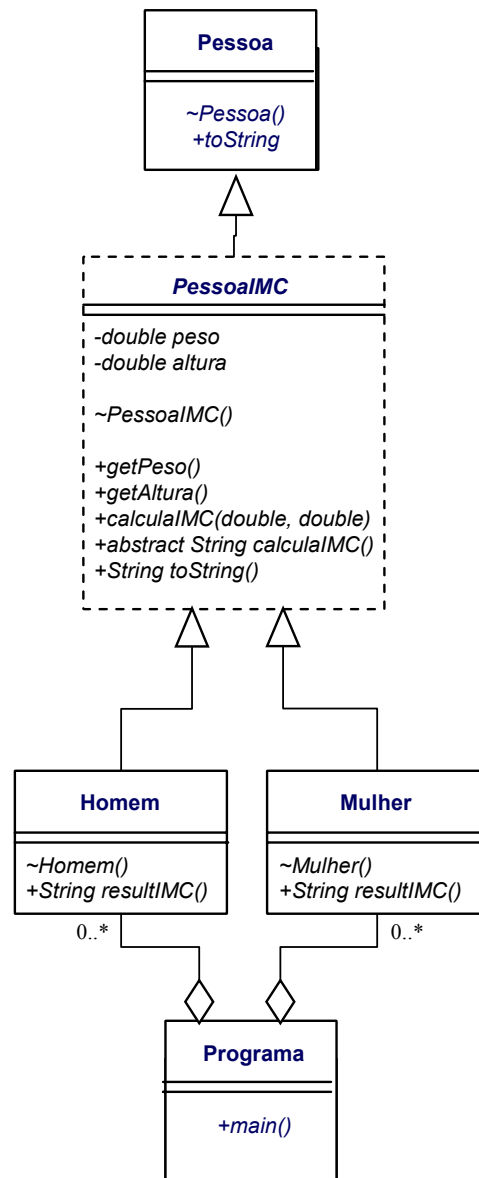


Figura4. Diagrama de classe

Crie uma classe para o programa principal, com o método *main()*, que crie instâncias das classes *Homem* e *Mulher* e armazene essas instâncias em um objeto array do tipo *PessoaIMC*. O programa deve perguntar ao usuário o tamanho do **array**, que tipo de objeto (Homem ou Mulher) deseja criar e os dados referentes a cada objeto. A leitura de dados deve ser feita através de fluxo de entrada. Após o armazenamento de todos os objetos, o programa deve ler cada posição do *array*, imprimindo os dados do objeto ali contido e calculando seu IMC.

Exemplo:

```

java PesoM
Digite o numero de pessoas:
2
Inserir homem (h) ou mulher(m)?
j
--- Opcao Invalida!!!
Inserir homem (h) ou mulher(m)?
h
  
```

```
Digite o nome:
Zezinho
Digite a data de nascimento:
01/01/1901
Digite o peso:
64.8
Digite a altura (em metros):
um m
--- A altura deve ser um numero real!!!
Digite a altura (em metros):
1.80
Inserir homem (h) ou mulher(m)?
m
Digite o nome:
Mariazinha
Digite a data de nascimento:
02/02/02/1902
Digite o peso:
64.8
Digite a altura (em metros):
1.8
-----
Nome: Zezinho
Data de Nascimento: 01/01/1901
Peso: 64.8
Altura: 1.8
IMC: 19.999999999999996  Abaixo do peso
-----
-----
Nome: Mariazinha
Data de Nascimento: 02/02/02/1902
Peso: 64.8
Altura: 1.8
IMC: 19.999999999999996  Peso ideal
-----
```