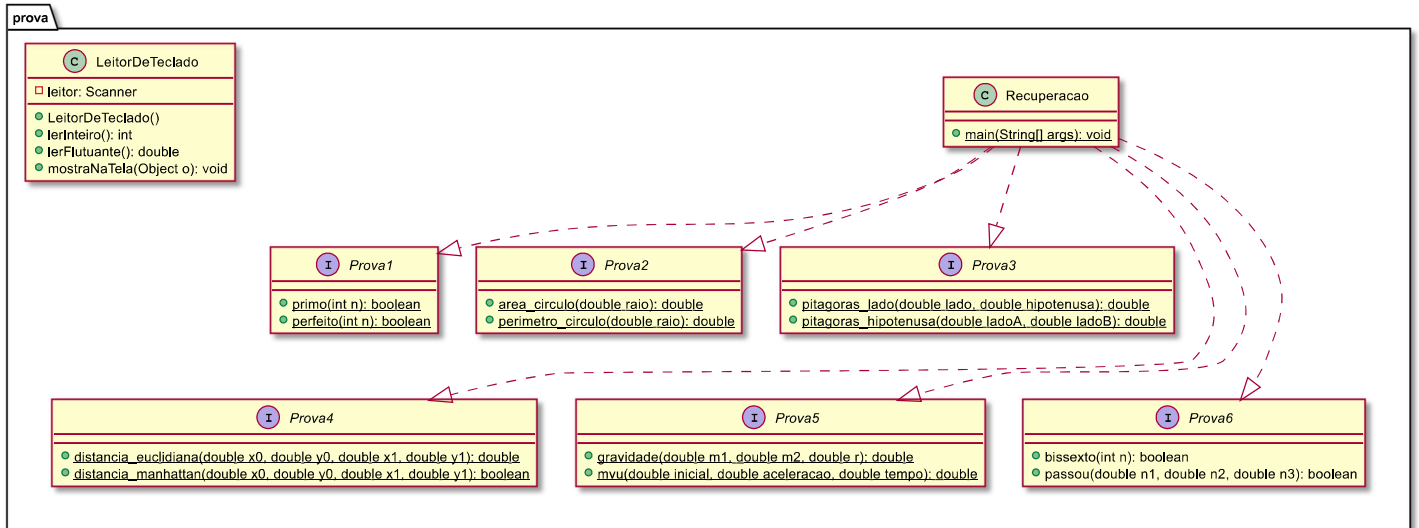


Recuperação



Você recebeu o seguinte diagrama de classe e deve codificá-lo em Java de acordo com seu número de prova. A classe **Recuperacao** **não** deve implementar todas as interfaces Prova1, Prova2, Prova3, Prova4, Prova5, Prova6. **Recuperacao** deve implementar apenas a interface cujo o número de prova foi sorteado para você, aluno.

Atenção: A implementação da classe **LeitorDeTeclado** é **obrigatória**. Toda a leitura de dados do teclado ou exibição de dados na tela deve ser feito usando essa classe.

Classe LeitorDeTeclado

Você deve:

- Instanciar um objeto Scanner([System.in](#)); no construtor de LeitorDeTeclado e guardá-lo no atributo leitor.
- O método lerInteiro tem como retorno a saída do método nextInt do atributo leitor.
- O método lerFlutuante tem como retorno a saída do método nextDouble do atributo leitor.
- O método mostraNaTela realiza a chamada para o método println de System.out passando o mesmo parâmetro de entrada.

Prova1

Questão 1. Na matemática, um número primo é aquele que pode ser dividido somente por 1 (um); e por ele mesmo. Por exemplo, o número 7 é primo, pois pode ser dividido apenas pelo número 1 e pelo número 7. Implemente o método primo que retorna true se o parametro de entrada é primo, e false se o parâmetro de entrada não é primo.

Para testar seu método, você pode fazer o seguinte no método main.

```
LeitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.primo(5)); //A saída é true
l.mostraNaTela(Recuperacao.primo(3)); //A saída é true
l.mostraNaTela(Recuperacao.primo(4)); //A saída é false
l.mostraNaTela(Recuperacao.primo(21)); //A saída é false
l.mostraNaTela(Recuperacao.primo(23)); //A saída é true
```

Questao 2. Na matemática, um número perfeito é um número inteiro para o qual a soma de todos os seus divisores positivos próprios (excluindo ele mesmo); é igual ao próprio número. Por exemplo o número 6 é perfeito, pois 1+2+3 é igual a 6. Sua tarefa é escrever o método "perfeito" que retorne true se o parâmetro de entrada é um número perfeito, e retorna false se não for número perfeito.

Para testar seu método, você pode fazer o seguinte no método main.

```
LeitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.perfeito(6)); //A saída é true
l.mostraNaTela(Recuperacao.perfeito(5)); //A saída é false
l.mostraNaTela(Recuperacao.perfeito(28)); //A saída é true
```

Prova2

Questão 1. A fórmula para calcular a área de um círculo é: $area = \pi * raio^2$. Considerando para este problema que $\pi = 3.14159$. Efetue o cálculo da área no método `area_circulo`, elevando o valor de raio ao quadrado e multiplicando por π .

Para testar seu método, você pode fazer o seguinte no método `main`.

```
LeitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.area_circulo(2.0)); //A saída é 12.5664
l.mostraNaTela(Recuperacao.area_circulo(100.64)); //A saída é 31819.3103
l.mostraNaTela(Recuperacao.area_circulo(150.00)); //A saída é 70685.7750
```

Questao 2. A fórmula para calcular o perímetro de um círculo é: $perimetro = 2 * \pi * raio$. Considerando para este problema que $\pi = 3.1415$. Efetue o cálculo do perímetro no método `perimetro_circulo`.

Para testar seu método, você pode fazer o seguinte no método `main`.

Para testar seu método, você pode fazer o seguinte no método `main`.

```
LeitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.perimetro_circulo(2.0)); //A saída é 12.566
l.mostraNaTela(Recuperacao.perimetro_circulo(0.4)); //A saída é 2.5132
l.mostraNaTela(Recuperacao.perimetro_circulo(100)); //A saída é 628.3
```

Prova3

Questão 1. Em um triângulo retângulo (com um ângulo de 90°), a soma dos quadrados dos catetos (os “lados” que formam o ângulo reto); é igual ao quadrado da hipotenusa (a aresta maior da figura). Efetue o teorema de Pitágoras no método `pitagoras_lado`, de modo que retorne o valor de um lado.

Para testar seu método, você pode fazer o seguinte no método `main`.

```
LeitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.pitagoras_lado(3.0, 5.0)); //A saída é 4.0
l.mostraNaTela(Recuperacao.pitagoras_lado(3.0, 5.831)); //A saída é 5.0
l.mostraNaTela(Recuperacao.pitagoras_lado(6.0, 7.211)); //A saída é 4.0
```

Questao 2. Em um triângulo retângulo (com um ângulo de 90°), a soma dos quadrados dos catetos (os “lados” que formam o ângulo reto); é igual ao quadrado da hipotenusa (a aresta maior da figura). Efetue o teorema de Pitágoras no método `pitagoras_hipotenusa`, de modo que retorne o valor da hipotenusa.

Para testar seu método, você pode fazer o seguinte no método `main`.

```
LeitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.pitagoras_hipotenusa(3.0, 4.0)); //A saída é 5.0
l.mostraNaTela(Recuperacao.pitagoras_hipotenusa(3.0, 5.0)); //A saída é 5.831
l.mostraNaTela(Recuperacao.pitagoras_hipotenusa(6.0, 4.0)); //A saída é 7.211
```

Prova4

Questão 1. Em matemática, distância euclidiana (ou distância métrica) é a distância entre dois pontos, que pode ser provada pela aplicação repetida do teorema de Pitágoras. Aplicando essa fórmula como distância, o espaço euclidiano torna-se um espaço métrico. Faça o cálculo da distância euclidiana no método `distancia_euclidiana` seguindo a equação:

$$distancia = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$$

Para testar seu método, você pode fazer o seguinte no método `main`.

```
LeitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.distancia_euclidiana(3.0, 5.0, 2.0, 3.0)); //A saída é 2.23607
l.mostraNaTela(Recuperacao.distancia_euclidiana(15.0, 13.0, 12.0, 15.0)); //A saída é 3.60555
l.mostraNaTela(Recuperacao.distancia_euclidiana(3.0, 2.0, 5.0, 13.0)); //A saída é 11.18034
```

Questao 2. A geometria de taxi, também conhecida como geometria pombalina, desenvolvida por Hermann Minkowski no século XIX, é uma forma de geometria em que a métrica habitual da geometria euclidiana é substituída por uma métrica nova em que a distância entre dois pontos

é a soma das diferenças absolutas entre as suas coordenadas. A distância de Manhattan também conhecida como distância pombalina, distância de quarteirões ou distância de taxi. Recebeu esse nome pois define a menor distância possível que um carro é capaz de percorrer numa malha urbana reticulada ortogonal, tal como se encontram em zonas como Manhattan ou a Baixa Pombalina. Faça o cálculo da distância manhattan no método `distancia_manhattan` seguindo a equação $distancia = |(x_1 - x_0)| + |(y_1 - y_0)|$.

Para testar seu método, você pode fazer o seguinte no método `main`.

```
LeitorDeTeclado l = new LeitorDeTeclado();
eitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.distancia_manhattan(3.0, 5.0, 2.0, 3.0)); //A saída é 3.0
l.mostraNaTela(Recuperacao.distancia_manhattan(15.0, 13.0, 12.0, 15.0)); //A saída é 5.0
l.mostraNaTela(Recuperacao.distancia_manhattan(3.0, 2.0, 5.0, 13.0)); //A saída é 13.0
```

Prova5

Questão 1. A Lei da Gravitação Universal (teorizada por Isaac Newton) diz que todos os corpos atraem-se mutuamente. E, essa força de atração é proporcional às massas dos corpos envolvidos e inversamente proporcional ao quadrado da distância que os separa. Faça o cálculo da força gravitacional no método `gravidade` usando a equação $F = G \cdot \frac{m_1 \cdot m_2}{r^2}$ com G sendo aproximadamente $6.7 * 10^{-11}$.

Para testar seu método, você pode fazer o seguinte no método `main`.

```
LeitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.gravidade(3000.0, 4000.0, 1.0)); //A saída é 0.000804
l.mostraNaTela(Recuperacao.gravidade(5000.0, 3000.0, 0.1)); //A saída é 0.10049999999999998
l.mostraNaTela(Recuperacao.gravidade(5000.0, 10000.0, 0.01)); //A saída é 33.5
```

Questao 2. Movimento uniformemente variado é o movimento no qual a velocidade escalar varia uniformemente no decorrer do tempo. O movimento caracteriza-se por haver uma aceleração escalar constante e diferente de 0. Faça o cálculo da velocidade no método `mov` seguindo a equação $velocidade = velocidade_{inicial} + aceleracao * tempo$

Para testar seu método, você pode fazer o seguinte no método `main`.

```
LeitorDeTeclado l = new LeitorDeTeclado();
eitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.muv(0.0, 3.0, 10.0)); //A saída é 30.0
l.mostraNaTela(Recuperacao.muv(15.0, 5.0, 21.0)); //A saída é 120.0
l.mostraNaTela(Recuperacao.muv(12.0, 9.0, 51.0)); //A saída é 471.0
```

Prova6

Questão 1. Para saber se um ano é bissexto ele deve atender um dos seguintes casos:

- É um número divisível por 4, mas não é divisível por 100
- É um número divisível por 4, por 100 e por 400.

Escreva o método `bissexto` que retorna `true` se o parâmetro de entrada é um ano bissexto, e `false` caso contrário.

Para testar seu método, você pode fazer o seguinte no método `main`.

```
LeitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.bissexto(1964)); //A saída é true
l.mostraNaTela(Recuperacao.bissexto(2000)); //A saída é true
l.mostraNaTela(Recuperacao.bissexto(2001)); //A saída é false
```

Questao 2. No IFBA, a nota de aprovação na média é 6.0.

Faça o cálculo da média para 3 notas passadas por parâmetro de entrada e retorne `true` se a média é maior ou igual 6.0, e `false` caso contrário.

Para testar seu método, você pode fazer o seguinte no método `main`.

```
LeitorDeTeclado l = new LeitorDeTeclado();
LeitorDeTeclado l = new LeitorDeTeclado();
l.mostraNaTela(Recuperacao.passou(4.0, 5.0, 6.0)); //A saída é false
l.mostraNaTela(Recuperacao.passou(4.5, 5.5, 6.5)); //A saída é false
l.mostraNaTela(Recuperacao.passou(4.9, 8.3, 9.3)); //A saída é true
```