***Exercícios***

***Exercício 1 – Pede Argumento***

Faça com que um programa Java leia um argumento passado para a Máquina Virtual Java. O programa deve ser formado por uma única classe e contém apenas o método main(). A classe deve se chamar PedeArgumento e você passará o seu nome como argumento. Por exemplo, se o seu nome é Nicolas, você deve executar o programa através do seguinte comando:

|  |
| --- |
| C:/> java PedeArgumento Nicolas  O programa deve exibir na tela como resultado:  > Olá Nicolas! |

*Dica:* Todo o parâmetro passado para a JVM é recebido pela array de String do método main(). Para acessar um argumento de uma array, forneça a posição da array que você deseja acessar. Por exemplo, como nós fornecemos apenas uma String, ela estará na primeira posição da array, que é zero: String nome=args [0]; Se você digitasse duas strings (por exemplo, o seu nome e sobrenome, seria necessário acessar as duas primeiras posições da array: 0 e 1).

***Exercício 2: Fatorial***

O fatorial de um número é calculado multiplicando-se os valores de um até o valor especificado. Um número fatorial é representado pelo número do fatorial seguido de um ponto de exclamação. Por exemplo, o fatorial de 4 é representado por 4! e é igual a : 1 x 2 x 3 x 4 = 24. Faça um programa java que calcule a fatorial de 2, 4, 6 e 10.

Dica : Use o laço for.

***Exercício 3: Distância entre dois pontos***

Escreva uma classe Java que contenha apenas o método main(). Você deve passar como parâmetro para o método main() as coordenadas x e y de pontos. O método calcula a distância Euclidiana ente estes dois pontos e exibe o resultado na tela.

Dicas: Distância Euclidiana entre 2 pontos P e Q :



Para calcular o quadrado de um número, utilize o método pow da classe Math:

32=9 --> double res = Math.pow(3,2);

Para calcular a raiz quadrada de um número, utilize o método sqrt da classe Math:

raiz quadrada de 9 --> double res = Math.sqrt(9);

***Exercício 4: Criando a conta do João da Silva***

Implemente uma classe ContaCorrente. Implemente um método print() que exiba na tela os dados do dono da conta corrente na classe ContaCorrente. Crie outra classe que se chama AcessaContaJoao. No método main(), crie uma conta para o “João da Silva” com saldo = 1000. Exiba os dados da conta chamando o método print(). Faça um saque de 135. Exiba novamente os dados.

***Exercício 5: Classe Pessoa***

Implemente a classe Pessoa. Uma pessoa possui: um nome; uma data de nascimento; um endereço. Escreva um método print() que exiba na tela os dados da pessoa. No método main() da classe, crie dois objetos do tipo Pessoa, cada um com seus dados. Exiba na tela os dados da primeira pessoa, apenas acessando os seus dados e os dados da segunda pessoa chamando o método print().

***Exercício 6: Encapsulamento***

Agora que você aprendeu o conceito de encapsulamento, encapsule os atributos da classe Pessoa. Os dados devem possuir o modificador private, e seus valores podem ser acessados e modificados por métodos mutator (set...) e acessor (get...).

***Exercício 7: Herança***

Implemente a classe Aluno. A classe Aluno é subclasse da classe Pessoa. A classe Aluno, além de conter os dados de uma Pessoa, vai ter também uma nota e uma turma. Crie métodos acessor e mutator para os atributos nota e turma. Escreva um método print2() que chame o método print() da classe pai (Pessoa) que irá exibir nome, data de nasc. e endereço do aluno e exiba na tela a nota e a turma do aluno. No método public static void main, crie 1 objeto aluno e aplique uma nota. Exiba os dados.

***Exercício 8: Sobreposição***

Na classe Aluno (criada no exercício anterior), faça a sobreposição do método print() da classe pai (classe Pessoa), substituindo o método print2().

***Exercício 9: Agregação***

Implemente a classe Ponto. Um ponto tem dois atributos do tipo inteiro: As coordenadas x e y. Além disso, um ponto também possui os métodos "construtor" (inicializa as coordenadas x e y com os valores passados como parâmetro para o método), "setaCoordenadas (int x, int y)" (recebe como parâmetro os novos valores das coordenadas) e "exibe()" (exibe os valores dos atributos x e y).

Implemente a classe Reta. Uma Reta possui 2 atributos do tipo Ponto: o ponto inicial e o ponto final da Reta. Uma Reta possui os métodos "construtor" (recebe 4 valores inteiros e inicializa as coordenadas x e y de cada ponto) e "exibe()" (exibe as coordenadas x e y de cada ponto).

Implemente a classe TestaReta que possui apenas o método main(). Dentro do método main(), crie um objeto do tipo Reta e inicialize o Ponto Inicial com x=2, y=3 e o Ponto Final com x=5, y=6. Chame o método exibe() da classe Reta.

*Nota:* Neste exemplo, podemos observar uma importante propriedade da orientação a objetos que é a agregação. A classe Reta tem um relacionamento do tipo agregação com a classe Ponto. Incluir objetos de classe como membros de outra classe é conhecido como agregação ou composição. A agregação permite a reutilização de código.

***Exercício 10: Classes Abstratas***

Implemente uma classe Hexagon. Essa classe também herda propriedades da classe abstrata Shape. A classe Hexagon possui apenas um dado que o comprimento do lado s (todos os lados tem comprimentos iguais) que é uma variável double. Implemente os métodos area() e circunference() do Hexagon:

A área de um hexágono é: 3/2\*Math.sqrt(3)\*s\*s.

A circunferência de um hexágono é: 6\*s.

Implemente o método main(), criando um hexágono de lado s=5. Mostre a área e circunferência deste hexágono na tela.

***Exercicio 11: StringTokenizer***

Faça um programa que busca palavras por palavra dentro de um texto e as imprima em maiúscula. Considere o texto uma string inicial.

***Exercício 12: Gerador de Frases***

Faça um programa que gere frases aleatoriamente. Uma frase é formada por uma combinação de um artigo + um substantivo + um verbo + uma preposição + um artigo + um substantivo, sendo que seus valores são lidos de arrays de String. As arrays devem ser povoadas com os seguintes valores:

|  |
| --- |
| String artigo[]={"o", "a", "um", "uma", "algum", "qualquer"};  String substantivo[]={"menino", "menina", "cachorro", "cidade", "carro"};  String verbo[]={"dirigiu", "pulou", "correu", "caminhou", "saltou"};  String preposicao[]={"para", "de", "acima de", "debaixo de", "sobre"}; |

Para formar uma frase, o programa deve escolher randomicamente um elemento da respectiva array. O texto deve gerar 10 frases diferentes.

*Dica:* Para gerar randomicamente um número inteiro, use o método nextInt(int) da classe Random. Como a classe Random pertence ao pacote java.util, você deve importá-lo. Você deve fornecer um parâmetro inteiro para o método, que é o valor máximo que pode ser gerado. Por exemplo, se você fornecer 5, ele vai gerar um número entre 0 (inclusive) e 5 (exclusive).

***Exercício 13: Array de Alunos (Dificuldade Fácil)***

Faça uma classe que contenha apenas um método main(). Nesse método, crie uma array de 5 posições do tipo Pessoa. Crie um objeto do tipo Aluno para cada posição da array. Especifique um nome, um endereço, uma data de nascimento, uma nota e uma turma para cada aluno. Imprima os dados de todos os alunos utilizando um laço for.

***Exercício 14: Classe Disciplina (Dificuldade Difícil)***

Faça a classe Disciplina, que é composta por um array de alunos e pelo nome da disciplina. Crie métodos para inserir alunos, remover alunos, listar todos os alunos e verificar o aluno mais velho.

***Exercício 15: Vector de Alunos***

Refaça o exercício 13 (Array de alunos) utilizando a classe Vector. Não esqueça de importar o pacote java.util.\*;