

ADS – IFPB – Campus Monteiro
LISTA SEMANAL – PROGRAMAÇÃO II - PROGRAMAÇÃO ORIENTADA A OBJETOS
Prof. Cleyton Caetano de Souza

Lista Semanal 2

1. Dada a classe CDF abaixo, escreva o código do método ePrimo (lê-se “é primo”), esse método recebe uma variável do tipo inteiro e retorna true se o número for primo e false se não for primo. Na matemática, um número é primo se ele tem apenas dois divisores, um e ele mesmo.

```
public class CDF {  
  
    public boolean ePrimo(int numero) {  
        |  
    }  
  
}
```

2. Crie um programa que servirá para testar o CDF da questão anterior. Comece solicitando ao usuário um número. Em seguida, use o objeto CDF para informar todos os números que são primos entre um e o número fornecido pelo usuário.

3. Escreva um programa onde o usuário vai informar uma quantidade indeterminada de valores. O usuário deverá sempre fornecer números no intervalo de zero a cem. Se o usuário fornecer um número fora desse intervalo, o programa deve informar que esse é um número inválido e ler um novo valor. Antes de cada leitura, o programa deve sempre perguntar se o usuário quer fornecer mais um número. Ao final, o programa deve informar quantas vezes cada número fornecido foi lido. Os números que não foram lidos não devem ser apresentados nessa contagem.

Console
Você deseja fornecer um número? Sim Informe o 1º número: 10 Você deseja fornecer um número? Sim Informe o 2º número: -4 Esse número não é válido Você deseja fornecer um número? Sim Informe o 2º número: 33 Você deseja fornecer um número? Sim Informe o 3º número: 11 Você deseja fornecer um número? Sim Informe o 4º número: 33 Você deseja fornecer um número? Não O número 10 foi lido 1 vez (es). O número 11 foi lido 1 vez (es). O número 33 foi lido 2 vez (es).

4. Crie a classe Contato, com os atributos nome, telefone e tipo. Os atributos nome e telefone serão Strings. O atributo tipo vai ser do tipo Enum **(você vai precisar pesquisar o**

que é um Enum e como usá-lo). Use para tipo um Enum com os valores **fixo** e **celular**. Siga as convenções de nomenclatura e crie os gets e sets para todos os atributos.

5. Crie um programa que começará pedindo os dados de contato de cinco colegas. Essas informações devem ser salvas em um vetor de Contatos. Em seguida, pergunte ao usuário um número e informe o nome e o telefone da pessoa guardada nessa posição do vetor. Caso o usuário forneça um número fora dos índices do vetor, peça para ele informar um novo número. Enquanto o usuário não digitar -1, o programa deve continuar a exibir o nome do contato guardado na posição informada pelo usuário.

Console

```
Digite o nome [0]: André
Digite o telefone [0]: 1111-1111
Digite o tipo [0]: fixo
Digite o nome [1]: Bianca
Digite o telefone [1]: 2222-2222
Digite o tipo [1]: fixo
Digite o nome [2]: Carlos
Digite o telefone [2]: 3333-3333
Digite o tipo [2]: celular
Digite o nome [3]: Daniela
Digite o telefone [3]: 4444-4444
Digite o tipo [3]: fixo
Digite o nome [4]: Elaine
Digite o telefone [4]: 5555-5555
Digite o tipo [4]: celular
---C-A-D-A-S-T-R-O---F-I-N-A-L-I-Z-A-D-O---
Digite uma posição: 4
Daniela, celular: 4444-4444
Digite uma posição: 2
Bianca, celular: 2222-2222
Digite uma posição: 9
Essa posição não é válida
Digite uma posição: 3
Carlos, fixo: 3333-3333
Digite uma posição: -1
---F-I-M---D-E---P-R-O-G-R-A-M-A---
```

6. Escreva a tabela de acompanhamento com o valor das variáveis x, y, z e w, ao final de cada linha, durante a execução do programa abaixo:

```

1 public class Programa {
2     public static void main(String[] args) {
3         int x = 10;
4         int y = x%4;
5         int z;
6         for(z = 1; x > y; y*=2)
7             z = x++ + z;
8         int w = x + y + z;
9         System.out.println(w);
10    }
11 }

```

#	x	y	z	w
3	10	0	0	0
4	10	?	0	0
5	?	?	0	0
6	?	?	1	0
7	?	?	?	?
6	?	?	?	?
?	?	?	?	?

7. Escreva um programa onde você lerá dois nomes fornecidos pelo usuário. Entretanto, o segundo nome fornecido não pode ser igual ao primeiro. **Enquanto** o usuário insistir em colocar o segundo nome igual ao primeiro, requisiite dele uma nova entrada para o segundo nome. Ao final, imprima os dois nomes em ordem alfabética. **Para isso, você precisará pesquisar sobre o método [compareTo](#) da classe String.**

Console I

Entrada 1: cachorro
Entrada 2: abacate
abacate, cachorro

Console II

Entrada 1: carro
Entrada 2: carro
As duas entradas devem ser diferentes
Entrada 2: carro
As duas entradas devem ser diferentes
Entrada 2: cachorro
cachorro, carro