



Disciplina	Turmas	Identificação da Tarefa	Prazo
Programação para Dispositivos Móveis	Manhã e Tarde	Entrega para Nota EN.A Segunda Lista de Exercícios – Java Básico Tarefa Individual	13/09/2020 23h59

### Instruções para Entrega Remota através do Teams

Prezados alunos, a entrega das tarefas será remota, através do recurso de Tarefas disponível no **Teams**.

Para tanto peço que leiam atentamente todos os itens a seguir e cumpram o que está solicitando.

1. Esta é a primeira entrega da disciplina. Eu a chamarei de Entrega EN.A;
2. A entrega será através do **Teams**;
3. Esta é uma tarefa individual;
4. São quatro programas a serem entregues nesta EN.A. Essas entregas devem ser feitas na forma de **arquivos separados**. Ou seja, obrigatoriamente, cada programa deverá ser um arquivo e peço que utilizem preferencialmente os nomes de classes e arquivos indicados aqui: Ex1.java, Ex2.java, Ex3.java, Ex4.java;
5. Entregar apenas o código fonte. O compilado não;
6. No código fonte de cada programa coloque no topo as linhas de comentário para identificação com as seguintes informações:  
/\* ILP506 – Turma Manhã(ou Tarde) – Nome: Zezinho de Oliveira \*/
7. **Atenção para o PRAZO LIMITE mostrado no cabeçalho deste documento.**  
Eu não vou bloquear entregas após o prazo. Porém, no Teams, fica registrado a data e a hora da entrega e aquelas que forem feitas fora do prazo são sinalizadas como atrasadas para mim.

### O que é a tarefa?

Esta tarefa é constituída pelos quatro programas a seguir.

1. Escreva um programa que leia um número inteiro **n** maior que 1 e mostre na tela se esse número é primo ou não. Lembrando: número primo é divisível apenas por 1 e por ele mesmo, ou seja, não é divisível por nenhum número entre 2 e **n-1**.
2. Escreva um programa Java que leia dois inteiros Min e Max. O valor de Min deve ser no mínimo 2. O valor de Max deve ser obrigatoriamente maior que Min e menor ou igual a 999999. O programa deve carregar um ArrayList de inteiros com todos os números primos existentes no intervalo fechado [Min, Max]. Ao final mostre na tela esses números primos e quantos foram encontrados. Para verificar se um número é primo crie um método que recebe n como parâmetro e retorna um boolean true se n for primo, ou false se n não for primo. O cabeçalho deste método está indicado abaixo.

```
private static boolean ePrimo(int n)
```

3. Escreva um programa que leia um número inteiro de 5 dígitos, ou seja, no intervalo fechado [10.000, 99.999]. Crie um método que receba esse número inteiro e calcule um dígito verificador segundo a regra a seguir. A regra de cálculo desse dígito verificador tem o nome de módulo 7. Exemplo: se o código for 21468, o dígito verificador será 3.

Em primeiro lugar deve-se tomar cada dígito individualmente e multiplicá-lo pelo peso apropriado. O peso do dígito menos significativo é 6, o peso do segundo dígito menos significativo é 5 e assim por diante, conforme mostrado na tabela. Os resultados das multiplicações devem ser somados e por fim calcula-se o resto dessa soma por 7

dígitos	2	1	4	6	8		
peso	2	3	4	5	6	soma	
multiplicação	4 + 3 + 16 + 30 + 48					= 101	Resto de 101 por 7 = 3

4. Escreva um programa que leia um número inteiro **n** e mostre na tela os **n** primeiros termos da sequência de Fibonacci. Essa sequência tem os dois primeiros termos 0 e 1, para os demais, cada termo é a soma dos dois anteriores.  
Exemplo: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, etc...  
(sugestão: use ArrayList<Integer> para praticar)