

LPOO1 - 2015/02 - IFPE

Linguagem de programação Orientada a Objetos - Plano da disciplina

Primeira lista de atividades de LPOO

Paulo Abadie Guedes, 10 September 2015 (created 6 September 2015)

no tags

Objetivo

1. O objetivo desta atividade é treinar o uso de tipos primitivos da linguagem e também de [Métodos estáticos](#).

Instruções e restrições

1. Todos os métodos desta lista devem ser **públicos** e **estáticos**, salvo orientação em contrário na própria questão.
2. Todos os nomes devem ser significativos e bem escolhidos.

Atividades

1. Crie um método que soma dois valores e retorna o seu resultado. Use valores inteiros.

Assinatura do método:

```
public static int soma(int a, int b);
```

2. Crie um método que soma três valores e retorna o seu resultado. Use um tipo diferente do método anterior. Exemplo de assinatura do método (use outro tipo):

```
public static long soma(long a, long b, long c);
```

3. Crie um método que soma três valores e retorna o seu resultado. Não use o operador de adição ('+').

Assinatura do método:

```
public static int soma(int a, int b, int c);
```

Use como apoio o método seguinte, de uma questão anterior.

```
public static int soma(int a, int b);
```

Note que aqui temos um exemplo de [Sobrecarga de método](#). Também é um exemplo de como podemos usar uma operação mais simples (soma de dois elementos) como ferramenta de suporte a operações mais complexas. É fundamental saber fazer a [Composição de operações](#).

4. Crie um método que determina se um dado valor é par ou não. Assinatura do método:

```
public static boolean verificaValorPar(int valor);
```

5. Crie um método que fatore um **Valor inteiro** e retorne todos os seus fatores em ordem crescente. Os fatores serão sempre [Números primos](#) naturais. Se um fator pode dividir o número várias vezes, ele deve aparecer várias vezes na resposta.

Exemplo: para a entrada **80**, a saída será: **[2, 2, 2, 2, 5]**.

Assinatura parcial do método:

```
public static TIPO_RETORNO fatoraValor(TIPO valor);
```

a. Responda nos comentários:

- i. Qual o tipo mais adequado para o parâmetro **valor**? substitua onde tem **TIPO**. Justifique.
- ii. Qual o tipo mais adequado para o **retorno** do método? substitua onde tem **TIPO_RETORNO**. Justifique.

6. Crie um método que retorna a sequência dos primeiros N **Números primos** naturais, em ordem crescente.
Assinatura do método:

```
public static long[] calculaPrimos(long quantidade);
```

7. Crie um método que faz a união de dois arrays de inteiros, criando um novo array.
- O tamanho do novo array será a soma dos tamanhos dos arrays recebidos como parâmetro.
 - O conteúdo do novo array será composto por uma cópia do conteúdo do primeiro array, seguido imediatamente do conteúdo do segundo array.
 - Se um dos parâmetros for nulo, ignore-o e empregue os dados do outro parâmetro. Se ambos forem nulos, retorne um array de tamanho zero.
 - Exemplo: A = [1,2,11]; B = [15, 16, 17, 18]. Saída = [1,2,11,15, 16, 17, 18].

Assinatura do método:

```
public static int[] calculaUniaoDeArrays(int[] primeiro, int[] segundo);
```

8. Crie um método que, dada uma sequência dos primeiros **Números primos** naturais, calcula os próximos N primos.
- Não precisa verificar se os valores fornecidos são realmente primos. Assuma que são primos e que estão em ordem crescente.
 - Os dados retornados no array devem ter, no início, os valores fornecidos no parâmetro **primos** e imediatamente depois, os novos valores calculados.

Assinatura do método:

```
public static long[] calculaMaisPrimos(long quantidade, long []primos);
```

9. Crie um método que faz a união de dois arrays de inteiros, criando um novo array.
- Assuma que os dados de cada entrada estarão em ordem crescente.
 - A saída irá conter a união das maiores sequências dos valores entre A e B, respeitando a ordem recebida.
 - Cada bloco de valores deve conter a quantidade máxima de itens entre os respectivos blocos de mesmo valor entre A e B.
 - Exemplo: A = [2,2,2,3,3,4,4,4]; B = [2,3,3,3,4,5,5]. Saída = [2,2,2,3,3,3,4,4,4,5,5].
 - Como A tem 3 itens com valor 2 (e B só tem uma ocorrência), a saída tem 3 itens com valor 2.

Assinatura do método:

```
public static long calculaUniaoDeFatores(long primeiroConjunto, long segundoConjunto);
```

10. Crie um método que calcula o MMC (**Mínimo múltiplo comum**) de dois valores e retorne.
- Determine o MMC através do método que usa uma lista de números primos.
 - Fatore cada número. Para cada fator em comum, descubra as maiores potências de seus termos.
 - Com base nestes dados, combine os termos e determine o MMC.
 - Use os métodos criados nas questões anteriores.

Assinatura do método:

```
public static long calculaMMCPorFatoracao(long valor1, long valor2);
```

Segundo a Wikipedia:

O mínimo múltiplo comum (mmc) de dois inteiros a e b é o menor inteiro positivo que é múltiplo simultaneamente de a e de b.

Se não existir tal inteiro positivo, por exemplo, se $a = 0$ ou $b = 0$, então $\text{mmc}(a, b)$ é zero por definição.

Fonte:

- https://pt.wikipedia.org/wiki/Mínimo_múltiplo_comum