

# Aprenda Java e Processo de software

*Baseado no Personal Software Process (PSP) para Engenheiros - Parte 1*

## Programa 2

### 1. Requisitos do software

Construa um programa para contar em linhas de código (LOC):

- tamanho total do programa,
- quantidade de cada parte do programa (quantidade de classes e de métodos),
- tamanho total de cada parte do programa (classe e métodos).

Não se preocupe em tratar todas as possíveis situações de um programa em Java. Afinal, para tanto seria necessário utilizar um analisador sintático do compilador Java, o que não é o esperado para este programa. Em relação à contagem de linhas de código, observe os seguintes requisitos:

- Uma linha de código deve ser contada se ele conter texto em uma linguagem de programação e tiver uma função no programa. Em outras palavras, uma linha que contenha apenas comentário ou que esteja em branco não deve ser contabilizada.
- Considere a criação de uma ou mais classes em cada arquivo com a extensão ".java", não sendo necessário tratar classes aninhadas (embora você possa fazê-lo, caso queira).
- Lembre-se que existe código da aplicação que não pertence a métodos que devem ser contabilizado como código da classe (por exemplo, declaração de atributos).
- Existe código da aplicação que não é declarado dentro da classe (por exemplo, declaração de pacote e comandos de importação de pacotes), mas que logicamente faz parte do código necessário à classe. Nesse caso, contabilize tal código como código da classe. Caso exista mais de uma classe definida no mesmo arquivo, contabilize tais linhas de código para cada classe contida no arquivo).
- Caso o programa tenha erros de sintaxe, não é necessário realizar a medição de tamanho do programa.

Teste adequadamente o seu programa. No mínimo, teste o programa contando o Programa 1. Um exemplo hipotético de resultado esperado seria o seguinte:

Nome do programa	Nome da classe	Nome da método	Tamanho
Programa 1			200
Programa 1	LinkedList		100
Programa 1	LinkedList	add	30
Programa 1	LinkedList	getNext	20
Programa 1	LinkedList	remove	40
Programa 1	Calculator		80
Programa 1	Calculator	average	30

Programa 1	Calculator	sampleStdDev	40
Programa 1	App		20
Programa 1	App	main	15

## 2. Processo de software: Visão geral

### 2.1. Propósito

Guiar o desenvolvimento de programas modulares.

### 2.2. Critérios de entrada

- Descrição do problema.
- Formulário de Resumo de Planejamento de Projeto.
- Registro de Tempo.
- Registro de Erros.
- **Padrão de estilo de codificação.**

### 2.3. Atividades

- **Planejar**
  - Produzir ou obter requisitos.
  - **Estimar o tempo necessário para desenvolvimento.**
  - Preencher os dados de planejamento no formulário de Resumo de Planejamento de Projeto.
  - Completar o Registro de Tempo.
- **Desenvolver**
  - Projetar o programa.
  - Implementar o projeto.
  - Compilar o programa, consertar e registrar todos os erros encontrados.
  - Testar o programa, consertar e registrar todos os erros encontrados.
  - Completar o Registro de Tempo.
- **Encerrar**
  - Completar o formulário de Resumo de Planejamento de Projeto com os dados de tempo efetivo, erros e tamanho.

### 2.4. Critérios de saída

- Programa rigorosamente testado.
- Formulário de Resumo de Planejamento de Projeto preenchido com dados estimados e efetivos.
- Registro de Tempo completo.
- Registro de Erros completo.

## 3. Processo de software: Planejar

### 3.1. Propósito

Guiar o processo de planejamento do PSP.

### 3.2. Critérios de entrada

- Descrição do problema.
- Formulário de Resumo de Planejamento de Projeto.
- Registro de Tempo.

### 3.3. Atividades

- **Analisar requisitos**
  - Produzir ou obter os requisitos para o programa.
  - Garantir que os requisitos estão claros e sem ambiguidade.
  - Resolver qualquer questão quanto aos requisitos.
- **Estimar recursos**
  - **Fazer sua melhor estimativa do tempo necessário para desenvolver o programa.**
  - **Preencher os dados de tempo estimado/planejado no formulário de Resumo de Planejamento de Projeto.**
  - **Usando como guia os dados totais de tempo consumido por fase do projeto (To Date %), distribuir o tempo total de desenvolvimento para cada fase do projeto do programa atual.**

### 3.4. Critérios de saída

- Requisitos documentados.
- Formulário de Resumo de Planejamento de Projeto preenchido com as estimativas de tempo de desenvolvimento.
- Registro de Tempo completo.
- Registro de Erros completo.

## 4. Processo de software: Desenvolver

### 4.1. Propósito

Guiar o desenvolvimento de programas pequenos.

### 4.2. Critérios de entrada

- Requisitos.
- Registro de Tempo.
- Registro de Erros.
- **Padrão de estilo de codificação.**

### 4.3. Atividades

- **Desenhar / Projetar**
  - Revisar os requisitos e produzir um projeto para atendê-los.
  - Registrar no registro de erros qualquer erro encontrado nos requisitos.
  - Registrar o tempo no Registro de Tempo.

- **Programar**
  - Implementar o projeto, **obedecendo o padrão de estilo de codificação.**
  - Registrar no Registro de Erros qualquer erro encontrado nos requisitos e no projeto.
  - Registrar o tempo no Registro de Tempo.
- **Compilar**
  - Compilar o programa até que não existam mais erros de compilação.
  - Corrigir todos os erros encontrados.
  - Registrar erros no Registro de Erros.
  - Registrar o tempo no Registro de Tempo.
- **Testar**
  - Testar o programa até que os casos de teste executem sem erros.
  - Corrigir todos os erros encontrados.
  - Registrar erros no Registro de Erros.
  - Registrar o tempo no Registro de Tempo.

#### 4.4. Critérios de saída

- Programa rigorosamente testado e **implementado de acordo com o padrão de estilo de codificação.**
- Registro de Tempo completo.
- Registro de Erros completo.

## 5. Processo de software: Encerrar

### 5.1. Propósito

Guiar o processo de encerramento do PSP.

### 5.2. Critérios de entrada

- Descrição do problema e requisitos.
- Formulário de Resumo de Planejamento de Projeto **com dados de tempo de desenvolvimento do programa.**
- Registro de Tempo.
- Registro de Erros.
- Programa testado e executável **que está em conformidade com o padrão de estilo de codificação.**

### 5.3. Atividades

- **Garantir a consistência de dados de erros**
  - Verificar que todos os erros encontrados em cada fase foram realmente registrados.
  - Verificar se os dados de cada erro no Registro de Erros estão corretos e completos.
  - Verificar se a quantidade de erros inseridos e removidos em cada fase é razoável e correta.

- Usando o melhor possível sua memória, corrigir qualquer dado omissos ou incorreto sobre os erros.
- **Garantir a consistência de dados de tempo**
  - Revisar o Registro de Tempo, procurando por erros e omissões.
  - Usando o melhor possível sua memória, corrigir qualquer dado omissos ou incorreto sobre os registros de tempo.
- **Medir tamanho do programa**
  - **Medir o tamanho do programa completo e de suas partes.**
  - **Determinar o tamanho do código base, apagado, adicionado, reutilizado (que não seja o código base) e total.**
  - **Preencher os dados de tamanho no Formulário de Resumo de Planejamento de Projeto.**

#### 5.4. Critérios de saída

- Programa rigorosamente testado **que está em conformidade com o padrão de estilo de codificação.**
- Formulário de Resumo de Planejamento de Projeto completo.
- Registro de Tempo completo.
- Registro de Erros completo.

## 6. Entrega

Quando você completar a fase de encerramento, envie os dados do pacote da tarefa, código fonte e resultados de teste para o professor.

O pacote de tarefa deve conter os seguintes a seguir, na ordem apresentada:

- Formulário de Resumo de Planejamento de Projeto,
- Registro de Tempo,
- Registro de Erros,
- Listagem do código fonte do Programa 2,
- Resultados dos testes.