


 <b>PCS 2302/2024</b> Laboratório de Fundamentos da Eng.de Computação	<b>PCS-2302 / PCS-2024</b> <b>Lab. de Fundamentos de Eng. de Computação</b>
	<b>Aula 04</b>  <b>Exercícios Turma 3</b>  <b>Professores:</b> Marcos A. Simplício Junior Paulo Sergio Muniz Silva

Aula 04:  
Introdução ao  
Paradigma de  
Objetos

Autores:  
Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2013 1

   <b>PCS 2302/2024</b> Laboratório de Fundamentos da Eng.de Computação	<b>Exercícios - Objetivos</b>
	<ul style="list-style-type: none"><li>Exercício 1: praticar os conceitos de herança e polimorfismo introduzidos na parte teórica</li><li>Exercício 2: praticar a interação entre o “hardware” da MVN (código em Java) e o software por ela executado (escrito em linguagem de montagem da MVN), para entender melhor o funcionamento do código Java que define a MVN</li></ul>

Aula 04:  
Introdução ao  
Paradigma de  
Objetos

Autores:  
Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2013 2



Aula 04:

Introdução ao  
Paradigma de  
Objetos

Autores:

Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2012

3

## Exercício 01

- Considere o domínio da geometria, onde polígonos representam figuras geométricas convexas com pelo menos 3 lados. A verificação de que um polígono é triângulo ou quadrilátero pressupõe avaliar o número de lados e uma propriedade geométrica que garanta convexidade. Triângulos podem ser categorizados dependendo das medidas de seus lados ou ângulos internos. O mesmo vale para quadriláteros.
- Este domínio é bastante adequado para treinarmos o entendimento dos fundamentos de herança e polimorfismo.



Aula 04:

Introdução ao  
Paradigma de  
Objetos

Autores:

Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2012

4

## Exercício 01

- O exercício consiste em:
  - a) Para os objetos Poligono, Triangulo, Quadrilatero, TrianguloIsosceles, TrianguloEquilatero, TrianguloEscaleno, Retangulo, Quadrado, construa um diagrama de classes indicando os relacionamentos de herança. Inclua na classe Poligono os métodos  

```
boolean validar(), float perimetro() e void imprime().
```
  - b) Baixe o arquivo codigos.zip que tem parte da implementação feita. Sua tarefa é completar o que estiver indicado com TODO (métodos “validar”) e criar as classes Retangulo e Quadrado Preencha a abstração procedimental (preparação para Javadoc) dos métodos a serem programados.
  - c) Submeta o exercício com o nome **TYGXXA04E01.zip**, incluindo todos os arquivos .java no arquivo zipado.(Opcional) Use VioletUML para modelar o diagrama de classes



DEPARTAMENTO DE ENGENHARIA DE  
COMPUTAÇÃO E SISTEMAS DIGITAIS  
**PCS**  
PCS 2302/2024  
Laboratório de  
Fundamentos da  
Eng.de Computação

Aula 04:

Introdução ao  
Paradigma de  
Objetos

Autores:

Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2012

5

## Exercício 02 : Chamada de Supervisor (SVC)

- O conjunto de instruções da MVN é bem limitado
  - Mas ela possui a instrução F (**Chamada de Supervisor**, representada pelo **mneumônico OS**), que a princípio não tem uma operação específica associada a ela.
- Para contornar esta limitação da MVN, podemos utilizar a instrução OS para estender a MVN.
  - Para isso, vamos adotar a seguinte estrutura:

OS	Parâmetros	Operação	Exemplo: OS 1FF
----	------------	----------	-----------------

**OS** Instrução F (1 byte)  
**Parâmetros** Número de parâmetros (1 byte)  
**Operação** Código de operação [00 a FF] (2 bytes)



DEPARTAMENTO DE ENGENHARIA DE  
COMPUTAÇÃO E SISTEMAS DIGITAIS  
**PCS**  
PCS 2302/2024  
Laboratório de  
Fundamentos da  
Eng.de Computação

Aula 04:

Introdução ao  
Paradigma de  
Objetos

Autores:

Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2012

6

## Exercício 02 : Chamada de Supervisor (SVC)



O valor dos parâmetros a serem passados para a chamada de supervisor devem ser posicionados no endereço de memória anterior ao da chamada, na forma de **pilha** (primeiro parâmetro vem logo acima da chamada OS)

Ex.: suponha que a operação **FF** tenha como parâmetros um conjunto de unidades lógicas (ULs).


Passando a UL 1 como parâmetro: UL1 K /0001  
OS /01FF

Passando ULs 1 e 2 como parâmetros: UL2 K /0002  
UL1 K /0001  
OS /02FF

Obs.: note que é necessário saltar a pilha de parâmetros para a correta execução do código

DEPARTAMENTO DE ENGENHARIA DE  
COMPUTAÇÃO E SISTEMAS DIGITAIS



PCS 2302/2024  
Laboratório de  
Fundamentos da  
Eng.de Computação

Aula 04:  
Introdução ao  
Paradigma de  
Objetos

Autores:  
Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2012

## Exercício 02 :

### Chamada de Supervisor (SVC)

---



#### TYGXXA04E02

Vamos estender a MVN para que ela possa interpretar a instrução F conforme a estrutura de chamada descrita, além de executar a operação solicitada.


Portanto, neste exercício é necessário fazer alterações no código em Java da máquina MVN para possibilitar tanto a interpretação da instrução F, quanto a execução da operação solicitada.

O que iremos fazer: criar uma nova classe de **Dispositivo**, denominada **Rand**, que gera números pseudoaleatórios usando como classe subjacente **java.util.Random**

7

DEPARTAMENTO DE ENGENHARIA DE  
COMPUTAÇÃO E SISTEMAS DIGITAIS



PCS 2302/2024  
Laboratório de  
Fundamentos da  
Eng.de Computação

Aula 04:  
Introdução ao  
Paradigma de  
Objetos

Autores:  
Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2012

## Exercício 02 :

### Chamada de Supervisor (SVC)

---

#### TYGXXA04E02 (Obrigatório)

Chamada SVC (em Java).

Altere o código da MVN para que, com uma *Chamada de Supervisor* (instrução OS), seja executada o método “addDispSimples”. Esse método cria um novo dispositivo do tipo Teclado (tipo 0), Monitor (tipo 1) ou Rand (tipo 4) e o adiciona ao GerenciadorDispositivos da MVN.

**Parâmetros:** type – o tipo de dispositivo (0, 1 ou 4, apenas)  
lu – o número da unidade lógica do dispositivo (0 a 255)



**Retorno (acumulador):** 0 em caso de sucesso, -1 em caso de erro (tipo inválido, ou unidade lógica inválida)

**Instrução de Chamada:** OS /02AD


**OBS:** Conforme slides anteriores, os parâmetros da instrução OS devem ser posicionados logo antes da chamada. Ex.:

LU	K	/0000
TYPE	K	/0001
OS		/02AD

8

DEPARTAMENTO DE ENGENHARIA DE  
COMPUTAÇÃO E SISTEMAS DIGITAIS



PCS 2302/2024  
Laboratório de  
Fundamentos da  
Eng.de Computação

Aula 04:  
Introdução ao  
Paradigma de  
Objetos

Autores:  
Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2012

## Exercício 02 :

### Chamada de Supervisor (SVC)

---

#### TYGXXA04E02 (Obrigatório)

#### Classe Rand (em Java).



Adicione uma classe “Rand” ao código da MVN. Use como base o arquivo Rand.java fornecido no pacote “mvn.dispositivo” do código fonte da MVN. O conteúdo deste arquivo é o dado abaixo, e deve ser complementado de acordo

(Importante: não altere o número de parâmetros do construtor).


```
package mvn.dispositivo;

public class Rand {

    /** Construtor: inicializa objeto (nenhum parametro recebido) */
    public Rand() {
        //...
    }
}
```

DEPARTAMENTO DE ENGENHARIA DE  
COMPUTAÇÃO E SISTEMAS DIGITAIS



PCS 2302/2024  
Laboratório de  
Fundamentos da  
Eng.de Computação

Aula 04:  
Introdução ao  
Paradigma de  
Objetos

Autores:  
Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2012

## Exercício 02 :

### Chamada de Supervisor (SVC)

---

#### TYGXXA04E02 – Requisitos da classe Rand

- Rand deve **implementar** a interface **Dispositivo**
- Rand deve **possuir** um objeto do tipo **Random s**
- O **construtor** e o método “**reset**” inicializa a semente de **s** com o valor 0
- “**escrever**” inicializa a semente de **s** com o valor passado
- “**ler**” deve gerar um **único byte** usando o método “**nextBytes**” da classe **Random**, o qual é retornado como resultado na forma de um objeto do tipo **Bits8**
- “**skip(val)**” deve **saltar “val” bytes** (ou nenhum se **val < 0**), retornando o próximo byte como resultado;
- “**podeLer**” e “**podeEscrever**” sempre retornam “**true**”
- “**position**” e “**size**” retornar 0, na forma de um objeto do tipo **Bits8**

## Exercício 02: Informações auxiliares

- **No exercício 2:** crie um código em linguagem MVN para testar o correto funcionamento da chamada OS. Nomeie-o **TYGXXA04E02.asm**
- Utilize os **diagramas e códigos** apresentados para entender o **funcionamento da MVN**.
- Você pode precisar **alterar a visibilidade** de métodos na MVN (e.g., de “private” para “public”)
- Consulte a **documentação das classes da MVN** (comentários no código) e a **documentação do Java** para lhe auxiliar no desenvolvimento
- **O código da MVN** disponibilizado pode carregar e executar programas. O mesmo **deve continuar funcionando após a modificação**.

## Exercício 02: Critérios de Correção

- Os seguintes arquivos devem ser entregues:
  - O código fonte completo da MVN com as suas alterações (arquivos “.java”), incluindo o Rand e todas as outras classes
  - O seu main usado para testar a instrução OS, com o nome **TYGXXA04E02.asm**
- O código da MVN será testado com as rotinas desenvolvidas nas aulas anteriores, para verificar que a MVN continua funcionando corretamente
- A correta adição de um dispositivo na MVN será feita usando o comando “s” da MVN, que permite ver os dispositivos nela presentes.



PCS 2302/2024  
Laboratório de  
Fundamentos da  
Eng.de Computação

Aula 04:

Introdução ao  
Paradigma de  
Objetos

Autores:

Anna H. R. Costa  
Jaime S. Sichman  
João José Neto  
Paulo S. Muniz Silva  
Ricardo L. A. Rocha

Reestruturação:  
Paulo S. Muniz Silva

v.1.0 ago. 2012

13

## Macro Arquitetura da MVN

