

# Programação Orientada a Objeto

## 08. Modificadores de Acesso e *Getters* e *Setters* (Prática)

Prof. Dr. Thiago L. T. da Silveira

`tltsilveira@inf.ufrgs.br`

2º Semestre de 2020

# Objetivos

- Exercitar os conceitos de POO vistos na aula anterior!
  - Escrevendo, compilando e interpretando (executando) aplicações em Java;
  - **Scanner** e Pacotes; Modificadores de Acesso, *Getters* e *Setters*.

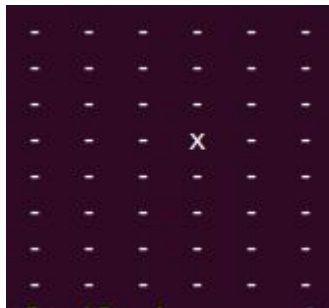
# Exercício #1 Parte #0

- Um refletor do tipo “painel de LEDs” contém vários pontos de luz, potencialmente independentes, que podem ser manipulados via programação de microcontroladores. Manter poucos LEDs ligados reduz o consumo de energia do painel; e a posição dos LEDs ligados causa efeitos diferentes: (i) LEDs próximos formam um feixe de luz concentrado; e (ii) LEDs distantes criam um efeito de luz difusa. Com essa abstração em mente, [...]



# Exercício #1 Parte #1

- Criar uma classe em Java (**PontoPainel8x6**) que representa a posição de um ponto de luz de um painel com 8 linhas e 6 colunas, conforme especificação:



PontoPainel8x6
- int x - int y
+ boolean setX(int x) + boolean setY(int y) + int getX() + int getY() + void imprime() + int distancia(PontoPainel8x6 p) - void reset()

# Exercício #1 Parte #1

- Definir os atributos e métodos conforme notação: (-) privados e (+) públicos;
- O método **setX** só deve atribuir um valor **x** ao atributo **this.x** se  $x \in [0, 6)$ ;
- O método **setY** só deve atribuir um valor **y** ao atributo **this.y** se  $y \in [0, 8)$ ;
- Os métodos **setX** e **setY** devem retornar **true** apenas se a atribuição for bem sucedida;
- Do contrário, os métodos **setX** e **setY** devem invocar o método **reset** e retornar **false** [...]

PontoPainel8x6
- int x - int y
+ boolean setX(int x) + boolean setY(int y) + int getX() + int getY() + void imprime() + int distancia(PontoPainel8x6 p) - void reset()

# Exercício #1 Parte #1

- O método **reset** atribui o valor zero aos atributos **x** e **y**;
- Os métodos **getX** e **getY** retornam os valores dos atributos **x** e **y**;
- O método **imprime** mostra na tela os valores dos atributos **x** e **y** na forma “(x, y)”;
- O método **distancia** recebe um objeto do tipo **PontoPainel8x6** e retorna a distância de Manhattan ( $dM$ ) para o objeto corrente. Dados os pontos  $p_1=(x_1, y_1)$  e  $p_2=(x_2, y_2)$ ,

$$dM(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|$$

PontoPainel8x6
- int x - int y
+ boolean setX(int x) + boolean setY(int y) + int getX() + int getY() + void imprime() + int distancia(PontoPainel8x6 p) - void reset()

# Exercício #1 Parte #1

- O método `reset` atribui o valor zero aos atributos `x` e `y`;
- Os métodos `getX` e `getY` retornam os valores dos atributos `x` e `y`;
- O método `imprime` mostra na tela os valores

*Podes importar "`import java.lang.Math`" para utilizar o método `Math.abs()` que implementa a funcionalidade "valor absoluto"; (desafio extra: implemente a sua versão da funcionalidade de `abs`)*

Manhattan ( $dM$ ) para o objeto corrente. Dados os pontos  $p_1=(x_1, y_1)$  e  $p_2=(x_2, y_2)$ ,

$$dM(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|$$

```
PontoPainel8x6
+ int getX()
+ int getY()
+ void imprime()
+ int distancia(PontoPainel8x6 p)
- void reset()
```

# Exercício #1 Parte #2

- Uma vez finalizada a implementação de **PontoPainel18x6**, a coloque em um pacote chamado “ponto”;
- Crie outra classe em Java (**AplicacaoTestaPonto**), que representa o programa principal e contém o método *main*. Coloque-a em um pacote chamado “teste”;
- Lembre-se de onde os arquivos devem estar e de incluir “**package ponto**” e “**package teste**” nos devidos lugares!



# Exercício #1 Parte #3

- O método *main* de **AplicacaoTestaPonto** deve declarar e instanciar um objeto **p1** do tipo **PontoPainel18x6**;
- Os valores dos atributos do objeto, **x** e **y**, devem ser informados pelo usuário;
- O método *main* deve pedir pares de valores, **x** e **y**, até que o usuário informe valores válidos para ambos, isto é, quando **p1.setX(x) && p1.setY(y)** é **true**;
- Assim que os atributos **x** e **y** de **p1** assumem valores válidos, deve-se invocar o método **imprime**;
- Crie um segundo objeto do tipo **PontoPainel18x6**, **p2**, e repita todos os passos desse slide, agora com **p2** [...]

# Exercício #1 Parte #3

*Instancie um objeto da classe "Scanner" para solicitar dados ao*

```
/* Definições de pacotes, outros imports ... */  
import java.util.Scanner;  
  
public class AplicacaoTestaPonto{  
  
    public static void main (String[] args){  
        /* Exemplo de como obter um valor  
        inteiro informado pelo usuário */  
        Scanner leitor = new Scanner(System.in);  
        int valor = leitor.nextInt();  
    }  
}
```

# Exercício #1 Parte #4

- O método *main* de **AplicacaoTestaPonto** deve invocar o método **distancia** do objeto **p1**, passando **p2** como argumento, e imprimir o resultado na tela;
- O método *main* de **AplicacaoTestaPonto** deve mostrar a localização de **p1** e **p2**, em um *grid* 8x6, utilizando o trecho de código abaixo e implementando o

```
int MAXY = 8, MAXX = 6;
for (y = 0; y < MAXY; y++){
    for (x = 0; x < MAXX; x++){
        if (/* implementar: verificar se a posição (y, x) coincide com a de um ponto */)
            System.out.print(" x ");
        else
            System.out.print(" - ");
    }
    System.out.println("");
}
```

# Exercício #1 - Exemplo de Execução

```
Informe a coordenada x do ponto 1: -1
Informe a coordenada y do ponto 1: 4
Informe a coordenada x do ponto 1: 2
Informe a coordenada y do ponto 1: 5
(2, 5)
Informe a coordenada x do ponto 2: 5
Informe a coordenada y do ponto 2: 3
(5, 3)
P1 e P2 estão a uma distância de 5 unidades
```

```
- - - - -
- - - - -
- - - - -
- - - - - x
- - - - -
- - x - -
- - - - -
- - - - -
```

# Atividades

- Entrega de um arquivo **.zip** (contendo a estrutura de diretórios da aplicação e os dois arquivos .java)
  - Entregue a implementação do Exercício #1 Partes #1 a #4
  - Entrega até às 23:55h de 25/02/2021



# Programação Orientada a Objeto

## 08. Modificadores de Acesso e *Getters* e *Setters* (Prática)

Prof. Dr. Thiago L. T. da Silveira

`tltsilveira@inf.ufrgs.br`

2º Semestre de 2020