PROGRAMAÇÃO ORIENTADA A OBJETOS

Prof. Delano M. Beder

Roteiro 01a

Paradigma Estruturado - Cálculo da distância entre dois pontos P1 (x,y) e P2 (x,y)

---

1. Crie um projeto (Aplicação C) denominado **Pontos2D-C**
2. Implementação da abstração **Ponto2D**

   Crie o arquivo de cabeçalho C denominado **Ponto2D.h**

```c
#ifndef PONTO2D_H
#define PONTO2D_H

typedef struct Ponto2D {
    float x, y;
} Ponto2D;

float distancia(Ponto2D p1, Ponto2D p2);

#endif /* PONTO2D_H */
```

   Crie o arquivo de código-fonte C denominado **Ponto2D.c**

```c
#include <math.h>
#include "Ponto2D.h"

float distancia(Ponto2D p1, Ponto2D p2) {
    float dx = p1.x - p2.x;
    float dy = p1.y - p2.y;
    return sqrt (dx * dx + dy * dy);
}
```

3. Atualize o arquivo de código-fonte C denominado **main.c** (programa principal)

```c
#include <stdio.h>
#include "Ponto2D.h"

int main() {
    Ponto2D p1, p2;

    p1.x = 4;
    p1.y = 4;

    p2.x = 7;
    p2.y = 8;

    printf("(%5.2f, %5.2f)\n", p1.x, p1.y);
```

```
    printf("(%5.2f, %5.2f)\n", p2.x, p2.y);
    printf("dist(P1, P2) = %5.2f\n", distancia(p1, p2));
    printf("dist(P2, P1) = %5.2f\n\n", distancia(p2, p1));

    p1.x += 10;
    p2.x += 10;

    printf("(%5.2f, %5.2f)\n", p1.x, p1.y);
    printf("(%5.2f, %5.2f)\n", p2.x, p2.y);
    printf("dist(P1, P2) = %5.2f\n", distancia(p1, p2));
    printf("dist(P2, P1) = %5.2f\n\n", distancia(p2, p1));

    p1.y += 5;
    p2.y += 5;

    printf("(%5.2f, %5.2f)\n", p1.x, p1.y);
    printf("(%5.2f, %5.2f)\n", p2.x, p2.y);
    printf("dist(P1, P2) = %5.2f\n", distancia(p1, p2));
    printf("dist(P2, P1) = %5.2f\n\n", distancia(p2, p1));

    return 0;
}
```

4. Crie o arquivo **Makefile** (opcional)

```
CXX         := gcc
CXX_FLAGS   :=

LIBRARIES   := -lm
EXECUTABLE  := Pontos2D-C

all: run
    @rm -f $(EXECUTABLE)

run: $(EXECUTABLE)
    @./$(EXECUTABLE)

$(EXECUTABLE): *.c
    $(CXX) $(CXX_FLAGS) -I. $^ -o $@ $(LIBRARIES)
```

5. Compile, execute e verifique a saída impressa
6. Fim