

PROGRAMAÇÃO ORIENTADA A OBJETOS

Prof. Delano M. Beder

Roteiro 01b

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

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

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

```
#ifndef PONTO2D_H
#define PONTO2D_H

class Ponto2D {
public:
    Ponto2D(float x, float y); // construtor
    virtual ~Ponto2D(); // destrutor

    float distancia(const Ponto2D p) const; // método da classe
    void moveX(float dx);
    void moveY(float dy);
    void imprime();
private:
    float x, y; // atributos da classe
};

#endif /* PONTO2D_H */
```

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

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

Ponto2D::Ponto2D(float x, float y) {
    this->x = x;
    this->y = y;
}

Ponto2D::~Ponto2D() {}

float Ponto2D::distancia(const Ponto2D p) const {
    float dx = this->x - p.x;
    float dy = this->y - p.y;
    return sqrt(dx * dx + dy * dy);
}
```

```

void Ponto2D::moveX(float dx) {
    this->x += dx;
}

void Ponto2D::moveY(float dy) {
    this->y += dy;
}

void Ponto2D::imprime() {
    printf("(%.2f, %.2f)\n", this->x, this->y);
}

```

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

```

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

int main() {

    Ponto2D p1(4, 4);
    Ponto2D p2(7, 8);

    p1.imprime();
    p2.imprime();
    printf("dist(P1, P2) = %.2f\n", p1.distancia(p2));
    printf("dist(P2, P1) = %.2f\n\n", p2.distancia(p1));

    p1.moveX(10);
    p2.moveX(10);

    p1.imprime();
    p2.imprime();
    printf("dist(P1, P2) = %.2f\n", p1.distancia(p2));
    printf("dist(P2, P1) = %.2f\n\n", p2.distancia(p1));

    p1.moveY(5);
    p2.moveY(5);

    p1.imprime();
    p2.imprime();
    printf("dist(P1, P2) = %.2f\n", p1.distancia(p2));
    printf("dist(P2, P1) = %.2f\n\n", p2.distancia(p1));

    return 0;
}

```

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

```
CXX      := g++
CXX_FLAGS := -Wall -Wextra -std=c++17 -ggdb
LIBRARIES :=
EXECUTABLE := Pontos2D-Cpp

all: run
    @rm -rf $(EXECUTABLE)

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

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

5. Compile, execute e verifique a saída impressa

6. Fim