Relações entre Classes

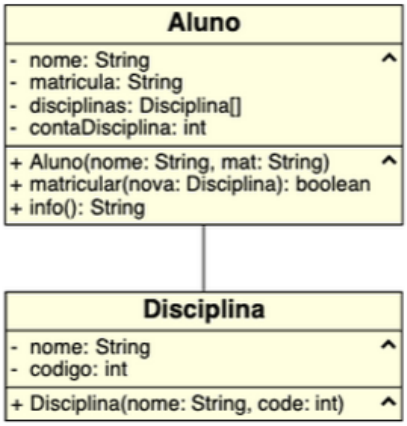
Associação

- Um objeto interagindo com outro de outro tipo para atingir determinada funcionalidade.
- Conexões entre classes.

```
public class Aluno { //Declaração da Classe
    //Atributos
    private String nome;
    private String mat;
    private Disciplina[] disciplinas;
    private int contaDisciplinas;

    //Construtor
    public Aluno(String nome, String mat) {
        this.disciplina = new Disciplina[6];
        this.nome = nome;
        this.mat = mat;
        this.contaDisciplinas = 0;
    }

    public void matricular(Disciplina nova) {
        if( podeMatricular() ){
            this.disciplinas[ contaDisciplinas++ ] = nova;
            return true;
        }
        return false;
}
```



Composição

- Associação do tipo todo-parte;
  - Um objeto dentro do outro para ser parte do todo.
- Acontece quando temos um objeto dentro do outro. Este objeto é parte do outro e não faz sentido que seja de outra forma.

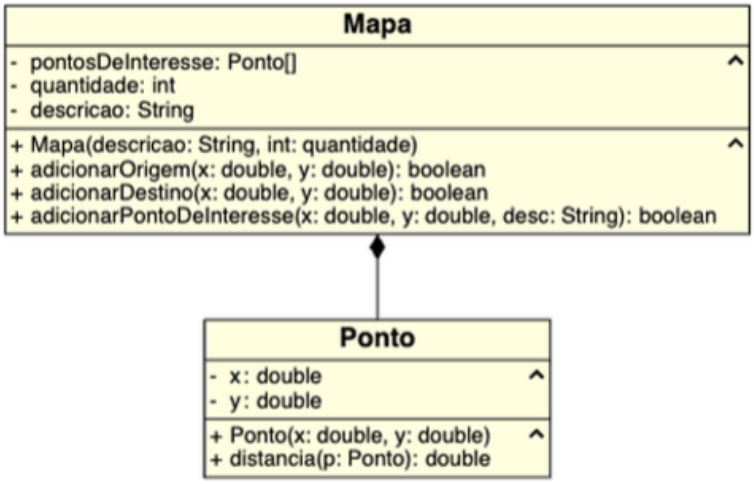
```
public class Mapa { //Declaração da Classe
  //Atributos
   private Ponto[] pontosDeInteresse; //Mapa tem um conjunto de
   private int quantidade;
   private String descricao;

  //Métodos
  public Mapa(String d, int n) {
      this.quantidade = n;
            this.descricao = d;

  public void adicionaPontoOrigem(double x, double y) {
        Ponto p = new Ponto(x, y);
            this.pontosDeInteresse[0] = p;
```

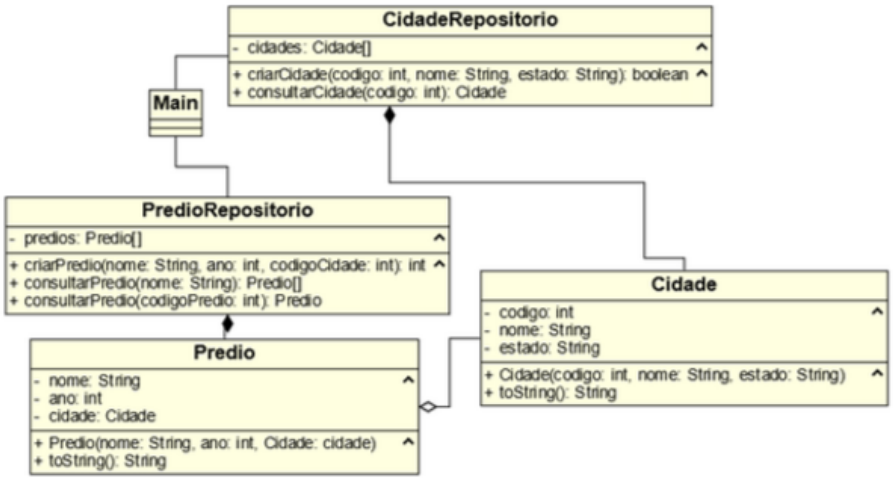
```
            // …

//Declaração da Classe
public class Ponto { //Ponto deve fazer parte de Mapa e não faz
    private double x;
    private double y;
    private String desc;
    public double distancia(Ponto p) {
        double xDist = this.x - p.x;
        double yDist = this.y - p.y;
        return Math.sqrt(xDist * xDist + yDist * yDist);
```



- Repositório: Armazena objetos de um determinado tipo, realiza consultas e buscas de objetos desse tipo.

Repositório



Encapsulamento

- Consiste em agrupar conceitos (características e operações) semanticamente relacionados, em um tipo, a classe;
    - É um dos princípios de OO.

```
public class Predio { //Declaração da Classe
    //Atributos
    private String nome;
    private int ano;
    private String rua;
    private String numero;
    private String cidade;
    private String estado;
    private String pais;
    private String cep;
    //…

   //Métodos
   public Predio(String nome, int ano, String rua, String numero
        this.nome = nome;
```

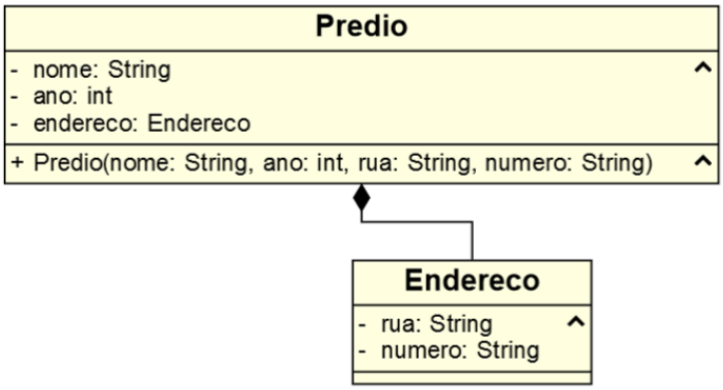
```
        this.ano = ano;
        this.rua = rua;
        ...
```

Encapsulando, prédio tem um endereço:

```
public class Predio { //Declaração da Classe
  //Atributos
  private String nome;
  private int ano;
  private Endereco end;
  //…

   //Métodos
   public Predio(String nome, int ano, String rua, int numero, S
        this.nome = nome;
        this.ano = ano;
        this.end = new Endereco(rua, numero, cidade, estato, pai
        //...


public class Endereco { //Declaração da Classe
    //Atributos
    private String rua;
    private int numero;
    private String cidade;
    //…
```



- Cada objeto PRÈDIO é composto de nome, ano e um endereço;
- Cada objeto de Prédio terá um endereço próprio;
- Nenhum endereço é compartilhado entre diferentes prédios nesse sistema;
- Endereço compõe prédio.

Agregação

- É um tipo especial de associação entre classes;
- Diferencia-se da composição por ser considerada uma relação todo-parte fraca.
    - As partes podem pertencer a outros agregados.

Agregação

Associação todo-parte fraca. Os objetos são fracamente relacionados ao todo.

Composição

Associação todo-parte forte. Os objetos são parte de um todo e não faz sentido existir sem ele.

```
public class Main {
    public static void main (String[] args) {
        Endereco e1 = new Endereco("Bedford Street", "90", …);
        Endereco e4 = new Endereco("Central Park West", "200", .

        Pessoa pessoa1 = new Pessoa("Rachel", "03456-6", e1);
        Pessoa pessoa2 = new Pessoa("Monica", "19345-5", e1);
   …
   Predio museu = new Predio("American Museum of Natural History
       pessoa3.setTrabalho( museu );


…
   // Mudando a rua deste endereço
        e1.setRua("Gramham Street");
        ...
// A alteração no objeto agregado afeta todos aqueles que o agre
```

```
//Exemplo de Agregação
public class Predio { //Declaração da Classe
    //Atributos
    private String nome;
    private int ano;
    private Endereco end;
    //…
   //Métodos
   public Predio(String nome, int ano, Endereco end) {
        this.nome = nome;
        this.ano = ano;
        this.end = end;
        ...

public class Pessoa { //Declaração da Classe
    //Atributos
    private String nome;
    Private String cpf
    private Endereco end;
    //…
   //Métodos
   public Pessoa(String nome, String cpf, Endereco end) {
        this.nome = nome;
        this.cpf = cpf;
        this.end = end;

 public class Endereco { //Declaração da Classe
    //Atributos
    private String rua;
    private String numero;
    private String cidade;
    //…
```

Representação UML