

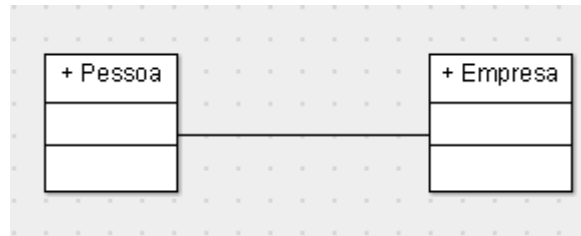
Associação de Objetos

Prof. Edwar Saliba Júnior
Fevereiro de 2011

Tipos de Associação

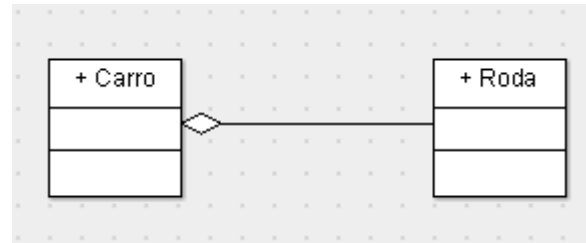
- Uma classe pode possuir um ou mais atributos que são objetos de outras classes;
- Este tipo de relacionamento é chamado de ASSOCIAÇÃO e se subdivide em três tipos:
 - Associação Simples;
 - Agregação e
 - Composição.

Associação Simples



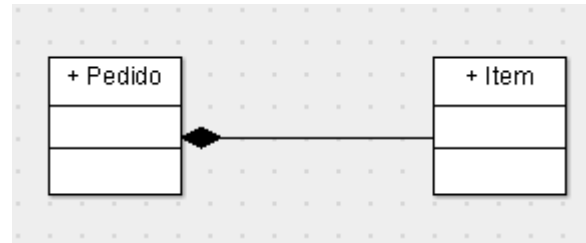
- Estabelece uma relação semântica estrutural entre as classes;
- Representada por um traço entre as classes;
- Exemplo:
 - uma pessoa trabalha para um empresa.

Agregação



- Estabelece uma relação “todo-parte” entre as classes, sendo que a parte pode existir sem o todo;
- Representada por uma linha com um losango aberto no lado todo;
- Exemplo: Carro e Roda. Roda é parte de um carro, porém ela pode existir sem o carro.

Composição



- Estabelece um relação “todo-parte” entre as classes. Sendo que a parte não existe sem o todo;
- Representada por uma linha com um losango cheio do lado todo;
- Exemplo: Pedido e Itens de pedido. Se o pedido for destruído, os itens que o compõem também deverão ser.

Exemplo de Agregação

Classe Ponto

```
1 package circulo;
2
3 public class Ponto {
4     private float x, y;
5
6     public Ponto(float x, float y) {
7         this.x = x;
8         this.y = y;
9     }
10
11     public void alterarX(float x) {
12         this.x = x;
13     }
14
15     public void alterarY(float y) {
16         this.y = y;
17     }
18
19     public float obterX() {
20         return x;
21     }
22
23     public float obterY() {
24         return y;
25     }
26 }
27
28
```

Classe Círculo

```
1 package circulo;
2
3 public class Circulo {
4
5     private float raio;
6     private Ponto centro;
7
8     public Circulo(float x, float y, float r) {
9         raio = r;
10        centro = new Ponto(x, y);
11    }
12
13    public void alterarCentro(float x, float y) {
14        centro.alterarX(x);
15        centro.alterarY(y);
16    }
17
18    public void alterarRaio(float r) {
19        raio = r;
20    }
21
22    public float obterCentroX() {
23        return centro.obterX();
24    }
25
26    public float obterCentroY() {
27        return centro.obterY();
28    }
29
30    public float obterRaio() {
31        return raio;
32    }
33 }
34
```


Classe Principal

```
1 package circulo;
2
3 import java.util.Scanner;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         float x, y, raio;
9         Circulo circ;
10        Scanner in = new Scanner(System.in);
11
12        System.out.println("Digite as coordenadas do centro: ");
13        System.out.println("Digite a coordenada X: ");
14        x = in.nextFloat();
15        System.out.println("Digite a coordenada Y: ");
16        y = in.nextFloat();
17        System.out.println("Digite o raio: ");
18        raio = in.nextFloat();
19
20        circ = new Circulo(x, y, raio);
21
22        System.out.println("Circulo criado: ");
23        System.out.println("Raio: " + circ.obterRaio());
24        System.out.println("Centro: (" + circ.obterCentroX() + ", "
25            + circ.obterCentroY() + ").");
26        System.out.println("");
27    }
28 }
29
```

Bibliografia

- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML Guia do Usuário**. Rio de Janeiro: Campus, 2000.
- DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**; tradução Edson Furmankiewicz; revisão técnica Fábio Lucchini. 6a. ed., São Paulo: Pearson, 2005.
- FERREIRA, Kecia Aline Marques. *Slides da disciplina de Programação de Computadores II*. CEFET-MG, 2009.