Exemplo de Associação de Objetos em Java

- Composição -

Neste documento será apresentado um exemplo do conceito de composição de objetos utilizando a linguagem de programação Java.

Na Figura 1 apresentamos o diagrama de classes (sem métodos) da regra de negócio do sistema.

A ideia central deste documento é apresentar um código simples, completo e funcionando de um sistema que contenha o conceito de "Associação de Objetos". Portanto, para se entender o que está neste documento, faz-se necessário o conhecimento prévio do leitor sobre os seguintes assuntos: linguagem de programação Java e conceitos de orientação a objetos (especificamente "associação de objetos").

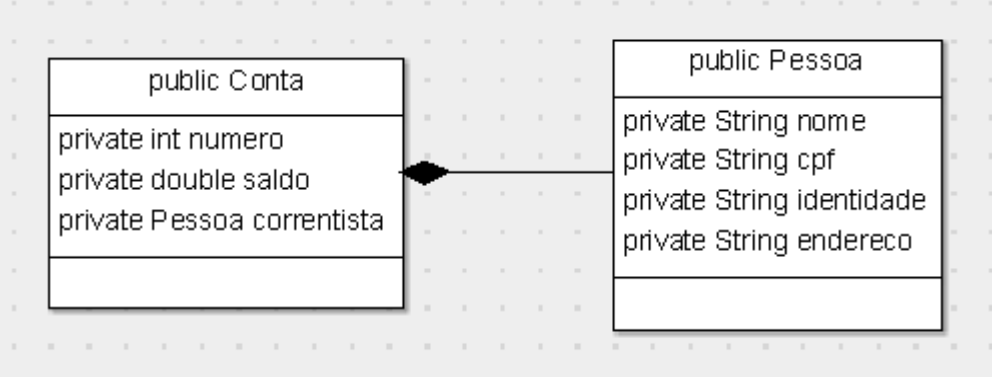

Figura 1: Diagrama de classe da regra de negócio.

O *software* em questão será constituído das seguintes classes: Pessoa, Conta e a classe Principal.

Sem mais delongas, vamos ao código:

Classe Pessoa

```java
package exemploassociacaodeobjetos;

/**
 *
 * @author Edwar Saliba Júnior
 */
public class Pessoa {
    private String nome;
    private String cpf;
    private String identidade;
    private String endereco;

    public Pessoa(String nome, String cpf, String identidade, String endereco) {
        this.nome = nome;
        this.cpf = cpf;
        this.identidade = identidade;
        this.endereco = endereco;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getCpf() {
        return cpf;
    }

    public void setCpf(String cpf) {
        this.cpf = cpf;
    }

    public String getIdentidade() {
        return identidade;
    }

    public void setIdentidade(String identidade) {
        this.identidade = identidade;
    }

    public String getEndereco() {
        return endereco;
    }

    public void setEndereco(String endereco) {
        this.endereco = endereco;
    }

    public void consultar(){
        System.out.println("Nome       : " + nome);
        System.out.println("C.P.F.     : " + cpf);
        System.out.println("Identidade: " + identidade);
        System.out.println("Endereço   : " + endereco);
    }
}
```

Figura 2: Classe Pessoa.

Classe Conta

```java
package exemploassociacaodeobjetos;

/**
 *
 * @author Edwar Saliba Júnior
 */
public class Conta {
    private int numero;
    private double saldo;
    private Pessoa correntista;

    public Conta(int numero, double saldo, Pessoa correntista) {
        this.numero = numero;
        this.saldo = saldo;
        this.correntista = correntista;
    }

    public int getNumero() {
        return numero;
    }

    public void setNumero(int numero) {
        this.numero = numero;
    }

    public double getSaldo() {
        return saldo;
    }

    public void setSaldo(double saldo) {
        this.saldo = saldo;
    }

    public Pessoa getCorrentista() {
        return correntista;
    }

    public void setCorrentista(Pessoa correntista) {
        this.correntista = correntista;
    }

    public void consultar(){
        System.out.println("Dados da Conta ----------------------");
        System.out.println("Número     : " + numero);
        System.out.println("Saldo      : " + saldo);
        System.out.println("Correntista -----------------------");
        correntista.consultar();
    }
}
```

Figura 3: Classe Conta.

Classe Principal

```
package exemploassociacaodeobjetos;

import java.util.Scanner;

/**
 *
 * @author Edwar Saliba Júnior
 */
public class ExemploAssociacaoDeObjetos {

    public static void main(String[] args) {
        Scanner e = new Scanner(System.in);
        final int TAM_VET_PESSOAS = 100;
        final int TAM_VET_CONTAS = 100;
        Pessoa pessoas[] = new Pessoa[TAM_VET_PESSOAS];
        Conta contas[] = new Conta[TAM_VET_CONTAS];
        String nome, cpf, identidade, endereco;
        int op, op2, i, pos, pos2, opCampo, resp, numero,
            quantidadeDePessoasCadastradas = 0,
            quantidadeDeContasCadastradas = 0;
        double saldo;

        do{
            System.out.println("\n\n-==[Menu Principal]==--");
            System.out.println("1 - Pessoa");
            System.out.println("2 - Conta");
            System.out.println("3 - Sair");
            System.out.println("Opção: ");
            op = e.nextInt();
            switch(op){
                case 1:
                    do{
                        System.out.println("\n\n-=[Pessoa]=-");
                        System.out.println("1 - cadastrar");
                        System.out.println("2 - alterar");
                        System.out.println("3 - excluir");
                        System.out.println("4 - consultar");
                        System.out.println("5 - relatório");
                        System.out.println("6 - voltar ao menu principal");
                        System.out.println("Opção: ");
                        op2 = e.nextInt();
                        e.skip("\n");

                        switch(op2){
                            case 1:
                                System.out.println("-[Cadastro de Pessoa]-");
                                System.out.println("Digite o nome: ");
                                nome = e.nextLine();
                                System.out.println("Digite o C.P.F.: ");
                                cpf = e.nextLine();
                                System.out.println("Digite a identidade: ");
                                identidade = e.nextLine();
                                System.out.println("Digite o endereço: ");
                                endereco = e.nextLine();
                                i = 0;
                                // Procura uma posição vazia no vetor.
                                while(i < TAM_VET_PESSOAS && pessoas[i] != null){
                                    i++;
                                }
                                if(i < TAM_VET_PESSOAS){
                                    pessoas[i] = new Pessoa(nome, cpf, identidade, endereco);
                                    quantidadeDePessoasCadastradas++;
                                    System.out.println("Pessoa cadastrada com sucesso.");
                                }
                                else{
                                    System.out.println("O vetor está cheio. Pessoa não cadastrada.");
                                }
                                break;
                            case 2:
                                if(quantidadeDePessoasCadastradas > 0){
```

Figura 4: Classe Principal - Parte 01.

```
71                                  System.out.println("-[Alteração de Pessoa]-");
72                                  System.out.println("Digite a posição do vetor: ");
73                                  pos = e.nextInt();
74                                  if(pessoas[pos] != null){
75                                      System.out.println("-- Dados Cadastrados --");
76                                      pessoas[pos].consultar();
77                                      System.out.println("\nQual campo deseja alterar? ");
78                                      System.out.println("1 - nome");
79                                      System.out.println("2 - C.P.F.");
80                                      System.out.println("3 - Identidade");
81                                      System.out.println("4 - Endereço");
82                                      System.out.println("Opção: ");
83                                      opCampo = e.nextInt();
84                                      e.skip("\n");
85
86                                      switch(opCampo){
87                                          case 1:
88                                              System.out.println("Digite o novo nome: ");
89                                              pessoas[pos].setNome(e.nextLine());
90                                              break;
91                                          case 2:
92                                              System.out.println("Digite o novo C.P.F.: ");
93                                              pessoas[pos].setCpf(e.nextLine());
94                                              break;
95                                          case 3:
96                                              System.out.println("Digite a nova identidade: ");
97                                              pessoas[pos].setIdentidade(e.nextLine());
98                                              break;
99                                          case 4:
100                                             System.out.println("Digite o novo endereço: ");
101                                             pessoas[pos].setEndereco(e.nextLine());
102                                     }
103                                     System.out.println("Alteração efetuada com sucesso.");
104                                 }
105                                 else{
106                                     System.out.println("Não existe pessoa na posição escolhida.");
107                                 }
108                             }
109                             else{
110                                 System.out.println("Não existem pessoas cadastradas.");
111                             }
112                             break;
113                         case 3:
114                             if(quantidadeDePessoasCadastradas > 0){
115                                 System.out.println("-[Exclusão de Pessoa]-");
116                                 System.out.println("Digite a posição do vetor: ");
117                                 pos = e.nextInt();
118                                 if(pessoas[pos] != null){
119                                     System.out.println("-- Dados Cadastrados --");
120                                     pessoas[pos].consultar();
121                                     System.out.println("\nConfirma exclusão? (1-sim/2-não)");
122                                     resp = e.nextInt();
123                                     if(resp == 1){
124                                         pessoas[pos] = null;
125                                         quantidadeDePessoasCadastradas--;
126                                         System.out.println("Exclusão efetuada com sucesso.");
127                                     }
128                                     else{
129                                         System.out.println("Exclusão não efetuada.");
130                                     }
131                                 }
132                                 else{
133                                     System.out.println("Não existe pessoa na posição escolhida.");
134                                 }
135                             }
136                             else{
137                                 System.out.println("Não existem pessoas cadastradas.");
138                             }
139                             break;
140                         case 4:
```

Figura 5: Classe Principal - Parte 02.

```
                                if(quantidadeDePessoasCadastradas > 0){
                                    System.out.println("-[Consulta de Pessoa]-");
                                    System.out.println("Digite a posição do vetor: ");
                                    pos = e.nextInt();
                                    if(pessoas[pos] != null){
                                        System.out.println("-- Dados Cadastrados --");
                                        pessoas[pos].consultar();
                                    }
                                    else{
                                        System.out.println("Não existe pessoa na posição escolhida.");
                                    }
                                }
                                else{
                                    System.out.println("Não existem pessoas cadastradas.");
                                }
                                break;
                            case 5:
                                if(quantidadeDePessoasCadastradas > 0){
                                    System.out.println("-[Relatório de Pessoas]-");
                                    i = 0;
                                    while(i < TAM_VET_PESSOAS){
                                        if(pessoas[i] != null){
                                            pessoas[i].consultar();
                                            System.out.println("-x-x-x-x-x-x-x-x-x-x-x-");
                                        }
                                        i++;
                                    }
                                }
                                else{
                                    System.out.println("Não existem pessoas cadastradas.");
                                }
                        }
                    }while(op2 != 6);
                    break;
                case 2:
                    do{
                        System.out.println("\n\n-=[Conta]=-");
                        System.out.println("1 - cadastrar");
                        System.out.println("2 - alterar");
                        System.out.println("3 - excluir");
                        System.out.println("4 - consultar");
                        System.out.println("5 - relatório");
                        System.out.println("6 - voltar ao menu principal");
                        System.out.println("Opção: ");
                        op2 = e.nextInt();

                        switch(op2){
                            case 1:
                                /* Testa se existem pessoas cadastradas no sistema
                                   antes de tentar efetuar um cadastro de conta, pois,
                                   para se cadastrar uma conta é necessário que se
                                   tenha um objeto Pessoa para ser associado ao
                                   objeto Conta.                                    */
                                if(quantidadeDePessoasCadastradas > 0){
                                    System.out.println("-[Cadastro de Conta]-");
                                    System.out.println("Digite o número: ");
                                    numero = e.nextInt();
                                    System.out.println("Digite o saldo: ");
                                    saldo = e.nextDouble();
                                    /* Mecanismo para possibilitar ao usuário, a
                                       seleção de uma pessoa já cadastrada no sistema. */
                                    System.out.println("-- Pessoa --");
                                    resp = 0;
                                    do{
                                        System.out.println("Escolha uma posição do vetor: ");
                                        pos = e.nextInt();
                                        if(pessoas[pos] != null){
                                            pessoas[pos].consultar();
                                            System.out.println("Confirma pessoa? (1-sim/2-não)");
                                            resp = e.nextInt();
```

Figura 6: Classe Principal - Parte 03

```
                                    }
                                    else{
                                        System.out.println("Não existe pessoa na posição " +
                                                "escolhida.");
                                    }
                                }while(resp != 1);

                                i = 0;
                                while(i < TAM_VET_CONTAS && contas[i] != null){
                                    i++;
                                }
                                if(i < TAM_VET_CONTAS){
                                    contas[i] = new Conta(numero, saldo, pessoas[pos]);
                                    quantidadeDeContasCadastradas++;
                                    System.out.println("Conta cadastrada com sucesso.");
                                }
                                else{
                                    System.out.println("O vetor está cheio. Conta não " +
                                            "cadastrada.");
                                }
                            }
                            else{
                                System.out.println("Não existem pessoas cadastradas no " +
                                        "sistema. Para se cadastrar uma conta é necessário " +
                                        "que a esta conta seja associada uma pessoa.");
                            }
                            break;
                        case 2:
                            if(quantidadeDeContasCadastradas > 0){
                                System.out.println("-[Alteração de Conta]-");
                                System.out.println("Digite a posição do vetor: ");
                                pos = e.nextInt();
                                if(contas[pos] != null){
                                    System.out.println("-- Dados Cadastrados --");
                                    contas[pos].consultar();
                                    System.out.println("\nQual campo deseja alterar? ");
                                    System.out.println("1 - número");
                                    System.out.println("2 - saldo");
                                    System.out.println("3 - Pessoa");
                                    System.out.println("Opção: ");
                                    opCampo = e.nextInt();
                                    e.skip("\n");

                                    switch(opCampo){
                                        case 1:
                                            System.out.println("Digite o novo número: ");
                                            contas[pos].setNumero(e.nextInt());
                                            break;
                                        case 2:
                                            System.out.println("Digite o novo saldo: ");
                                            contas[pos].setSaldo(e.nextDouble());
                                            break;
                                        case 3:
                                            System.out.println("-- Escolha a nova Pessoa --");
                                            resp = 0;
                                            do{
                                                System.out.println("Escolha uma posição do " +
                                                        "vetor: ");
                                                pos2 = e.nextInt();
                                                if(pessoas[pos2] != null){
                                                    pessoas[pos2].consultar();
                                                    System.out.println("Confirma pessoa? " +
                                                            "(1-sim/2-não)");
                                                    resp = e.nextInt();
                                                }
                                                else{
                                                    System.out.println("Não existe pessoa na " +
                                                            "posição escolhida.");
                                                }
                                            }while(resp != 1);
```

Figura 7: Classe Principal - Parte 04.

```
281                                         contas[pos].setCorrentista(pessoas[pos2]);
282                                     }
283                                     System.out.println("Alteração efetuada com sucesso.");
284                                 }
285                                 else{
286                                     System.out.println("Não existe conta cadastrada nesta " +
287                                             "posição.");
288                                 }
289                             }
290                             else{
291                                 System.out.println("Não existem contas cadastradas.");
292                             }
293                             break;
294                         case 3:
295                             if(quantidadeDeContasCadastradas > 0){
296                                 System.out.println("-[Exclusão de Conta]-");
297                                 System.out.println("Digite a posição do vetor: ");
298                                 pos = e.nextInt();
299                                 if(pessoas[pos] != null){
300                                     System.out.println("-- Dados Cadastrados --");
301                                     contas[pos].consultar();
302                                     System.out.println("\nConfirma exclusão? (1-sim/2-não)");
303                                     resp = e.nextInt();
304                                     if(resp == 1){
305                                         contas[pos] = null;
306                                         quantidadeDeContasCadastradas--;
307                                         System.out.println("Exclusão efetuada com sucesso.");
308                                     }
309                                     else{
310                                         System.out.println("Exclusão não efetuada.");
311                                     }
312                                 }
313                                 else{
314                                     System.out.println("Não existe conta na posição escolhida.");
315                                 }
316                             }
317                             else{
318                                 System.out.println("Não existem contas cadastradas.");
319                             }
320                             break;
321                         case 4:
322                             if(quantidadeDeContasCadastradas > 0){
323                                 System.out.println("-[Consulta de Conta]-");
324                                 System.out.println("Digite a posição do vetor: ");
325                                 pos = e.nextInt();
326                                 if(contas[pos] != null){
327                                     System.out.println("-- Dados Cadastrados --");
328                                     contas[pos].consultar();
329                                 }
330                                 else{
331                                     System.out.println("Não existe conta na posição escolhida.");
332                                 }
333                             }
334                             else{
335                                 System.out.println("Não existem contas cadastradas.");
336                             }
337                             break;
338                         case 5:
339                             if(quantidadeDeContasCadastradas > 0){
340                                 System.out.println("-[Relatório de Contas]-");
341                                 i = 0;
342                                 while(i < TAM_VET_CONTAS){
343                                     if(contas[i] != null){
344                                         contas[i].consultar();
345                                         System.out.println("-x-x-x-x-x-x-x-x-x-x-x-");
346                                     }
347                                     i++;
348                                 }
349                             }
350                             else{
```

Figura 8: Classe Principal - Parte 05.

```
351                                         System.out.println("Não existem contas cadastradas.");
352                                     }
353                                 }
354                             }while(op2 != 6);
355                         }
356                     }while(op != 3);
357             }
358     }
```

Figura 9: Classe Principal - Parte 06.

E assim termina o exemplo de *software* orientado a objetos, utilizando o conceito de “Associação de Objetos – Composição”.

Qualquer dúvida, entre em contato.

Edwar Saliba Júnior