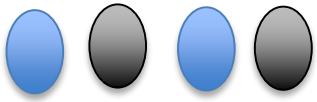
CIÊNCIAS DA COMPUTAÇÃO

FACULDADE DE CIÊNCIAS | UNIVERSIDADE AGOSTINHO NETO



1º Ano

FUNDAMENTOS DE PROGRAMAÇÃO

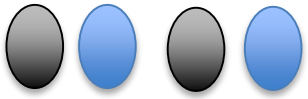
Docente:

- ✓ Mateus Padoca Calado - PhD

Monitores:

- ✓ Nsimba Kiafuka
- ✓ Anacleto Mimoso
- ✓ Mariano Calelua

Conteúdo



Cap. VIII

- Classes

CAP. VIII – Classes

- Uma classe é uma estrutura de dados que agrega variaveis(atributos) e métodos(operações), ou seja é a definição do comportamento e estrutura de um tipo de objectos.

Sintaxe	Exemplo

`public class nomeDaClasse {`
 	
`/*Declaração de variáveis*/`
 	
`/*definição de métodos*/`
 	
`}`
	`public class Pessoa {`
	
	`/*Declaração de variáveis*/`
	
	`/*definição de métodos*/`
	`}`

CAP. VIII – Classes

- Basicamente cada classe pode ter dois tipos de membros:
  - **campo(s) –** são variáveis declarada no escopo global da classe, podendo ser de tipo primitivo ou de tipo por referencia.
  - **método(s) -** método que executa uma acção para instâncias da classe (Objecto).

CAP. VIII – Classes

Criação de uma classe pessoa que agrega os seguintes membros: Três campos: nome, idade e genero.
Dois métodos:

- uma função **public int tamanho()** que retorna o tamanho do nome
- um procedimento **public void anoNascimento (int anoActual)** que informa o ano de nascimento da pessoa.

Resolução

```
public class Pessoa {
     /*Declaração  de variáveis*/
     String nome;
     int idade;
     char genero;

     /*definição dos métodos*/
     public int tamanhoNome(){
          return nome.length();
     }
     public void anoNascimeto(int anoActual){
          int nascimento = anoActual – idade;
          System.out.println(“Ano de Nascimento = ”+ nascimento);
     }
}
```

CAP. VIII – Objectos

- Um objecto é a instância de uma classe, ou seja, é cada uma das variáveis criadas a partir da definição de uma classe.

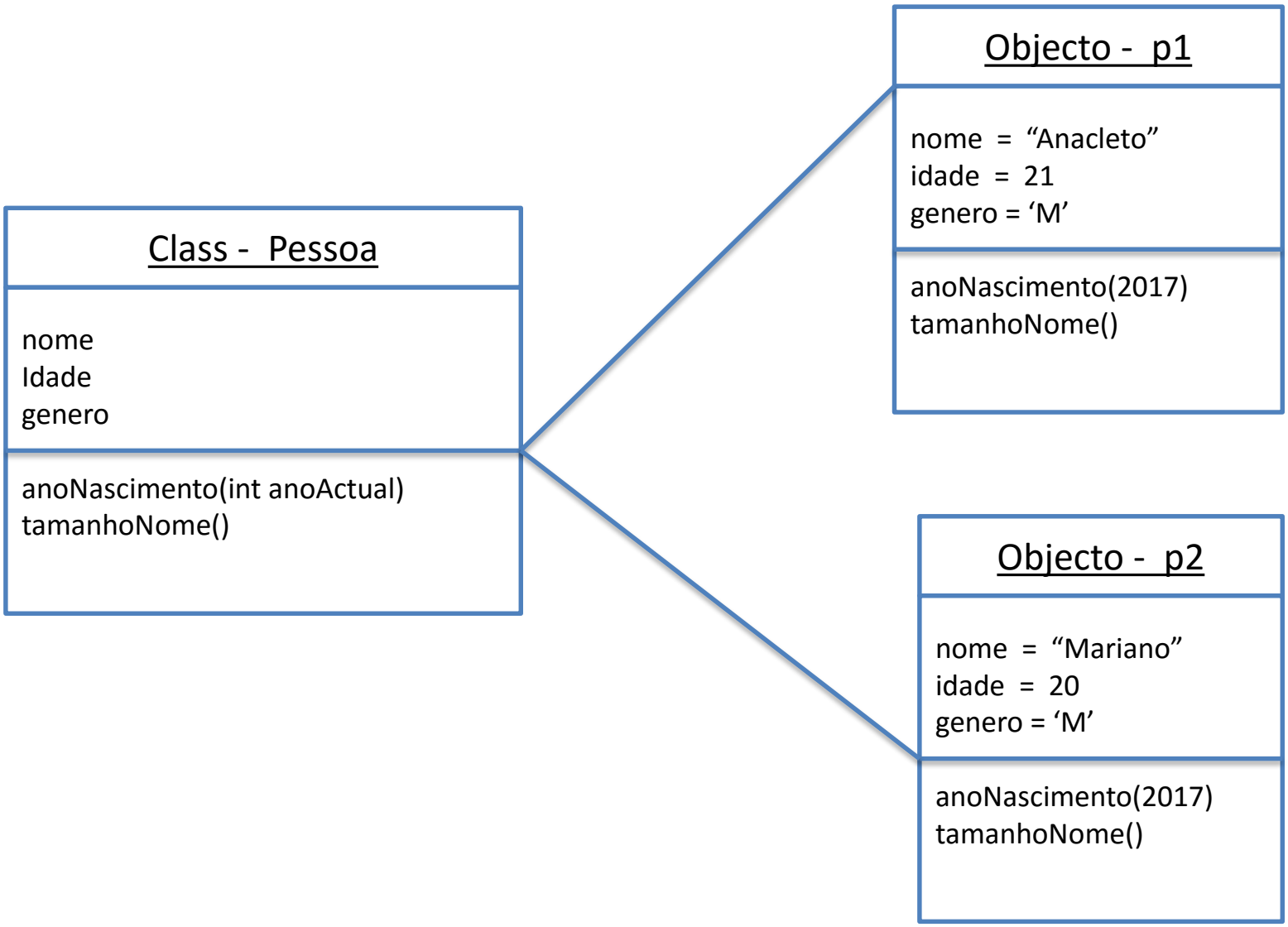
Sintaxe
`NomeDaClass nomeObjecto = new NomeDaClass();`

- Exemplo:

```
class TestePessoa{
     public static void main(String[] args) {
          Pessoa p1 = new Pessoa();
     }
}
```

- p1 é uma variável do tipo Pessoa, ou seja, p1 tem acesso as variaveis e métodos declarados em Pessoa.

CAP. VIII – Representações gráficas



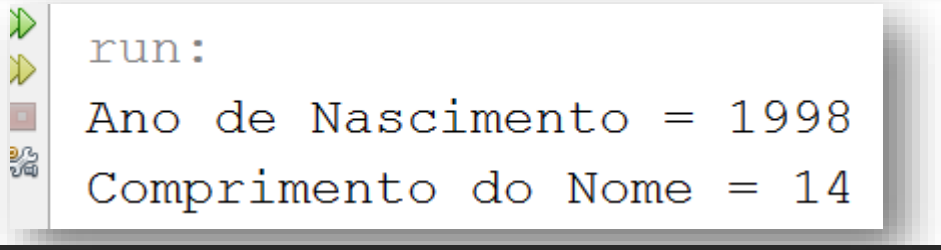
CAP. VIII – Objectos

- Implementação da class TestePessoa que cria e utiliza um objecto do tipo Pessoa.

EXEMPLO

```
public class TestePessoa {
    public static void main(String[] args) {

        /*Declarar o objecto */
        Pessoa p1 = new Pessoa();

        /*Preencher os dados do Objecto */
        p1.nome = "Nsimba Paulina";
        p1.idade = 19;
        p1.genero = 'F';

        /* Utilizar os métodos do objecto*/
        p1.anoNascimeto(2017);
        System.out.println(p1.tamanhoNome());
    }

}
```

- **RESULTADO:**



CAP. VIII – Array de Objectos

- Também é possível criar um array de uma classe.

```
Pessoa todasPessoas[] = new Pessoa[10]; /*criou-se um vector de pessoa com 10 posições*/
```

- Para preencher o vector deve-se sempre instanciar a posição em causa:

```
todasPessoas[0] = new Pessoa();
todasPessoas[0].nome = "Pedro Kondo";
todasPessoas[0].idade = 19;
todasPessoas[0].genero = 'M';
```

Ou:

```
Pessoa p = new Pessoa(); /*variavel auxiliar*/
p.nome = "Pedro Kondo";
p.idade = 19;
p.genero = 'M';
todasPessoas[0] = p;
```

CAP. VIII – ArrayList de Objectos

- Também é possível criar um ArrayList de uma classe.

```
ArrayList<Pessoa> todasPessoas = new ArrayList<>();
```

- Para preencher o ArrayList utiliza-se uma variável auxiliar:

```
Pessoa p = new Pessoa(); /*variavel auxiliar*/
p.nome = "Pedro Kondo";
p.idade = 19;
p.genero = 'M';
todasPessoas.add(p);
```

CAP. VIII – Modificadores de acesso

- Os **modificadores de acesso** são padrões de visibilidade de acessos às **classes, variaveis e métodos.**
- Alguns modificadores de acesso frequentemente utilizados em Java são:
  - public - os elementos public da classe podem ser a cessada e visualizada em qualquer lugar e por qualquer classe;
  - protected – os elementos definidos como Protected na class podem ser acessado por todas as classes do mesmo pacote;
  - private – os elementos definidos como private na class não podem ser acessados ou usados por outras classes;
  - padrão (sem nenhum modificador) – todas as classes do mesmo pacote têm acesso as variáveis, métodos ou classe;
  - static - os elementos definidos como static será o mesmo para todos as instância criada.

CAP. VIII – Modificadores de acesso

- Classe pessoa com modificadores de acesso private na variável idade.

Resolução

```
public class Pessoa {
     /*Declaração  de variáveis*/
     String nome;
     private int idade; /*variaveis privada*/
     char genero;

     /*definição dos métodos para acesso a variavel privada idade*/
     void setIdade(int idade){
         this.idade = idade;
     }
     int getIdade(){
         return idade;
     }
     /*definição dos outros métodos*/
     int tamanhoNome(){
          return nome.length();
     }
     void anoNascimeto(int anoActual){
          int nascimento = anoActual – idade;
          System.out.println("Ano de Nascimento = "+ nascimento);
     }  }
```

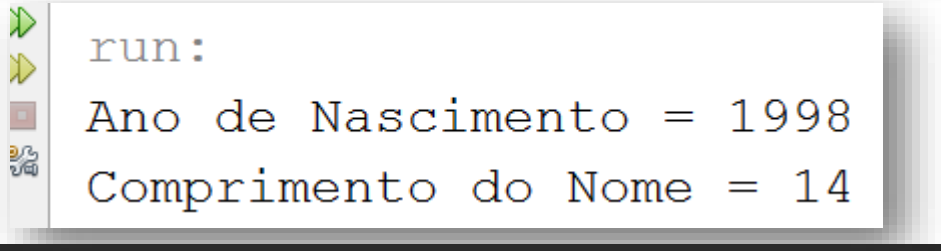
CAP. VIII – Modificadores de acesso

- Implementação da class TestePessoa que cria e utiliza um objecto do tipo Pessoa.

EXEMPLO

```
public class TestePessoa {
    public static void main(String[] args) {

        /*Declarar o objecto */
        Pessoa p1 = new Pessoa();

        /*Preencher os dados do Objecto */
        p1.nome = "Nsimba Paulina";
        p1.setIdade(19); //preencher a idade
        p1.genero = 'F';

        /* Utilizar os métodos do objecto*/
        p1.anoNascimeto(2017);
        System.out.println(p1.tamanhoNome());
    }

}
```

- **RESULTADO:**



CAP. VIII. Exercícios

1. Problema – Aluno

   1.1 Crie uma classe Aluno que agrega os seguintes membros: Três campos: nome, idade e o número de estudante (com modificador padrão).

   Dois métodos:

   - uma função **public int tamanho()** que retorna o tamanho do nome
   - um procedimento **public void imprimirDados()** que imprimi o nome a idade e o número do aluno.

   1.2 Crie a classe TesteAluno com os seguintes membros:

   a) Um vector todosAlunos[50] do tipo Aluno
   b) Um procedimento **public static void inserirAluno()** – que insere um novo aluno no vector.
   c) Um procedimento **public static void consultarAluno()** – que solicita o numero do aluno e apresenta os seu dados.
   d) Um procedimento **public static void listarTodosAlunos()** – que lista os dados de todos alunos armazenados no vector.
   e) Implemente um menu interativo no método main e chama as operações :
      - 1 – Inserir Aluno
      - 2 – Consultar Aluno
      - 3 – Listar Alunos
      - 4 - Sair

CAP. VIII. Exercícios

2. Problema – Conta Bancaria

2.1 Crie uma classe Conta que agrega os seguintes membros: Três campos: numero, nomeDono e o saldo da conta(com modificador padrão).

- Um procedimento public void **lenvatamento(double valor) -** que efectua um levantamento na conta em função do valor especificado.
- um procedimento **public void depositar(double valor)** que efectua um deposito na conta em função do valor especificado.
- um procedimento **public void imprimirDados()** que imprimi o número, o nome e o saldo da conta bancaria.

2.2 Crie a classe TesteConta com os seguintes membros:

a) Um vector todascontas[50] do tipo Conta
b) Um procedimento **public static void criarConta()** – que cria uma nova conta no vector.
c) Um procedimento **public static void consultarConta()** – que solicita o numero do conta e apresenta os seu dados.
d) Um procedimento **public static void listarContas()** – que imprimi todas as conta armazenadas no vector.
e) Implemente um menu interativo no método main e chama as operações :
   - 1 – Criar conta
   - 2 – Consultar Conta
   - 3 – Listar Contas
   - 4 - Sair