



Programação Orientada a Objetos (POO)

Prof. Dr. Alan Souza

alan.souza@unama.br

2020

POO



VANTAGENS DE PROGRAMAR COM ORIENTAÇÃO À OBJETOS

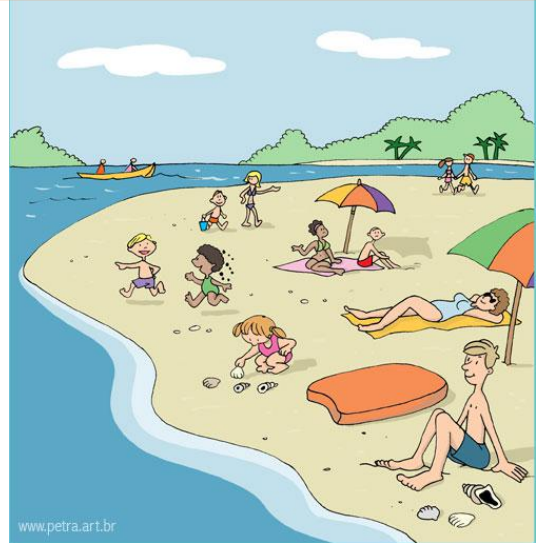
1. Reuso do código;
2. Aumento da produtividade;
3. Redução das linhas do projeto;
4. Separação de responsabilidades;
5. Encapsulamento;
6. Polimorfismo;
7. Componentização;
8. Facilita o teste do sistema.

3. Classes e Objetos

DEFINIÇÃO DE CLASSE

Uma classe é a descrição de um conjunto de entidades (reais ou abstratas) do mesmo tipo e com as mesmas características e comportamentos.

Identifique as classes nesta imagem



3. Classes e Objetos

DEFINIÇÃO DE CLASSE

A classe também pode ser considerada como sendo um **tipo de variável criado pelo programador**.

COMPONENTES DA CLASSE

Atributos da Classe

Representam as características da classe.

Métodos/funções

Representam os comportamentos da classe.

3. Classes e Objetos

EXEMPLO



Qual o nome desta classe?

Crianca

Quais os atributos/características?

cor do cabelo, cor da pele, altura, idade...

Quais as funções?

brincar, nadar, correr, sorrir...

3. Classes e Objetos

CONVENÇÃO CAMEL CASE

O nome das classes devem iniciar com letra maiúscula. Se a classe for composta por duas palavras, cada início de palavra inicia com letra maiúscula.

Exemplos: **Coqueiro, GuardaSol.**

O nome dos atributos e das funções devem iniciar com letra minúscula. Caso seja composto por duas ou mais palavras, a primeira palavra começa com minúsculo e as outras com maiúsculo.

Exemplos: **cor, numeroGalhos.**

3. Classes e Objetos



Atributos - características da classe:

- Estão associados à declaração de variáveis;
- Cada atributo precisa de um tipo e de um nome (identificador);
- Exemplos:
 - String** corCabelo;
 - String** corPele;
 - double** altura;
 - int** idade;

3. Classes e Objetos



Métodos/funções - ações da classe

- Podem receber e/ou retornar argumentos (valores).
- A variável que receber o valor retornado pelo método deve ser do mesmo tipo do retorno do método.
- O tipo de variável enviado para o método deve ser igual ou equivalente ao tipo de variável de recebimento do método.

3. Classes e Objetos



DEFINIÇÃO DE OBJETO

- Conceito semelhante ao de variável (espaço na memória principal);
- Através dele, torna-se possível **acessar atributos e/ou métodos de uma classe**, desde que estes sejam acessíveis;
- Um objeto é aquilo que se pode criar a partir da classe;
- Possui um valor inicial.

3. Classes e Objetos



PARTE PRÁTICA

3. Classes e Objetos



CRIANDO A CLASSE “CRIANCA” EM JAVA

```
public class Crianca { //início da classe  
  
    //corpo da classe Crianca  
  
} //fim da classe Crianca
```

3. Classes e Objetos



DECLARANDO OS ATRIBUTOS DA CLASSE “CRIANCA” EM JAVA

```
public class Crianca {  
    String corCabelo;  
    String corPele;  
    double altura;  
    int idade;  
}
```

3. Classes e Objetos



DECLARANDO OS MÉTODOS/FUNÇÕES DA CLASSE “CRIANCA” EM JAVA

```
public class Crianca {
    //atributos omitidos
    void brincar(){
        System.out.println("Brincando...");
    }
    void correr(double metros){
        System.out.println("Correndo " + metros +
" metros");
    }
}
```

3. Classes e Objetos



DECLARANDO OS MÉTODOS/FUNÇÕES DA CLASSE “CRIANCA” EM JAVA

```
public class Crianca {
    //códigos omitidos
    void nadar(double metros){
        System.out.println("Nadando " + metros + " m");
    }
    String sorrir(){
        return ":";
    }
}
```

3. Classes e Objetos



CRIANDO OBJETO DA CLASSE “CRIANCA” EM JAVA

```
public class ProjetoPraia {
    public static void main(String args[]) {
        Crianca c1;
        c1 = new Crianca();
        //ou...
        Crianca c2 = new Crianca();
    }
}
```

3. Classes e Objetos

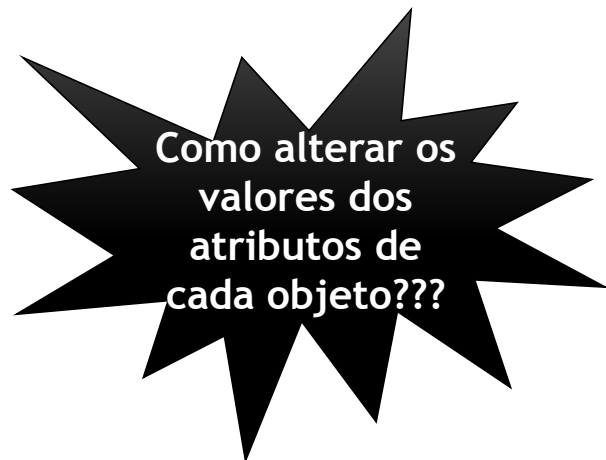
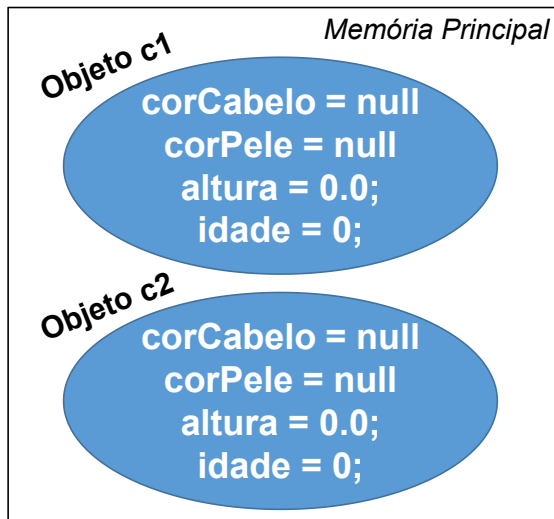


CRIANDO OBJETO DA CLASSE “CRIANCA” EM JAVA

```
public class ProjetoPraia {
    public static void main(String args[]) {
        Crianca c1; //declaração de variável
        c1 = new Crianca(); //instanciação do objeto
        //ou...
        Crianca c2 = new Crianca(); //declaração e
        instanciação ao mesmo tempo
    }
}
```


3. Classes e Objetos

ANALISANDO OS OBJETOS CRIADOS



3. Classes e Objetos

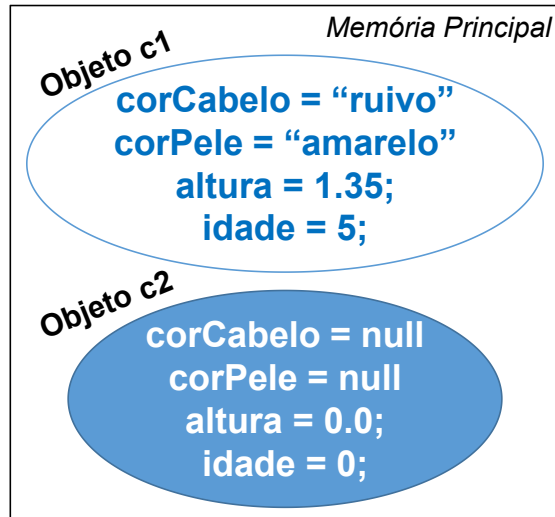
ALTERANDO OS ATRIBUTOS DOS OBJETOS CRIADOS

```
class ProjetoPraia {
    public static void main(String args[]) {
        //instanciação dos objetos omitidos
        c1.corCabelo = "ruivo";
        c1.corPele = "amarelo";
        c1.altura = 1.35;
        c1.idade = 5;
    }
}
```

Operador ponto acessa atributos ou métodos da classe através do objeto

3. Classes e Objetos

ANALISANDO OS OBJETOS CRIADOS (VALORES MUDADOS)



3. Classes e Objetos

MOSTRANDO OS VALORES DO OBJETOS c1

```
class ProjetoPraia {
    public static void main(String args[]) {
        //instanciação dos objetos omitidos
        System.out.println("Cor do cabelo: " + c1.corCabelo);
        System.out.println("Cor da pele: " + c1.corPele);
        System.out.println("Altura: " + c1.altura);
        System.out.println("Idade: " + c1.idade);
    }
}
```

3. Classes e Objetos



Acessando métodos pelo objeto c1

```
class ProjetoPraia {
    public static void main(String args[]) {
        //códigos omitidos
        //acessando métodos:
        c1.brincar();
        String sorriso = c1.sorrir();
        System.out.println( sorriso );
    }
}
```

3. Classes e Objetos



EXERCÍCIOS

1) Criar um projeto de nome ProjetoPraia no Netbeans e construir nele a classe Crianca, de acordo com o que foi mostrado anteriormente.

Em seguida:

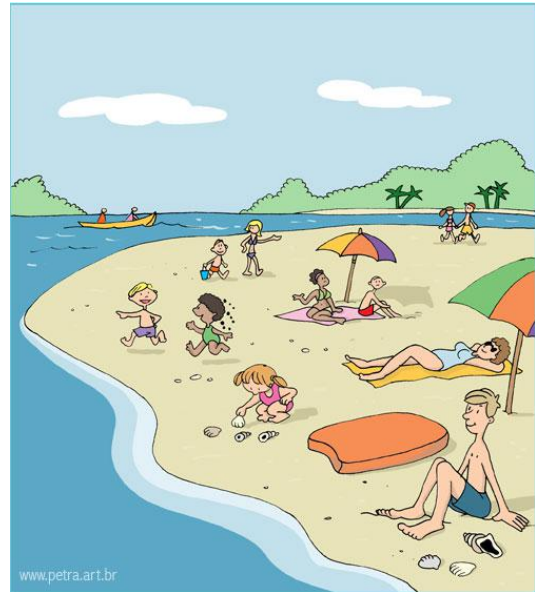
- crie dois objetos da classe Crianca,
- altere os atributos através de um dos objetos,
- mostre os valores dos atributos modificados na tela,
- chame os métodos da classe através do objeto.

3. Classes e Objetos

EXERCÍCIOS

2) Analise novamente a imagem da praia e extraia dela mais uma classe. Além disso, defina para ela:

- três atributos,
- dois métodos.



3. Classes e Objetos

EXERCÍCIOS

3) Crie a classe do exercício 2 no ProjetoPraia no Netbeans. Depois, faça:

- Instanciação de dois objetos da classe;
- Alteração dos atributos através do objeto da classe;
- Impressão dos atributos na tela;
- Invocação dos métodos da classe.



Programação Orientada a Objetos (POO)

Prof. Dr. Alan Souza

alan.souza@unama.br

2020

Abstração



- Um dos pontos mais importantes em qualquer linguagem de programação orientada a objetos;
- Imaginar o que a entidade (classe) a ser programada irá realizar dentro do sistema, a partir da necessidade do sistema;
- **IDENTIDADE:** deve ser única (sem conflitos)
- **PROPRIEDADES:** características da entidade, ou seja, os atributos.
- **MÉTODOS/FUNÇÕES:** ações/eventos que a entidade irá realizar.

Abstração

A partir de um modelo geral, as especificidades são criadas.

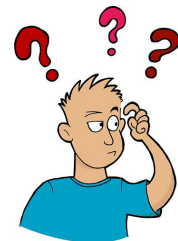


Abstração

- Exemplo: criar um sistema para editar textos no computador.

Abstração:

- Este sistema seria formado por quais classes?
- Quais os atributos de cada classe?
- Quais ações de cada classe?



Muitas dessas dúvidas é o usuário do sistema que precisa responder, pois o sistema será feito para ele.

-> **REUNIÕES**