

Programação Orientada a Objetos

Alexandre César Muniz de Oliveira



Desenvolvimento

Parte II



Desenvolvimento

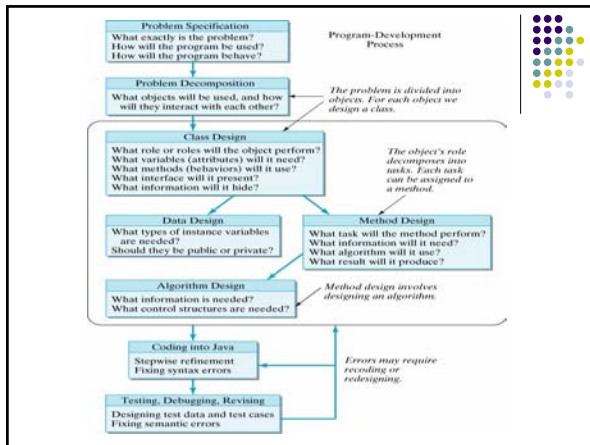
“The sooner you begin to type code, the longer the program will take to finish...”



Desenvolvimento

“... because careful design of the program must precede coding. This is particularly true of object-oriented programs”.





Desenvolvimento

• Programa *Riddle*

- Entendimento do problema
 - O que exatamente se quer resolver?
 - Como o programa será usado?
 - Como o programa deverá se comportar?

Desenvolvimento

- Especificação
 - Desenhe uma classe que represente um charada
 - pergunta
 - resposta
 - Armazena diferentes charadas
 - Recupera pergunta e resposta independentemente

Desenvolvimento

- Decomposição
 - Divisão em partes torna a solução mais gerenciável
 - Em objetos: cada objeto é responsável por uma parte do comportamento global
 - Dividir para conquistar
 - Como dividir?
 - Quais objetos?
 - Efetivamente:
 - Buscar por nomes no domínio do problema
 - Dividir o problema em um conjunto de objetos cooperativos
 - Buscar por verbos que significam ações realizadas pelos objetos

Desenvolvimento



• Projeto de objetos

- Que papel o objeto executará no programa?
- Qual os dados que ele precisará?
- Que ações ele tomará?
- Que interface ele apresentará para os demais objetos?
- Que dados ele ocultará?
- Pensar sobre o comportamento de um objeto na realidade é pensar sobre o comportamento de uma classe deles

Desenvolvimento



• Projeto de classes

- Atributos e métodos comuns a todos os objetos de uma mesma classe
- Um conjunto de estados que cada objeto pode assumir
- Comportamento comum

Desenvolvimento



• Projeto de classes

- Uma classe pode ser usada para
 - conter a rotina de execução principal de uma aplicação iniciada pelo sistema operacional (método main)
 - conter funções globais (métodos estáticos)
 - conter constantes e variáveis globais (campos de dados estáticos)
 - especificar e criar objetos (contém construtores, métodos e atributos de dados)

Desenvolvimento



• Projeto de classes

- Um único arquivo pode conter diversas classes, porém apenas uma delas pode ser definida como pública
- O nome do arquivo Java deve ter o mesmo nome da classe pública
- A Sun recomenda que nomes de classes sejam iniciados com uma letra maiúscula, para diferenciar nomes de objetos

Desenvolvimento



- Nome: *Riddle*
- Papel: Armazenar e recuperar questão e resposta
- Atributos (variáveis de instância)
 - *Question*: variável ou objeto? Pública ou privada
 - *Answer*
- Comportamento (verbos)
 - *Riddle()*: armazena pergunta e resposta
 - *getQuestion()*: recupera separadamente
 - *getAnswer()*: recupera separadamente

Desenvolvimento



- Comportamento:
 - Busca por verbos
 - Ações serão encapsuladas por métodos
 - Chamar um método
 - Corresponde a uma interação entre objetos
 - Envio de uma mensagem de um objeto para outro
 - Exemplo:
 - *getAnswer()* "Dê-me sua resposta"
 - *Riddle()*: Construtor é um método especial invocado quando da criação do objeto

Desenvolvimento



- Métodos
 - públicos (interface)
 - privativos (comportamento oculto)
- Ocultação
 - Excetuando a interface, tudo o mais no objeto deve ser oculto
 - Detalhes de implementação expostos são indesejáveis
 - Exemplo: classe *Racional*

Desenvolvimento



Riddle
- question : String
- answer : String
+ Riddle(q : String, a : String)
+ getQuestion() : String
+ getAnswer() : String

Desenvolvimento



• Dados, métodos e algoritmos

- Qual o tipo de dado dos atributos?
- Qual o algoritmo?
- Que entrada os métodos necessitam?
- Que saída os métodos produzem?

Codificação



```
/*
 * File: Riddle.java * Author: Java, Java, Java * Description: Defines a simple riddle.
 */
public class Riddle extends Object // Class header
{ // Begin class body
    private String question; // Instance variables
    private String answer;
    public Riddle(String q, String a) // Constructor method
    {
        question = q;
        answer = a;
    } // Riddle()

    public String getQuestion() // Instance method
    { return question; }

    public String getAnswer() // Instance method
    { return answer; }
} // End class body
```

Codificação



```
public class RiddleUser
{ public static void main(String argv[])
{ Riddle riddle1 = new Riddle("What is black and white and red all over?",
    "An embarrassed zebra.");
  Riddle riddle2 = new Riddle("What is black and white and read all over?",
    "A newspaper.");
  System.out.println("Here are two riddles:");
  System.out.println(riddle1.getQuestion());
  System.out.println(riddle2.getQuestion());
  System.out.println("The answer to the first riddle is:");
  System.out.println(riddle1.getAnswer());
  System.out.println("The answer to the second is:");
  System.out.println(riddle2.getAnswer());
} // main()
} // RiddleUser class
```

Codificação

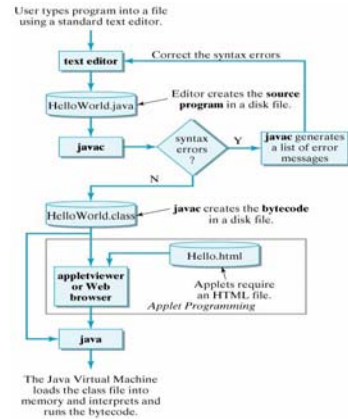


```
Java Language Elements
/* * File: HelloWorld.java * Author: Java Java Java * Description: Prints Hello
World greeting.
 */
public class HelloWorld extends Object // Class header
{ // Start class body
    private String greeting = "Hello World!"; // Method definition
    public void greet() // Start method body
    { // Output statement
        System.out.println(greeting); // End method body
    } // Method header
    public static void main(String args[]) // Method call
    {
        HelloWorld helloworld; // declare
        helloworld = new HelloWorld(); // create or instantiate
        helloworld.greet(); // Method call
    } // End class body
}
```

Codificação

```
/** HelloWorldApplet program */
import java.applet.Applet;           // Import class names
import java.awt.Graphics;
public class HelloWorldApplet extends Applet // Class header
{
    public void paint(Graphics g)      // The paint method
    {
        g.drawString("Hello World!", 10, 10);
    }
}
```

• Execução



Codificação

```
public class StringPuns
{
    public static void main(String args[])
    {
        String s = new String("string");
        String s1 = s.concat(" puns.");
        System.out.println("Here are the top 5 " + s1);
        String s2 = "5. Hey baby, wanna ";
        String s3 = s + " along with me.";
        System.out.println(s2 + s3);
        System.out.println("4. I've got the world on a " + s + ".");
        String s4 = new String("two");
        String s5 = ". You have more class than a ";
        System.out.println(s4.length());
        System.out.println(s5 + s + " of pearls.");
        System.out.println("2. It is ");
        System.out.println(s.equals("string"));
        System.out.println(" that I am no " + s + " bean.");
        String s6 = " quintet.";
        System.out.println("1. These puns form a " + s + s6);
    }
} // StringPuns class
```

Codificação

```
import java.awt.*;
import java.applet.*;
public class HelloWorldGraphic extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.cyan);           // Set color
        g.fillRect(25, 25, 140, 40);      // Fill rectangle
        g.setColor(Color.blue);          // Set color
        g.drawRect(25, 25, 140, 40);     // Outline rectangle
        g.setColor(Color.black);         // Set color
        g.drawString("Hello World", 50, 50); // Display string
        g.setColor(Color.yellow);        // Set color
        g.fillOval(25, 75, 140, 40);     // Fill oval
        g.setColor(Color.red);           // Set color
        g.drawOval(25, 75, 140, 40);     // Outline oval
        g.setColor(Color.black);         // Set color
        g.drawString("Welcome to Java", 50, 100);
    }
}
```

Codificação

