PERSISTÊNCIA DE

OBJETOS EM JAVA

UM BREVE TUTORIAL

(Versão 1.0)

ALCIDES CALSAVARA

PUCPR

2018

# Classe não persistente

Crie um projeto Java.

Crie a classe **Jogador**:

public class Jogador {  
 private String nome;  
 private int pontuacao;  
 private float distancia;  
  
 public Jogador(String nome, int pontuacao) {  
 this.nome = nome;  
 this.pontuacao = pontuacao;  
 }  
  
 public void posicionar(float distancia) {  
 this.distancia = distancia;  
 }  
  
 public void pontuar() {  
 pontuacao++;  
 distancia = distancia / 2;  
 }  
  
 public void exibir() {  
 System.*out*.println(nome + ": " + pontuacao + ", " + distancia);  
 }

}

Crie e execute uma classe de teste que instancie a classe Jogador e chame os seus métodos.

# Classe persistente: Interface Serializable

Faça a classe Jogador implementar a interface Serializable:

public class Jogador implements Serializable

É preciso fazer a seguinte importação:

import java.io.\*;

Faça o atributo distancia ser transiente (não persistente):

transient private float distancia;

# Persistência de um objeto

Adicione o método salvar na classe Jogador:

public void salvar(String nome\_arquivo) throws IOException {  
 FileOutputStream arquivo = new FileOutputStream(nome\_arquivo);  
 ObjectOutputStream gravador = new ObjectOutputStream(arquivo);  
  
 gravador.writeObject(this);  
  
 gravador.close();  
 arquivo.close();  
}

Crie e execute a classe CriarJogador:

import java.io.\*;  
  
public class CriarJogador {  
 public static void main(String[] args) {  
 Jogador jogador = new Jogador("Blue Jedi", 100);  
 jogador.posicionar(5.0f);  
 jogador.exibir();  
  
 try {  
 jogador.salvar("BlueJedi.ser");  
 System.*out*.println("Jogador criado e salvo com sucesso!");  
 } catch (IOException e) {  
 System.*out*.println("Excecao de I/O");  
 e.printStackTrace();  
 }  
 }  
}

Verifique o novo arquivo criado em disco.

# Recuperação de um objeto persistente

Adicione o método abrir na classe Jogador:

public static Jogador abrir(String nome\_arquivo) throws IOException, ClassNotFoundException {  
 Jogador jogador = null;  
  
 FileInputStream arquivo = new FileInputStream(nome\_arquivo);  
 ObjectInputStream restaurador = new ObjectInputStream(arquivo);  
  
 jogador = (Jogador) restaurador.readObject();  
  
 restaurador.close();  
 arquivo.close();  
  
 return jogador;  
}

Crie e execute a classe RecuperarJogador:

import java.io.\*;  
  
public class RecuperarJogador {  
  
 public static void main(String[] args) {  
 try {  
 Jogador jogador = Jogador.*abrir*("BlueJedi.ser");  
 System.*out*.println("Jogador recuperado com sucesso!");  
 jogador.exibir();  
 } catch (IOException e) {  
 System.*out*.println("Excecao de I/O");  
 e.printStackTrace();  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("Excecao de classe desconhecida");  
 e.printStackTrace();  
 }  
 }  
}

Observe o que aconteceu com o atributo distancia.

# Atualização de um objeto persistente

Crie e execute a classe AtualizarJogador:

import java.io.\*;  
  
public class AtualizarJogador {  
 public static void main(String[] args) {  
 String nome\_arquivo = "BlueJedi.ser";  
  
 try {  
 Jogador jogador = Jogador.*abrir*(nome\_arquivo);  
 System.*out*.println("Jogador recuperado com sucesso!");  
 jogador.exibir();  
  
 jogador.posicionar(20.0f);  
  
 jogador.pontuar();  
 jogador.exibir();  
  
 jogador.salvar(nome\_arquivo);  
 System.*out*.println("Jogador salvo com sucesso!");  
 } catch (IOException e) {  
 System.*out*.println("Excecao de I/O");  
 e.printStackTrace();  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("Excecao de classe desconhecida");  
 e.printStackTrace();  
 }  
 }  
}

Observe o que aconteceu com os atributos pontuacao e distancia.

# Persistência de objetos associados

Crie a classe Time:

import java.io.\*;  
  
public class Time implements Serializable {  
 private String nome;  
 private Jogador jogador1;  
 private Jogador jogador2;  
 private Treinador treinador;  
  
 public Time(String nome\_time, String nome\_1, int pontuacao\_1,  
 String nome\_2, int pontuacao\_2, String nome\_treinador) {  
 this.nome = nome\_time;  
 jogador1 = new Jogador(nome\_1, pontuacao\_1);  
 jogador2 = new Jogador(nome\_2, pontuacao\_2);  
 treinador = new Treinador(nome\_treinador);  
 jogador1.posicionar(8.0f);  
 jogador2.posicionar(14.0f);  
 }  
  
 public void exibir() {  
 System.*out*.println(nome);  
 jogador1.exibir();  
 jogador2.exibir();  
 treinador.exibir();  
 }  
  
 public void salvar(String nome\_arquivo) throws IOException {  
 FileOutputStream arquivo = new FileOutputStream(nome\_arquivo);  
 ObjectOutputStream gravador = new ObjectOutputStream(arquivo);  
  
 gravador.writeObject(this);  
  
 gravador.close();  
 arquivo.close();  
 }  
  
 public static Time abrir(String nome\_arquivo) throws IOException, ClassNotFoundException {  
 Time time = null;  
  
 FileInputStream arquivo = new FileInputStream(nome\_arquivo);  
 ObjectInputStream restaurador = new ObjectInputStream(arquivo);  
  
 time = (Time) restaurador.readObject();  
  
 restaurador.close();  
 arquivo.close();  
  
 return time;  
 }  
}

Cria a classe Treinador:

import java.io.Serializable;  
  
public class Treinador implements Serializable {  
 private String nome;  
 public Treinador(String nome) {  
 this.nome = nome;  
 }  
 public void exibir() {  
 System.*out*.println("Treinador " + nome);  
 }  
}

Crie e execute a classe CriarTime:

import java.io.\*;  
  
public class CriarTime {  
 public static void main(String[] args) {  
 Time time = new Time("Bugs", "Spider", 30, "Flea", 60, "Snake");  
 time.exibir();  
  
 try {  
 time.salvar("Bugs.ser");  
 System.*out*.println("Time criado e salvo com sucesso!");  
 } catch (IOException e) {  
 System.*out*.println("Excecao de I/O");  
 e.printStackTrace();  
 }  
 }  
}

Observe quantos objetos foram persistidos.

Observe quantos arquivos foram criados.

# Recuperação de objetos persistentes associados

Crie e execute a classe RecuperarTime:

import java.io.\*;  
  
public class RecuperarTime {  
 public static void main(String[] args) {  
 try {  
 Time time = Time.*abrir*("Bugs.ser");  
 System.*out*.println("Time recuperado com sucesso!");  
 time.exibir();  
 } catch (IOException e) {  
 System.*out*.println("Excecao de I/O");  
 e.printStackTrace();  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("Excecao de classe desconhecida");  
 e.printStackTrace();  
 }  
 }  
}

# Persistência de uma coleção de objetos

Crie a classe Liga:

import java.util.ArrayList;  
import java.io.\*;  
  
public class Liga implements Serializable {  
 ArrayList<Time> times;  
 public Liga() {  
 times = new ArrayList<Time>();  
 times.add(new Time("Big Cats", "Lion", 13, "Jaguar", 21, "Felix"));  
 times.add(new Time("Flyers", "Bat", 30, "Eagle", 22, "Dumont"));  
 times.add(new Time("Swimmers", "Shark", 70, "Dolphin", 40, "Verne"));  
 }  
  
 public void exibir() {  
 for (Time t: times) t.exibir();  
 }  
  
 public void salvar(String nome\_arquivo) throws IOException {  
 FileOutputStream arquivo = new FileOutputStream(nome\_arquivo);  
 ObjectOutputStream gravador = new ObjectOutputStream(arquivo);  
  
 gravador.writeObject(this);  
  
 gravador.close();  
 arquivo.close();  
 }  
  
 public static Liga abrir(String nome\_arquivo) throws IOException, ClassNotFoundException {  
 Liga liga = null;  
  
 FileInputStream arquivo = new FileInputStream(nome\_arquivo);  
 ObjectInputStream restaurador = new ObjectInputStream(arquivo);  
  
 liga = (Liga) restaurador.readObject();  
  
 restaurador.close();  
 arquivo.close();  
  
 return liga;  
 }  
}

Crie e execute a classe CriarLiga:

import java.io.IOException;  
  
public class CriarLiga {  
 public static void main(String[] args) {  
 Liga liga = new Liga();  
 liga.exibir();  
  
 try {  
 liga.salvar("Champions.ser");  
 System.*out*.println("Liga criada e salva com sucesso!");  
 } catch (IOException e) {  
 System.*out*.println("Excecao de I/O");  
 e.printStackTrace();  
 }  
 }  
}

Verifique quantos objetos foram persistidos.

Verifique quantos arquivos foram criados em disco.

# Recuperação de uma coleção de objetos persistentes

Crie e execute a classe RecuperarLiga:

import java.io.IOException;  
  
public class RecuperarLiga {  
 public static void main(String[] args) {  
 try {  
 Liga liga = Liga.*abrir*("Champions.ser");  
 System.*out*.println("Liga recuperada com sucesso!");  
 liga.exibir();  
 } catch (IOException e) {  
 System.*out*.println("Excecao de I/O");  
 e.printStackTrace();  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("Excecao de classe desconhecida");  
 e.printStackTrace();  
 }  
 }  
}