ORIENTAÇÃO A OBJETOS 2

Serialização Java (Disco Rígido)

Prof. Érick Oliveira Rodrigues

SERIALIZAÇÃO

- O Java **provê um mecanismo chamado serialização** de objetos, onde o mesmo é convertido em uma sequência de bytes que contém informação sobre:
  - Dados do objeto
  - Tipo do objeto
  - Tipos de dados armazenados no objeto

UTILIZAÇÃO

- Dêem exemplos de sistemas que podem utilizar cada um dos exemplos abaixo:

1. Objeto -> Serialização -> Arquivo
2. Arquivo -> Desserialização -> Objeto
3. Objeto -> Serialização -> Rede -> Desserialização -> Objeto

SERIALIZAÇÃO

- A serialização é um processo independente da JVM.
  - Um objeto pode ser serializado em Java e desserializado em outra plataforma ou linguagem.

- As classes ObjectInputStream e ObjectOutputStream são streams de alto nível que contém métodos para serializar e converter de volta objetos.

SERIALIZAÇÃO

- Na classe ObjectOutputStream temos o método:

```
public final void writeObject(Object x) throws IOException
```

Serializa um objeto (no caso, x) e o manda pelo output stream. De forma similar, o ObjectInputStream:

```
public final Object readObject() throws IOException, ClassNotFoundException
```

para desserializar um objeto.

SERIALIZAÇÃO

- O método readObject retorna o próximo objeto e o desserializa. O valor de retorno é um objeto da classe **Object**, dessa forma você precisa fazer o downcast do objeto. Vamos supor que temos a seguinte classe:

```
class Employee implements java.io.Serializable {
    public String name;
    public String address;
    public transient int SSN;
    public int number;

    public void mail () {
        System.out.println("Enviando uma correspondencia para " + name + "
" + address);
    }
}
```

SERIALIZAÇÃO

- Para serializar esta classe de forma satisfatória, duas condições precisam ser atingidas:
  - A classe precisa implementar a interface Serializable.
  - Todos os atributos da classe precisam ser serializables (ou serializável). Se um atributo não for serializável, ele deve ser marcado como **transient**.
- Se estiver curioso para saber se uma classe é serializável ou não, a forma de saber é simples: Veja se a mesma implementa a interface Serializable. Se implementar, então sim.

TRANSIENT

- Você pode marcar uma variável ou atributo como **transient** caso você não queira que o mesmo seja serializado.

- Exemplo:
  - Em um jogo online, você não precisa transmitir todas as animações para todos os outros jogadores. Dessa forma, para economizar processamento, tempo de envio pela rede, etc, você pode marcar os atributos de algumas animações como **transient**. Assim, apenas você verá a animação.

SERIALIZANDO UM OBJETO

```
Employee e = new Employee();
e.name = "Reyan Ali";
e.address = "Phokka Kuan, Ambehta Peer";
e.number = 101;

try {
    FileOutputStream fileOut =
            new FileOutputStream("/tmp/employee.ser");
    ObjectOutputStream out = new ObjectOutputStream(fileOut);
    out.writeObject(e);
    out.close();
    fileOut.close();
    System.out.printf("Serialized data is saved in /tmp/employee.ser");
} catch (IOException i) {
    i.printStackTrace();
}
```

SERIALIZANDO UM OBJETO

```
Employee e = null;
try {
    FileInputStream fileIn = new FileInputStream("/tmp/employee.ser");
    ObjectInputStream in = new ObjectInputStream(fileIn);
    e = (Employee) in.readObject();
    in.close();
    fileIn.close();
} catch (IOException i) {
    i.printStackTrace();
    return;
} catch (ClassNotFoundException c) {
    System.out.println("Employee class not found");
    c.printStackTrace();
    return;
}

System.out.println("Deserialized Employee...");
System.out.println("Name: " + e.name);
System.out.println("Address: " + e.address);
System.out.println("Number: " + e.number);
```

EXEMPLO DE SERIALIZAÇÃO

- Para baixar um código completo com exemplo de leitura e escrita, favor acessar:

CR3-0OB

UTILIZAÇÃO DA BIBLIOTECA KRYO

- Caso tenham curiosidade de como construir um banco de dados NoSQL utilizando a biblioteca kryo, que é mais eficiente que a serialização do Java, segue o código de exemplo do meu jogo:

J52-Y7X

EXERCÍCIO / TRABALHO

- Criar um banco de dados não relacional utilizando serialização.

- O banco deverá ter a função de guardar um novo arquivo e achar os arquivos posteriormente.
  - Seria interessante trabalhar com alguma **cache** dos arquivos na memória RAM?