Orientação a objetos

Serialização

Arquivos

- Em Java, a manipulação de arquivos se dá através de algumas classes existentes dentro do java.io.
  - FileWriter
  - BufferedWriter
  - FileReader
  - BufferedReader

Arquivos

- FileWriter:
  - A classe FileWriter é usada para escrever caracteres em um arquivo.
  - Ela é uma classe especializada que deriva da classe abstrata Writer.
  - Ao criar uma instância do FileWriter, você precisa fornecer o caminho do arquivo que deseja escrever.
  - O FileWriter permite gravar dados de caracteres diretamente em um arquivo, substituindo o conteúdo existente, se houver.

Arquivos

- BufferedWriter:
  - A classe BufferedWriter é usada para escrever grandes quantidades de dados de caracteres em um fluxo de saída com melhor desempenho.
  - Ela também é uma classe especializada que deriva da classe abstrata Writer.
  - O BufferedWriter melhora o desempenho escrevendo os dados em um buffer interno antes de gravá-los fisicamente no dispositivo de armazenamento.
  - Isso minimiza o número de operações de gravação física, tornando a gravação mais eficiente.

Arquivos

- FileReader:
  - A classe FileReader é usada para ler caracteres de um arquivo.
  - Ela é uma classe especializada que deriva da classe abstrata Reader.
  - Ao criar uma instância do FileReader, você precisa fornecer o caminho do arquivo que deseja ler.
  - O FileReader permite ler os caracteres do arquivo de forma sequencial.

Arquivos

- BufferedReader:
  - A classe BufferedReader é usada para ler grandes quantidades de caracteres de um fluxo de entrada com melhor desempenho.
  - Ela também é uma classe especializada que deriva da classe abstrata Reader.
  - O BufferedReader melhora o desempenho lendo os dados em um buffer interno antes de retorná-los ao chamador.
  - Isso minimiza o número de operações de leitura física, tornando a leitura mais eficiente.

Arquivos - Exemplo

```
3  public class Aluno {
4      private String nome;
5      private int idade;
6
7      public Aluno(String nome, int idade) {
8          this.nome = nome;
9          this.idade = idade;
10     }
11
12     public String getNome() {
13         return nome;
14     }
15
16     public int getIdade() {
17         return idade;
18     }
19 }
```

Arquivos - Exemplo

```
2
3  import java.io.FileWriter;
4  import java.io.IOException;
5  import java.io.BufferedWriter;
6  import java.io.FileReader;
7  import java.io.BufferedReader;
8
```

Arquivos - Exemplo

```
 9 public class Principal {
10     public static void main(String[] args) {
11         // Criando objetos Aluno
12         Aluno aluno1 = new Aluno("João", 20);
13         Aluno aluno2 = new Aluno("Maria", 22);
14         Aluno aluno3 = new Aluno("Pedro", 19);
15 
16         // Escrevendo os alunos em um arquivo de texto
17         try {
18             FileWriter arquivo = new FileWriter("alunos.txt");
19             BufferedWriter escritor = new BufferedWriter(arquivo);
20 
21             // Escrevendo os alunos no arquivo
22             escritor.write(aluno1.getNome() + "," + aluno1.getIdade());
23             escritor.newLine();
24 
25             escritor.write(aluno2.getNome() + "," + aluno2.getIdade());
26             escritor.newLine();
27 
28             escritor.write(aluno3.getNome() + "," + aluno3.getIdade());
29             escritor.newLine();
30 
31             escritor.close();
32             arquivo.close();
33 
34             System.out.println("Alunos salvos no arquivo alunos.txt");
35         } catch (IOException e) {
36             e.printStackTrace();
37         }
```

```java
public class Principal {
    public static void main(String[] args) {
        // Criando objetos Aluno
        Aluno aluno1 = new Aluno("João", 20);
        Aluno aluno2 = new Aluno("Maria", 22);
        Aluno aluno3 = new Aluno("Pedro", 19);

        // Escrevendo os alunos em um arquivo de texto
        try {
            FileWriter arquivo = new FileWriter("alunos.txt");
            BufferedWriter escritor = new BufferedWriter(arquivo);

            // Escrevendo os alunos no arquivo
            escritor.write(aluno1.getNome() + "," + aluno1.getIdade());
            escritor.newLine();

            escritor.write(aluno2.getNome() + "," + aluno2.getIdade());
            escritor.newLine();

            escritor.write(aluno3.getNome() + "," + aluno3.getIdade());
            escritor.newLine();

            escritor.close();
            arquivo.close();

            System.out.println("Alunos salvos no arquivo alunos.txt");
        } catch (IOException e) {
            e.printStackTrace();
        }
```

Arquivos - Exemplo

```
39          // Lendo os alunos do arquivo
40          try {
41              FileReader arquivo = new FileReader("alunos.txt");
42              BufferedReader leitor = new BufferedReader(arquivo);
43
44              System.out.println("Alunos lidos do arquivo:");
45
46              String linha;
47              while ((linha = leitor.readLine()) != null) {
48                  String[] campos = linha.split(",");
49
50                  String nome = campos[0];
51                  int idade = Integer.parseInt(campos[1]);
52
53                  Aluno aluno = new Aluno(nome, idade);
54
55                  System.out.println("Nome: " + aluno.getNome() + ", Idade: " + aluno.getIdade());
56              }
57
58              leitor.close();
59              arquivo.close();
60          } catch (IOException e) {
61              e.printStackTrace();
62          }
63      }
64  }
```

```java
        // Lendo os alunos do arquivo
        try {
            FileReader arquivo = new FileReader("alunos.txt");
            BufferedReader leitor = new BufferedReader(arquivo);

            System.out.println("Alunos lidos do arquivo:");

            String linha;
            while ((linha = leitor.readLine()) != null) {
                String[] campos = linha.split(",");

                String nome = campos[0];
                int idade = Integer.parseInt(campos[1]);

                Aluno aluno = new Aluno(nome, idade);

                System.out.println("Nome: " + aluno.getNome() + ", Idade: " + aluno.getIdade());
            }

            leitor.close();
            arquivo.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Arquivo - Desafio

- Faça uma classe para manipulação de arquivos, que tenha métodos:
  - Além de possíveis construtores e getters e setters (não obrigatório), que grave no arquivo o que for passado por parâmetro e outro que retorne uma lista de objetos (Alunos, por exemplo).

Arquivo - Desafio

```java
public class Arquivo {
    private FileWriter arqw;
    private BufferedWriter escritor;
    private FileReader arqr;
    private BufferedReader leitor;
    private List<Aluno> listAlunos;
    public String nomeArquivo;

    public Arquivo(String nomeArquivo) {
        this.nomeArquivo = nomeArquivo;
        listAlunos = new ArrayList<>();
    }

```

```java
public void gravaArquivo(Aluno a) {
    // Escrevendo os alunos em um arquivo de texto
    try {
       // Escrevendo os alunos no arquivo
        arqw = new FileWriter(nomeArquivo+".txt", true);
        escritor = new BufferedWriter(arqw);
        escritor.write(a.getNome() + "," + a.getIdade());
        escritor.newLine();
        escritor.close();
        arqw.close();

        System.out.println("Alunos salvos no arquivo alunos.txt");
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Arquivo - Desafio

```
42      public List<Aluno> leArquivo() {
43           // Lendo os alunos do arquivo
44          System.out.println("Alunos lidos do arquivo:");
45          try {
46              arqr = new FileReader(nomeArquivo+".txt");
47              leitor = new BufferedReader(arqr);
48              String linha;
49              while ((linha = leitor.readLine()) != null) {
50                  String[] campos = linha.split(",");
51
52                  String nome = campos[0];
53                  int idade = Integer.parseInt(campos[1]);
54                  Aluno aluno = new Aluno(nome, idade);
55                  listAlunos.add(aluno);
56              }
57
58              leitor.close();
59              arqr.close();
60
61          } catch (IOException e) {
62              e.printStackTrace();
63          }
64          return listAlunos;
65      }
66  }
```

Arquivo - Desafio

```
11 public class Principal {
12     public static void main(String[] args) {
13         // Criando objetos Aluno
14         Aluno aluno1 = new Aluno("João", 20);
15         Aluno aluno2 = new Aluno("Maria", 22);
16         Aluno aluno3 = new Aluno("Alziras", 19);
17         List<Aluno> lista = new ArrayList<>();
18         Arquivo arquivo = new Arquivo("alunos");
19         arquivo.gravaArquivo(aluno1);
20         arquivo.gravaArquivo(aluno2);
21         arquivo.gravaArquivo(aluno3);
22 
23         lista = arquivo.leArquivo();
24 
25         for(Aluno a : lista) {
26             System.out.println("Nome: " + a.getNome() + ", Idade: " + a.getIdade());
27         }
28 
29     }
30 }
```