

Unidade Curricular:

Integração de Sistemas de Informação

Tema da Ficha Prática:

Manipulação de Ficheiros

Objectivos:

Pretende-se com esta ficha prática que os alunos interajam com o conceito de ficheiros nas linguagens de programação: JAVA, C# e VB . Net.

Bibliografia:

Para apoio a esta ficha os alunos devem consultar os apontamentos teóricos e práticos da disciplina bem como de outros recursos online.

Índice

| | |
|--|---|
| 1. Manipulação de ficheiros usando a linguagem de programação JAVA..... | 2 |
| 1.1 Verificar se um ficheiro ou directoria existe | 2 |
| 1.2 Criar um ficheiro..... | 2 |
| 1.3 Obter o tamanho de um ficheiro | 3 |
| 1.4 Remover um ficheiro | 3 |
| 1.5 Renomear um ficheiro ou directoria..... | 4 |
| 1.6 Mover um ficheiro ou directoria para outra directoria | 4 |
| 1.7 Obter a data de modificação de um ficheiro ou directoria | 5 |
| 2. Exercícios a elaborar..... | 6 |
| 2.1 Implemente as funcionalidades apresentadas no ponto 1 mas utilizando C# .Net..... | 6 |
| 2.2 Implemente as funcionalidades apresentadas no ponto 1 mas utilizando VB .Net | 6 |

1. Manipulação de ficheiros usando a linguagem de programação JAVA

Teste as seguintes operações utilizando ficheiros:

1.1 Verificar se um ficheiro ou directoria existe

```
import java.io.File;

public class DetermineFileDirExists {

    private static void doTest() {

        // Create a File object
        File file1 = new File("README_InputFile.txt");
        File file2 = new File("BlaBlaBla.txt");

        boolean b = file1.exists();
        System.out.println();
        System.out.println("Does File/Dir " + file1 + " exist? (" + b + ")\n");

        b = file2.exists();
        System.out.println();
        System.out.println("Does File/Dir " + file2 + " exist? (" + b + ")\n");

    }

    public static void main(String[] args) {
        doTest();
    }

}
```

1.2 Criar um ficheiro

```
import java.io.File;
import java.io.IOException;

public class CreateFile {

    private static void doCreate() {

        // Create a File object
        File file = new File("NewFile.txt");

        boolean success = false;

        try {
            // Create file on disk (if it doesn't exist)
            success = file.createNewFile();
        } catch (IOException e) {
            e.printStackTrace();
        }

        if (success) {
            System.out.println("File did not exist and was created.\n");
        } else {
            System.out.println("File already exists.\n");
        }
    }

}
```

```
    }

    }

    public static void main(String[] args) {
        doCreate();
    }

}
```

1.3 Obter o tamanho de um ficheiro

```
import java.io.File;
import java.io.IOException;

public class SizeOfFile {

    private static void doCheckSize() {

        // Create a File object
        File file = new File("README_InputFile.txt");

        // Get the number of bytes in the file
        long fileLength = file.length();
        System.out.println(
            "The length (in bytes) of file " + file +
            " is " + fileLength + ".\n");

    }

    /**
     * Sole entry point to the class and application.
     * @param args Array of String arguments.
     */
    public static void main(String[] args) {
        doCheckSize();
    }

}
```

1.4 Remover um ficheiro

```
import java.io.File;

public class DeletingFile {

    private static void doCreate() {

        // Create a File object
        File file = new File("NewFile.txt");

        boolean success = file.delete();

        if (success) {
            System.out.println("File was successfully deleted.\n");
        } else {
            System.out.println("File was not successfully deleted.\n");
        }

    }

}
```

```
}

public static void main(String[] args) {
    doCreate();
}

}
```

1.5 Renomear um ficheiro ou directoria

```
import java.io.File;
public class RenamingFileOrDir {

    private static void doRename() {

        // A File (or Directory) with the old name
        File file1 = new File("OldFile.txt");
        System.out.println("File1 = " + file1);

        // A File (or Directory) with the new name
        File file2 = new File("NewFile.txt");
        System.out.println("File2 = " + file2);

        // Rename File (or Directory)
        boolean success = file1.renameTo(file2);

        if (success) {
            System.out.println("File was successfully renamed.\n");
        } else {
            System.out.println("File was not successfully renamed.\n");
        }

    }

    public static void main(String[] args) {
        doRename();
    }

}
```

1.6 Mover um ficheiro ou directoria para outra directoria

```
import java.io.File;

public class MoveFileOrDir {

    private static void doMove() {

        // File (or Directory) to be moved
        File file = new File("NewFile.txt");

        // Destination directory
        File dir = new File("./new_dir");

        // Move file to a new directory
        boolean success = file.renameTo(new File(dir, file.getName()));

        if (success) {
            System.out.println("File was successfully moved.\n");
        }

    }

}
```

```

    } else {
        System.out.println("File was not successfully moved.\n");
    }

}

/**
 * Sole entry point to the class and application.
 * @param args Array of String arguments.
 */
public static void main(String[] args) {
    doMove();
}
}

```

1.7 Obter a data de modificação de um ficheiro ou directoria

```

import java.io.File;
import java.util.Date;

public class GetSetModificationTime {

    private static void doModifyTime() {

        // Create a File object
        File file = new File("README_InputFile.txt");

        // Get the last modified time.
        // 0L is returned if the file does not exist.
        long modifiedTime = file.lastModified();

        System.out.println();
        System.out.println("The last modified time of file " + file);
        System.out.println("-----");
        System.out.println(" - milliseconds since midnight, January 1, 1970, GMT = "
+ modifiedTime);
        System.out.println(" - date = " + new Date(modifiedTime));
        System.out.println();

        // Set the last modified time
        long newModifiedTime = System.currentTimeMillis();
        boolean success = file.setLastModified(newModifiedTime);

        System.out.println("Setting a new modified time for the file " + file);
        System.out.println("-----");
        System.out.println(" - new milliseconds = " + newModifiedTime);
        System.out.println(" - date = " + new Date(newModifiedTime));
        System.out.println();

    }

    public static void main(String[] args) {
        doModifyTime();
    }

}

```

2. Exercícios a elaborar

2.1 Implemente as funcionalidades apresentadas no ponto 1 mas utilizando C#.Net

2.2 Implemente as funcionalidades apresentadas no ponto 1 mas utilizando VB.Net