File Handling

- Creating file
- Writing & appending to a file
- Reading file data
- Deleting file

You Must Know

- **Import java.io.*;**

  it import's the input and output function's, and more we are used in file operation's like open close save etc....

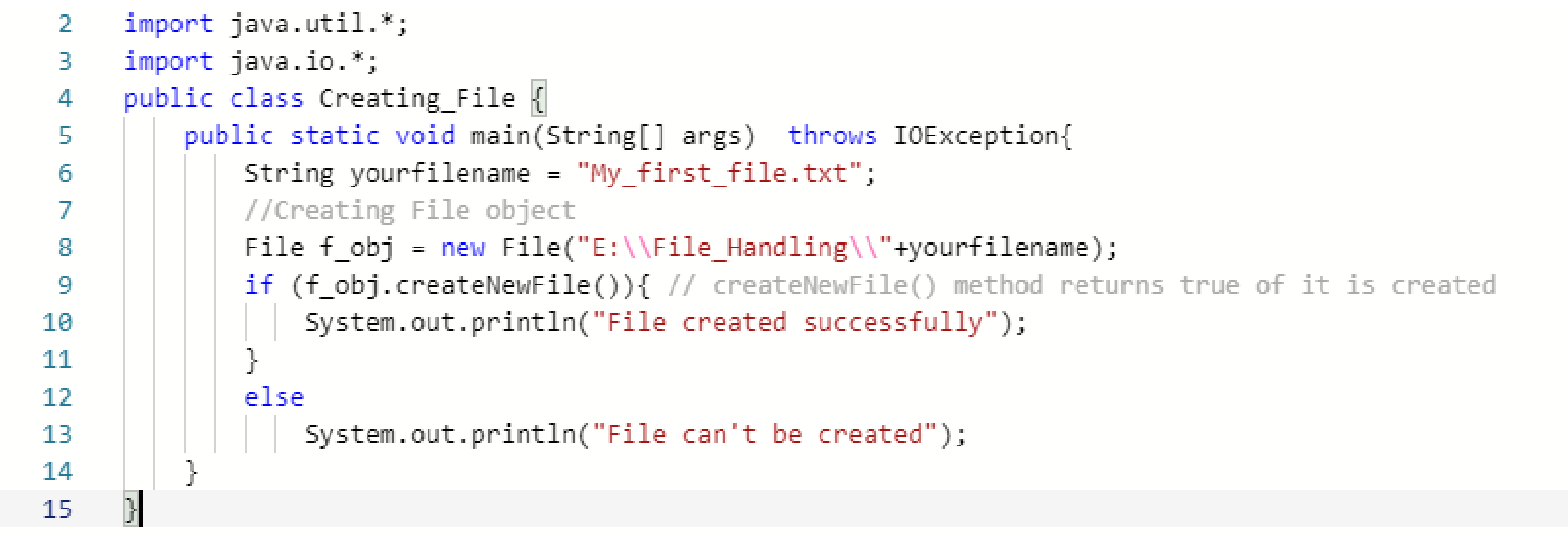
- **throws IOException**

  The **throws** keyword indicates that a certain method can potentially "**throw**" a certain exception. ... When is **IOException thrown IOException** is the base exception class used for handling the failures. Here we will use this for File Handling.

Creating File:

- **File file_object = new File(filename);**
  Java File class is used for creation of **files** and directories, **file** searching, **file** deletion, etc. The **File object** represents the actual **file**/directory on the disk
- **.createNewFile() method:**
  This function creates new empty file. The function returns true if the abstract file path does not exist and a new file is created. It returns false if the filename already exists.

Source Code



```java
import java.util.*;
import java.io.*;
public class Creating_File {
    public static void main(String[] args)  throws IOException{
        String yourfilename = "My_first_file.txt";
        //Creating File object
        File f_obj = new File("E:\\File_Handling\\"+yourfilename);
        if (f_obj.createNewFile()){ // createNewFile() method returns true of it is created
            System.out.println("File created successfully");
        }
        else
            System.out.println("File can't be created");
    }
}
```

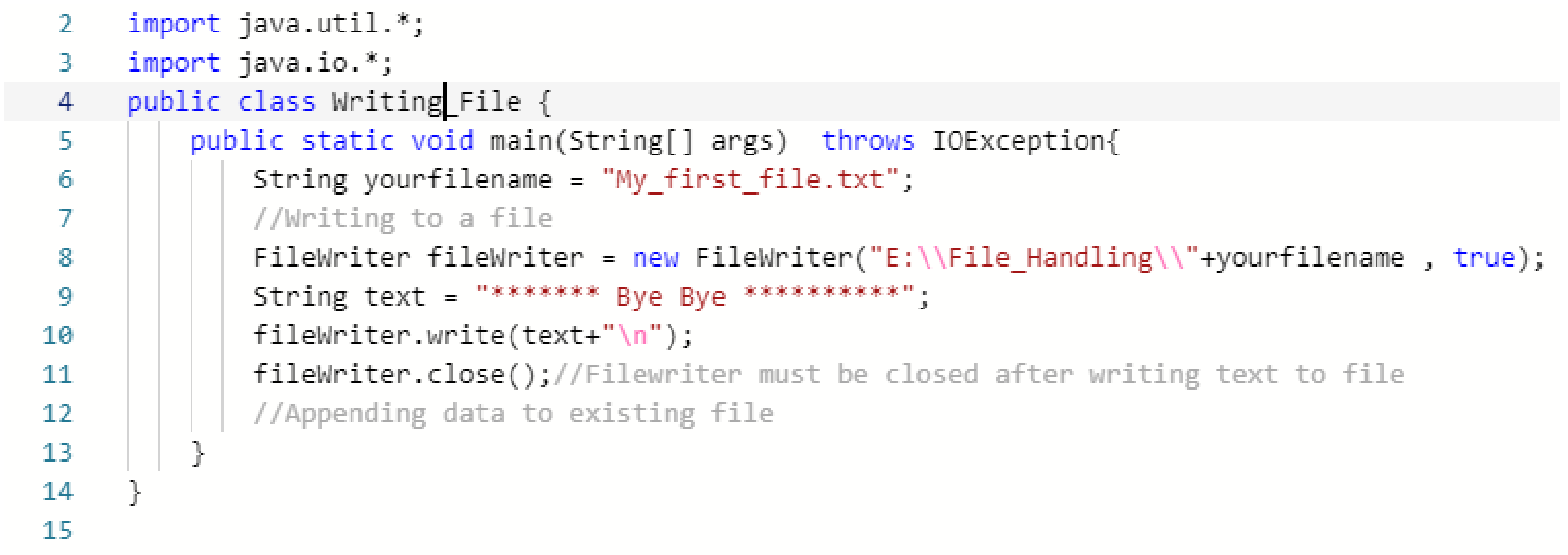
Writing content to an existing file

- FileWriter fileWriter = new FileWriter (yourfilename , true);

  Java FileWriter class is used to write character-oriented data to a file. It is character-oriented class which is used for file handling in **java**

- **write() method:**
  write () methods allow you to write character(s) or strings to a file.

Source Code:

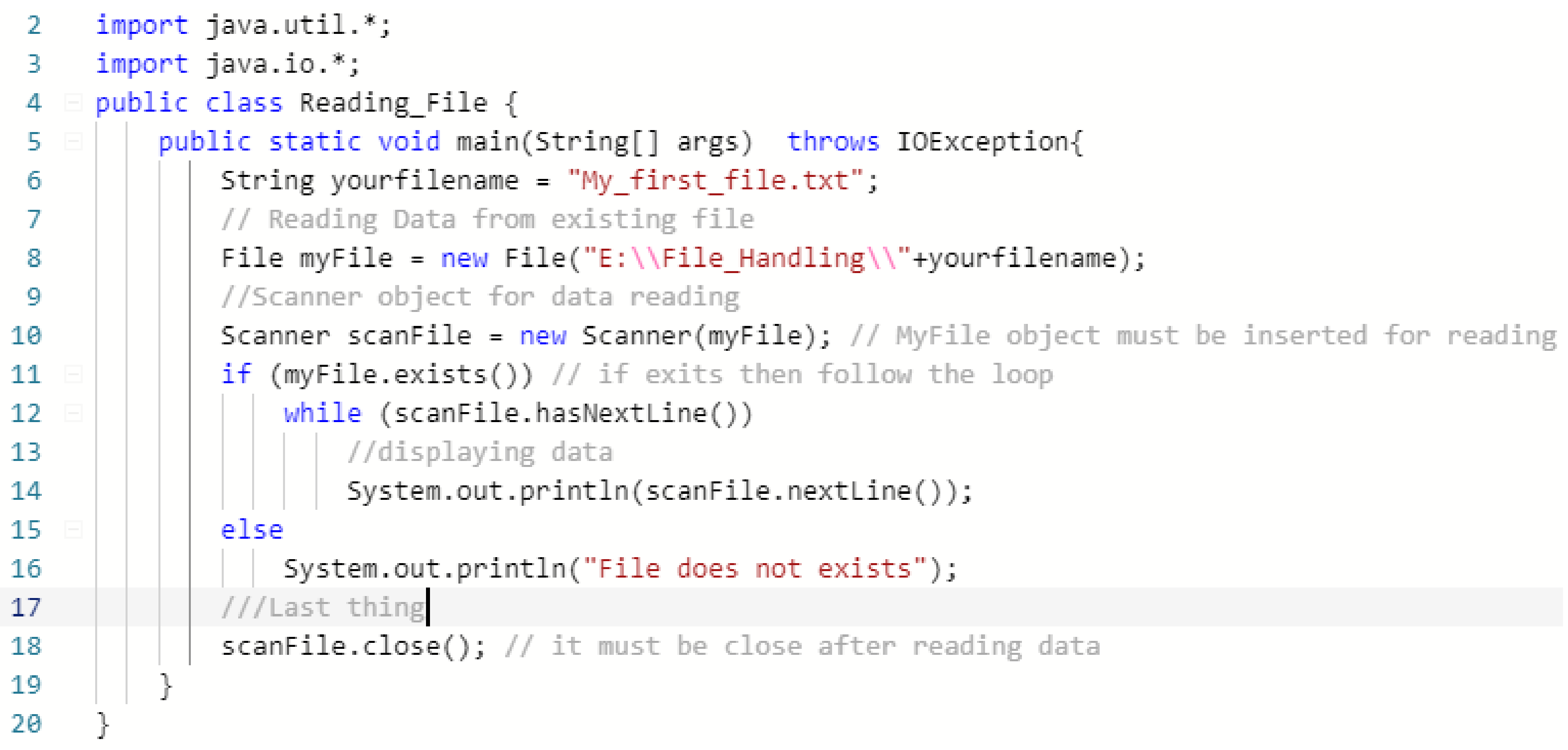


```java
import java.util.*;
import java.io.*;
public class Writing_File {
    public static void main(String[] args)  throws IOException{
        String yourfilename = "My_first_file.txt";
        //Writing to a file
        FileWriter fileWriter = new FileWriter("E:\\File_Handling\\"+yourfilename , true);
        String text = "******* Bye Bye **********";
        fileWriter.write(text+"\n");
        fileWriter.close();//Filewriter must be closed after writing text to file
        //Appending data to existing file
    }
}
```

Method For Reading Data:

- The **hasNextLine()** is a method of **Java Scanner** class which is used to check if there is another line in the input of this scanner. It returns true if it finds another line, otherwise returns false.
- To read the contents of a **file**, **Scanner** class provides various **constructors**. Used to read data from the file represented by the given File object.
- Scanner must be closed (***scanner.close ()***) after reading data.

Source Code:



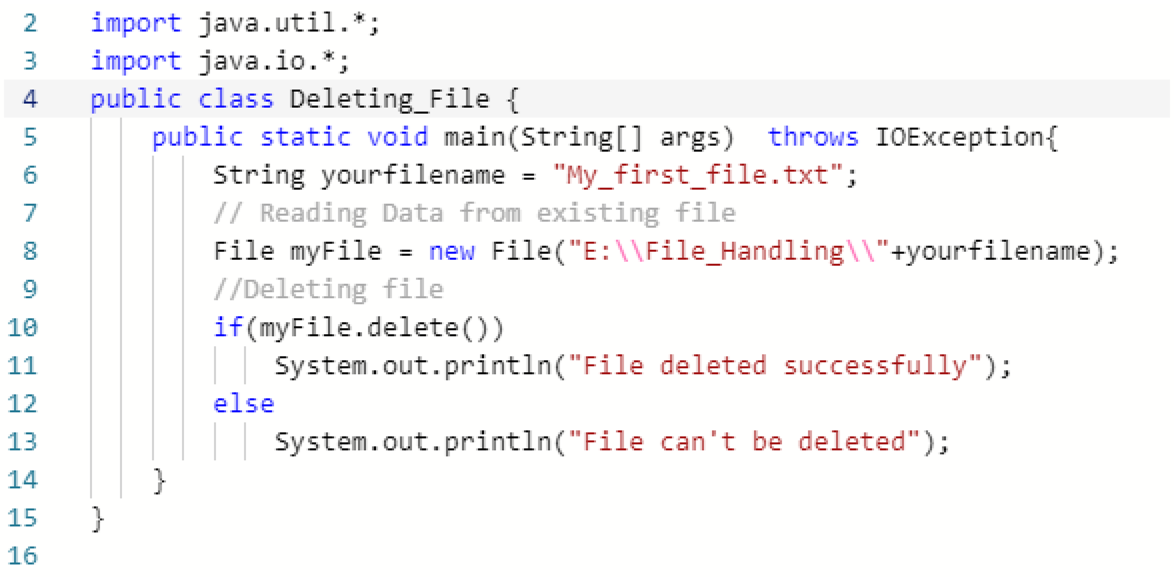
```java
import java.util.*;
import java.io.*;
public class Reading_File {
    public static void main(String[] args)  throws IOException{
        String yourfilename = "My_first_file.txt";
        // Reading Data from existing file
        File myFile = new File("E:\\File_Handling\\"+yourfilename);
        //Scanner object for data reading
        Scanner scanFile = new Scanner(myFile); // MyFile object must be inserted for reading
        if (myFile.exists()) // if exits then follow the loop
            while (scanFile.hasNextLine())
                //displaying data
                System.out.println(scanFile.nextLine());
        else
            System.out.println("File does not exists");
        ///Last thing
        scanFile.close(); // it must be close after reading data
    }
}
```

<u>Method:</u> *File.delete()*

- Using **java**.io.*

  File.**delete() function**: Deletes the **file** or directory denoted by this abstract path name. Syntax: public Boolean **delete ()** Returns: true if and only if the **file** or directory is successfully **deleted**; false otherwise.

  Source Code:



```
import java.util.*;
import java.io.*;
public class Deleting_File {
    public static void main(String[] args)  throws IOException{
        String yourfilename = "My_first_file.txt";
        // Reading Data from existing file
        File myFile = new File("E:\\File_Handling\\"+yourfilename);
        //Deleting file
        if(myFile.delete())
            System.out.println("File deleted successfully");
        else
            System.out.println("File can't be deleted");
    }
}
```

Description

- PDF source code

Like Share

Directed By:

RUSLAN BABAR

For Video tutorial visit

YouTube: https://youtu.be/qCrEVUY-z3I