Java OOP 10128

Files

Pini shlomi

Java Files

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Files are an essential part of any programming language as they allow us to store and retrieve data.
- we can work with different types of files, including text file, CSV file, and binary file.
- **Text file**: store data in plain text format.
- **CSV** (Comma Separated Values) file : store data in tabular format.
- **Binary file**: store data in binary format.

File Methods

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
import java.io.File;

File f = new File("file_example.txt");
System.out.println("file name: " + f.getName());
System.out.println("is file exist? " + f.exists());
System.out.println("is file?  " + f.isFile());
System.out.println("is directory? " + f.isDirectory());
System.out.println("can read? " + f.canRead());
System.out.println("can write? " + f.canWrite());
System.out.println("can execute? " + f.canExecute());
System.out.println("file path: " + f.getPath());
System.out.println(" " + f.getAbsolutePath());
```

```
file name: file_example.txt
is file exist? true
is file? true
is directory? false
can read? true
can write? true
can execute? true
file path: C:\pini\Afeka…\file_example.tx
C:\pini\Afeka…\file_example.txt
```

Get File Path

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
import java.io.File;

String packagePath =
     this.getClass().getPackage().getName()
     .replace('.', File.separatorChar);

filePath = System.getProperty("user.dir")
          + File.separator + "src"
          + File.separator + packagePath
          + File.separator;
```

Text file - Reading

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
File f = new File(filePath + filename);

Scanner s = new Scanner(f);

StringBuffer sb = new StringBuffer();

while (s.hasNext()) {
    sb.append(s.nextLine() + "\n");
}

s.close();
```

throws FileNotFoundException

Text file - Writing

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
File f = new File(filePath + filename);

PrintWriter pw = new PrintWriter(f);

pw.print(content);

s.close();
```

throws FileNotFoundException

Text file

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
class TextFile {
    private final String filePath;
    private final String filename;
    public TextFile(String filename) {
        String packagePath = this.getClass().getPackage().getName().replace('.', File.separatorChar);
        filePath = System.getProperty("user.dir") + File.separator + "src"
                + File.separator + packagePath + File.separator;
        this.filename = filename;
    }
    public String readFileContent() throws FileNotFoundException {
        File f = new File(filePath + filename);
        Scanner s = new Scanner(f);
        StringBuilder sb = new StringBuilder();
        while (s.hasNext()) {
            sb.append(s.nextLine()).append("\n");
        }
        s.close();
        return sb.toString();
    }
    public void writeFileContent(String content) throws FileNotFoundException {
        File f = new File(filePath + filename); // new FileWriter(filePath + filename, true); throws IOException
        PrintWriter pw = new PrintWriter(f);
        pw.print(content);
        pw.close();
    }
}
```

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל : כתיבה וקריאה קבצי טקסט

עליכם לממש את החסר בקבצי ה-starter.

בקבצים קיים main אשר לאחר ההרצה מפיק את הפלט הבא:

```
File content before adding:
Start file content...

File content after adding:
Start file content...
some content
```

קישור ל-starter

אתר להורדת ספריות מ-github.

CSV File (Comma Separated Values)

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Technically it possible to use a Scanner to read a CSV.
- It's recommended to use BufferedReader.

- Why?
  - Efficiency: BufferedReader class is specifically designed for reading large amounts of text efficiently. It reads data from a file in chunks, rather than one character at a time like the Scanner class..
  - Handling of line endings: BufferedReader class automatically handles different line endings, while the Scanner class does not.

CSV file - Reading

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
BufferedReader  reader = new BufferedReader(new FileReader(filename));
StringBuilder sb = new StringBuilder();
String line = reader.readLine();
String[] header = line.split(",");
line = reader.readLine();
while (line != null) {
      String[] fields = line.split(",");
      for (int i = 0; i < header.length; i++) {
            sb.append(header[i])
            .append(": ").append(fields[i]).append("\n");
      }
      line = reader.readLine();
}
reader.close();
```

throws IOException

persons.csv

```
Name,Age,Gender
Ron Shalev,27,M
Taly Dor,25,F
```

CSV file - Writing

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
BufferedWriter writer = new BufferedWriter(
                        new FileWriter(filename, true));

writer.write(content);


writer.close();
```

throws IOException

append

persons.csv(after)

```
Name,Age,Gender
Ron Shalev,27,M
Taly Dor,25,F
Yael Mor,30,F
```

CSV file

```
class CSVFile {
    private final String filePath;
    private final String filename;
    public CSVFile(String filename) {
        String packagePath = this.getClass().getPackage().getName().replace('.', File.separatorChar);
        filePath = System.getProperty("user.dir") + File.separator + "src"
              + File.separator + packagePath + File.separator;
        this.filename = filename;
    }
    public String readFileContent() throws IOException {
        BufferedReader  reader = new BufferedReader(new FileReader(filePath+filename));
        StringBuilder sb = new StringBuilder();
         String line = reader.readLine();
         String[] header = line.split(",");
        line = reader.readLine();
         while (line != null) {
            String[] fields = line.split(",");
            for (int i = 0; i < header.length; i++) {
                sb.append(header[i]).append(": ").append(fields[i]).append("\n");
            }
            line = reader.readLine();
         }
        reader.close();
        return sb.toString();
    }
    public void writeFileContent(String content) throws IOException {
        BufferedWriter writer = new BufferedWriter(new FileWriter(filePath + filename, true));
        writer.write(content);
        writer.close();
    }
}
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

persons.csv

```
Name,Age,Gender
Ron Shalev,27,M
Taly Dor,25,F
```

console

```
Name: Ron Shalev
Age: 27
Gender: M
Name: Taly Dor
Age: 25
Gender: F
```

persons.csv(after)

```
Name,Age,Gender
Ron Shalev,27,M
Taly Dor,25,F
Yael Mor,30,F
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל : כתיבה וקריאה קבצי CSV

עליכם לממש את החסר בקבצי ה-starter.

בקבצים קיים main אשר לאחר ההרצה מפיק את הפלט הבא:

File content before adding:
Name: Ron Shalev, Age: 27, Gender: M,
Name: Taly Dor, Age: 25, Gender: F,

File content after adding:
Name: Ron Shalev, Age: 27, Gender: M,
Name: Taly Dor, Age: 25, Gender: F,
Name: Yael Mor, Age: 30, Gender: F,

קישור ל-starter

אתר להורדת ספריות מ-github.

Binary File

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Binary files store data as a sequence of binary digits (bits) that represent specific values or instructions.
- Binary files include image files (such as JPEG or PNG), audio files (such as MP3 or WAV), video files (such as MPEG or AVI), and executable files (such as EXE or DLL).
- To read and write binary files we are using ObjectInputStream and ObjectOutputStream classes, which are used for reading and writing bytes

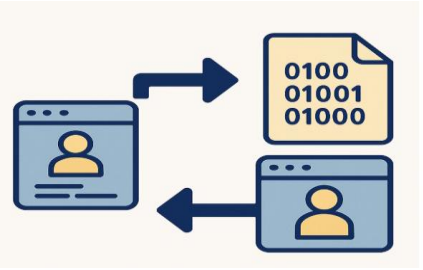
Serialization and deserialization

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- **Serialization**
  converting an object's state (its instance variables) into a stream of bytes, which can then be saved to a file, sent over a network, or stored in a database.

- **Deserialization**
  reverse process of converting the stream of bytes back into an object's state.

Serializable interface



אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- indicates that its state can be serialized.

- Serialization is useful for:
  - Storing and retrieving objects from a file or database
  - Sending objects over a network
  - Sharing objects between different applications or systems

Student

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public class Student implements Serializable{
    private static final long serialVersionUID = -4585905327894841518L;
    private String name;
    private int id;
    private float average;

    public Student( int id, String name, float average) {
        this.name = name;
        this.id = id;
        this.average = average;
        System.out.println("The Student " + name + " is created");
    }

    . . .  Getter's and Setter's

    @Override
    public String toString() {
          return "Id=" + id + "\t Name=" + name + "\t Average=" + average;
    }
}
```

Binary file - Writing

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
Student[] students = {
      new Student(1, "Yosi Mor", 79),
      new Student(2, "Hila Zur", 84)
};
```

```
ObjectOutputStream file = new ObjectOutputStream(
      new FileOutputStream(filePath + filename));
```

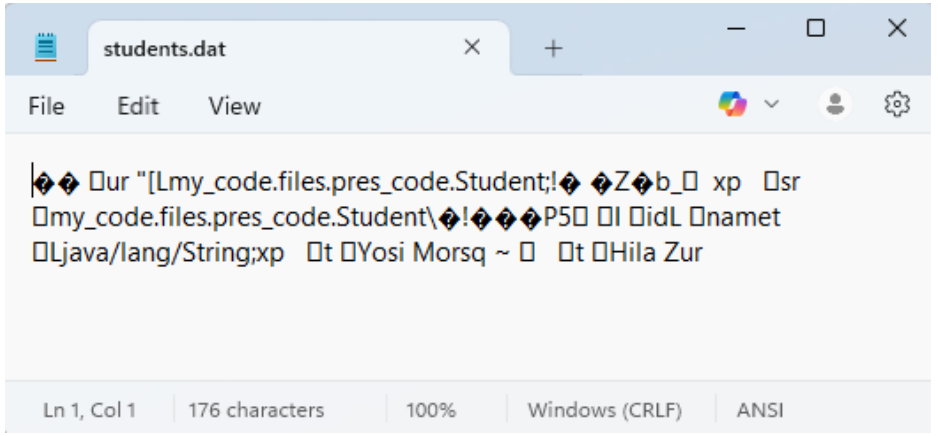
throws IOException

```
file.writeObject(students);
```

```
file.close();
```

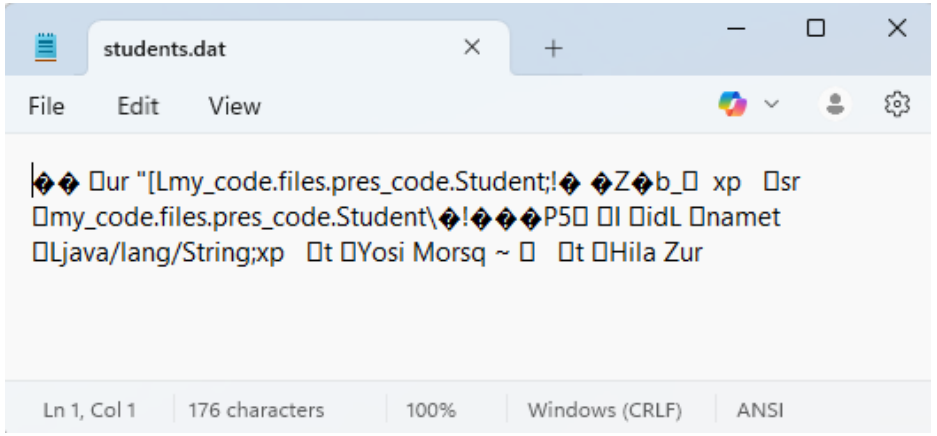
console

```
The Student Yosi Mor is created
The Student Hila Zur is created
```



Binary file - Reading

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992




```
ObjectInputStream file = new ObjectInputStream(
        new FileInputStream(filePath + filename));

Student[] students = (Student[])file.readObject();


file.close();
```

throws IOException, ClassNotFoundException

Binary File write and read

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public class BinaryFileReadWrite{
   public static void main(String[] args) throws IOException, ClassNotFoundException {
      String filename = "students.dat";
      BinaryFile binary File = new BinaryFile(filename);
      Student[] students = {
           new Student(1, "Yosi Mor", 79),
           new Student(2, "Hila Zur", 84)
      };
      binary binary File(students);
      students = binary File.readStudents();
      System.out.println(Arrays.toString(students));
   }
}
```

console

```
The Student Yosi Mor is created
The Student Hila Zur is created
[Id=1 Name=Yosi Mor Average=79.0, Id=2 Name=Hila Zur Average=84.0]
```

Binary File write and read

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
class BinaryFile {
   private final String filePath;
   private final String filename;
   public BinaryFile(String filename) {
     String packagePath = this.getClass().getPackage()
         .getName().replace('.', File.separatorChar);
     filePath = System.getProperty("user.dir") + File.separator + "src"
         + File.separator + packagePath + File.separator;
     this.filename = filename;
   }
   public void saveStudents(Student[] students) throws IOException {
     ObjectOutputStream file = new ObjectOutputStream(
         new FileOutputStream(filePath + filename));
     file.writeObject(students);
     file.close();
   }
   public Student[] readStudents() throws IOException, ClassNotFoundException {
     ObjectInputStream file = new ObjectInputStream(
         new FileInputStream(filePath + filename));
     Student[] students = (Student[])file.readObject();
     file.close();
     return students;
   }
}
```

Transient

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Used to indicate that a field should not be serialized when an object is written to a binary file.
- Useful when there are fields that should not be persisted or when the serialized data size needs to be minimized.
- When an object is deserialized from a binary file, the value of a transient field will be set to its default value.

Static

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Used to indicate that a field should not be serialized because it belongs to the class, not to an instance of the class.
- Static fields are not serialized because they are not part of an object's state.
- When an object is deserialized from a binary file, the value of a static field will be the value it had when the class was loaded.

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל : יצירת ספר טלפון בקובץ בינארי

עליכם לממש את החסר בקבצי ה-starter.

בקבצים קיים main אשר לאחר ההרצה מפיק את הפלט הבא:

```
PhoneBook list:
Contact [phoneNumber=054-4652124, name=Taly, group=Family]
Contact [phoneNumber=050-1234124, name=Mor, group=Friends]
Contact [phoneNumber=054-8594521, name=Kobi, group=Work]
```

קישור ל-starter

אתר להורדת ספריות מ-github.