Report On The Spa

Task 6

This pdf file is dedicated on reporting about the code for the spa. First thing that stands out is the interface “LegalEntity”. This basically acts as the face of the organization where the address and Vat number are located:

```
public interface LegalEntity {
    String getAddress();
    String getVatNumber();

}
```

this is a test build so the address and Vat number aren’t given, it’s just a structure.

The next important piece of code is the “Therapist”. This class contains the basic information about the employed therapists: their names, surnames and license numbers

```
public Therapist(String name, String surname, String licenseNumber) {
    this.name = name;
    this.surname = surname;
    this.licenseNumber = licenseNumber;
```

due to this being a test build, there is only the structure where one can write the details about the therapists

```
public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getSurname() {
    return surname;
}

public void setSurname(String surname) {
    this.surname = surname;
}

public String getLicenseNumber() {
    return licenseNumber;
}

public void setLicenseNumber(String licenseNumber) {
    this.licenseNumber = licenseNumber;
}
```

After this comes the spa class itself. The primary use of this class is to combine everything else and be used by the test runner. For this task the test runner wasn't requested:

```
public class Spa implements LegalEntity {
    private String address;
    private String vatNumber;
    private List<Therapist> therapists = new ArrayList<>();

    public Spa(String address, String vatNumber) {
        this.address = address;
        this.vatNumber = vatNumber;
    }
```

Another important thing this class does is save information. Down lower the is a set of code where the details about therapist appointments are saved:

```
public void saveTherapistsToFile(String filename) throws IOException {
    try (BufferedWriter writer = new BufferedWriter(new
FileWriter(filename))) {
        for (Therapist t : therapists) {
            writer.write(t.getName() + "," + t.getSurname() + "," +
t.getLicenseNumber());
            writer.newLine();
        }
    }
}

public void loadTherapistsFromFile(String filename) throws IOException {
    therapists.clear();
    try (BufferedReader reader = new BufferedReader(new
FileReader(filename))) {
        String line;
        while ((line = reader.readLine()) != null) {
            String[] parts = line.split(",");
            if (parts.length == 3) {
                therapists.add(new Therapist(parts[0], parts[1], parts[2]));
            }
        }
    }
```

This code allows us to save previous requests and upload them whenever needed. This does require the test runner to actually make and save appointments.