

hw22-FixThePitfalls

Solution:

Oddball Solution

Simplify the isValidId method in the Validations class.

```
public boolean isValidId(String value) {
    String regex = "^[LOlo][a-zA-Z0-9]+$";
    Pattern pattern = Pattern.compile(regex);
    Matcher matcher = pattern.matcher(value);
    return matcher.matches();
}
```

Type embedded in names:

Change it to employeeFilePath to avoid having the type (file) in the name.

```
public class EmployeeManager {
    private final String employeeFilePath = "employees.json";
}
```

Conditional Complexity:

Use a stream to simplify

```
@Override
public Date deserialize(JsonElement json, Type typeOfT, JsonDeserializationContext context) throws JsonParseException {
    String dateStr = json.getAsString();
    return Arrays.stream(inputFormats)
        .map(format -> {
            try {
                return format.parse(dateStr);
            } catch (ParseException e) {
                return null;
            }
        })
        .filter(Objects::nonNull)
        .findFirst()
        .orElseThrow(() -> new JsonParseException("Unparseable date: " + dateStr));
}
```

Long Methods:

Split into smaller methods

```

private List<Employee> loadEmployees() {
    try (Reader reader = new FileReader(employeeFilePath)) {
        return parseEmployeesFromJson(reader);
    } catch (IOException e) {
        handleLoadError(e);
        return new ArrayList<>();
    }
}

private List<Employee> parseEmployeesFromJson(Reader reader) {
    Type listType = new TypeToken<List<Employee>>(){}.getType();
    return GSON.fromJson(reader, listType);
}

private void handleLoadError(IOException e) {
    System.out.println("Error al cargar los empleados: " + e.getMessage());
}

```

Dead Code:

Remove code from getIntInput

Non-descriptive variable names:

They should have clearer names that indicate their exact purpose, such as gsonInstance or employeeList or employees.

```

public class EmployeeManager {
    private final String employeeFilePath = "employees.json";
    private Gson gsonInstance = new GsonBuilder().setPrettyPrinting().create();
    private final List<Employee> employeeList;
    public EmployeeManager() {
        gsonInstance = new GsonBuilder()
            .setPrettyPrinting()
            .registerTypeAdapter(Date.class, new DateAdapter())
            .create();
        List<Employee> loadedEmployees = loadEmployees();
        employeeList = (loadedEmployees != null) ? loadedEmployees : new ArrayList<>();
    }
}

```