Petre Flaviu-Mihai
Grupa 10LF332

Proiect TaskOrganization

Lab1: Introducere în Java (output, tipuri de valori, funcții):

La începutul aplicației afișez un mesaj de întâmpinare în funcția “StartApp()” din clasa “App” și descriu funcționalitățile aplicației:

```
//region Methodes
public void StartApp() {  1 usage
    _scanner = new Scanner(System.in);
    System.out.println("Bine ai venit în aplicatia de organizare a task-urilor!");
    System.out.println("Alege o optiune:");
    System.out.println("1. Adaugă un task.");
    System.out.println("2. Afișează task-urile.");
    System.out.println("3. Șterge un task.");
    System.out.println("4. Setează task finalizat");
    System.out.println("5. Verifică dacă termenul limită al unui task a expirat.");
    System.out.println("6. Sortează task-urile după prioritatea");
    System.out.println("7. Verifică dacă o persoană este implicată într-un task.");
    System.out.println("8. Afișează task-urile care sunt urgente în cadrul locului de muncă.");
    System.out.println("9. Ieșire.");
}
```

Pe parcursul aplicației voi folosi următoarele tipuri de valori: int, String, boolean, LocalDate.

```
private String _title;  3 usages
private String _description;  3 usages
private int _priority;  3 usages
private LocalDate _deadline;  3 usages
private boolean _isCompleted;  3 usages
private final String _type = "simple";  1 usage
```

Lab2: Introducere în Java (input, for, while, switch, if):

- Utilizatorul introduce o opțiune (de la 1 la 9);
- Folosesc switch pentru a gestiona acțiunile;
- Utilizez while pentru a permite repetarea până când utilizatorul alege să închidă aplicația;
- Folosesc for pentru a parcurge lista de task-uri;
- Folosesc if pentru a valida, de exemplu, valabilitatea termenului limită al unui task.

Petre Flaviu-Mihai
Grupa 10LF332

Lab3: Colecții Java (Array, List, Map):

- Folosesc Map<String, ArrayList<Task>> pentru a stoca toate task-urile după titlul acestora;
- Am implementat metode precum:
  - Sortare: Se pot sorta task-urile după prioritate;
  - Filtrare: Afișeaz doar task-urile care sunt urgente în cadrul locului de muncă.

```
@Override 1 usage
public void sortTasksByPriority(){
    List<Task> allTasks = new ArrayList<>();

    for (Map.Entry<String, ArrayList<Task>> entry : _taskMap.entrySet()) {
        allTasks.addAll(entry.getValue());
    }

    allTasks.sort(Comparator.comparingInt(Task::get_priority).reversed());

    for (Task task : allTasks) {
        System.out.println(task.get_description() + " " + task.get_priority());
    }
}
```

Sortarea task-urilor după prioritate

Lab4: Clase Java (clasă cu atribute și metode):

- Am realizat clasa Task, care se ocupă de task-urile simple, cu următoarele atribute:

```
public class Task extends TaskManager{
    //region Variables
    private String _title; 3 usages
    private String _description; 3 usages
    private int _priority; 3 usages
    private LocalDate _deadline; 3 usages
    private boolean _isCompleted; 3 usages
    private final String _type = "simple"; 1 usage
    //endregion
```

- Clasa conține constructori cu parametri și fară parametri, getters și setters.

Lab5: Moștenire în Java:

- Am realizat două clase derivate din clasa Task:
  - WorkTask;
  - PersonalTask.

WorkTask este o clasă dedicată pentru task-urile legat de muncă, iar PersonalTask este o clasă dedicată pentru task-urile legate de viața utilizatorului.

Petre Flaviu-Mihai
Grupa 10LF332

Lab6: Interfețe în Java:

- Am implementat interfața ITaskManager cu metodele din clasa TaskManager:

```
public interface ITaskManager {  2 usages  5 implementations
    void AddTask(Task newTask);  3 usages  1 implementation
    boolean RemoveTask(String title);  2 usages  1 implementation
    void PrintTask(String title, String description);  5 usages  3 implementations
    boolean CompleteTask(String title, String description);  2 usages  1 implementation
    boolean IsDeadlineExpired(String title, String description);  1 usage  1 implementation
    void sortTasksByPriority();  1 usage  1 implementation
    void saveToFile(String filePath);  6 usages  1 implementation
    Map<String, ArrayList<Task>> readFromFile(String filePath);  2 usages  1 implementation

}
```

Lab7: Teste pentru fiecare metodă:

- Am realizat teste pentru următoarele clase:
  - Task;
  - PersonalTask;
  - WorkTask;
  - TaskManager.

Lab8: Persistența datelor:

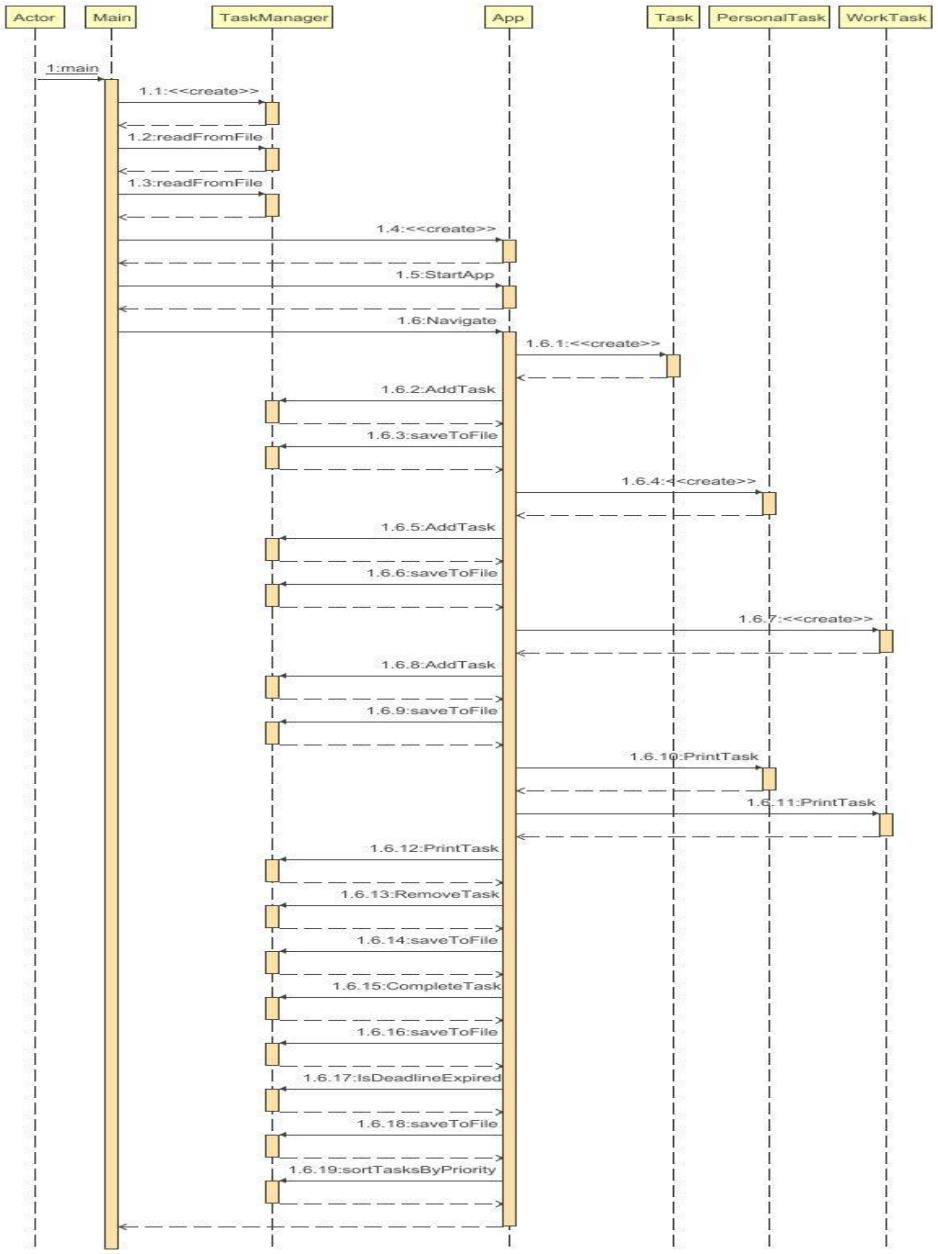
- Salvare și încărcare datelor:
  - Salvez lista de task-uri într-un fișier .json folosind biblioteca Jackson;
  - La pornirea aplicației se salvează task-urile existente din fișier.
- Structură JSON:
  - Task-urile sunt salvate cu diferențiere clară între atribute.


```
{
  "Curatenie" : [ {
    "_title" : "Curatenie",
    "_description" : "Trebuie sa dau ce aspiratorul",
    "_priority" : 4,
    "_deadline" : [ 2025, 1, 15 ],
    "_isCompleted" : false,
    "_type" : "simple"
  }, {
    "_title" : "Curatenie",
    "_description" : "Trebuie sa duc gunoiul",
    "_priority" : 3,
    "_deadline" : [ 2025, 2, 12 ],
    "_isCompleted" : false,
    "_type" : "simple"
  } ],
  "Proiect" : [ {
    "_title" : "Proiect",
    "_description" : "Trebuie sa finalizez proiectul urgent",
    "_priority" : 5,
    "_deadline" : [ 2025, 1, 22 ],
    "_isCompleted" : false,
    "_type" : "work",
    "_workSpace" : "Departament IT"
  } ]
}
```

Petre Flaviu-Mihai
Grupa 10LF332

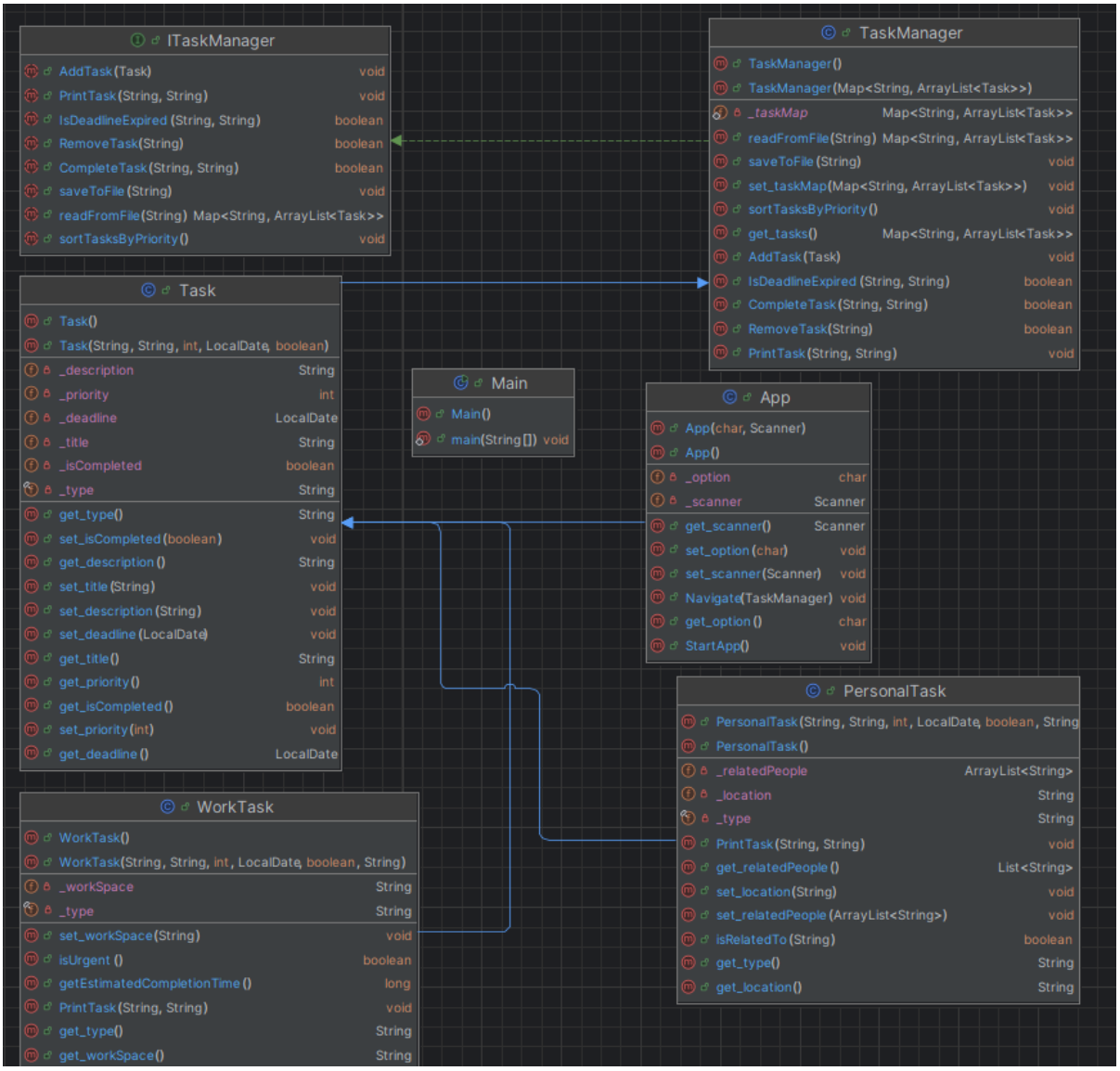
Lab9: Realizați 3 diagrame UML:

- Diagramă pe secvențe



Petre Flaviu-Mihai
Grupa 10LF332

- Diagramă pe clase



Petre Flaviu-Mihai
Grupa 10LF332

- Diagramă flux de date