Proiect To-Do List în Java

Cotet Mihaita-Cornel

**Structura proiectului**

ToDoListProject

│

└── src

├── Main.java

├── Task.java

└── ToDoList.java

**Descriere:**

Acest program Java implementează o aplicație simplă de tip To-Do List folosind o structură de date ArrayList pentru stocarea și gestionarea sarcinilor. Utilizatorii pot adăuga sarcini noi, le pot marca ca finalizate și pot vizualiza lista actuală de sarcini. Programul citește date de la tastatură, iar interacțiunea cu utilizatorul este realizată printr-un meniu interactiv. Logica programului este implementată folosind principiile programării orientate pe obiecte (OOP)

1. Task.java

**Rol**: Task.java definește o **clasă model** pentru o sarcină individuală în lista de activități. Aceasta reprezintă un obiect de bază în proiectul nostru, permițându-ne să gestionăm informații specifice despre fiecare sarcină.

**Atribute**:

* String title: Stochează titlul sarcinii.
* String description: Stochează descrierea sarcinii, oferind mai multe detalii.
* boolean isCompleted: Indicator care arată dacă sarcina a fost completată sau nu, implicit fiind false.

**Constructorul**:

* public Task(String title, String description): Initializează o nouă sarcină cu titlul și descrierea date, setând starea isCompleted la false.

**Metode getter**:

* getTitle() și getDescription(): Returnează titlul și descrierea sarcinii, respectiv, permițând accesul la aceste valori într-un mod sigur.

**Metodă pentru actualizarea stării**:

* markAsCompleted(): Schimbă valoarea lui isCompleted la true, indicând că sarcina a fost finalizată.

**Metodă toString()**:

* Metoda suprascrisă toString() formatează o descriere a obiectului Task pentru o ieșire clară, specificând dacă este Completed sau Pending.

1. TodoList.java

**Rol**: ToDoList.java gestionează lista de sarcini și oferă metode pentru a adăuga, marca și afișa sarcinile.

**Atribut**:

* ArrayList<Task> tasks: Stochează toate sarcinile într-o listă dinamică. Folosirea ArrayList permite adăugarea și accesarea sarcinilor în mod eficient.

**Constructor**:

* public ToDoList(): Inițializează lista de sarcini, creând un nou ArrayList.

**Metodă addTask**:

* public void addTask(Task task): Adaugă o sarcină în lista tasks și afișează un mesaj de confirmare. Această metodă folosește tasks.add(task), care inserează obiectul Task la sfârșitul listei.

**Metodă markTaskAsCompleted**:

* public void markTaskAsCompleted(int index): Primește un index al sarcinii și o marchează ca finalizată. Verificarea if (index >= 0 && index < tasks.size()) asigură că indexul este valid, evitând erori de accesare a listei.

**Metodă displayTasks**:

* public void displayTasks(): Iterează prin toate sarcinile și le afișează împreună cu indicele corespunzător. Utilizează toString() din clasa Task pentru a obține detalii despre fiecare sarcină.

1. Main.java

**Rol**: Main.java conține punctul de intrare al aplicației și un meniu pentru a interacționa cu utilizatorul. Aici gestionăm input-ul utilizatorului pentru a adăuga, marca și afișa sarcini în lista de to-do.

**Inițializări**:

* ToDoList toDoList = new ToDoList(): Creează o instanță a clasei ToDoList pentru gestionarea sarcinilor.
* Scanner scanner = new Scanner(System.in): Creează un obiect Scanner pentru a primi input de la utilizator.

**Loop-ul de Meniu**:

* Folosim un while(true) pentru a repeta afișarea meniului și a primi input de la utilizator până când utilizatorul alege să iasă.
* În funcție de alegerea utilizatorului (choice), sunt executate diferite operațiuni.

**Opțiunile din Meniu**:

* **1**: Adăugarea unei noi sarcini:
  + Solicite titlul și descrierea sarcinii și creează un nou obiect Task, adăugându-l în toDoList.
* **2**: Marcarea unei sarcini ca completă:
  + Primește numărul sarcinii de la utilizator (indexul) și apelează markTaskAsCompleted din ToDoList.
* **3**: Afișarea tuturor sarcinilor:
  + Apelează metoda displayTasks din ToDoList, care afișează lista curentă de sarcini.
* **4**: Ieșirea din aplicație:
  + Închide scanner și oprește execuția programului prin return.