****SEMINAR 2****

**Creați o aplicație cu interfață de tip consolă care gestionează o listă de TO-DOs. Fiecare task este reprezentat de un string după cum urmează: *DescriereTask\_ZiDeadline\_Status.***

**Aplicația are următorul meniu:**

1. **Citire listă de TO-DOs**

**Ex. *BookSkydivingAdventure\_18\_in-progress, LearnJuggling\_25\_pending, StartVegetableGarden\_27\_in-progress***

1. **Adăugare TO-DO la lista de task-uri cu citirea informațiilor pentru câmpuri**

**Ex. Descriere: *HostMovieMarathonNight***

**Zi deadline: *17***

**Status: *done***

1. **Afișare task-uri din listă care au un status dat (*pending, in-progress, done*). Dacă nu există task-uri cu status-ul cerut, afișați un mesaj corespunzător.**
2. **Afișați task-urile pe zile.**

**Ex. Considerând cele 4 task-uri de mai sus, plus: *TravelToUnchartedIsland\_31\_pending, ReadBook\_25\_in-progress, BuildTreehouse\_18\_done,*rezultatul acestei acțiuni din meniu va fi:**

**Ziua 17: *HostMovieMarathonNight***

**Ziua 18: *BookSkydivingAdventure, BuildTreehouse***

**Ziua *25: LearnJuggling, ReadBook***

**Ziua 27: *StartVegetableGarden***

**Ziua 31: *TravelToUnchartedIsland***

1. **Ieșire din aplicație**

FUNCȚIONALITĂȚI

|  |  |
| --- | --- |
| F1 | Citire listă de task-uri |
| F2 | Adăugare task în listă |
| F3 | Filtrare task-uri după status |
| F4 | Afișare task-uri în funcție de zile |
| F5 | Ieșire din aplicație |

F1. Citire listă de task-uri

* Modelare: scenariu de rulare
* Activități: cum implementăm efectiv funcționalitatea?
* Cazuri de test

1. Modelare

|  |  |  |
| --- | --- | --- |
| Program | Utilizator | Descriere |
| 1. Citire lista  2. Adaugare task  3. Filtrare dupa status  4. Task-uri pe zile  5. Iesire  10. Afisare lista de task-uri |  | Programul afișează un meniu din care utilizatorul poate selecta opțiunea dorită. |
|  | 1 | Utilizatorul alege opțiunea 1 |
| Introduceți lista: |  | Programul cere introducerea listei |
|  | ***BookSkydivingAdventure\_18\_in-progress, LearnJuggling\_25\_pending, StartVegetableGarden\_27\_in-progress*** | Utilizatorul introduce lista conform unui format definit (informații task-uri separate cu ”\_”, task-uri separate cu ”,”) |
| Lista a fost citită cu succes. |  | Programul afișează un mesaj prin care informează utilizatorul cu privire la statusul acțiunii efectuate. |

1. Activități

|  |  |
| --- | --- |
| A1 | Citire lista de la tastatură |
| A2 | Formare lista de task-uri (procesarea informațiilor citite, separare în task-uri etc) |
| A3 | Interfața cu utilizatorul |

1. Cazuri de test

|  |  |
| --- | --- |
| **Input** | **Output** |
| ***BookSkydivingAdventure\_18\_in-progress, LearnJuggling\_25\_pending, StartVegetableGarden\_27\_in-progress*** | Listă citită cu succes, lungimea listei de task-uri este 3 |
| ***BookSkydivingAdventure\_18\_in-progress*** | Listă citită cu succes, lungimea listei de task-uri este 1 |
| “” (string vid) | INVALID |
| ***BookSkydivingAdventure\_18\_in-progress\_*** | INVALID |
| ***BookSkydivingAdventure\_in-progress*** | INVALID |
| ***BookSkydivingAdventure\_100000\_in-progress*** | INVALID |
| ***BookSkydivingAdventure\_18\_abcde*** | INVALID |
| ***\_18\_in-progress*** | INVALID |
| ***....*** |  |

**TO DO:**

* Implementare validări
  + Validare string listă de task-uri
  + Validare string-uri care reprezintă un task (trebuie să ne decidem ce reprezintă un task invalid cu privire la descriere, zi, status)
* Reprezentare task:
  + Cât de ușor este de implementat funcționalități ce se folosesc de informație despre task-uri (e.g. filtrare după status)? Ar ajuta să folosim o altă reprezentare? Ce fel de reprezentare?
  + **Q**: De ce am introdus funcția *obtine\_atribute\_task* în cod, care este utilitatea ei?