Projects

Monday, July 03, 2023 1:41 PM

Mini Projects List

1. Checkpoint-1
2.

Individual Projects:

Individual Project: You can select one from two options

1. **Todo List :** Files / Projects / Individual Project / Project-I

2. **Money Tracking**: Files / Projects / Individual Project / Project-II

Info About the Project:

- The Project will be a ConsoleApp.
- Before the coding search info for the planning step of project.
- Before start the Coding you should planning the Project (What classes you need, UML diagrams and other)
  - UML diagram Tool : https://www.lucidchart.com/pages/examples/uml_diagram_tool
  - UML diagram Tool : https://www.smartdraw.com/uml-diagram/uml-diagram-tool.htm
  - Drawio : https://app.diagrams.net/

UML Tutorials

- UML Diagrams Full Course (Unified Modeling Language) - YouTube
- How to Make a UML Sequence Diagram - YouTube
- UML Tutorial: How to Draw UML Class Diagram - YouTube
- UML Use Case Diagram Tutorial - YouTube
- UML class diagrams - YouTube
- All About UML Activity Diagrams - YouTube
-

Keep it on mind

- Don't think complex just do it simple
- Just follow the pdf instruction
- Of course if you want you can add more properties to your project
- Publish it on GitHub

Deadline: 2024.10.18

1. Planning

1. Choose your project (C# ToDo List or Money Tracking)
2. Plan your project & Create UML diagram (optional) (which classes, which fields and methods etc. will be use in project)

2. Coding

- Try to use separate class files

  - Create Classes

  - Create Objects

  - Create Methods (class methods or functions that you need)

  - Create List(s)

  - Coding Menu part (Console App)

  - Add objects to list

  - Show on screen (sorted by Date and Sorted by Project)

  - Edit-Mark down- Delete operations

  - Save & Quit part (This will be send to file and then will read from file) - (You can think and manage it in the last)

3. Keep it simple as much as you can. Don't think complex

- Don't think complex just do it simple
- Just follow the pdf instruction
- Create your project skeleton first when you code it
- Of course if you want you can add more properties to your project.
- Publish it on GitHub

****

You can think like this about save to file and read from file:

For example ToDoList Project

- Each task includes the following properties:
    - Task title: A short description of the task.
    - Due date: The date by which the task should be completed.
    - Status: Indicates whether the task is completed or pending.
    - Project: Specifies the project to which the task belongs.

  You can think like this;

  1. Get Info from User (Also you can added already some info into the file)
  2. Create your objects
  3. Add them to the List
  4. You can show, sort, update them.
  5. Send(Write) to file (you can use just txt file or any other format)
  6. Think ==> It will write to file Each object (title, date, project name and status) will be added to a separate line in the txt file.
  7. Think ==> When you read data from File, it will read line by line and each line will create again object and will be added in to the list(or array)

- You can sort it when you get the info again from file. Just think and create your appropriate functions and methods.