This document describes the **team work assignment** for Telerik Academy students studying Object-Oriented Programming (OOP) – March 2015.

Project Description

Design and implement an object-oriented application by choice. Here are some suggestions:

- A game:
    - The world can be fantasy, sci-fi, modern, etc.
    - You may have one or more heroes, gaining experience, having skills, wearing items, etc.
    - You may have one or more players, fighting against a computer or against each other
    - You may have enemies like creatures, machines, warriors, etc.
    - You may have items like swords, guns, armor, space-suits, etc.
    - You may have skills like double-damage, teleportation, etc.
- A line-of-business application
    - TODO list application, TODOs can have a deadline, priorities, types etc.
    - Calendar application, keeping meetings with places and date, having notifications, etc.
    - Social network, with different users, different roles, friend list, user information, chat between users, groups of users, etc.
    - A chat application, supporting user-to-user chats, channels, group chats, friends-list, starred users, etc.

General Requirements

Please define and implement the following object-oriented assets in your project:

- At least **5 interfaces** (with one or more implementations)
- At least **15 classes** (implementing the application logic)
- At least **3 abstract classes** (with inheritors)
- At least **1 custom exception class** (with usage in your code)
- At least **3 levels of depth in inheritance**
- At least **1 polymorphism usage**
- At least **1 structure**
- At least **1 enumeration**
- At least **1 event** (with subscribers)
- At least **1 design pattern** (e.g. Composite, Singleton, Factory, Wrapper, Bridge, Command, etc.)

You might read about design patterns in Wikipedia, Sourcemaking, DoFactory and others.

Additional Requirements

- Follow the **best practices for OO design**: use data encapsulation, use exception handling properly, use delegates and events like it is recommended in MSDN, use inheritance, abstraction and polymorphism properly, follow the principles of strong cohesion and loose coupling.
- Obligatory use Git to keep your source code and for team collaboration (you might use https://github.com/). TFS or SVN are **not allowed**. **Use Git**.
- Provide a class diagram (to visualize all types).

Optional Requirements

If you have a chance, time and a suitable situation, you might add some of the following to your project:

- **Static members** (fields, properties, constructor, etc.)
- **Constants**, **generic types**, **indexers**, **operators**
- **Lambda expressions** and **LINQ**
- Implementation of **IEnumerable<T>**, **ICloneable**, **ToString()** override
- **Namespaces** (if your classes are too much)
- **User interface (UI)** – console, graphical, web or mobile

Non-Required Work

- **Completely finished project** is not obligatory required. It will not be a big problem if your project is not completely finished or is not working greatly. This team work project is for educational purpose. Its main purpose it to experience object-oriented modeling and OOP in a real-world project and to get some experience in team working and team collaboration with Git.

Deliverables

Put the following in a **ZIP archive** and submit it (each team member submits the same file):

- The complete **source code**.
- Brief **documentation** of your project (2-3 pages). It should provide the following information (in brief):
  - Team name and list of team members (with http://telerikacademy.com usernames)
  - Project purpose – what problem do you solve?
  - Class diagram of your types
  - The URL of your Git repository
  - Any other information (optionally)
- Optionally provide a PowerPoint presentation designed for the project defense.

Public Project Defense

Each team will have to deliver a public defense of its work in front of the other students and trainers. You will have only 10 minutes for the following:

- **Demonstrate** the application (very shortly).
- Show the **class diagram** (just a glance)
- Show the **source code** in the Git web-based source code browser.
- Show the **commits logs** to confirm that each team member has contributed.
- Optionally you might prepare a PowerPoint presentation (3-4 slides).

Please be strict in timing! Be **well prepared** for presenting maximum of your work for minimum time. Bring your own laptop. Test it preliminary with the multimedia projector. Open the project assets beforehand to save time. **You have 10 minutes**, no more. It is not necessary for all team members to participate in the project defense.

Give Feedback about Your Teammates

You will be invited to **provide feedback** about all your teammates, their attitude to this project, their technical skills, their team working skills, their contribution to the project, etc. The feedback is important part of the project evaluation so **take it seriously** and **be honest**.