Seminar 10
week 10 (14-20 November 2017)

1. Questions from **Lab-Assignment 6 from Laboratory 8**.

2. Solve some simple problems in C#. Please discuss some possible implementations for each of the following topics:

- 2.1. **An example of using properties which are declared in an interface:** Declare and implement an interface Employee with two properties: name and unique identifier.
- 2.2. **An example of using inherited properties:** Define two classes Cube and Square which implement an abstract class, Shape, and override its abstract Area property. Please use of the override modifier on the properties. The program accepts the side as an input and calculates the areas for the square and cube. It also accepts the area as an input and calculates the corresponding side for the square and cube.
- 2.3. **An example of using indexer that is declared in an interface.**
- 2.4. **An example of using delegates:** Implement a class BookDB that encapsulates a bookstore database that maintains a database of books. It exposes a method, ProcessPaperbackBooks, which finds all paperback books in the database and calls a delegate for each one. The delegate type that is used is named ProcessBookDelegate. The Test class uses this class to print the titles and average price of the paperback books.
- 2.5. **An example of using events declared in the base class:** When you create a class that can be used as a base class for other classes, you should consider the fact that events are a special type of delegate that can only be invoked from within the class that declared them. Derived classes cannot directly invoke events that are declared within the base class. Although sometimes you may want an event that can only be raised by the base class, most of the time, you should enable the derived class to invoke base class events. To do this, you can create a protected invoking method in the base class that wraps the event. By calling or overriding this invoking method, derived classes can invoke the event indirectly.