Course code: GOC2741

The course is intended for everyone who wants to practically try programming with design patterns (Design Patterns). The course will solve examples from initial brief through design solutions to the application using code. The proposed solution create participants together under the guidance of an instructor, then they have time to convert the design into a real application code. On the course is devoted much time to the real solution of the task and the application of theoretical knowledge of OOP and design patterns. Minimum time is devoted to the theory.

What we teach you:
- Understand practical examples of OOP principles
- Use design patterns for creating high-quality solutions
- Learn to take account of possible requests for changes in the future

Required skills:
- Knowledge of object oriented programming in .NET
- Knowledge of C # or VB .NET

Teaching methods:
- Professional explanation with practical samples and examples.

Teaching materials:
- Books published by offer professional publishing.

Course syllabus:
Unification of concepts and initial knowledge
- Inheritance, polymorphism
- Interface
- Aggregation
- Open / Closed Principle
- Single Responsibility Principle

Design Patterns in practice

Kreacion?ln?
- Factory - factory buildings
- Singleton - damned design pattern
- Builder - creating more complex objects

Structural
- Adapter - the road to unification
- Composite - as they have to worry about the type of object
- Decorator - otherwise recursion
- Facade - take your system outside handsome
- Proxy - to supplement what is needed without inheriting

Behavioral
- Command - encapsulation events
- Chain of responsibility - whether someone decides
- Iterator - love foreach
- Observer - distribution and reception of information
- Strategy - inheritance and interface
- Template method - to do the same or else

Design patterns for information only

- Prototype
- Bridge
- Flyweight
- Interpreter
- Mediator
- Memento
- State
- Visitor