(2 курс) ООП на C# (углубленно)

1. Интерфейсы
2. **S.O.L.I.D**.
3. Patterns (10-15)

Модули

1. Создание модулей
2. Способы подключения
3. Система «Плагинов»
4. Nuget модули – подключить NLog
5. Работати подключение пакетов
6. Reflections
7. Libraries (dll), loading
8. Потоки, основы многопоточного программирования + основы СЕТЕВОГО обмена данными.
9. C# version features 3.0, 4.0, 5.0, … 8.0
10. Linq
11. Windows Forms (2001), modern – WPF (2007)
12. АТРИБУТЫ
13. Database, MS SQL Server, (Ado.NET), Entity Framework
14. Курсовая работа по ООП на C# + WPF + с испл Б.Д.

Gang of Four = GOF

Сборник «рецептов» ОО проектирования, ПАТТЕРНЫ проектирования/ программирования ОО (универсальные) – рецепты как следовать S.O.L.I.D.

Паттерны разделили на 3 ГРУППЫ

1. Creational (порождающие)
2. Structural (структурные)
3. Behavior (поведенческие)

Из первой группы Creational самый известный и самый простой из всех сущ.

(редко используемый)

Паттерн «**Singleton**» (одиночка) –

«анти-паттерн»

Суть? – гарантирует на 100% что для некоторого класса в системе ВСЕГДА будет сущ. Только ОДИН ЕДИНСТВ экземпляр (объект, более одного создать НЕВОЗМОЖНО.

Зачем? – случай когда два таких объекта могут конфликтовать, слишком «тяжёлые» много кушают. В том числе и ЗАЩИТА от «дурака».

АРХИТЕКТУРА паттерна (устройство) –

ОДИН КЛАСС и в нём необх реализовать 3 условия

1. Приватный конструктор **private**
2. Приватное Статическое ПОЛЕ хранящее объект этого класса (будет единственный) **private static field;**
3. Публичный статический метод (или свойство) возвращающий то самое закрытое поле из п (2)

Как правило, в наследовании не участвует, но в C# это можно обойти с помощью рефлексии.

В реальной жизни (не так часто, но)

1. В синглтоне можно хранить Глобальные настройки всего приложения, которые можно смотреть из любой части / модуля системы.
2. Специальный объект для ведения журнала в программе – logger (лоджер – логер) – журнал ошибок возникающих, действий юзера и т.д.