SOLID

S - все методы и классы выполняют лишь одну функцию.

Например,

public class GuessService : IGuessService

{

private int \_minInterval, \_maxInterval;

private int? \_randomNumber;

private int \_maxTries;

private int \_currentTries;

public GuessService(int minInterval, int maxInterval, int maxTries)

{

\_minInterval = minInterval;

\_maxInterval = maxInterval;

\_maxTries = maxTries;

}

public void SetUpGame()

{

\_randomNumber =

\_minInterval + new Random().Next(\_maxInterval - \_minInterval);

\_currentTries = \_maxTries;

}

public bool CheckNumber(int number)

{

if (\_randomNumber == null)

throw new Exception("random number is not configured");

\_currentTries--;

if (number == \_randomNumber)

return true;

if (\_currentTries == 0)

throw new Exception("no tries left");

return false;

}

public void Configure(ConfigureInput input)

{

\_minInterval = input.Min;

\_maxInterval = input.Max;

\_maxTries = input.Tries;

}

}

O - принцип открытость-закрытость соблюдается.

Например, если сейчас мы захотим добавить сюда вывод в WPF, нужно будет добавить просто еще один проект и там использовать функционал из Core.service. Менять другие классы не надо будет.

L - принцип барбара-лискова соблюдается. На вход в конструкторы везде принимаются интерфейсы, а значит можно легко использовать разные реализации.

public GuessService(int minInterval, int maxInterval, int maxTries)

{

\_minInterval = minInterval;

\_maxInterval = maxInterval;

\_maxTries = maxTries;

}

I - принцип разделения интерфейсов соблюдается. Каждый интерфейс содержит только нужны еему методы.

public interface IGuessService

{

public void SetUpGame();

public bool CheckNumber(int number);

public void Configure(ConfigureInput input);

}

D - принцип инверсии зависимостей соблюдается.

public static void Main(string[] args)

{

Config config = Config.ParseConfig("config.yml");

GuessService guessService =

new GuessService(config.Min, config.Max, config.Tries);

Cli cli = new Cli(guessService, "quit");

cli.Run();

}