**Сборная солянка по теории**

Модификаторы доступа:

Private – закрытый, только свой класс.

Private protected – свой класс или производный в одной и той же сборке.

Protected – класс или производный, производные могут быть в других сборках.

Internal – только в одной сборке.

Protected internal – текущая сборка любое место, другая сборка производные классы.

Public – везде.

File – только файл кода.

Принципы ООП:

►Наследование (Inheritance); процесс, благодаря которому один объект может наследовать свойства от другого объекта.

►Инкапсуляция (Encapsulation); механизм, связывающий вместе данные и код, обрабатывающий эти данные, и сохраняющий их от внешнего воздействия и ошибочного использования.

►Полиморфизм (Polymorphism) — способность к изменению функций, унаследованных от базового класса

Статический – перегрузка методов и операторов.

Динамический – наследование, реализация интерфейсов, переопределение методов

►Абстракция подразумевает разделение и независимое рассмотрение интерфейса и реализации. Абстракция - уровень описания/представления модели чего либо

► Класс – шаблон, по которому определяется форма объекта

► Объект – это физическая реализация класса.

Свойства – специальные методы класса, служат для организации доступа к полям класса. Как правило, свойство связано с закрытым полем класса и определяет методы его получения и установки (предоставляет инкапсуляцию).

Интерфейсы позволяют определить некоторый функционал, не имеющий конкретной реализации. Затем этот функционал реализуют классы, применяющие данные интерфейсы.

Ключевые слова в классах: static, abstract, sealed, partial, модификаторы.

Интерфейс – правила, которые определяют набор методов, свойств, событий, которые должны быть реализованы классами, все методы интерфейса являются public.

Исключение генерируется, когда возникает некоторая ошибка или необычная ситуация, которая нарушает выполнение программы. Генерация происходит при помощи оператора throw. Использование блоков try-catch-finally.

Делегат – объект, предназначенный для хранения ссылок на методы (указатель на методы) и можно вызвать данные методы.

Обработчики событий — это не что иное, как методы, вызываемые через делегатов.

**Событие** – элемент класса, позволяет ему посылать другим объектам уведомления об изменении своего состояния. События построены на основе делегатов: с помощью делегатов вызываются методы-обработчики событий.

Сериализация и десериализация - это процессы преобразования объектов в определенный формат данных (например, JSON, XML или бинарный формат) и обратно, соответственно.

Windows Forms - платформа пользовательского интерфейса для создания разнообразных клиентских приложений рабочего стола в Windows.

Форма — это визуальная поверхность, на которой выводится информация для пользователя.

Элемент управления — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.

При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления создается событие.

**Вопросы Лаб2**

1. Какое основное назначение технологии Windows Forms, ее особенности, преимущества и недостатки?

Основное назначение: создание графических пользовательских интерфейсов (GUI) для приложений на платформе Windows.

Особенности: использует объектно-ориентированный подход, позволяет легко создавать интерфейсы с помощью визуального дизайнера.

+ широкий выбор ЭУ, быстрая разработка, интеграция с другими технологиями .NET.

- только для Windows норм работает, меньшая гибкость по сравнению с WPF.

1. Зачем используется класс Form? Назовите основные методы, свойства и события данного класса.

Для создания окна приложения. Контейнер для всех элементов. Show(), Hide(), Close(). Свойства Text (заголовок), Size (размер), BackColor (цвет фона), Width, Height, Location, FormBorderStyle. События Load (при загрузке), Closed (перед закрытием), Closing (после закрытия), Activated, Deactivated, Resize.

1. Поясните структуру проекта и назначение всех файлов?

Файл формы (.cs или .vb): содержит код формы, включая определение элементов управления, обработку событий и другую логику.

Файл дизайна формы (.designer.cs или .designer.vb): автоматически сгенерированный код дизайнера формы.

Program.cs – файл с точкой входа, где создается и запускается главная форма.

Файл ресурсов (.resx): содержит ресурсы, такие как строки, изображения.

App.config – файл конфигурации приложения.

1. Зачем нужен атрибут STAThreadAttribute?

Приложение должно быть запущено в однопоточной COM .

1. Как в вашем проекте используются события и делегаты?

События для обработки различных действий, таких как нажатие кнопки.

Делегаты для связи событий с их обработчками

1. Объясните схему работы цепочек делегатов.

Цепочка делегатов - последовательность делегатов, которые вызываются последовательно при возникновении события.

1. Объясните механизм подписки и отмены подписки на события.

Механизм подписки и отмены подписки на события в C# основан на использовании операторов += и -= для добавления и удаления обработчиков событий соответственно.

1. Как создать вторую форму и передать туда данные? Есть ли другие способы?

Создание: Form2 form2 = new Form2(); передача данных через свойста или методы формы

1. Как во время выполнения приложения добавить/удалить элемент управления?

Добавить Controls.Add(new Button()).

Удалить Controls.Remove(button).