

Рекомендуемые темы проектов.

Требования к проектам:

- Windows Forms. C#.
- Наличие всех заявленных функций.
- Корректность работы программы (без ошибок и сбоев).
- Интерфейс приложения должен быть интуитивно понятным и удобным для пользователя.
- Структура базы данных должна быть описана в виде ER-диаграммы и соответствовать ЗНФ.
- Код должен быть чистым, структурированным, читаемым и документированным (комментарии).
- Корректная обработка ошибок и исключений.
- Проект должен быть протестирован на различных сценариях.
- Проект должен быть собран в виде исполняемого файла (.exe или .msi)

Примечания:

- Работа в команде (2-3 участника) приветствуется.
- Темы проектов ориентировочные, функционал можно изменять.
- Можно предложить свою тему, согласовав с преподавателем.
- Использование ИИ-ассистентов разрешено в рамках проекта как инструмента для повышения эффективности работы, а не для замены самостоятельного мышления. Бездумное использование сгенерированного кода без понимания его функционирования не допускается.

Мониторинг курсов валют.

Описание: Приложение для отслеживания курсов валют в реальном времени с возможностью построения графиков изменения курса.

Используемые технологии:

Элементы управления: Chart, DataGridView, ComboBox, Label, ScottPlot 5 (построение графиков).

База данных MySQL для хранения истории курсов.

Класс Timer для периодического обновления данных.

Использование HttpClient для получения данных о курсах валют с API (например, Open Exchange Rates).

Система учета клиентов фитнес-клуба.

Описание: Разработка программы для учета клиентов фитнес-клуба, их абонементов, посещений и тренировок.

Используемые технологии:

Элементы управления: DataGridView, DateTimePicker, PictureBox, TabControl.

База данных MySQL для хранения данных о клиентах и тренировках.

Класс Timer для напоминания о предстоящих тренировках.

Использование HttpClient для отправки SMS-уведомлений клиентам (через сторонний API).

P2P-чат в локальной сети.

Разработать P2P-чат для локальной сети, где каждый участник может обмениваться сообщениями напрямую через TCP/IP, используя UDP broadcast для обнаружения пользователей; локальное хранение истории переписки (MySQL или SQLite) и

уведомления о новых сообщениях, а также создать простой интерфейс с элементами управления RichTextBox, TextBox, ListBox и Button.

Дополнительно: реализовать шифрование сообщений (AES).

Приложение для управления библиотекой.

Описание: Создание программы для учета книг, выдачи и возврата литературы, а также отслеживания задолженностей читателей.

Используемые технологии:

Элементы управления: TreeView, DataGridView, ComboBox, Label.

База данных MySQL для хранения информации о книгах, читателях и операциях.

Класс Timer для автоматического обновления статуса просроченных книг.

Использование HttpClient для поиска книг в интернете (например, через Google Books API).

Клиент для погодного сервиса.

Создание приложения для отображения текущей погоды и прогноза на несколько дней вперед для выбранного города.

Элементы управления: PictureBox, TrackBar, NotifyIcon

HttpClient для получения данных о погоде

Timer для регулярного обновления прогноза

MySQL для сохранения истории погоды

Приложение для мониторинга криптовалют.

Элементы управления: DataGridView, Chart, SplitContainer

HttpClient для получения данных с криптобирж

Timer для обновления котировок

MySQL для хранения исторических данных

Приложение для изучения иностранных слов.

Элементы управления: FlowLayoutPanel, RichTextBox, RadioButton

HttpClient для получения переводов и произношений

Timer для режима интервальных повторений

MySQL для хранения словаря и прогресса обучения

Планировщик задач с напоминаниями.

Элементы управления: TreeView, NotifyIcon, DateTimePicker.

Timer: Проверка дедлайнов каждую минуту.

MySQL: Хранение задач, меток и пользователей.

Дополнительно: Синхронизация с Google Calendar через API.

Система тестирования знаний.

Элементы управления: RadioButton, CheckBox, ProgressBar.

Timer: Ограничение времени на вопрос.

MySQL: База вопросов, тестов и результатов.

HttpClient: Загрузка тестов с облачного хранилища.

Дополнительно: Рейтинговая таблица пользователей.

Доска Канбан.

Разработать приложение "Канбан-доска" на C# с использованием Windows Forms для управления проектными задачами, включающее функциональность перетаскивания карточек между колонками (минимум 3 колонки: "К выполнению", "В процессе", "Завершено"), создание и редактирование задач с указанием сроков и приоритетов, хранение данных в MySQL, систему уведомлений с использованием Timer о

приближающихся дедлайнах, возможность синхронизации через HttpClient, а также поиск, фильтрацию и цветовую маркировку карточек по различным параметрам.

Приложение "Таймер для продуктивности".

Описание:

Приложение, которое помогает пользователю управлять временем, используя технику "Помидоро".

Элементы управления: ProgressBar для отслеживания сессии, кнопки Start, Pause, Reset.

Класс Timer для отсчета времени.

MySQL для хранения статистики (дата, количество завершенных сессий).

Дополнительно: отправка напоминаний через локальные уведомления.

Приложение "Новостной агрегатор".

Описание:

Программа, которая загружает и отображает последние новости из нескольких источников.

Элементы управления: ListBox для списка новостей, WebBrowser для отображения полного текста новости, кнопка Refresh.

Класс HttpClient для получения данных через API (например, NewsAPI, Guardian API).

Timer для автоматического обновления новостей.

MySQL для хранения загруженных новостей (заголовок, дата, источник, ссылка).

Приложение "Менеджер заметок".

Описание:

Приложение для создания и хранения заметок.

Элементы управления: RichTextBox для ввода текста, кнопки для сохранения/удаления/редактирования заметок, ListBox для отображения списка заметок.

MySQL для хранения заметок (заголовок, текст, дата создания).

Дополнительно: поиск по заметкам, экспорт в PDF.

Приложение "Курсы валют".

Описание:

Программа для отображения текущих курсов валют.

Элементы управления: ComboBox для выбора валюты, Label для отображения текущего курса.

Класс HttpClient для получения данных через API (например, Центробанк РФ или другой источник).

Timer для периодического обновления данных.

MySQL для хранения истории курсов.

Приложение "Калькулятор калорий".

Описание:

Приложение для подсчета калорий, потребленных за день.

Элементы управления: ComboBox для выбора продукта, NumericUpDown для ввода количества, Label для отображения общего числа калорий.

MySQL для хранения базы данных продуктов (название, калорийность на 100 г).

Дополнительно: возможность добавления новых продуктов в базу.

Социальная сеть (мини-версия).

Описание: Регистрация пользователей, публикация сообщений, просмотр ленты новостей, отправка личных сообщений.

Используемые элементы: TextBox, Button, ListBox, RichTextBox, PictureBox.

Дополнительно: Хранение данных о пользователях и сообщениях в базе данных MySQL, обмен сообщениями через HttpClient, использование Timer для обновления ленты новостей.

Игра 'Морской бой' с сетевым режимом.

Описание: Классическая игра "Морской бой" с возможностью игры с другим игроком по сети.

Используемые элементы: Button, PictureBox, Label.

Дополнительно: Использование HttpClient для обмена данными между игроками, хранение статистики игр в базе данных MySQL, использование Timer для контроля времени хода.

Приложение для управления расписанием занятий.

Описание: Составление и просмотр расписания занятий для студентов и преподавателей.

Используемые элементы: DataGridView, TextBox, DateTimePicker, Button, ComboBox, TreeView.

Дополнительно: Хранение данных в базе данных MySQL, возможность синхронизации расписания с календарем Google или Outlook через HttpClient, использование Timer для напоминаний о предстоящих занятиях.

Приложение для поиска рецептов (Recipe Finder).

Описание: Разработка приложения, которое позволяет пользователям искать рецепты блюд по ингредиентам или названию.

Используемые технологии:

Элементы управления: TextBox, ListBox, PictureBox, Button.

Публичное API: Spoonacular API для поиска рецептов.

База данных MySQL для сохранения избранных рецептов.

Класс HttpClient для взаимодействия с API.

Приложение для поиска фильмов и сериалов (Movie Finder).

Описание: Создание приложения, которое позволяет искать фильмы и сериалы, просматривать их описание, рейтинги и трейлеры.

Используемые технологии:

Элементы управления: ListView, PictureBox, WebBrowser, Button.

Публичное API: The Movie Database (TMDb) для получения данных о фильмах.

База данных MySQL для сохранения списка "Любимые фильмы".

Класс HttpClient для работы с API.

Приложение для перевода текста (Translator App).

Описание: Разработка программы для перевода текста на разные языки с поддержкой автодетекта исходного языка.

Используемые технологии:

Элементы управления: TextBox, ComboBox, Button, Label.

Публичное API: Google Translate API или DeepL API для перевода текста.

Класс HttpClient для отправки запросов к API.

База данных MySQL для сохранения истории переводов.

Приложение для поиска музыки (Music Searcher).

Описание: Создание программы для поиска информации о песнях, альбомах и исполнителях, а также просмотра текстов песен.

Используемые технологии:

Элементы управления: TextBox, ListBox, PictureBox, Button.

Публичное API: Spotify Web API или Lyrics.ovh для получения данных о музыке.
База данных MySQL для сохранения любимых песен.
Класс HttpClient для взаимодействия с API.

Приложение для поиска мест отдыха (Travel Planner).

Описание: Создание программы для поиска популярных мест отдыха (города, достопримечательности, отели) с информацией о них.
Используемые технологии:
Элементы управления: PictureBox, DataGridView, ComboBox, Button.
Публичное API: TripAdvisor API или Foursquare API для получения данных о местах.
Класс HttpClient для работы с API.
База данных MySQL для сохранения избранных мест.

Приложение для анализа криптовалют (Crypto Tracker).

Описание: Разработка программы для отслеживания текущих цен на криптовалюты, построения графиков и отправки уведомлений о достижении определенного уровня.
Используемые технологии:
Элементы управления: Chart, DataGridView, NumericUpDown, Label, ScottPlot 5 (построение графиков).
Публичное API: CoinGecko API или CoinMarketCap API для получения данных о криптовалютах.
Класс Timer для периодического обновления данных.
База данных MySQL для хранения истории цен.

Приложение для поиска авиабилетов (Flight Finder).

Описание: Создание программы для поиска доступных авиабилетов по заданным параметрам (город вылета, город прибытия, даты).
Используемые технологии:
Элементы управления: DateTimePicker, ComboBox, DataGridView, Button.
Публичное API: Skyscanner API или Amadeus API для получения данных о рейсах.
Класс HttpClient для работы с API.
База данных MySQL для сохранения предпочтений пользователя.

Приложение для анализа землетрясений (Earthquake Monitor).

Описание: Создание программы для мониторинга последних землетрясений по всему миру с отображением их местоположения и силы.
Используемые технологии:
Элементы управления: DataGridView, Chart, PictureBox, Button.
Публичное API: USGS Earthquake API для получения данных о землетрясениях.
Класс Timer для периодического обновления данных.
База данных MySQL для хранения истории событий.

Приложение для поиска и отслеживания фильмов/сериалов.

Элементы управления: ListView с режимами отображения, Rating Control
HttpClient для работы с TMDb API или OMDB API
Timer для уведомлений о новых эпизодах

MySQL для хранения просмотренного контента и оценок.

Персональный ассистент с использованием API искусственного интеллекта
Элементы управления: RichTextBox с форматированием, SpeechRecognition

HttpClient для запросов к OpenAI API или аналогичным сервисам
Timer для периодических напоминаний
MySQL для хранения истории запросов и ответов

Приложение для анализа социальных сетей.

Элементы управления: Chart, TreeView, ToolStrip
HttpClient для работы с VK API, Twitter API или аналогичными
Timer для сбора статистики
MySQL для хранения данных анализа

Система мониторинга экологических данных.

Элементы управления: Map, Chart, ColorScale
HttpClient для работы с API экологического мониторинга (OpenWeatherMap AQI)
Timer для обновления показателей
MySQL для хранения исторических данных

Клиент для Reddit.

API: Reddit API.
Элементы управления: RichTextBox (посты), TabControl (разделы).
Функционал:
Просмотр популярных постов, комментариев.
MySQL: Кэширование данных для офлайн-доступа.
Дополнительно: Аутентификация через OAuth2.

Генератор мемов.

API: Imgflip API.
Элементы управления: PictureBox, TextBox (ввод текста), Slider (настройка размера).
Функционал:
Создание мемов из шаблонов.
MySQL: Сохранение пользовательских шаблонов.
Дополнительно: Публикация в социальные сети.

Каталог фильмов с рекомендациями.

API: The Movie Database (TMDB).
Элементы управления: DataGridView, Rating Control, Search Bar.
Функционал:
Поиск по жанру, актерам, рейтингу.
MySQL: История просмотров и избранное.
Дополнительно: Трейлеры через YouTube API.

Приложение для поиска книг.

API: Google Books API
Элементы управления: TextBox, Button, ListBox, PictureBox, RichTextBox, DataGridView
Описание:
Поиск книг по автору или названию.
Отображение обложки, описания, рейтинга.
Возможность добавления книги в избранное и сохранения в MySQL.

Приложение для конвертации валют.

API: ExchangeRatesAPI, Open Exchange Rates API

Элементы управления: ComboBox, TextBox, Button, Label, ScottPlot 5 (построение графиков).

Описание:

Выбор валют и ввод суммы.

Получение актуального курса обмена.

График изменения курса за неделю/месяц.

Автообновление данных через Timer.

Приложение для поиска вакансий.

API: HeadHunter API, LinkedIn API (ограниченный доступ)

Элементы управления: TextBox, Button, ListBox, RichTextBox, DataGridView

Описание:

Поиск вакансий по ключевым словам.

Фильтрация по зарплате, городу, компании.

Сохранение избранных вакансий в MySQL.

Отправка уведомлений о новых вакансиях через Timer.

Конвертер файлов.

Разработать приложение для конвертации файлов из одного формата в другой, например, TXT в CSV, JSON в XML или другие популярные форматы.

Требования:

Реализовать интерфейс с использованием элементов управления OpenFileDialog (для выбора файла), SaveFileDialog (для сохранения результата) и ComboBox (для выбора формата конвертации).

Поддерживать конвертацию между следующими форматами:

TXT ↔ CSV

JSON ↔ XML

Дополнительные форматы (например, JSON ↔ YAML или XML ↔ TXT) – по желанию.

Обеспечить корректную обработку ошибок (например, неверный формат входного файла).

Добавить возможность просмотра содержимого файла перед и после конвертации (например, через элемент RichTextBox).

Шифрование и дешифрование файлов.

Разработать приложение для шифрования и дешифрования файлов с использованием пароля. Пользователь может выбирать файл, указывать пароль и сохранять зашифрованный/расшифрованный результат.

Требования:

Реализовать интерфейс с использованием элементов управления:

OpenFileDialog для выбора файла.

SaveFileDialog для сохранения результата.

TextBox для ввода пароля.

Button для запуска операций шифрования и дешифрования.

Использовать алгоритм шифрования AES для обеспечения безопасности данных.

Обеспечить проверку корректности ввода пароля (например, минимальная длина).

Добавить возможность просмотра статуса операции (например, через элемент Label или ProgressBar).

Обеспечить корректную обработку ошибок (например, неверный пароль или поврежденный файл).

Опционально: добавить поддержку работы с большими файлами и индикацию процесса (например, через ProgressBar).

Учет сотрудников.

Разработать приложение для учета сотрудников компании. Пользователь может добавлять сотрудников, назначать их на должности и отделы, а также просматривать и редактировать данные.

Требования:

Реализовать интерфейс с использованием элементов управления:

DataGridView для отображения списка сотрудников.

TextBox для ввода данных (например, имя, фамилия, должность).

ComboBox для выбора отдела.

Button для выполнения операций (добавление, удаление, редактирование).

Хранить данные о сотрудниках в базе данных MySQL или SQLite:

Таблица Employees с полями: Id, FirstName, LastName, Position, Department.

Реализовать следующие функции:

- Добавление нового сотрудника.
- Удаление сотрудника.
- Редактирование данных сотрудника.
- Фильтрация сотрудников по отделу или должности.

Добавить возможность экспорта данных в файл (например, CSV или Excel) через элемент SaveFileDialog.

Обеспечить корректную обработку ошибок (например, дублирование данных или некорректный ввод).

Опционально: добавить функционал для генерации отчетов (например, количество сотрудников в каждом отделе).

Программа для поиска и скачивания треков с YouTube.

Разработать приложение на языке C# с использованием Windows Forms, которое позволяет пользователю:

Искать музыкальные треки на YouTube по названию (через YouTube Data API).

Скачивать выбранный трек в формате MP3.

Воспроизводить скачанные треки прямо в приложении.

Приложение "Интернет-радио".

Создать приложение для прослушивания интернет-радио на языке C# с использованием Windows Forms. Приложение должно предоставлять доступ к различным радиостанциям, поддерживать управление воспроизведением (воспроизведение, пауза, остановка), а также отображать информацию о текущей радиостанции и транслируемом треке (если доступно).

Игры

Крестики-нолики.

Создать игру "Крестики-нолики" на поле 3x3 или больше, где игроки по очереди ставят свои знаки.

Требования:

Поле игры отображается с помощью кнопок (Button) или элемента TableLayoutPanel.

Реализовать проверку победы (три одинаковых знака в ряд, столбец или диагональ).

Добавить возможность игры против компьютера.

Реализовать счетчик побед для каждого игрока.

Опционально: добавить уровни сложности для ИИ.

Змейка (Snake).

Разработать классическую игру "Змейка", где игрок управляет движением змейки, собирая еду и увеличивая её длину. Игра завершается при столкновении со стеной или собственным телом.

Требования:

Поле игры отображается в элементе управления Panel.

Управление змейкой через клавиши стрелок.

Еда появляется случайным образом на поле.

Реализовать счетчик очков.

Использовать класс Timer для движения змейки.

Опционально: добавить уровни сложности (увеличение скорости).

Пятнашки.

Создать игру "Пятнашки", где игрок перемещает фишки на поле 4x4, чтобы расположить их в правильном порядке.

Требования:

Поле игры отображается с помощью кнопок (Button) или элемента TableLayoutPanel.

Игрок может перемещать фишки, кликая на них.

Добавить возможность генерации случайного начального состояния поля.

Реализовать проверку победы (все фишки в правильном порядке).

Опционально: добавить таймер для подсчета времени.

Сапер (Minesweeper).

Разработать игру "Сапер", где игрок открывает клетки на поле, избегая мин, и использует подсказки (числа) для определения их расположения.

Требования:

Поле игры отображается с помощью кнопок (Button) или элемента TableLayoutPanel.

Реализовать генерацию мин случайным образом.

При открытии клетки отображать число мин в соседних клетках.

Добавить функционал пометки клеток флажками.

Реализовать проверку победы (все клетки без мин открыты).

Опционально: добавить уровни сложности (размер поля и количество мин).

Арканойд (Arkanoid).

Создать игру "Арканойд", где игрок управляет платформой, отбивая мяч и разбивая блоки.

Требования:

Поле игры отображается в элементе управления Panel.

Реализовать движение мяча с использованием класса Timer.

Игрок управляет платформой с помощью клавиш стрелок.

Добавить блоки, которые разрушаются при ударе мяча.

Реализовать счетчик очков за разбитые блоки.

Опционально: добавить уровни сложности (увеличение скорости мяча, новые типы блоков).

Тетрис.

Разработать классическую игру "Тетрис", где игрок управляет падающими фигурами, чтобы заполнять горизонтальные линии.

Требования:

Поле игры отображается в элементе управления Panel.

Реализовать движение фигур с использованием класса Timer.

Игрок может поворачивать и перемещать фигуры с помощью клавиш.

Добавить проверку заполненных линий и их удаление.

Реализовать счетчик очков и уровень сложности (увеличение скорости падения).

Опционально: добавить режим паузы и сохранение рекордов.

Гонки.

Создать простую игру "Гонки", где игрок управляет машиной, избегая препятствий и собирая бонусы.

Требования:

Поле игры отображается в элементе управления Panel.

Реализовать движение машины с помощью клавиш стрелок.

Препятствия и бонусы появляются случайным образом и движутся сверху вниз.

Реализовать счетчик очков за избегание препятствий и сбор бонусов.

Опционально: добавить уровни сложности (увеличение скорости).

Камень, ножницы, бумага.

Разработать игру "Камень, ножницы, бумага", где игрок соревнуется с компьютером.

Требования:

Интерфейс включает кнопки для выбора варианта ("Камень", "Ножницы", "Бумага").

Компьютер делает случайный выбор.

Реализовать подсчет побед игрока и компьютера.

Добавить возможность играть несколько раундов.

Опционально: добавить анимации и эффекты для улучшения визуального восприятия.

Судоку.

Создать игру "Судоку", где игрок решает головоломку, заполняя числа в сетке 9x9.

Требования:

Поле игры отображается с помощью текстовых полей (TextBox) или элемента DataGridView.

Реализовать генерацию случайной головоломки Судоку.

Проверять корректность ввода чисел.

Реализовать проверку победы (все числа расставлены правильно).

Опционально: добавить подсказки и режим решения головоломки автоматически.

Виселица.

Разработать игру "Виселица", где игрок угадывает слово, вводя буквы.

Требования:

Интерфейс включает отображение загаданного слова (пустые места для букв), кнопки для ввода букв и рисунок виселицы.

Реализовать выбор случайного слова из списка.

Отображать ошибочные буквы и количество оставшихся попыток.

Реализовать проверку победы (слово угадано) или поражения (исчерпаны попытки).

Опционально: добавить категории слов (например, животные, города) и анимированный рисунок виселицы.

Пинг-Понг (Pong).

Создать классическую игру "Пинг-Понг", где два игрока управляют ракетками, отбивая мяч друг другу.

Требования:

Поле игры отображается в элементе управления Panel.

Реализовать движение мяча с использованием класса Timer.

Игроки управляют ракетками с помощью клавиш (например, W/S для одного игрока, стрелки вверх/вниз для другого).

Добавить счетчик очков за каждый пропущенный мяч.

Опционально: добавить режим игры против компьютера.