**Общие требования к реализации работ**

* Организация дружелюбного пользовательского интерфейса
  + проверка вводимых пользователем данных
  + вывод адекватных и понятных пользователю сообщений (в особенности – об ошибках).
* Отсутствие, по возможности, жёстко закодированных значений (например, «зашитых» в код путей к файлам, магических чисел).
* Использование идентификаторов, передающих назначение элементов (например, переменных, методов, и т.д.). Рекомендуется использовать слова или аббревиатуры на английском языке.
* Соблюдение принципа инкапсуляции (например, поля класса, как правило, не должны быть общедоступными для изменения).
* Соблюдение принципа единственной обязанности (например, методы класса не должны осуществлять ввод данных с консоли или вывод их на неё, если такое поведение не характерно для сущности, которую он описывает).
* Разработанная программа должна получать все необходимые данные из внешнего источника (например, с консоли, из файла, базы данных, от удалённого сервера, и т.д.), то есть при запуске не содержит никакой информации.
* После окончания работы приложения, оно должно сохранять информацию о прошлом сеансе.

**Лабораторная работа №1**

**Цель работы**

Изучить принципы построения диалоговых консольных приложений, применив на практике знания базовых синтаксических конструкций языка C# и объектно-ориентированного программирования.

**Описание работы**

Диалоговые консольные приложения – консольные программы, работа с которыми ведётся в диалоговом режиме («запрос-ответ»). В качестве запроса может выступать ввод команды, либо выбор пункта меню, выведенного программой в консоль. После получения запроса программа может потребовать ввода необходимых для выполнения запрошенной операции данных. После получения необходимой информации программа осуществляет соответствующее действие, выводит результат работы в консоль и ожидает следующего запроса. Данные о работе приложения необходимо сохранять в файл заданного формата.

В процессе выполнения работы необходимо:

1. понять, какие существуют сущности и процессы в проектируемой системе;
2. выделить основные функции системы;
3. продумать логику работы пользователя с программой: предусмотреть набор действий пользователя и множество состояний программы;
4. реализовать объектно-ориентированную модель в виде программы.
5. Реализовать сериализацию.

**Варианты хранения файла:**

1. JSON
2. XML
3. CSV

**19. Интерактивный туристический гид**

Разработайте программу для путешественников, которая предоставляет информацию о достопримечательностях, маршрутах и советах для туристов. Программа выбирает наиболее интересные места по маршрутам, а также хранит историю рекомендованных путей

**Поиск достопримечательностей:**

* Пользователь выбирает маршрут или указывает интересующий регион.
* Программа предлагает список наиболее популярных достопримечательностей в этом регионе или на маршруте.

**Планирование маршрутов:**

* Пользователь может создавать маршруты, добавляя интересующие точки или достопримечательности.
* Программа помогает оптимизировать маршрут и предлагает последовательность посещения мест.

**Хранение истории рекомендованных путей:**

* Программа сохраняет историю маршрутов и посещенных мест для каждого пользователя.
* Пользователь может просматривать предыдущие маршруты, их описание и рекомендации.

**Советы для туристов:**

* Программа предоставляет полезные советы и рекомендации для каждой достопримечательности или маршрута.
* Это включает информацию о лучшее время для посещения, советы по безопасности и особенностям местного климата.

Добро пожаловать в программу для путешественников!

Меню:

1. Поиск достопримечательностей по региону

2. Планирование маршрута

3. Просмотр истории рекомендованных путей

4. Выйти из программы

Выберите действие (1-4): 1

Введите регион или город, который вас интересует: Париж

Наиболее популярные достопримечательности в Париже:

1. Эйфелева башня

2. Лувр

3. Нотр-Дам де Пари

4. Шанз-Элизе

5. Монмартр

Меню:

1. Поиск достопримечательностей по региону

2. Планирование маршрута

3. Просмотр истории рекомендованных путей

4. Выйти из программы

Выберите действие (1-4): 2

Добавьте точки для маршрута:

1. Эйфелева башня

2. Лувр

Выберите точку (1-2) или введите 'готово', чтобы завершить: 1

Точка 'Эйфелева башня' добавлена к маршруту.

Выберите точку (1-2) или введите 'готово', чтобы завершить: 2

Точка 'Лувр' добавлена к маршруту.

Выберите точку (1-2) или введите 'готово', чтобы завершить: готово

Оптимальный маршрут:

1. Эйфелева башня

2. Лувр

Меню:

1. Поиск достопримечательностей по региону

2. Планирование маршрута

3. Просмотр истории рекомендованных путей

4. Выйти из программы

Выберите действие (1-4): 3

История рекомендованных путей:

1. Маршрут по Парижу (Эйфелева башня -> Лувр)

- Оптимальный маршрут для первого посещения города.

Меню:

1. Поиск достопримечательностей по региону

2. Планирование маршрута

3. Просмотр истории рекомендованных путей

4. Выйти из программы

Выберите действие (1-4): 4

Спасибо за использование программы! Приятного путешествия!

**Материалы для выполнения работы**

1. Гайд по работе с JSON: <https://learn.microsoft.com/ru-ru/dotnet/standard/serialization/system-text-json/how-to?pivots=dotnet-7-0>
2. XML:
   1. <https://learn.microsoft.com/ru-ru/dotnet/standard/serialization/xml-and-soap-serialization>
   2. <https://learn.microsoft.com/ru-ru/dotnet/standard/serialization/how-to-serialize-an-object>
   3. <https://learn.microsoft.com/ru-ru/dotnet/standard/serialization/how-to-deserialize-an-object>

**Лабораторная работа №4**

**Цель работы**

Изучить основы работы с базами данных в языке C#, а также освоить TASK & async абстракции языка.

**Описание работы**

Модифицировать программу, полученную в результате выполнения работы №2 таким образом, чтобы она хранила данные в sqlite. (В качестве провайдера к базе необходимо использовать EF\Dapper\Linq2db согласно варианту) для максимального балла необходимо использовать PSQL или MySQL

**Лабораторная работа №5**

**Цель работы**

Изучить основы веб разработки

**Описание работы**

Модифицировать программу, полученную в результате выполнения работы №3 таким образом, чтобы она предоставляла доступ к данным посредством REST API. Требуется использовать ASP.NET Core Web API.

**Лабораторная работа №6**

**Цель работы**

Изучить основы создания приложений с помощью (WPF\Avalonia\MAUI) по вариантам, применив навыки программирования на C# и понимание концепций разработки интерактивных графических приложений.

**Описание работы**

Необходимо реализовать программу, функционально аналогичную (как минимум) тем, что были разработаны в результате выполнения лабораторных работ №2 и №3.

Отображаемое при запуске окно программы предоставляет пользователю доступ к основной функциональности.

Связь данных в элементах управления и обработчиков взаимодействий с пользователем должна быть реализована с помощью таких возможностей как привязки данных (Data Bindings), шаблоны (Data Templates) и стили (Styles).