Варианты проекта:

* Картотека электронных книг. *Элементы данных: книга – название, ISBN, авторы, количество страниц, тег(и), год издания, издательство; автор – имя, фото, дата рождения, список книг; издательство – название, город, список книг.*
* Каталог фильмов пункта проката. *Элементы данных: фильм – название, постер, жанр(ы),возрастное ограничение, актеры, режиссер, год выпуска; актер – имя, дата рождения, краткая биография, список фильмов; режиссер – имя, список фильмов.*
* Система тестирования знаний (с редактором тестов). *Элементы данных: тест – название, предмет, количество времени, режим измерения времени (на весь тест, на каждый вопрос, …), список вопросов; вопрос – текст вопроса, ценность в баллах, тест, к которому он относится; вопрос первого типа – дополнительно варианты ответа, некоторые из которых правильные; вопрос второго типа – дополнительно левый и правый список, между элементами которых нужно установить правильное соответствие; вопрос третьего типа – без вариантов ответа, предусматривающий текстовый ответ.*
* Система управления группами и студентами на факультете. *Элементы данных: студент – фамилия, имя, отчество, фото, год первого поступления, оценки, группа, группы, в которых он обучался раньше, и периоды обучения в других группах; группа – номер, староста (студент), имена преподавателей в каждом семестре по каждому предмету, расформирована ли.*
* Складской учёт компьютеров и компьютерных комплектующих. *Элементы данных: компьютер – инвентарный номер, материнская плата, блок питания, винчестеры, приводы оптических дисков; материнская плата – серийный номер, производитель, модель, тип сокета процессора, установленный процессор, количество и типы слотов ОЗУ, установленные ОЗУ, установленная видеокарта, список внешних разъемов; блок питания – серийный номер, производитель, номинальная мощность; винчестер – серийный номер, производитель, емкость, тип SATA; привод оптических дисков – серийный номер, производитель, тип (CD-R, DVD-RW, …); процессор – серийный номер, производитель, модель, тип сокета, тактовая частота; ОЗУ – серийный номер, производитель, тип, объем, частота; видеокарта – серийный номер, производитель, модель; дополнительно все комплектующие – инвентарный номер компьютера, в котором они установлены.*
* Программа для управления персоналом предприятия. *Элементы данных: работник – фамилия, имя, отчество, фото, должность, дата приема на работу на предприятие и дата увольнения, предыдущие должности и время, проведенное на этих должностях, отдел, начальник работника и его подчиненные из других отделов; отдел – название, год образования, описание деятельности, начальник.*
* Каталог данных о звёздах и планетах. *Элементы данных: созвездие – название, изображение, положение на звездной карте, звезды; звезда – название, радиус, масса, светимость, тип, планеты-спутники; планета – название, радиус, масса, период обращения вокруг своей оси, тело, вокруг которого планета вращается, период ее обращения вокруг этого тела, радиус орбиты.*
* Генеалогическое дерево (редактирование и просмотр). *Элементы данных: человек – фамилия, имя, отчество, пол, дата рождения, фото, муж/жена, мать, отец, дети, братья, сестры, двоюродные братья, двоюродные сестры (последние четыре – только для чтения).*

**Лабораторная работа 1. Создание модели и классов для коллекций**

**Постановка задачи**. Необходимо выполнить следующее:

1) Разработать функциональные требования к выбранному проекту – определить список возможностей, которые будут реализованы программой.

2) Создать классы для представления элементов данных и коллекций, с которыми работает приложение. Реализовать стандартные операции для коллекций (добавление, удаление, поиск элемента), а также операции, специфичные для предметной области. Реализовать интерфейс IEnumerable<T> для коллекций. Рассмотреть возможность использования нестандартного базового класса для создания коллекции (KeyedCollection, OrderedDictionary и т. п.).

**Лабораторная работа 2. Создание приложения с помощью WPF**

**Постановка задачи**. Необходимо выполнить следующее:

1) Разработать общую архитектуру проекта, в том числе определить необходимый набор форм для приложения WPF.

2) Создать необходимый набор форм для приложения. Применить различные элементы управления и несколько вариантов компоновки окна.

**Лабораторная работа 3. Связывание с данными**

**Постановка задачи**. Необходимо реализовать отображение данных приложения с помощью привязки данных. Воспользоваться такими возможностями, как конвертирование значений, проверка данных (при редактировании). Рассмотреть различные способы задания источника и различные режимы привязки.

**Лабораторная работа 4. Файлы и потоки данных**

**Постановка задачи**. Необходимо реализовать методы для сохранения и чтения данных приложения с помощью текстовых и двоичных файлов. Применить при чтении/за­писи потоки, поддерживающие сжатие информации (по желанию – дополнительно шифровать информацию). Продемонстрировать в создаваемом приложении работу с классами для объектов файловой системы (копирование и переименование файлов, отслеживание изменения состояния файла). *(Например, создание резервной копии файла перед перезаписью; или, если программа загрузила данные из файла, и потом этот файл изменился, предложить загрузить новое содержимое.)*

**Лабораторная работа 5. Работа с технологией отражения**

**Постановка задачи**. Необходимо разработать для приложения систему плагинов. Плагины приложения являются классами, реализующими заранее оговоренный интерфейс (содержимое этого интерфейса на ваше усмотрение). При запуске программа должна зайти в поддиректорию /plugins, найти в ней все сборки .dll и в этих сборках все классы, реализующие интерфейс плагина.

В главном окне программы необходимо создать главное меню. Первая часть этого меню может представлять операции, типичные для оконных приложений, работающих с данными (открыть файл, сохранить файл, выход…). Вторая часть должна генерироваться динамически и позволять пользоваться всеми плагинами, которые приложение обнаружило при запуске. По желанию можно создать контекстные меню для основных компонентов окна.

**Лабораторная работа 6. Использование технологии LINQ to Objects**

**Постановка задачи**. Необходимо реализовать с помощью LINQ to Objects методы для выборки, упорядочения и группировки данных в классах модели, вычисления агрегатных функций. Воспользоваться LINQ-запросами для отображения в приложении необходимой части данных..

**Лабораторная работа 7. Программирование с использованием задач**

**Постановка задачи**. Необходимо выполнить следующее:

1) Осуществлять поиск и загрузку плагинов в фоновом режиме с помощью асинхронных операций. Приложение должно запускаться и быть доступным для работы без подменю плагинов, а меню плагинов должно появляться после того, как оно будет полностью построено. Обратите внимание на то, что приложения WPF используют модель многопоточности Single Thread Affinity.

2) Переписать некоторые операции обработки данных в приложении с помощью PLINQ или TPL.

3) Реализовать постраничный просмотр для некоторых используемых коллекций. Минимальным постраничным просмотром являются кнопки «В начало», «Предыдущий», «Следующий», «В конец». При переключении страниц программа должна в фоновой задаче подготовить следующую и предыдущую страницу для новой позиции, чтобы при нажатии пользователем на кнопку запрошенная страница отображалась без задержки.

**Лабораторная работа 8. Работа с ресурсами и конфигурацией**

**Постановка задачи**. Необходимо дополнить меню форм горячими клавишами и пиктограммами. Пиктограммы каждого пункта меню должны храниться в XML-файле ресурсов и загружаться при запуске приложения. Настройки горячих клавиш сохранить в XML-файле конфигурации (эти настройки также должны считываться и применяться при запуске приложения). Писать отдельную форму настройки необязательно, но при сдаче работы нужно продемонстрировать, как изменения в конфигурационном файле отражаются в меню приложения. Также в конфигурационном файле нужно сохранить имя папки (или несколько имен папок), в которой приложение при запуске должно искать свои плагины.

**Лабораторная работа 9. Графика и анимация**

**Постановка задачи**. Необходимо создать визуальный стиль для некоторых (минимум двух) элементов управления. Создать шаблоны для элементов данных, используемых в проекте. Создать анимации для элементов управления (эффект «свечения», плавного изменения размеров при выделении и т.п.). Использовать разные виды анимации (с использованием интерполяции, ключевых кадров, пути). Реализовать элементы 2D-графики / трансформаций / графических эффектов (можно представить график или диаграмму, построенную по неким данным проекта; нарисовать логотип или векторный рисунок).