Весенний семестр 1 курса

1. Простая программа на языке C#

Постановка задачи. Необходимо разработать простейшую консольную программу на языке C# для своего варианта задания. Ознакомиться с возможностями MS Visual Studio по созданию приложений для платформы .Net Framework. Скомпилировать написанную программу, используя компилятор командной строки csc.exe.

Варианты:

* Решение уравнения третьей или четвертой степени
* Вычисление арифметического выражения
* Решение физической или алгоритмической задачи
* Написание простейшей игры (например, текстовый вариант «21»)

2. Работа с числами, датами и текстом

Постановка задачи. Необходимо ознакомиться с базовыми возможностями классов DateTime, String, StringBuilder, Char, Math, Random, BigInteger, Decimal, Double, Single, UInt64, Int64, UInt32, Int32, UInt16, Int16, Byte, SByte, Boolean. Выполнить три задания из предложенных на выбор.

Варианты:

* Получить текущее время и дату в двух разных форматах и вывести на экран количество нулей, единиц, …, девяток в их записи.
* В заданной строке поменять порядок слов на обратный (слова разделены пробелами).
* Дана строка, слова которой разделены пробелами. Распознать в ней слова, являющиеся числами в шестнадцатеричной системе счисления, и вывести их десятичный эквивалент.
* Рассчитать максимальную степень двойки, на которую делится произведение подряд идущих чисел от a до b (числа целые 64-битные без знака).
* Дана строка. Найти в ней все заглавные буквы, не входящие в английский алфавит.
* Реализовать эффективное перемешивание символов строки.
* Дана строка. Записать номера символов, входящих в нее, в виде шестнадцатеричных чисел через пробелы. Числа должны быть ровно из четырех цифр (возможно, с нулями в начале).
* С помощью класса DateTime вывести на консоль названия месяцев на французском языке. По желанию обобщить на случай, когда язык задается с клавиатуры.
* Сгенерировать равновероятно случайную строку длиной не более четырех строчных английских букв.
* Дана строка, содержащая число с десятичной точкой. Преобразовать эту строку в число действительного типа (не пользуясь стандартным Parse/TryParse).
* Дана строка, слова в которой разделены пробелами. Есть знаки препинания, которые записаны сразу после слова. Добавить перед каждым словом тот знак препинания, который стоит после него.
* Дана строка, слова в которой разделены пробелами. Вывести все слова, содержащие буквы, не входящие в английский алфавит. Вывод должен быть выровнен по правому краю и иметь ширину самого длинного слова.
* Дана строка из 256 английских букв. Записать через пробел 30 символов этой строки, стоящих на случайных местах. Желательно сделать только одно обращение к классу Random.
* Реализовать вычисление параметров треугольника (стороны, углы, периметр, площадь, радиусы вписанной и описанной окружностей, …) по трем заданным параметрам.
* Дана строка, состоящая из строчных английских букв. Заменить в ней все буквы, стоящие после гласных, на следующие по алфавиту (z заменяется на a).

3. Создание класса на языке C#

Постановка задачи. Необходимо реализовать первый из классов предметной области согласно выбранному варианту. Предусмотреть необходимый набор методов, полей, свойств, конструкторов и индексаторов в реализуемом классе. Реализовать статические элементы класса (например, создание уникального Id), перегрузку методов. Продемонстрировать работу с созданным классом.

Варианты:

* Человек – Спортсмен – Специалисты по отдельным видам спорта.
* Человек – Студент – Студенты отдельных специальностей.
* Предмет мебели – Шкаф – Виды шкафов (для посуды, для обуви и т. д.).
* АЛУ – Процессор – Процессоры разных производителей.
* Транспортное средство – Автомобиль – Автомобили разных марок.
* Помещение – Жилая комната – Виды комнат (гостиная, спальня, детская и т. д.).

4. Взаимодействие с неуправляемым кодом

Постановка задачи. Необходимо реализовать в коде .NET взаимодействие с неуправляемым кодом. Выполнить задание согласно выбранному варианту. Дополнительно ознакомиться с понятием конвенций вызова, написать динамическую библиотеку на языке C++ и воспользоваться ее возможностями в коде .NET. Библиотека должна предоставлять функции с различными конвенциями вызова.

Варианты:

* Простое рисование на рабочем столе с помощью GDI
* Аналог утилиты WinSight (просмотр информации об окнах)
* Утилита для просмотра информации о компьютере (CPU, RAM, …)
* Проигрывание музыкальных файлов с помощью MCI
* Утилита для записи всех нажатий клавиш (Key logger)
* Просмотр информации о процессах (упрощенный аналог менеджера задач)

5. Наследование, полиморфизм, структуры, перечисления

Постановка задачи. Необходимо создать классы, производные от класса, созданного в работе 3 (на нижнем уровне иерархии достаточно трех классов). Выделить и реализовать полиморфные и (по возможности) абстрактные элементы классов. Создать структуры и перечисления, подходящие для описания предметной области, и воспользоваться ими в написанных классах.

6. Интерфейсы и универсальные шаблоны

Постановка задачи. Необходимо создать интерфейсы, подходящие для описания исследуемой предметной области. Изучить стандартные интерфейсы (IComparable<T>, IEquatable<T>, IFormattable и т. д.) и реализовать какие-нибудь из них в созданных классах. Изучить принципы обобщенного программирования .NET Framework.

7. Преобразование и сравнение объектов

Постановка задачи. Необходимо разработать и протестировать класс для представления рационального числа. Реализовать в этом классе перекрытие математических операций. Реализовать несколько методов для представления объекта класса в виде строки. Реализовать метод для получения объекта класса по строковому представлению (по желанию использовать для этого регулярные выражения). Реализовать в классе стандартные интерфейсы для проверки двух объектов на равенство и порядкового сравнения. Перекрыть явные или неявные операторы преобразования к типам целых и действительных чисел. Рекомендуется обратить внимание на правильные шаблоны реализации этих методов и операций.

8. Делегаты, события, перегрузка операций, исключительные ситуации

Постановка задачи. Необходимо дополнить классы, созданные в ходе лабораторных работ, событиями. Тестирование событий выполнить при помощи назначения обработчиков, записанных в форме анонимных методов и лямбда-выражений. Реализовать в методах классов генерацию исключительных ситуаций (в тех случаях, когда что-либо мешает нормальному выполнению операции) и предусмотреть обработку этих исключений в коде, использующем эти классы.

Осенний семестр 2 курса

1. Создание модели и классов для коллекций

Постановка задачи. Необходимо создать классы для представления указанных коллекций. Реализовать стандартные операции для коллекции (добавление, удаление, поиск элемента). Реализовать интерфейс IEnumerable<T> для коллекций. Рассмотреть возможность использования нестандартного базового класса для создания коллекции.

Варианты:

* *Факультет* объединяет студенческие *группы*, а группа – *студентов*.
* *Покупка* в магазине ‑ это коллекция *товаров*. Товары в покупке могут быть в нескольких экземплярах.
* *Тест* содержит коллекцию *вопросов*, а вопрос ‑ *варианты ответов.*
* *Компьютерная сеть* содержит *компьютеры*, связанные с некими другими компьютерами в сети.
* *Матрица* – это коллекция *строк* (числовых векторов).
* *Генеалогическое дерево* содержит *людей*.

2. Простое приложение на WPF

Постановка задачи. Необходимо реализовать простое однооконное приложение WPF для просмотра и редактирования коллекции, созданной в работе 1. Применить различные элементы управления и несколько вариантов компоновки окна.

3. Связывание с данными

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

4. Файлы и потоки данных

Постановка задачи. Необходимо реализовать возможность сохранения и чтения коллекции с помощью текстовых и двоичных файлов. Применить при чтении/записи потоки, поддерживающие сжатие информации (по желанию – дополнительно шифровать информацию).

5. Сериализация

Постановка задачи. Необходимо реализовать класс, позволяющий сериализовать объекты произвольных типов. Класс может производить сериализацию в любой формат данных, поддерживать один из стандартных механизмов сериализации или предоставлять новый. Применить созданный класс для сохранения данных приложения в файл.

6. LINQ to Objects

Постановка задачи. Необходимо реализовать с помощью LINQ to Objects методы для выборки из коллекции, упорядочения и группировки элементов коллекции, вычисления агрегатных функций. Воспользоваться LINQ-запросами для отображения в приложении необходимой части коллекции. По желанию сделать свою реализацию оператора упорядочения, группировки, объединения или работы с множествами, применяя концепцию отложенных вычислений там, где это возможно.

7. Многопоточное программирование

Постановка задачи. Необходимо создать класс для представления музыкальной композиции со следующими атрибутами: ID (уникальный); название (строка длиной до 256 символов); длина (минуты, секунды); исполнитель (строка длиной до 256 символов); жанр (значение из перечисления допустимых жанров); рейтинг (целое число от 0 до 10). Создать класс для представления плей-листа, представляющего коллекцию неповторяющихся композиций и обладающего следующими элементами: ID (уникальный); название (строка длиной до 256 символов); общая длина (минуты, секунды); рейтинг (высчитывается как среднее рейтинга композиций). Обеспечить хранение плей-листа в файле (в произвольном формате). Создать «плеер» для загрузки и «проигрывания» плей-листов. Проигрывание должно происходить в отдельном потоке, при этом допускается одновременное проигрывание нескольких плей-листов. Проигрыванием можно управлять с консоли (пауза/стоп). При «проигрывании» композиции выводятся текстовые метки или звуковые сигналы для каждых 5 секунд проигрывания.

8. Графика и анимация

Постановка задачи. Необходимо создать визуальный стиль для некоторых (минимум двух) элементов управления. Создать шаблон для выбранного элемента управления или элемента данных. Создать в проекте анимации для элементов управления (эффект «свечения», плавного изменения размеров при выделении и т.п.). Использовать несколько видов анимации (с использованием линейной интерполяции, ключевых кадров, пути). Применить в проекте работу с картинками. По желанию освоить основы работы с аудио или видео в WPF.

9. Работа с базами данных. ADO.NET

Постановка задачи. Необходимо создать базу данных для хранения данных приложения. При разработке базы уделить внимание следующим аспектам:

* наличие в таблицах ограничений (уникальность значений, первичные ключи);
* наличие связей между таблицами;
* индексы таблиц.

Написать SQL-скрипты для создания объектов базы (таблиц, связей). Написать SQL-скрипты для вставки в таблицы базы тестовых значений. Реализовать основные операции с сущностями проекта (создание, чтение, изменение, удаление). По желанию изучить и применить шаблон проектирования MVVM для реализации визуальной части проекта.

Весенний семестр 2 курса

1. Анимация на CSS3, HTML и JS

Постановка задачи. Необходимо разработать простое браузерное приложение, включающее работу с HTML, CSS, анимацией в CSS. Применить различные варианты селекторов CSS. Обратить внимание на модель компоновки, используемую в веб, особенности структуры HTML по сравнению с XML. По желанию можно дополнить приложение возможностью взаимодействия с пользователем, реализовав логику взаимодействия на языке JavaScript.

Варианты:

* Анимация вращения тетраэдра / гексаэдра / октаэдра / додекаэдра / икосаэдра
* Анимация движения человечка
* Анимация оригами (складывания фигурки из листа бумаги)
* Различные варианты анимации «двустороннего» плоского элемента (информация на одной стороне, по определенному событию элемент поворачивается обратной стороной, на которой также есть информация)

2. Простейшее WebForms приложение

Постановка задачи. Необходимо реализовать простое WebForms приложение, предоставляющее возможность просмотра данных по схеме «список-подробности» и возможность добавления информации пользователем. Применить различные элементы управления, проверочные элементы при вводе данных пользователем. По желанию ознакомиться с элементами для представления источников данных.

Варианты:

* Поваренная книга с рецептами
* Список участников спортивной команды
* Список заданий или мероприятий
* Ассортимент товаров
* Список маршрутов городского транспорта
* Словарь иностранного языка

Темы проектов:

* **Простая социальная сеть.** Регистрация пользователей. Настройка персональной информации. «Друзья» пользователя. Обмен сообщений с другими пользователями. Поиск пользователей по определенным критериям. Управление пользователями. Модерирование сообщений.
* **Интернет-фотоальбом.** Регистрация, загрузка фотографий. Возможность просматривать и оценивать фотографии других пользователей. Поиск фотографий. Управление пользователями. Модерирование материалов.
* **Персональный блог/блогхост.** Регистрация пользователей. Создание блога (блогов). Создание и редактирование статей блога. Тэги статей. Поиск по тэгам, тексту. Комментирование статей. Управление пользователями. Модерирование статей и комментариев.
* **Интернет-аукцион.** Просмотр, поиск, и «покупка» лотов. Регистрация и выставление лотов на аукцион. Управление пользователями и модерирование списка лотов.
* **Система тестирования знаний.** Регистрация, выбор (поиск) теста. Прохождение теста с контролем времени. Статистика тестирования. Управление пользователями. Редактирование тестов. Расширенная статистика тестирования.
* **Файловое хранилище.** Доступ к файлу по короткой ссылке. Общие файлы и файлы с ограниченным доступом. Для зарегистрированных пользователей – возможность размещения файлов и управление файлами. Управление пользователями и их файлами. Поиск файлов.
* **Картотека текстовых материалов (вариант: аудио или видео).** Поиск в картотеке по различным критериям. Оценка материалов пользователями. Управление пользователями. Модерирование материалов.
* **Система учета знаний.** Зарегистрированные пользователи («программисты») указывают и оценивают свои знания в различных (сгруппированных) областях. Пользователь - «менеджер» осуществляет отбор программистов по заданным критериям. Возможность генерирования отчетов. Управление пользователями и ролями. Управление списками областей знаний.
* **Система отслеживания заданий.** Выдача задания менеджером. Статус задания, согласно рабочему процессу. Процент выполнения. Почтовые уведомления клиентам системы. Управление пользователями и их ролями.
* **Форум.** Стандартные операции, присущие любому форуму, – добавление тем, сообщений. Модерирование записей. Работа с пользователями форума.

(синим цветом отмечены возможности администраторов и модераторов)

3. Разработка каркаса MVC приложения

Постановка задачи. Необходимо создать начальный набор контроллеров и представлений сайта, а также реализовать базовые операции с моделью. Применить при написании представлений помощники написания HTML. Продумать систему эталонных и частичных представлений. По желанию использовать в одном проекте несколько движков представлений.

4. Настройка маршрутизации

Постановка задачи. Необходимо расширить функциональность веб-сайта добавлением маршрутов, ограничений на параметры, фильтров действий. Разделить логику обработки пользовательских запросов на несколько контроллеров и предусмотреть соответствующие маршруты. Применить несколько видов ограничений на маршрут, в том числе создание собственного ограничения. По желанию создать собственный обработчик маршрута и ознакомиться с маршрутизацией атрибутами.

5. Безопасность веб-приложений

Постановка задачи. Необходимо реализовать на сайте страницы, доступные пользователям определенной роли (администраторы, модераторы и т. д.), и предусмотреть ограничение доступа на эти страницы. Реализовать возможность регистрации и аутентификации пользователей.

6. Веб-службы WCF и Web API

Постановка задачи. Необходимо создать веб-службу WCF или Web API для работы с данными веб-проекта. Протестировать службу, используя консольное приложение или небольшой вспомогательный веб-проект.

7. Использование AJAX

Постановка задачи. Необходимо реализовать избранные функции веб-сайта с применением AJAX. Требуется продемонстрировать два подхода – применение стандартных компонентов AJAX и реализация AJAX-функционала «вручную», по базовым принципам этой технологии. По желанию освоить на практике третий подход – реализацию AJAX с помощью библиотеки jQuery или другой Javascript-платформы.

8. Использование SignalR или развертывание

Постановка задачи. Необходимо выполнить одно из двух заданий (по желанию – оба):

* Реализовать избранные функции веб-сайта с применением SignalR;
* Создать для сайта пакет развертывания и продемонстрировать развертывание сайта на веб-сервере без применения IDE.