Условия лабораторных работ и домашнего задания по курсу ПиК ЯП (языки C# и F#)

Условия лабораторных работ по языку C#

Лабораторная работа №1

Разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Программа осуществляет ввод с клавиатуры коэффициентов A, B, C, вычисляет дискриминант и корни уравнения (в зависимости от дискриминанта).
3. Если коэффициент A, B, C введен некорректно (не приводится к действительному числу), то необходимо проигнорировать некорректное значение и ввести коэффициент повторно.
4. Корни уравнения выводятся зеленым цветом. Если корней нет, то сообщение выводится красным цветом.
5. Коэффициенты A, B, C задаются в виде параметров командной строки. Если они не указаны, то вводятся с клавиатуры в соответствии с пунктом 2. Проверка из пункта 3 в этом случае производится для параметров командной строки без повторного ввода с клавиатуры.

Контрольные вопросы:

1. Какими способами можно преобразовать значение строкового типа в значение числового типа?
2. Какие символьные и строковые типы данных существуют в C#?
3. Какие средства консольного ввода/вывода существуют в C#?
4. Какие целочисленные типы данных существуют в C#?
5. Как задаются и обрабатываются аргументы командной строки в консольном приложении?
6. Как работает механизм обработки исключений в C#?
7. Какие условные операторы существуют в C#?
8. Как работает цикл foreach?
9. Как работает форматированный вывод в консоль?

10. Какие операторы цикла существуют в C#?

Лабораторная работа №2

Разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Абстрактный класс «Геометрическая фигура» содержит виртуальный метод для вычисления площади фигуры.
3. Класс «Прямоугольник» наследуется от «Геометрическая фигура». Ширина и высота объявляются как свойства (property). Класс должен содержать конструктор по параметрам «ширина» и «высота».
4. Класс «Квадрат» наследуется от «Прямоугольник». Класс должен содержать конструктор по длине стороны.
5. Класс «Круг» наследуется от «Геометрическая фигура». Радиус объявляется как свойство (property). Класс должен содержать конструктор по параметру «радиус».
6. Для классов «Прямоугольник», «Квадрат», «Круг» переопределить виртуальный метод Object.ToString(), который возвращает в виде строки основные параметры фигуры и ее площадь.
7. Разработать интерфейс IPrint. Интерфейс содержит метод Print(), который не принимает параметров и возвращает void. Для классов «Прямоугольник», «Квадрат», «Круг» реализовать наследование от интерфейса IPrint. Переопределяемый метод Print() выводит на консоль информацию, возвращаемую переопределенным методом ToString().

Контрольные вопросы:

1. В чем разница между ключевыми словами «override» и «new» при переопределении виртуального метода?
2. Как переопределить виртуальный метод?
3. Как реализуется наследование класса от класса?
4. Как объявить конструктор класса в C#?
5. Как из конструктора класса вызвать конструктор базового класса?
6. Что такое свойства и для чего они используются?
7. Что такое опорная переменная свойства?
8. Как задаются области видимости для свойств и аксессоров?
9. Что такое абстрактный класс?

10. Что такое интерфейсы и для чего они используются в C#?

Лабораторная работа №3

Разработать программу, реализующую работу с коллекциями.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создать объекты классов «Прямоугольник», «Квадрат», «Круг».
3. Для реализации возможности сортировки геометрических фигур для класса «Геометрическая фигура» добавить реализацию интерфейса IComparable. Сортировка производится по площади фигуры.
4. Создать коллекцию класса ArrayList. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.
5. Создать коллекцию класса List<Figure>. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.
6. Модифицировать класс разреженной матрицы (проект SparseMatrix) для работы с тремя измерениями – x,y,z. Вывод элементов в методе ToString() осуществлять в том виде, который Вы считаете наиболее удобным. Разработать пример использования разреженной матрицы для геометрических фигур.
7. Реализовать класс «SimpleStack» на основе односвязного списка. Класс SimpleStack наследуется от класса SimpleList (проект SimpleListProject). Необходимо добавить в класс методы:
    - public void Push(T element) – добавление в стек;
    - public T Pop() – чтение с удалением из стека.
8. Пример работы класса SimpleStack реализовать на основе геометрических фигур.

Контрольные вопросы:

1. В чем сходство и различие между обобщенными и необобщенными коллекциями?
2. Как осуществляется работа с кортежем?
3. Как осуществляется сортировка коллекций?
4. Как можно реализовать класс разреженной матрицы на основе класса словаря?

5. Как можно реализовать классы списка и стека без использования стандартных коллекций?
6. Как осуществляется работа с обобщенным списком?
7. Как осуществляется работа с необобщенным списком?
8. Как осуществляется работа с обобщенным стеком?
9. Как осуществляется работа с обобщенной очередью?
10. Как осуществляется работа с обобщенным словарем?

Лабораторная работа №4

Разработать программу, реализующую работу с файлами.

1. Программа должна быть разработана в виде приложения Windows Forms на языке C#. По желанию вместо Windows Forms возможно использование WPF.
2. Добавить кнопку, реализующую функцию чтения файла в список слов List<string>.
3. Для выбора имени файла используется класс OpenFileDialog, который открывает диалоговое окно с выбором файла. Ограничить выбор только файлами с расширением «.txt».
4. Для чтения из файла рекомендуется использовать статический метод ReadAllText() класса File (пространство имен System.IO). Содержимое файла считывается методом ReadAllText() в виде одной строки, далее делится на слова с использованием метода Split() класса string. Слова сохраняются в список List<string>.
5. При сохранении слов в список List<string> дубликаты слов не записываются. Для проверки наличия слова в списке используется метод Contains().
6. Вычислить время загрузки и сохранения в список с использованием класса Stopwatch (пространство имен System.Diagnostics). Вычисленное время вывести на форму в поле ввода (TextBox) или надпись (Label).
7. Добавить на форму поле ввода для поиска слова и кнопку поиска. При нажатии на кнопку поиска осуществлять поиск введенного слова в списке. Слово считается найденным, если оно входит в элемент списка как подстрока (метод Contains() класса string).
8. Добавить на форму список (ListBox). Найденные слова выводить в список с использованием метода «название_списка.Items.Add()». Вызовы метода «название_списка.Items.Add()» должны находится

между вызовами методов «название_списка.BeginUpdate()» и «название_списка. EndUpdate()».

9. Вычислить время поиска с использованием класса Stopwatch. Вычисленное время вывести на форму в поле ввода (TextBox) или надпись (Label).

Контрольные вопросы:

1. Как создать приложение Windows Forms?
2. Как реализовать измерение времени выполнения программы с использованием класса Stopwatch?
3. Для чего используются события FormClosed и FormClosing?
4. Как используется класс OpenFileDialog?
5. Как используется класс SaveFileDialog?
6. Как используется элемент TextBox?
7. Как осуществить чтение текстового файла в виде единой строки?
8. Как осуществить запись текстового файла в виде единой строки?
9. Как осуществить чтение текстового файла в виде массива строк?
10. Как осуществить запись текстового файла в виде массива строк?

Лабораторная работа №5

Разработать программу, реализующую вычисление расстояния Левенштейна с использованием алгоритма Вагнера-Фишера.

1. Программа должна быть разработана в виде библиотеки классов на языке C#.
2. Использовать самый простой вариант алгоритма без оптимизации.
3. Дополнительно возможно реализовать вычисление расстояния Дамерау-Левенштейна (с учетом перестановок соседних символов).
4. Модифицировать предыдущую лабораторную работу, вместо поиска подстроки используется вычисление расстояния Левенштейна.
5. Предусмотреть отдельное поле ввода для максимального расстояния. Если расстояние Левенштейна между двумя строками больше максимального, то строки считаются несовпадающими и не выводятся в список результатов.

Контрольные вопросы:

1. Что такое расстояние Левенштейна?

2. Приведите пример вычисления расстояния Левенштейна.
3. Что такое расстояние Дамерау-Левенштейна?
4. Приведите пример вычисления расстояния Дамерау-Левенштейна.
5. В чем состоит поправка Дамерау?
6. Для чего используется транспозиция в поправке Дамерау?
7. Объясните алгоритм Вагнера-Фишера вычисления расстояния Дамерау-Левенштейна (на основе матрицы).
8. Как инициализируются начальные значения в матрице при вычислении расстояния Левенштейна? Почему?
9. Как задать различные веса для операций удаления, добавления и замены?
10. Как осуществить интеграцию разработанного метода в приложение Windows Forms.

Лабораторная работа №6

Часть 1. Разработать программу, использующую делегаты.

(В качестве примера можно использовать проект «Delegates»).

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Определите делегат, принимающий несколько параметров различных типов и возвращающий значение произвольного типа.
3. Напишите метод, соответствующий данному делегату.
4. Напишите метод, принимающий разработанный Вами делегат, в качестве одного из входным параметров. Осуществите вызов метода, передавая в качестве параметра-делегата:
    - метод, разработанный в пункте 3;
    - лямбда-выражение.
5. Повторите пункт 4, используя вместо разработанного Вами делегата, обобщенный делегат Func< > или Action< >, соответствующий сигнатуре разработанного Вами делегата.

Часть 2. Разработать программу, реализующую работу с рефлексией.

(В качестве примера можно использовать проект «Reflection»).

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создайте класс, содержащий конструкторы, свойства, методы.

3. С использованием рефлексии выведите информацию о конструкторах, свойствах, методах.
4. Создайте класс атрибута (унаследован от класса System.Attribute).
5. Назначьте атрибут некоторым свойствам классам. Выведите только те свойства, которым назначен атрибут.
6. Вызовите один из методов класса с использованием рефлексии.

Контрольные вопросы:

1. Что такое делегат?
2. Что такое неявная типизация и как она используется для делегатов?
3. Как используется обобщенный делегат Func?
4. Как используется обобщенный делегат Action?
5. Что такое лямбда-выражения и как они используются?
6. Что такое рефлексия и для чего она используется?
7. Как реализуется работа с атрибутами?
8. Как реализуются динамические действия с объектами классов?
9. Как реализуется работа со сборками?
10. Как получить информацию о типе на основе класса и на основе инициализированного объекта класса?

Лабораторная работа №7

Разработать программу, реализующую работу с LINQ to Objects. В качестве примера используйте проект «SimpleLINQ» из примера «Введение в LINQ».

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создайте класс «Сотрудник», содержащий поля:
    - ID записи о сотруднике;
    - Фамилия сотрудника;
    - ID записи об отделе.
3. Создайте класс «Отдел», содержащий поля:
    - ID записи об отделе;
    - Наименование отдела.
4. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением один-ко-многим разработайте следующие запросы:
    - Выведите список всех сотрудников и отделов, отсортированный по отделам.

- Выведите список всех сотрудников, у которых фамилия начинается с буквы «А».
- Выведите список всех отделов и количество сотрудников в каждом отделе.
- Выведите список отделов, в которых у всех сотрудников фамилия начинается с буквы «А».
- Выведите список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А».

5. Создайте класс «Сотрудники отдела», содержащий поля:
   - ID записи о сотруднике;
   - ID записи об отделе.
6. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением много-ко-многим с использованием класса «Сотрудники отдела» разработайте следующие запросы:
   - Выведите список всех отделов и список сотрудников в каждом отделе.
   - Выведите список всех отделов и количество сотрудников в каждом отделе.

Контрольные вопросы:

1. Для чего используется технология LINQ?
2. Что такое провайдер в технологии LINQ и для чего он используется?
3. Для каких видов данных существуют провайдеры в технологии LINQ?
4. В чем особенность технологии LINQ to Objects?
5. Как осуществляется сортировка данных в технологии LINQ?
6. Как осуществляется группировка данных в технологии LINQ?
7. Как осуществляется соединение данных (join) в технологии LINQ?
8. Как осуществляется соединение данных (join) для связи много-ко-многим в технологии LINQ?
9. Как работают операции над множествами в технологии LINQ?
10. Что такое анонимные типы и как они используются в технологии LINQ?

Условия лабораторных работ по языку F#

Лабораторная работа №1

Составить программу на функциональном языке программирования для решения биквадратного уравнения с использованием алгоритма рассмотренного в разделе «Биквадратное уравнение» статьи https://ru.wikipedia.org/wiki/Уравнение_четвёртой_степени. Программа должна использовать алгебраические типы и механизм сопоставления с образцом.

В случае комплексных корней их вычисление не обязательно, можно выводить информацию о том, что корни комплексные.

Контрольные вопросы:

1. В чем недостатки реализации квадратного уравнения на языке C#, которая использует список корней?
2. В чем недостатки реализации квадратного уравнения на языке C#, которая использует список корней и enum?
3. В чем преимущество использования алгебраических типов (discriminated unions) в F#?
4. Как написать функцию, которая возвращает значение алгебраического типа в F#?
5. Что такое механизм сопоставления с образцом и как он используется в F#?
6. Приведите пример использования механизма сопоставления с образцом для алгебраического типа.
7. Приведите пример использования механизма сопоставления с образцом для интерфейса и наследуемых классов.
8. Как на языке C# реализовать решение квадратного уравнения с использованием интерфейса и наследуемых классов?
9. Как на языке F# реализовать решение квадратного уравнения с использованием интерфейса и наследуемых классов?
10. В чем сходства и различия между решениями на основе алгебраического типа и на основе интерфейса и наследуемых классов?

Лабораторная работа №2

1. Создайте два варианта функции, которая возвращает кортеж значений. Первый вариант принимает на вход параметры в виде кортежа, второй вариант параметры в каррированном виде.

2. Выберите простой алгоритм, который может быть реализован в виде рекурсивной функции и реализуйте его в F#. Пример – вычисление суммы целых чисел в заданном диапазоне.

3. Преобразуйте разработанную рекурсивную функцию в форму хвостовой рекурсии.

4. Разработайте конечный автомат из трех состояний и реализуйте его в виде взаимно-рекурсивных функций.

5. Разработайте функцию, которая принимает 3 целых числа и лямбда-выражение для их суммирования в виде кортежа и в каррированном виде.

Контрольные вопросы:

1. Что такое кортеж?
2. Что такое каррирование? В чем его преимущества?
3. Приведите пример каррирования на языке C#.
4. Приведите пример каррирования на языке F#.
5. Что такое рекурсия?
6. В чем отличие хвостовой рекурсии от обычной рекурсии?
7. Каждое ли рекурсивное выражение можно преобразовать в форму хвостовой рекурсии?
8. Как на основе взаимно-рекурсивных функций реализовать конечный автомат?
9. Как записываются лямбда-выражения в F#?
10. Как используются символы «*» и «->» для описания типов в F#?

Лабораторная работа №3

1. Разработайте функцию, которая принимает три параметра обобщенных типов и возвращает их в виде кортежа. Модифицируйте функцию: не указывая явно типы параметров, задавая выражения в теле функции, сделайте так, чтобы параметры были типов int, float, string.

2. С использованием двухэтапного создания обобщенных функций реализуйте функции, которые осуществляют сложение:

- трех аргументов типа int;
- трех аргументов типа float;
- трех аргументов типа string.

3. С использованием list comprehension для четных элементов списка [1..10] верните список кортежей. Каждый кортеж содержит элемент списка, его квадрат и куб.

4. Напишите два варианта функции, которая принимает на вход список и возвращает квадраты его значений. Необходимо использовать свойства списка Head и Tail. Первый вариант функции использует оператор if, второй вариант использует сопоставление с образцом на уровне функции.

5. Последовательно примените к списку функции map, sort, filter, fold, zip, функции агрегирования. Функции применяются в любом порядке и произвольно используются в трех комбинациях.

- Первая комбинация заканчивается функцией агрегирования (например, сумма элементов списка). Список предварительно может быть отсортирован, отфильтрован и т.д.
- Вторая комбинация заканчивается функцией fold, которая осуществляет свертку списка. Вторая комбинация выполняет те же действия, что и первая комбинация и должна возвращать такой же результат.
- Третья комбинация заканчивается функцией zip, которая соединяет два списка.

6. Реализуйте предыдущий пункт с использованием оператора потока « |> ».

7. Реализуйте предыдущий пункт с использованием оператора композиции функций « >> ».

Контрольные вопросы:

1. Что такое кортеж?
2. Как работает двухэтапное создание обобщенных функций?
3. Как работает механизм list comprehension?
4. Как работает функция sort?
5. Как работает функция filter?
6. Как работает функция fold?
7. Как работает функция zip?
8. В чем особенности работы функции zip для списков разной длины?
9. Что такое оператор потока « |> » и как он работает?

10.Что такое оператор композиции функций « >> » и как он работает?

Лабораторная работа №4

Часть 1. Использование классов, интерфейсов и наследования.

Разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке F#.

2. Абстрактный класс «Геометрическая фигура» содержит виртуальный метод для вычисления площади фигуры.

3. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Ширина и высота объявляются как свойства (property). Класс должен содержать конструктор по параметрам «ширина» и «высота».

4. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.

5. Класс «Круг» наследуется от класса «Геометрическая фигура». Радиус объявляется как свойство (property). Класс должен содержать конструктор по параметру «радиус».

6. Для классов «Прямоугольник», «Квадрат», «Круг» переопределить виртуальный метод Object.ToString(), который возвращает в виде строки основные параметры фигуры и ее площадь.

7. Разработать интерфейс IPrint. Интерфейс содержит метод Print(), который не принимает параметров и возвращает void. Для классов «Прямоугольник», «Квадрат», «Круг» реализовать наследование от интерфейса IPrint. Переопределяемый метод Print() выводит на консоль информацию, возвращаемую переопределенным методом ToString().

Часть 2. Использование алгебраического типа и сопоставления с образцом.

1. Реализуйте класс геометрическая фигура в виде алгебраического типа (discriminated union), который содержит варианты (дискриминаторы) «Прямоугольник», «Квадрат», «Круг» с необходимыми параметрами.

2. Разработайте для данного класса функцию вычисления площади. Функция должна принимать параметр типа «геометрическая фигура» и вычислять различные варианты площади в зависимости от дискриминатора. Необходимо использовать механизм сопоставления с образцом.

Контрольные вопросы:

1. Как объявить в F# класс, который содержит члены класса и несколько конструкторов?
2. Как объявляются абстрактные методы в классах F#?
3. Как объявляются виртуальные методы в классах F#?
4. Как реализовать перегрузку виртуального метода в классе F#?
5. Как объявить в F# абстрактный класс?
6. Как объявить в F# интерфейс?
7. Как описать реализацию интерфейса в классе F#?
8. Что такое записи (record types) и как они используются?
9. Как описать алгебраический тип (discriminated union)?
10. Что такое взаимно рекурсивные типы? Как объявить такой тип данных в F#?

Лабораторная работа №5

Для произвольно выбранного типа данных (например, Maybe) реализуйте функции функтора, аппликативного функтора, монады.

Проверьте для Вашей реализации справедливость соответствующих законов для функтора, монады и аппликативного функтора (тех законов, которые можно проверить с использованием F#). Некоторые законы могут не выполняться. Это означает что данный тип не является в полной мере функтором, аппликативным функтором, монадой.

Контрольные вопросы:

1. Для чего используются функторы, аппликативные функторы и монады?
2. Что такое функтор? Его основные операции?
3. Как работает функция fmap?
4. Сформулируйте два основных закона функторов. Приведите примеры.
5. Что такое аппликативный функтор? Его основные операции?
6. Как работает функция apply?

7. Сформулируйте четыре основных закона аппликативных функторов. Приведите примеры.
8. Что такое монада? Ее основные операции?
9. Как работает функция bind?
10. Сформулируйте три основных закона монад. Приведите примеры.

Лабораторная работа №6

Разработайте программу, которая осуществляет разбор текста с использованием библиотеки FParsec. Результатом разбора должны быть значения алгебраического типа.

Контрольные вопросы:

1. Как работает подход Parser Combinator для анализа текста?
2. Что такое парсеры в подходе Parser Combinator?
3. Приведите пример работы готового парсера из библиотеки FParsec.
4. Приведите пример работы механизма сопоставления с образцом в парсер-комбинаторном подходе.
5. Что такое комбинаторы в подходе Parser Combinator?
6. Как работает комбинатор « >>. »?
7. Как работает комбинатор « .>> »?
8. Как реализовать разбор списков?
9. Как работает комбинатор выбора <|> ?
10. Как реализовать разбор данных с применением алгебраического типа?

Лабораторная работа №7

С использованием класса MailboxProcessor реализуйте агента, который реагирует на внешние события и выполняет различные действия (например, выдает результаты в консоль).

Контрольные вопросы:

1. Что такое мультиагентный (многоагентный) подход?
2. Что такое акторный подход?
3. В чем сходства и различия между многоагентным и акторным подходами?
4. Как реализовать актор с использованием класса MailboxProcessor?
5. Приведите пример реализации актора на основе класса MailboxProcessor.