# Список лабораторных работ (ИСП, 1 семестр изучения)

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Тема работы | К-во  занятий | Баллы |
| 1 | Основные концепции ООП | 2 | 8 |
| 2 | Простая программа на языке C# | 1 | 4 |
| 3 | Создание класса на языке C# | 1 | 6 |
| 4 | Наследование, полиморфизм, интерфейсы | 1 | 6 |
| 5 | Универсальные шаблоны | 1 | 4 |
| 6 | События, исключительные ситуации, перегрузка операций, пространства имён | 1 | 6 |

# Работа №1. Основные концепции ООП

Цель работы.

Ознакомиться с концепциями ООП в среде программирования Delphi.

Постановка задачи.

Разработать консольное приложение в среде программирования Delphi, используя объектно-ориентированный подход:

1. Предусмотреть необходимый набор классов для описания предметной области.

2. Предусмотреть необходимый набор методов, свойств и полей в проектируемых классах.

3. Предусмотреть возможности тестирования созданных классов.

Варианты классов.

1. Класс для представления списка товаров. Товар характеризуется уникальным названием, количеством единиц, ценой. Реализовать методы для добавления и редактирования товаров, поиска товаров, «покупки» товара.

2. Класс для представления телефонного справочника. Учесть, что у человека может быть несколько телефонов. Реализовать добавление, удаление, поиск информации (по номеру и по фамилии).

3. Класс для представления данных о дате и времени. Реализовать доступ к отдельным компонентам информации о времени (секунды, минуты, дни). Реализовать возможность арифметических операций с датами (сложение, вычитание). Предоставить возможность преобразования даты в строку.

4. Класс для представления массива бит. Реализовать методы для чтения или установки произвольного бита в массиве, возможность представления массива бит как строки, побитовые логические операции.

5. Класс для представления матриц целых чисел. Размер матрицы произволен. Реализовать операции над матрицами: сложение, умножение матрицы на скаляр, транспонирование, вычисление определителя (в случае, если это возможно).

6. Класс для представления многоугольников на плоскости. Многоугольник характеризуется набором координат вершин. Реализовать возможность вычисления площади многоугольника, перемещения многоугольника на произвольный вектор, поворот многоугольника относительно произвольной точки.

7. Класс для представления таблицы, элементы которой представляет собой пару «ключ-значение ключа». «Ключ» – строка, «значение ключа» – целое число. Реализовать поиска значения по ключу, подтверждение наличия ключа или значения в таблице, сортировку таблицы по ключу.

8. Класс для представления алгебраических многочленов от одной переменной. Реализовать возможность алгебраических операций с многочленами, реализовать метод для вычисления значения многочлена в данной точке.

9. Класс для представления множества. Элементами множества являются фразы из слов. Два элемента считаются идентичными, если представляющие их фразы состоят из одинаковых слов (регистр не учитывать) (Например: «too many» и «Many too»). Необходимо реализовать операции над такими множествами (объединение, пересечение, разность).

10. Класс для представления разреженных числовых матриц (только малая часть элементов отлична от нуля). Реализовать некоторые операции над такими матрицами (сложение, транспонирование, умножение).

11. Класс для представления больших целых чисел (несколько десятков цифр). Реализовать арифметические операции с такими числами. Предоставить возможность представления числа в виде строки.

12. Класс для представления неориентированных графов. Поместить в класс методы добавления и удаления вершин и рёбер графа, просмотра существующих вершин и рёбер.

# Работа №2. Простая программа на языке C#

Цели работы.

1. Изучить структуру простой программы на языке C#.

2. Изучить возможности среды программирования Visual Studio 2010 (или другой IDE) по созданию программ для платформы .Net Framework.

3. Изучить возможности компилятора командной строки csc.exe.

Постановка задачи.

Ознакомится с возможностями Visual Studio 2010 по созданию программ для платформы .Net Framework 4.0. Разработать простую консольную программу на языке C# для вычисления функции при заданных значениях аргументов. Результаты вычислений сохранять в массиве. Скомпилировать написанную программу, используя компилятор командной строки csc.exe.

Варианты заданий.

1.

2.

3.

4.

5.

6.

7.

8.

# Работа №3. Создание класса на языке C#

Цели работы.

1. Изучить синтаксис описания класса в языке C#.

2. Изучить использование элементов класса (полей, методов, свойств).

3. Изучить понятие статического элемента и статического класса.

4. Изучить синтаксис описания и использования расширяемых методов.

Постановка задачи.

В задании необходимо реализовать класс согласно указанному варианту. Предусмотреть необходимый набор методов, свойств и индексаторов в реализуемых классах. Изучить статические классы и статические элементы класса. Расширить созданные ранее классы статическими элементами (например, для работы с уникальным числовым идентификатором объекта) и методами расширения.

Варианты заданий.

1. Стек для хранения строк.

2. Очередь для хранения целых чисел.

3. Дек (deque, двунаправленная очередь) для строк.

4. Двусвязный список для хранения объектов, описывающих человека.

5. Хеш-таблица, хранящая строки.

6. Граф с числовой информацией в вершинах и рёбрах.

7. Разреженная числовая матрица.

8. Матрица целых чисел.

9. Множество (как тип данных) объектов, описывающих человека.

10. Бинарное дерево поиска, хранящее строки.

11. Односвязный список для хранения целых чисел.

12. Очередь для хранения строк.

# Работа №4. Наследование, полиморфизм, интерфейсы

Цели работы.

1. Изучить синтаксис и механизмы наследования и полиморфизма в C#.

2. Изучить понятия структуры и перечисления.

Постановка задачи.

Необходимо реализовать классы, согласно указанному варианту. Выделить общие элементы классов в отдельный класс (или интерфейс).

Варианты заданий.

1. Односвязный список для хранения строк. Двусвязный список для хранения строк.

2. Хеш-таблица, хранящая строки. Хеш-таблица, хранящая числа.

3. Граф с числовой информацией в вершинах. Граф с числовой информацией на рёбрах.

4. Разреженная матрица. Разреженная диагональная матрица.

5. Матрица целых чисел. Верхняя треугольная матрица чисел.

6. Множество (как тип данных) строк. Упорядоченное множество строк.

7. Произвольное дерево. Бинарное дерево.

8. Односвязный список для хранения целых чисел. Односвязный список для хранения строк.

9. Дек для строк. Стек для хранения строк.

10. Стек для хранения строк. Стек для хранения коротких строк (длина меньше 10 символов).

11. Матрица целых чисел. Ленточная матрица чисел.

12. Дек для строк. Очередь для хранения строк.

# Работа №5. Универсальные шаблоны

Цель работы.

Изучить понятие универсального шаблона.

Постановка задачи.

В лабораторной работе требуется изучить принципы обобщённого программирования в C# и .Net Framework. Необходимо разработать обобщённые версии классов, созданных в рамках работы №3.

# Работа №6. События, исключительные ситуации, перегрузка операций, пространства имён

Цели работы:

1. Изучить понятия делегата и λ-выражения.

2. Изучить понятие исключительной ситуации.

3. Ознакомится с перегрузкой операций в языке C#.

4. Получить представление о пространствах имён.

Постановка задачи.

В работе требуется расширить классы, созданные в работе №5, событиями. По возможности, заменить некоторые из методов класса методами, принимающими в качестве параметра делегат. Реализовать в классах перегрузку операций (обязательными являются операции неявного и явного приведения типов). Предусмотреть в методах класса возможность генерации исключительных ситуаций (как системных, так и пользовательских). Создать собственные пространства имён, в которые поместить разработанные классы.