Программа курса «Языки программирования»
мехмат, 1 курс, направление «Фундаментальная информатика и информационные технологии» 2 семестр 2021–2022 уч. г.

Раздел 1. Регулярные выражения. Файлы. Абстрактные типы данных и классы

Регулярные выражения

1. Недостатки методов строк s.Split и s.IndexOf. Понятие регулярного выражения. Методы Regex.Match, Regex.IsMatch, Regex.Matches, Regex.Split и примеры их использования. Аналогичные экземплярные методы и их отличие от статических. Классы Match и MatchCollection.
2. Квантификаторы. Примеры использования.
3. Директивы нулевой длины. Примеры использования.
4. Классы символов. Примеры использования.
5. Группы. Именованные группы. Ссылки на группы. Ссылки на именованные группы. Примеры использования.
6. Жадная и ленивая квантификация.
7. Утверждения положительного просмотра вперед нулевой ширины. Утверждения положительного просмотра назад нулевой ширины. Пример: текст внутри тега.
8. Утверждения отрицательного просмотра вперед нулевой ширины. Утверждения отрицательного просмотра назад нулевой ширины.
9. Замена с помощью регулярных выражений. Замена с вычислением выражений. Замена именованных групп. Примеры использования.

Файлы

10. Определение файла. Преимущества и недостатки файлов. Классификация файлов по типу компонент и по способу доступа.
11. Основные операции при работе с закрытыми и открытыми файлами.
12. Основные пространства имён и классы .NET, связанные с файлами. Класс файлового потока FileStream, операции чтения и записи для FileStream. Режимы открытия файла.
13. Механизм обработки исключений. Оператор try – catch, алгоритм его работы. Оператор try – finally. Обработка исключений при работе с файлами – примеры.
14. Оператор using и его использование при открытии-закрытии файла.
15. Файловый указатель. Действия с файловым указателем. Цикл по FileStream. Буферизация файлов.
16. Потоковые адаптеры для текстовых файлов и работа с ними: чтение, запись, цикл по текстовому файлу. Вывод числовой информации в текстовые потоки.
17. Кодировки текстовых файлов. Методы класса File для создания текстовых файлов.
18. Потоковые адаптеры для двоичных файлов. Считывание-запись с помощью двоичных адаптеров. Цикл по двоичному файлу на чтение и на запись. Исключение EndOfStreamException. Возведение всех элементов в квадрат.
19. Текстовые файлы как последовательности строк. Примеры преобразования текстовых файлов. Парсер строк текстового файла, хранящего данные о персонах.
20. Сериализация. Бинарная сериализация. SOAP-сериализация. JSON-сериализация.

Введение в классы. Классы как абстрактные типы данных. Стандартные классы .NET

21. Классы и объекты. Модификаторы защиты доступа. Свойства в C#. Автоматически реализуемые свойства. Отличие класса от структуры. Представление в памяти.
22. Record-классы. Инициализатор объекта record-класса. Сравнение на равенство. Оператор with. Деконструктор. Короткий синтаксис record-классов.

23. Типы значений, допускающие значение null. Операции ?. и ?? Примеры использования: вызов процедуры, вызов функции для ссылочного типа, вызов функции для размерного типа.
24. Абстрактный тип данных (АТД). Класс как конкретный и как абстрактный тип данных. Принцип отделения интерфейса от реализации, его преимущества.
25. АТД Стек, его интерфейс. Вычисление значения выражения в польской инверсной записи с помощью стека: описание алгоритма и его реализация.
26. АТД и класс Очередь, его интерфейс. Примеры использования очереди: алгоритм заливки области.
27. Стандартные классы коллекций .NET: Stack<T>, Queue<T>, List<T>, HashSet<T>, SortedSet<T>, Dictionary<K,V>, SortedDictionary<K,V>, LinkedList<T>. Коллекции как последовательности.
28. Стандартный класс словаря (ассоциативного массива) Dictionary<K,V>. Ключи и значения. Инициализация, цикл по словарю. Индексное свойство.
29. Реализация словаря на базе списка пар. Делегирование.
30. Исключение KeyNotFoundException. Другие методы словаря.
31. Примеры использования: словарь количества встречаемости слов в файле.

Раздел 2. Динамические структуры данных и рекурсия

Динамические структуры данных

32. Статические и динамические структуры данных. Списки: линейные и циклические, односвязные и двусвязные. Класс узла списка.
33. Основные операции с линейными односвязными списками: вставка в начало, удаление из начала, вставка после текущего, удаление следующего, проход по списку, поиск. Недостатки линейного односвязного списка.
34. Класс двусвязного списка MyLinkedList<T>. Основные операции с линейными двусвязными списками: инициализация, вставка элемента в начало и конец, вставка элемента в середину перед и после данного, удаление элемента в начале, середине и конце списка, проход по списку, поиск элемента в списке.
35. Стандартные классы LinkedList<T>, LinkedListNode<T>, их основные методы и свойства. Пример: удаление четных в LinkedList<T>.
36. Сравнение асимптотической сложности операций двусвязных списков и массивов.
37. Реализация стека на базе структуры данных «линейный односвязный список». Реализация стека на базе LinkedList<T>.
38. Реализация очереди на базе структуры данных «линейный односвязный список». Реализация очереди на базе LinkedList<T>.

Рекурсия

39. Рекурсия, примеры. Праворекурсивные и леворекурсивные определения, примеры. Прямая и косвенная рекурсия. Грамматика языка программирования и формы Бэкуса-Наура. Рекурсивные подпрограммы. Прямая и косвенная рекурсия.
40. Примеры рекурсивного зацикливания. Простейшие примеры рекурсии. Глубина рекурсии, текущий уровень глубины рекурсии. Рекурсивный спуск и рекурсивный возврат.
41. Графическое изображение рекурсии (2 способа).
42. Примеры рекурсии: факториал числа, степень числа (2 способа, отличающиеся глубиной рекурсии), нахождение минимального элемента в массиве (2 способа), анализ глубины рекурсии в каждом случае.
43. Рекурсивный обход линейного односвязного списка. Концевая рекурсия и итерация.

44. Доказательство завершимости рекурсии.
45. Программный стек: определения, рисунок. Вызов подпрограммы на этапе выполнения. Запись активации, её содержимое. Что происходит при вызове подпрограммы и при выходе из подпрограммы. Пример использования программного стека при рекурсии. Переполнение программного стека.
46. Виды рекурсивных подпрограмм (5 шт.). Функция Аккермана.
47. Примеры плохого использования рекурсии: числа Фибоначчи. Каскадная рекурсия и дерево рекурсивных вызовов. Оптимизация рекурсивного алгоритма вычисления чисел Фибоначчи: метод с концевой рекурсией.
48. Примеры использования рекурсии: ханойские башни. Глубина рекурсии, количество рекурсивных вызовов.
49. Бинарный поиск в отсортированном массиве.
50. Быстрая сортировка Хоара. Алгоритмы Partition и QuickSort. Оценка количества операций при быстрой сортировке в среднем. Сравнение с пузырьковой сортировкой. Быстрая сортировка в худшем случае.
51. Генерация всех перестановок. Глубина рекурсии и количество рекурсивных вызовов.
52. Генерация всех подмножеств. Глубина рекурсии и количество рекурсивных вызовов.

Деревья

53. Деревья, примеры. Основные определения: вершины и ребра, корень, листья, крона, глубина дерева.
54. Бинарные (двоичные) деревья. Рекурсивное определение бинарного дерева. Идеально сбалансированное и полное бинарное дерево.
55. Обходы деревьев: инфиксный, префиксный, постфиксный. Вывод элементов, полученных в результате обхода. Определение количества элементов в бинарном дереве, поиск элемента.
56. Создание идеально сбалансированного бинарного дерева.
57. Определение количества листьев в бинарном дереве.
58. Поиск элемента в бинарном дереве.
59. Нерекурсивный обход бинарного дерева в ширину с помощью очереди. Нерекурсивный обход бинарного дерева в глубину с помощью стека.
60. Связь деревьев и рекурсии.
61. Бинарные деревья поиска (БДП). Основные операции при работе с БДП: добавление, поиск.
62. Сортировка деревом. Оценка количества операций при добавлении и поиске в БДП, при сортировке деревом. Сортировка деревом в худшем случае.
63. Удаление из БДП (только алгоритм).
64. Реализация множества на базе бинарного дерева поиска.
65. Реализация ориентированного графа в виде набора инцидентных вершин. Остовное дерево графа. Алгоритм обхода графа в глубину с получением глубинного остовного дерева. Алгоритм обхода графа в ширину с получением остовного дерева в ширину.
66. Бинарные деревья – алгоритм определения минимальной суммы от корня к листу. Отсечение заведомо неверных частичных решений.
67. Алгоритм перебора с возвратом. Обход конем шахматной доски. Оптимизация обхода конем шахматной доски.

Раздел 3. Объектно-ориентированное программирование

Наследование. Исключения

68. Наследование, примеры. Диаграмма наследования на UML. Пример наследования Person – Student. Переопределяющие методы. Вызов унаследованного конструктора.

Цели наследования. Наследование как расширение и как сужение. Иерархия наследования – примеры. Хранение потомка в памяти.

69. Принцип «Открыт-закрыт» и его роль при проектировании сложных систем. Учет будущих изменений. Пример: очередь с фильтрацией.
70. Наследование и включение. Пример: очередь с фильтрацией, реализованная включением; недостатки.
71. Когда следует использовать наследование, а когда – включение. Примеры.
72. Виды отношений между классами: ассоциация, агрегация, композиция, наследование, реализация интерфейсов. UML-диаграммы классов. Пример UML-диаграммы Персона-Преподаватель-Студент-Старшекурсник-Группа-Кафедра-Институт.
73. Наследование и выявление общего предка.
74. Модификатор доступа protected.
75. Принцип подстановки Барбары Лисков.
76. Класс System.Exception, его свойства. Иерархия наследования исключений .NET. Обработка исключений разных типов – примеры. Порядок записи обработчиков исключений. Создание собственных типов исключений. Обработка всех исключений. Генерация исключения.
77. Приведение типов в иерархии предок-потомок. DownCast и UpCast.
78. Операции is и as. Расширенный is.

Полиморфизм и интерфейсы

79. Определение полиморфизма. Раннее и позднее связывание.
80. Позднее связывание и виртуальные методы. Переопределение виртуального метода. Полиморфные переменные, статический и динамический тип. Цепочка виртуальности и её разрыв. Алгоритм поиска в цепочке виртуальности.
81. Виртуальные методы как блоки для замены кода.
82. Полиморфные контейнеры. Обработка всех элементов вызовом виртуальных методов (на примере метода Work()). Абстрактные типы. Полиморфные контейнеры и операции as-is.
83. Класс Object – неявный предок всех классов в NET. Метод GetType() и его сравнение с операцией is. Пример – родословная объекта. Полиморфные контейнеры и метод GetType(). Pattern matching и оператор switch по типам.
84. Переопределение методов ToString(), Equals() и GetHashCode() в классе Person. Возможность определения HashSet<Person>.
85. Интерфейсы. Что может и что не может присутствовать в интерфейсе. Реализация интерфейсов классами.
86. Совместимость по присваиванию и операции is as для интерфейсов. Интерфейсы и полиморфизм.
87. Интерфейсы как роли.
88. SOLID – пять основных принципов ООП. Принцип инверсии зависимости – примеры.
89. Стандартные интерфейсы NET. Интерфейс сравнения IComparable<T> и его использование для создания SortedSet<Person>. Интерфейс сравнения IComparer<T> и его использование для сортировки списка персон.
90. Внутренняя реализация интерфейсов. Интерфейс IEnumerable<T> и его реализация.
91. Ограничения, налагаемые на параметры обобщений в методах и классах. Секция where, виды ограничений. Примеры: обобщенная функция поиска минимального элемента в массиве, обобщённый класс MySortedSet<T> на базе бинарного дерева поиска.