

Программа курса «Основы программирования»
мехмат, 1 курс, направление «Фундаментальная информатика и
информационные технологии»
2 семестр 2015–2016 уч. г.

Модуль 1. Файлы. Введение в классы

Файлы

1. Определение файла. Преимущества файлов. Классификация файлов по типу компонент и по способу доступа. Текстовые и двоичные файлы, их основные характеристики.
2. Файловая переменная. Файловый указатель. Основные операции при работе с файлами. Способы открытия файлов, различия между ними. Буферизация файлов.
3. Подпрограммы для работы с закрытыми файлами.
4. Типизированные файлы, подпрограммы для работы с типизированными файлами. Добавление элемента в конец файла. Возведение всех элементов в квадрат. Использование типизированных файлов для работы с простейшими базами данных. ~~Сортировка элементов файла.~~
5. Бестиповые двоичные файлы.
6. Текстовые файлы. Открытие текстового файла, указание кодировки при открытии.
7. Функции для работы с текстовыми файлами. Сумма чисел в текстовом файле. Обработка строк в текстовом файле.
8. Работа с текстовым файлом как с последовательностью строк.

Введение в классы

9. Классы и объекты. Отличие класса от записи. Представление в памяти. Ссылочная объектная модель.
10. Сборка мусора.
11. Особенности ссылочной объектной модели: присваивание переменных классов, переменные классов в качестве параметров подпрограмм и в качестве возвращаемого значения функций, сравнение переменных классов. Переменная Self.
12. Обобщенные классы.
13. ~~Пространства имен .NET, их подключение к программе на PascalABC.NET. Пространство имен System. Вложенные пространства имен .NET на примере пространства имен System.IO.~~

Модуль 2. Динамические структуры данных и рекурсия

Динамические структуры данных

14. Статические и динамические структуры данных. Списки: линейные и циклические, односвязные и двусвязные. Класс узла списка.
15. Основные операции с линейными односвязными списками: вставка в начало, удаление из начала, вставка после текущего, удаление следующего, проход по списку, ~~носок~~.
16. 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. Добавление второго списка в конец первого.
17. Сравнение линейных двусвязных списков и массивов по производительности операций.

Рекурсия

18. Рекурсия, примеры. Праворекурсивные и леворекурсивные определения. Прямая и косвенная рекурсия. Рекурсивные подпрограммы. Необходимость опережающих объявлений при косвенной рекурсии.
19. Простейшие примеры рекурсии. Глубина рекурсии. Рекурсивный спуск и рекурсивный возврат.
20. Графическое изображение рекурсии (2 способа).
21. ~~Концевая рекурсия и ее замена на итерацию.~~
22. Примеры рекурсии: факториал числа, степень числа (2 способа, отличающиеся глубиной рекурсии), нахождение минимального элемента в массиве (2 способа), вывод списка. Анализ глубины рекурсии в каждом случае. Концевая рекурсия и ее замена итерацией.
23. Доказательство завершимости рекурсии.
24. Использование программного стека при рекурсивных вызовах. Запись активации. Что происходит при вызове подпрограммы и при выходе из подпрограммы. Переполнение программного стека. Рекурсивное заикливание.
25. Формы рекурсивных подпрограмм (5 шт.). Каскадная рекурсия и дерево рекурсивных вызовов.
26. Функция Аккермана.
27. Примеры плохого использования рекурсии: числа Фибоначчи. Оптимизация рекурсивного алгоритма вычисления чисел Фибоначчи: метод динамического программирования и метод с концевой рекурсией.
28. Примеры использования рекурсии: ханойские башни. Глубина рекурсии, количество рекурсивных вызовов.
29. Быстрая сортировка Хоара. Оценка количества операций при быстрой сортировке в среднем. Сравнение с пузырьковой сортировкой. Быстрая сортировка в худшем случае.
30. Генерация всех перестановок.
31. Генерация всех подмножеств.
32. Алгоритм перебора с возвратом. Обход конем шахматной доски.

Деревья

33. Деревья, примеры. Основные понятия: вершины и ребра, корень, листья, глубина дерева, идеально сбалансированное и полное дерево.
34. Рекурсивное определение дерева. Связь деревьев и рекурсии.
35. Обходы деревьев: инфиксный, префиксный, постфиксный. Обратная польская бесскобочная запись выражения.
36. Бинарные деревья. Идеально сбалансированное бинарное дерево, его создание. Задачи на бинарные деревья: вывод элементов в инфиксном, префиксном и постфиксном порядке, определение количества элементов в дереве, поиск элемента.
37. Определение минимальной суммы от корня к листу. Алгоритм перебора с возвратом. Метод ветвей и границ.
38. Бинарные деревья поиска (БДП). Основные операции при работе с БДП: добавление, поиск. Удаление из БДП (только алгоритм).
39. Сортировка деревом. Оценка количества операций при добавлении и поиске в БДП, при сортировке деревом. Сортировка деревом в худшем случае.

Модуль 3. Объектно-ориентированное программирование

Абстрактные типы данных и классы

40. Абстрактный тип данных (АТД), его интерфейс. Класс как реализация АТД. Принцип отделения интерфейса от реализации, его преимущества. Защита доступа. Модификаторы защиты доступа.
41. АТД Стек, его интерфейс. Реализация стека в виде класса на базе массива.
42. Реализация стека в виде класса на базе односвязного списка.
43. Вычисление значения выражения в польской инверсной записи с помощью стека.
44. АТД Очередь, его интерфейс. Реализация очереди на базе односвязного линейного списка.
45. Примеры использования очереди: обход дерева в ширину и алгоритм заливки области.
46. Класс Динамический массив, его спецификация, интерфейс. Реализация класса Динамический массив на базе встроенных динамических массивов. Методы, связанные с выделением памяти.
47. Свойства классов. Свойства Count и Capacity класса Динамический массив.
48. Индексные свойства классов. Свойство Elems класса Динамический массив. Индексные свойства по умолчанию. Генерация и обработка собственных исключений.
49. Класс Множество, его интерфейс. Реализация на базе класса динамического массива. Понятие делегирования.
50. Стандартные классы множеств .NET: HashSet и SortedSet, их сравнение. Цикл по стандартным множествам.
- ~~51. Реализация множества на базе бинарного дерева поиска.~~
52. Класс Словарь (ассоциативный массив), его интерфейс. Ключи и значения. Доступ по индексу. Реализация на базе списка пар. Стандартные классы словарей .NET Dictionary и SortedDictionary, их сравнение. Цикл по стандартным словарям.
53. Частотный словарь слов в файле.
- ~~54. Пространство имен System.Collections.Generic. Стандартные классы коллекций библиотеки .NET: стек, очередь, динамический массив, двусвязный список, ассоциативный массив, множество. Пример использования LinkedList. Цикл foreach по стандартным контейнерам. Сравнение HashSet и SortedSet. Сравнение Dictionary и SortedDictionary.~~
55. Класс Ориентированный граф, реализация в виде списка дуг. Пример использования.
56. Реализация графа в виде списка дуг.
57. Реализация графа в виде списка инцидентных вершин.
58. Алгоритм обхода графа в глубину.

Наследование. Исключения

59. Наследование, примеры. Цели наследования. Наследование как расширение и как сужение. Пример наследования Person – Student. Замещающие методы. Вызов унаследованного конструктора. Вид доступа protected.
60. Принцип «Открыт-закрыт» и его роль при проектировании сложных систем. Учет будущих изменений. Пример: очередь с подсчетом элементов.
61. Наследование и включение. Пример: очередь с подсчетом элементов, реализованная включением; недостатки.
62. Когда следует использовать наследование, а когда – включение. Примеры.
63. Виды зависимостей между классами: ассоциация, агрегация, композиция, наследование, реализация интерфейсов. UML-диаграммы классов. Пример UML-диаграммы Персона-Преподаватель-Студент-Старшекурсник-Группа.

- 64. Наследование и выявление общего предка.
- 65. Присваивание в иерархии предок-потомок. DownCast и UpCast.
- 66. Принцип подстановки.
- 67. Операции is и as.
- 68. Недостатки Assert. Класс System.Exception, его свойства. Иерархия исключений в NET.
- 69. Секции on обработки исключений в блоке try. Порядок записи обработчиков исключений. Примеры.

Полиморфизм и интерфейсы

- 70. Определение полиморфизма. Раннее и позднее связывание.
- 71. Позднее связывание и виртуальные методы. Переопределение виртуального метода. Полиморфные переменные, статический и динамический тип.
- 72. Виртуальные методы как блоки для замены кода. Полиморфные подпрограммы.
- 73. Класс Object – неявный предок всех классов в NET.
- 74. Переопределение методов Equals, GetHashCode и ToString в классах Person и Student.
- 75. ~~Пример редословная переменной.~~
- 76. Цепочка виртуальности и ее разрыв. Алгоритм поиска в цепочке виртуальности.
- 77. Полиморфные контейнеры. Обработка всех элементов вызовом виртуальных методов (на примере контейнера графических фигур и метода Print).
- 78. ~~Таблица виртуальных методов – внутренний механизм реализации полиморфизма.~~
~~Цена вызова виртуальных методов.~~
- 79. Интерфейсы. Что может и что не может присутствовать в интерфейсе. Реализация интерфейсов.
- 80. Совместимость по присваиванию и операции is as для интерфейсов. Интерфейсы и полиморфизм.
- 81. ~~Сравнение наследования интерфейса и наследования реализации.~~
- 82. ~~Стандартные интерфейсы NET. Интерфейс IComparable и его использование для сортировки массива студентов.~~
- 83. Условия, налагаемые на параметры шаблонов. Секция where. Пример: обобщенная функция поиска минимального элемента в массиве.
- 84. ~~Интерфейс IEnumerable<T> и цикл foreach. Классы, реализующие интерфейс IEnumerable<T>.~~