

Программа курса «Основы программирования»
мехмат, 1 курс, направление «Фундаментальная информатика и
информационные технологии»
1 семестр 2018–2019 уч. г.

Модуль 1. Основные конструкции. Циклы

1. Понятие алгоритма. Свойства алгоритма. Эквивалентность алгоритмов. Исполнитель алгоритма. Примеры алгоритмов.
2. Способы описания алгоритмов: словесный, псевдокод, блок-схемы, языки программирования.
3. Критерии сравнения программ.
4. Компиляторы и интерпретаторы.
5. Общая структура программы на языке Паскаль. Понятие типа данных. Основные типы языка Паскаль. Разделы описания переменных и констант. Внутриблочные описания. Множественное описание с инициализацией.
6. Оператор присваивания. Множественное присваивание. Примеры использования. Совместимость типов по присваиванию. Расширенные операторы присваивания: $+=$ $-=$ $*=$ $/=$. Задача о перемене местами двух значений.
7. Оператор вывода. Процедура `writelnFormat` и форматная строка. Интерполированная строка.
8. Функции ввода, в т.ч. `ReadReal`, `ReadReal2`. Обработка ошибок ввода и функция `TryRead`.
9. Ошибки в программе, их виды. Примеры ошибок каждого вида.
10. Условный оператор. Вложенные условные операторы. Составной оператор. Задача о поиске минимума и максимума двух чисел. Задача о выборе одного из трех чисел, значение которого находится между двумя другими. Условная операция.
11. Арифметические выражения. Операции в арифметических выражениях. Тип арифметического выражения. Стандартные арифметические функции, процедуры и методы.
12. Операции `div` и `mod`, `shl` и `shr`, побитовые операции `and`, `or`, `xor`.
13. Логические выражения и операции. Таблицы истинности. Сокращенное вычисление логических выражений. Таблица приоритетов операций языка Паскаль.
14. Оператор выбора варианта. Перечисление и диапазон в метках выбора.
15. Цикл с предусловием. Цикл с постусловием. Примеры: вычисление суммы нечетных двузначных и вычисление $n!$ Моделирование цикла `repeat` с помощью цикла `while` и наоборот. Зацикливание и бесконечные циклы.
16. Цикл `loop`. Цикл `for` (цикл с параметром).
17. Примеры использования циклов: Вывод арифметической прогрессии, вычисление суммы нечетных двузначных, вычисление $n!$ с помощью цикла `for` или `loop`.
18. Считывание последовательности с помощью `ReadSeqInteger` и цикл `foreach` по последовательности. Свойства последовательностей. Преимущества подхода с использованием последовательностей.
19. Примеры использования циклов: подсчет количества чисел, удовлетворяющих некоторому условию, защита от неверного ввода с помощью `TryRead` и цикла.
20. Примеры использования циклов: табулирование функции. Решения с помощью цикла `for` и цикла `while`.
21. Хранение вещественного значения в памяти. Мантисса и порядок. Величина `real.Epsilon`. Погрешность округления и вычислительная погрешность. Сравнение вещественных чисел на равенство.
22. Рекуррентные последовательности. Арифметическая и геометрическая прогрессии как частные случаи рекуррентных последовательностей. Числа Фибоначчи.
23. Примеры использования циклов: алгоритм Евклида нахождения НОД, в т.ч. использование множественного присваивания.
24. Примеры использования циклов: сумма и произведение цифр целого числа.
25. Предусловия и постусловия. Процедура `assert`.

26. Примеры использования циклов: поиск максимума (минимума), поиск условного максимума (минимума). Понятие инварианта цикла. Доказательство корректности алгоритма поиска максимума с помощью инвариантов.
27. Примеры использования циклов: суммирование конечного ряда, вычисление $\text{Sqrt}(x)$ как предела последовательности.
28. Операторы break и continue. Использование флагов при решении задач. Эквивалентные циклы без break и continue. Примеры использования циклов: поиск заданного значения среди введенных (решение с break и эквивалентный алгоритм без break).
29. Зависимость времени работы алгоритма от размера задачи. Понятие асимптотической сложности алгоритма. Асимптотически эквивалентные последовательности ($a_n \sim b_n, n \rightarrow \infty$). Нотация Θ . Таблица наиболее часто встречающихся асимптотических сложностей, соответствующие графики.
30. Примеры использования циклов: алгоритм определения простоты числа, его асимптотическая сложность.
31. Примеры использования циклов: разложение целого числа на простые сомножители.
32. Примеры использования циклов: вычисление корня функции на отрезке. Метод половинного деления. Оценка скорости сходимости: формула количества итераций для достижения указанной точности.

Модуль 2. Подпрограммы, модули, библиотеки

33. Подпрограммы. Цели введения подпрограмм в язык программирования.
34. Процедуры. Описание и вызов процедуры. Формальные и фактические параметры, их соответствие при вызове. Примеры использования процедур. Короткие описания процедур.
35. Способы передачи параметров: по значению и по ссылке. Понятие ссылки. Внутренний механизм передачи параметра по ссылке. Входно-выходные параметры – примеры.
36. Функции. Описание и вызов функции. Короткие описания функций. Переменная Result, ошибка неинициализированной переменной Result. Примеры использования функций: функция $\text{Sign}(x)$, сумма квадратов чисел от a до b. Сравнение функций и процедур.
37. Функции, возвращающие кортежи. Тип кортежа. Распаковка кортежа в переменные.
38. Оператор exit.
39. Локальные и глобальные переменные.
40. Пространство имен и описание переменных с одинаковыми именами.
41. Обращение к нелокальной переменной на чтение и запись и побочный эффект.
42. Принцип локальности при описании переменных.
43. Перегрузка имен подпрограмм. Какую версию перегруженной подпрограммы выбирать?
44. Параметры по умолчанию. Предварительное объявление подпрограмм.
45. Обобщенные подпрограммы. Вывод типа обобщенной подпрограммы при ее вызове.
46. Процедурный тип и процедурные переменные. Процедурные переменные как параметры подпрограмм. Подпрограммы обратного вызова (callback). Лямбда-выражения.
47. Типы процедурных переменных. Процедурные переменные-функции. Примеры: бинарные операции, функция суперпозиции функций. Предикаты, примеры их использования.
48. ~~Методы разработки программ сверху вниз и снизу вверх. Преимущества и недостатки каждого метода. Разработка через тестирование.~~
49. Модули. Цели введения модулей. Структура модуля. Разделы интерфейса и реализации. Необходимость разделения на интерфейс и реализацию. Принцип сокрытия информации. Упрощенная структура модуля. Документирующие комментарии.
50. Схема компиляции и перекompиляции программы с модулями.
51. Модуль как пространство имен. Алгоритм поиска имен в модулях. Системный модуль PABCSysSystem.
52. Библиотеки. Отличие библиотек от модулей. Подключение библиотек в PascalABC.NET. Представление о директивах компилятора.

Модуль 3. Массивы, последовательности, записи, строки

53. Перечислимый и диапазонный типы. Стандартные подпрограммы для работы с перечислимым типом. Использование перечислимого типа в `for` и `case`.
54. Динамические массивы. Выделение памяти под динамический массив. Хранение динамических массивов в памяти. Присваивание динамических массивов. Сравнение динамических массивов. Изменение длины динамического массива по ходу работы программы. Цикл `foreach` по динамическому массиву. Возвращение памяти, занимаемой динамическим массивом: представление о сборке мусора.
55. Передача динамических массивов как параметров подпрограмм на примере процедуры `Transform`. Динамические массивы как возвращаемые значения функций (на примере функции `Convert`). Проблема изменения динамического массива, переданного как параметр, внутри подпрограммы.
56. Представление о методах расширения.
57. Стандартные подпрограммы ввода, вывода, заполнения динамических массивов (заполнение случайными, заполнение по заданному правилу, заполнение рекуррентной последовательностью). Внутреннее устройство функции `ArgGen`, возвращающей рекуррентную последовательность.
58. Переменное число параметров подпрограмм.
59. Операции сложения динамических массивов и умножения динамического массива на число, операция `in`. Срезы динамических массивов.
60. Задачи на одномерные массивы: сумма и произведение элементов, инвертирование массива, поиск (реализации с `for` и `while`), поиск по условию, поиск с барьером, количество по условию, минимальный/максимальный элемент и их индексы, условный минимум/максимум, сдвиг влево/вправо, циклический сдвиг влево/вправо, фильтрация элементов по условию.
61. Решение задач на массивы с помощью срезов: инвертирование, элементы с нечетными индексами, сдвиги, циклические сдвиги, удаление и вставка элемента по заданному индексу.
62. Сортировки массивов. Сортировка выбором. Пузырьковая сортировка (обычная и с флагом). Сортировка вставками (алгоритм). Асимптотическая сложность алгоритмов сортировки.
63. Слияние двух упорядоченных массивов в один упорядоченный. Бинарный поиск в упорядоченном массиве. Его асимптотическая сложность.
64. Списки `List<T>`, их инициализация. Цикл по спискам. Индексация списков, срезы списков. Методы списков, отсутствующие у динамических массивов: вставка, удаление одного элемента, удаление элементов по условию. Распределение памяти под список: размер и ёмкость. Передача списков как параметров.
65. Примеры задач с использованием списков: вставка нескольких элементов по условию, алгоритм `Merge` слияния двух отсортированных списков в один отсортированный.
66. Двумерные динамические массивы (матрицы). Элементы строки, столбца. Создание, заполнение случайными, заполнение по заданному правилу, вывод.
67. Задачи на двумерные массивы: сумма элементов в j -том столбце; сумма элементов в каждом столбце; минимальный элемент в каждом столбце.
68. Поиск элемента во всём двумерном массиве (использование оператора `goto`).
- ~~69. Обнуление элементов выше/ниже главной/побочной диагонали в квадратной матрице.~~
70. Последовательности. Тип последовательности, цикл по последовательностям. Вывод последовательности. Стандартные генераторы последовательностей `Range` и `PartitionPoints`. Генераторы последовательностей с лямбда-выражениями: `SeqGen`.
71. Сравнение последовательностей и массивов. Последовательности и ленивость. Последовательности и занимаемая память.
72. Оператор `yield` и создание собственного генератора последовательности. Пример: генератор арифметической прогрессии.
73. Массивы как последовательности. Принцип подстановки Барбары Лисков.2 Преимущества принципа подстановки.
74. Бесконечные последовательности. Использование функции `Take` с бесконечными последовательностями.

75. Стандартные методы работы с последовательностями: фильтрация, проекция, сортировка. Цепочки методов (запросов) и комбинирование запросов в цепочке.
76. Другие операции над последовательностями: вычисление минимума, максимума, суммы, среднего, первого, последнего. Взятие n первых элементов, пропуск n первых элементов. Реверсирование последовательности, последовательность без повторяющихся элементов, превращение последовательности в массив.
77. Функция Range как замена цикла for. Примеры: суммы, суммы квадратов.
78. Методы последовательностей All и Any. Определение простоты числа.
79. Методы Aggregate и Zip для последовательностей и примеры их использования.
80. Записи. Поля записей. Хранение записей в памяти. Создающая функция для записей. Передача записей как параметров подпрограмм. Сравнение записей на равенство.
81. ~~Методы в записях. Описание методов внутри и вне записей. Обращение к полям записей из методов. Записи как пространства имен. Записи как модули. Объектно-ориентированный и процедурно-ориентированный подходы, их сравнение.~~
82. Классовые (статические) методы записей. Пример.
83. Классы. Конструктор класса. Методы класса. Хранение объектов класса в памяти. Отличие классов от записей: присваивание, сравнение, передача в подпрограммы. Классы как пространства имен. Обращение к полю класса из метода класса.
84. Автоклассы и их преимущества.
85. Возврат динамической памяти, занимаемой объектом класса, и сборка мусора.
86. Записи и классы как средство упаковки параметров.
87. Кортежи как легковесные записи. Обращение к элементам кортежей. Распаковка элементов кортежей в переменные. Операции с кортежами.
88. Массивы объектов. Сортировка массива записей по критерию сравнения, передаваемому в качестве параметра. Сортировка по ключу: стандартные методы OrderBy и ThenBy. Примеры использования в комбинации с проекцией и фильтрацией.
89. Группировка и метод GroupBy. Обход элементов, возвращаемых GroupBy, с помощью цикла foreach. Выполнение групповой операции над элементами группы.
90. Методы последовательностей с кортежами: ZipTuple, UnzipTuple, Partition, Interleave, Pairwise, Cartesian. Примеры использования.
91. Индексная сортировка массивов записей и необходимость её использования.
92. Множества, основные операции над множествами. Типы HashSet<T> и SortedSet<T>. Вывод множеств. Цикл по множеству. Отличие HashSet<T> от SortedSet<T>. Асимптотическая сложность операций для HashSet<T> и SortedSet<T>. Создающие функции HSet и SSet. Операции пересечения, объединения, разности множеств, операция принадлежности элемента множеству. Добавление элемента в множество, удаление элемента из множества.
93. Решето Эратосфена.
94. Символы. Коды символов. Стандартные подпрограммы работы с символами. Статические методы типа char. Константы #.
95. Строки. Хранение строк в памяти. Операции со строками. Передача строк в качестве параметров. Преобразование строка ↔ число.
96. Срезы строк, безопасные срезы строк. Примеры использования.
97. Основные методы типа string. Разбиение строки на слова и склейка строки из массива строк. Строки как последовательности символов.
98. ~~Некоторые задачи на строки: формирование строки 'ABCDEF...Z'; удаление всех вхождений подстроки в строку;~~ сортировка строки слов, разделенных пробелами; выделение из строки вещественных чисел и их суммирование.
99. Функция ReadLines. Текстовые файлы как последовательности строк. Считывание из базы данных, сохраненной в текстовый файл, в массив объектов. Пример запроса к базе данных. ~~Пример преобразования базы данных: процедура WriteLines.~~
100. Задача об определении количества слов в файле на каждую букву.