Лабораторная работа № 1. Линейные алгоритмы

***По теме рекомендуется прочитать дополнительно раздел «Лабораторная работа №1» из файла Osnovy\_programirovanniya\_na\_С#***

*Цель работы*: выработка умений составления, отладки и тестирования программ линейной структуры и оформления отчетов.

**Порядок выполнения лабораторных работ.**

* Прочитать теоретический материал
* Разобрать определения по теме
* Выполнить задания в соответствии с описанием
* Написать отчет по лабораторной работе
* Защитить отчет

**Определения по теме**

***Сущность понятия Тип данных.***

Данные хранятся в памяти компьютера в двоичном коде. Минимальная единица памяти – бит. Память компьютера – непрерывная область бит, каждый из которых может хранить значение 0 или 1. Группа из 8 бит образует ячейку для хранения данных, называемую байтом. Каждая ячейка памяти имеет физический адрес. Физический адрес – это числовой код, который центральный процессор (или другое устройство) может использовать для доступа к соответствующей ячейке памяти. Минимальная адресуемая единица памяти – БАЙТ.

Тип данных, хранящихся в байте или последовательности байтов, определяет способ интерпретации последовательности нулей и единиц, хранящихся в них. Допустим в байте памяти содержится набор из 8 единиц. Что представляет собой эта последовательность, как ее нужно интерпретировать (трактовать), какие действия с ней можно выполнять, какие сведения она в себе несет?

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Ответить на эти вопросы без дополнительной информации о характере этих данных невозможно. Этой дополнительной информацией и является Тип данных. Например, дополнительно известно (задан тип данных), что в данной ячейке записано однобайтное целое число. Тогда мы можем однозначно сказать, что это число 255 в двоичной системе счисления. Если же это целое однобайтное число со знаком, то получим число -1 (почему не -127?). Если эта последовательность обозначает символ в кодировке ASCII, то это символ. Русская заглавная буква «А» имеет код 128 (10000000) . Если мы будем интерпретировать значение байта памяти как логический тип данных, то нулевое значение байта будет соответствовать значению false, не нулевое – true.

Кроме того, что тип данных несет информацию о том, как нужно интерпретировать данные, он так же определяет, какое количество байт памяти занимает единица данных. Так, для хранения целочисленных значений может использоваться 1 (тип byte), 2 (int 16), 4 (int 32), 8 (int 64) байт памяти. Символьный тип данных может занимать 1 или 2 байта, в зависимости от типа кодировки ASCII или Unicode (c#). Логические данные всегда занимают в памяти 1 байт.

***Итак, что определяет тип данных?***

1. Объем памяти в байтах, занимаемый единицей данных.
2. Внутреннее представление данных (способ их интерпретации) при обработке.
3. Множество возможных значений (диапазон).
4. Набор допустимых действий (операций) над данными.

***Пример***. Данные типа int в c# занимают в памяти 4 байта. Интерпретируются как двоичной представление целого числа со знаком. Старший бит определяет знак числа (0 – положительное, 1 – отрицательное). Оставшиеся 31 бит – двоичное представление числа. С данными этого типа допустимы арифметические операции, операции отношений и ряд других.

***Переменная*** – поименованная, либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным. Данные, находящиеся в переменной (то есть по данному адресу памяти), называются значением этой переменной. В различные моменты времени переменная может иметь разные значения.

***Значение переменной по умолчанию –*** это значение, которое переменная получает в определенных ситуациях, когда значение не задано пользователем или не может быть получено из исходных данных, заданных пользователем. Например, при преобразовании строкового представления значения методом TryParse(), когда формат данных в строке не соответствует формату, к которому они должны быть преобразованы. Значение по умолчанию можно определить литералом default, например int a = default(int);Значения по умолчанию C# представлены в таблице 1.

Таблица 1 – значения по умолчанию *C#*

|  |  |
| --- | --- |
| ***Тип*** | ***Значение по умолчанию*** |
| Любой ссылочный тип | ***null*** |
| Любой [встроенный целочисленный тип](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/builtin-types/integral-numeric-types) | ***0 (Ноль)*** |
| Любой встроенный тип с плавающей запятой | ***0 (Ноль)*** |
| bool | ***false*** |
| char | ***'\0' (U+0000)*** |

***Тип переменной*** – тип данных, для доступа к которым переменная создана.

***Объявление переменной*** – указание компилятору (декларация) о намерении использовать переменную в дальнейшем коде. Объявление включает в себя указание идентификатора, и, возможно, типа и значения. Например, int a; double b=3.142. Во втором случае переменной присваивается значение, то есть переменная инициализируется. В c# можно объявить переменную без указания ее типа. Для этого используется оператор var. Тип переменной будет определен при ее инициализации по типу присваиваемого значения.

***Инициализация переменной*** – операция присваивания начального значения переменной, означающая начало использования переменной для хранения данных, приведение ее в состояние готовности к использованию. Объявление и инициализации могут быть разнесены в коде программы. Пример. В одной части кода объявляется int a, а затем ниже по тексту производится инициализация, например a=10.

***Определение переменной*** – объявление переменной с указанием типа данных. В процессе определения переменной выделяется память для ее размещения. В процессе объявления память не выделяется.

***Шутка.*** В бар входит ковбой и требует выпивку – **это объявление**, декларация намерений. Бармен спрашивает – чем будешь платить: деньгами, отдашь пистолет или отработаешь мытьем посуды? – Я заплачу, – отвечает ковбой, – у меня есть деньги. – Это **определение** типа данных.

Тогда плати 5 баксов и получай выпивку. Ковбой платит и веселье начинается – Это**инициализация**, присвоение начального значения, запуск процесса в работу.

**Лабораторная работа № 1.**

Выполнить индивидуальное задание в соответствии с выданным вариантом. Задание взять из файла «Лабораторная № 1.pdf».

**Оформите отчет по лабораторной работе**

Содержимое отчета описано в файле «МУ Содержимое отчета по лабораторной работе.docx».