**Задания для дистанционного обучения по информатике на**

**08.06.2020 г.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Преподаватель: Бондарева Е.А.**

**Почта:** elen.bondarevva@yandex.ru

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Здравствуйте, уважаемые обучающиеся, как Вы знаете на время сложной эпидемиологической обстановки, наш техникум перешел на дистанционное обучение, в связи с этим отправляю Вам задания для самостоятельного изучения.

Практическое занятие № 13 по теме:

**Разработка несложного алгоритма решения задачи.**

**Цель:** Разработка несложных алгоритмов.

Исключительно важно использовать язык блок-схем при разработке алгоритма решения задачи. Решение одной и той же задачи может быть реализовано с помощью различных алгоритмов, отличающихся друг от друга как по времени счета и объему вычислений, так и по своей сложности. Запись этих алгоритмов с помощью блок-схем позволяет сравнивать их, выбирать наилучший алгоритм, упрощать, находить и устранять ошибки.

Отказ от языка блок-схем при разработке алгоритма и разработка алгоритма сразу на языке программирования приводит к значительным потерям времени, к выбору неоптимального алгоритма. Поэтому необходимо изначально разработать алгоритм решения задачи на языке блок-схем, после чего алгоритм перевести на язык программирования.

При разработке алгоритма сложной задачи используется метод пошаговой детализации. На первом шаге продумывается общая структура алгоритма без детальной проработки отдельных его частей. Блоки, требующие детализации, обводятся пунктирной линией и на последующих шагах разработки алгоритма продумываются и детализируются.

В процессе разработки алгоритма решения задачи можно выделить следующие этапы:

* *Этап 1*. Математическое описание решения задачи.
* *Этап 2*. Определение входных и выходных данных.
* *Этап 3*. Разработка алгоритма решения задачи.

Базовые алгоритмические конструкции

В теории программирования доказано, что для записи любого, сколь угодно сложного алгоритма достаточно *трех базовых структур*:

* следование (линейный алгоритм);
* ветвление (разветвляющийся алгоритм);
* цикл-пока (циклический алгоритм).

Линейные алгоритмы

*Линейный алгоритм* образуется из последовательности действий, следующих одно за другим. Например, для определения площади прямоугольника необходимо сначала задать длину первой стороны, затем задать длину второй стороны, а уже затем по формуле вычислить его площадь.



Пример

**ЗАДАЧА. Разработать алгоритм вычисления гипотенузы прямоугольного треугольника по известным значениям длин его катетов a и b.**

На примере данной задачи рассмотрим все три этапа разработки алгоритма решения задачи:

***Этап 1. Математическое описание решения задачи.***

Математическим решением задачи является известная формула:

,

где с-длина гипотенузы, a, b – длины катетов.

*Этап 2. Определение входных и выходных данных.*

Входными данными являются значения катетов a и b. Выходными данными является длина гипотенузы – c.

*Этап 3. Разработка алгоритма решения задачи.* 

Разветвляющиеся алгоритмы

*Алгоритм ветвления*содержит условие, в зависимости от которого выполняется та или иная последовательность действий.



Пример.

ЗАДАЧА. Разработать алгоритм вычисления наибольшего числа из двух чисел x и y.

Этап 1. Математическое описание решения задачи.

Из курса математики известно, если x > y, то наибольшее число x, если x < y, то наибольшее число y, если x = y, то число x равно числу y.

Этап 2. Определение входных и выходных данных.

Входными данными являются значения чисел x и y. Выходным данными являются:

* наибольшее число
* любое из чисел, если числа равны

Для решения задачи нам необходимо знать значения x и y.

Этап 3. Разработка алгоритма решения задачи.

Введем обозначения: S – сумма последовательности, i – значение натурального числа. Начальное значение цикла i=1, конечное значение цикла i =100, шаг цикла 1.

В схеме алгоритма решения задачи цифрами указаны номера элементов алгоритма, которые соответствуют номерам шагов словесного описания алгоритма

В рассматриваемом алгоритме (рис.3) имеются три ветви решения задачи:

* первая: это элементы 1, 2, 3, 4, 8.
* вторая: это элементы 1, 2, 3, 5, 6, 8
* третья: это элементы 1, 2, 3, 5, 7, 8.

Выбор ветви определяется значениями x и y в элементах 3 и 5, которые являются условиями, определяющими порядок выполнения элементов алгоритма. Если условие (равенство), записанное внутри символа «решение», выполняется при введенных значениях x и y, то следующими выполняется элементы 4 и 8. Это следует из того, что они соединены линией с надписью «да» и направление (последовательность) вычислений обозначена стрелочкой.

Если условие в элементе 3 не выполняется, то следующим выполняется элемент 5. Он соединен с элементом 3 линией с надписью «нет». Если условие, записанное в элементе 5, выполняется, то выполняется элементы 6 и 8, в противном случае выполняются элементы 7 и 8.

Циклические алгоритмы

*Циклический алгоритм*– определяет повторение некоторой части действий (операций), пока не будет нарушено условие, выполнение которого проверяется в начале цикла. Совокупность операций, выполняемых многократно, называется телом цикла.



Алгоритмы, отдельные действия в которых многократно повторяются, называются *циклическими алгоритмами,* Совокупность действий, связанную с повторениями, называют *циклом*.

При разработке алгоритма циклической структуры выделяют следующие понятия:

* параметр цикла – величина, с изменением значения которой связано многократное выполнение цикла;
* начальное и конечное значения параметров цикла;
* шаг цикла – значение, на которое изменяется параметр цикла при каждом повторении.

Цикл организован по определенным правилам. Циклический алгоритм состоит из подготовки цикла, тела цикла и условия продолжения цикла.

 

В подготовку цикла входят действия, связанные с заданием исходных значений для параметров цикла:

* начальные значения цикла;
* конечные значения цикла;
* шаг цикла.

В тело цикла входят:

* многократно повторяющиеся действия для вычисления искомых величин;
* подготовка следующего значения параметра цикла;
* подготовка других значений, необходимых для повторного выполнения действий в теле цикла.

В условии продолжения цикла определяется допустимость выполнения повторяющихся действий. Если параметр цикла равен или превысил конечное значение цикла, то выполнение цикла должно быть прекращено.

 Пример

ЗАДАЧА. Разработать алгоритм вычисления суммы натуральных чисел от 1 до 100.

*Этап 1. Математическое описание решения задачи*.

Обозначим сумму натуральных чисел через S. Тогда формула вычисления суммы натуральных чисел от 1 до 100 может быть записана так:



где Xi – натуральное число X c номером i, который изменяется от 1 до n, n=100 – количество натуральных чисел.

Этап 2. Определение входных и выходных данных.

*Входными данными являются натуральные числа:* 1, 2, 3, 4, 5, …, 98, 99, 100.

*Выходные данные* – значение суммы членов последовательности натуральных чисел.

*Параметр цикла –*величина, определяющая количество повторений цикла. В нашем случае i – номер натурального числа.

*Подготовка цикла* заключается в задании начального и конечного значений параметра цикла.

* *начальное значение параметра цикла равно 1,*
* *конечное значение параметра цикла равно*n*,*
* *шаг цикла равен 1.*

Для корректного суммирования необходимо предварительно задать начальное значение суммы, равное 0.

*Тело цикла.*В теле цикла будет выполняться накопление значения суммы чисел, а также вычисляться следующее значение параметра цикла по формулам:

S=S+i;              I=I+1;

*Условие продолжения цикла:* цикл должен повторяться до тех пор, пока не будет добавлен последний член последовательности натуральных чисел, т.е. пока параметр цикла будет меньше или равен конечному значению параметра цикла.

Этап 3. Разработка алгоритма решения задачи.

Введем обозначения: S – сумма последовательности, i – значение натурального числа.

Начальное значение цикла i=1, конечное значение цикла i =100, шаг цикла 1.

