

2 курс

**ПЛАН – КОНСПЕКТ**  
проведения практического занятия № 2  
по дисциплине «Информатика»

**Раздел 1. «Автоматизированная обработка информации.»**

**Тема № 1.3: «Технология обработки информации»**

Подготовил: преподаватель  
В.Н. Борисов

Рязань 2024

**Практическое занятие № 2 «Ознакомление с этапами подготовки и обработки информации на вычислительной технике.»  
по Теме № 1.3. «Технология обработки информации».**

**Цель занятия:** изучить со студентами основные сведения об этапах подготовки и обработки информации на вычислительной технике, об основных структурах алгоритмов, практическое применение полученных знаний – выполнение задания, включающего в себя составление алгоритма решения задачи (блок-схемы) по условию задачи.

**Вид занятия:** классно-групповое, комбинированное (по проверке знаний, умений по пройденному материалу, по изучению и первичному закреплению нового материала, применению на практике полученных знаний).

**Методы проведения занятия:** доведение теоретических сведений, выполнение практического задания.

**Время проведения:** 2 ч (90 мин.)

**Основные вопросы:**

1. Ознакомление с этапами подготовки и обработки информации на вычислительной технике (ВТ).
2. Знакомство с основными структурами алгоритмов.
3. Применение на практике изученного материала (выполнение практического задания – составление алгоритма решения задачи (блок-схемы) по условию задачи.

**Литература:**

1. [1 учебник раздела «Дополнительной учебной литературы» рабочей программы изучения дисциплины]: Гаврилов, М. В. Информатика и информационные технологии : учебник для среднего профессионального образования / М. В. Гаврилов, В. А. Климов. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 355 с. — (Профессиональное образование). — ISBN 978-5-534-15930-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/viewer/informatika-i-informacionnye-tehnologii-510331#page/1>, глава 2.

**Примерный расчет времени:**

1. Вступительная часть – 20 мин.
2. Основная часть – 60 мин.
3. Заключительная часть – 10 мин.

### **Вступительная часть:**

Занятия начать с объявления темы занятия, основных рассматриваемых вопросов, времени изучения темы (нового материала), закрепления на практике полученных знаний, перечисления литературы.

### **Основная часть (доведение теоретических сведений):**

**Первый вопрос: Ознакомление с этапами подготовки и обработки информации на вычислительной технике (ВТ).**

Сведения по данному вопросу представлены в план-конспекте лекционного занятия по теме 1.3.

**Второй вопрос: Знакомство с основными структурами алгоритмов. Основные алгоритмические конструкции.**

#### **Понятие и свойства алгоритмов.**

*Алгоритм – заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.*

Это – не определение в математическом смысле слова, а, скорее, описание интуитивного понятия алгоритма, раскрывающее его сущность.

Понятие алгоритма является не только одним из главных понятий математики, но одним из главных понятий современной науки. Основные свойства алгоритмов следующие:

1. **Понятность** для исполнителя – исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.
2. **Дискретность** (прерывность, раздельность) – алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).
3. **Определенность** – каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.
4. **Результативность** (или конечность) состоит в том, что за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить

решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.

**5. Массовость** означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

### Способы описания алгоритма.

На практике наиболее распространены следующие формы представления алгоритмов:

- *словесная* (запись на естественном языке);
- *графическая* (изображения из графических символов);
- *псевдокоды* (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- *программная* (тексты на языках программирования).

**Словесный способ** записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке.

Словесный способ не имеет широкого распространения, так как такие описания:

- строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.

**Графический способ** представления алгоритмов является более компактным и наглядным по сравнению со словесным. *При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.*

Такое графическое представление называется схемой алгоритма или **блок–схемой**. В блок–схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде *блочного символа*. Блочные символы

соединяются *линиями*  
выполнения действий.

*переходов*, определяющими

очередность

### Таблица блочных символов.

В таблице приведены наиболее часто употребляемые блочные символы.

В таблице приведены наиболее часто употребляемые символы:

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме
Ввод–вывод		Ввод–вывод в общем виде
Пуск–останов		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать

Блок "**процесс**" применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок. Представление отдельных операций достаточно свободно.

Блок "**решение**" используется для обозначения переходов управления по условию. В каждом блоке "решение" должны быть указаны вопрос, условие или сравнение, которые он определяет.

Блок "**модификация**" используется для организации циклических конструкций. (Слово модификация означает видоизменение, преобразование). Внутри блока записывается параметр цикла, для которого указываются его начальное

значение, граничное условие и шаг изменения значения параметра для каждого повторения.

Блок "**предопределенный процесс**" используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам.

### Базовые алгоритмические структуры.

Алгоритмы можно представлять как некоторые структуры, состоящие из отдельных **базовых** (т.е. основных) **элементов**. Естественно, что при таком подходе к алгоритмам изучение основных принципов их конструирования должно начинаться с изучения этих базовых элементов. Для их описания будем использовать язык схем алгоритмов и простейший псевдокод.

Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур: следование, ветвление, цикл. Характерной особенностью базовых структур является наличие в них **одного входа и одного выхода**.

**1.Базовая структура "следование"**. Образуется последовательностью действий, следующих одно за другим:

Псевдокод	Язык блок–схем
действие 1 действие 2 . . . . . действие n	

**2.Базовая структура "ветвление"**. Обеспечивает в зависимости от результата проверки условия (**да** или **нет**) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к *общему выходу*, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

Структура *ветвление* существует в четырех основных вариантах:

- если–то;
- если–то–иначе;
- выбор;
- выбор–иначе.

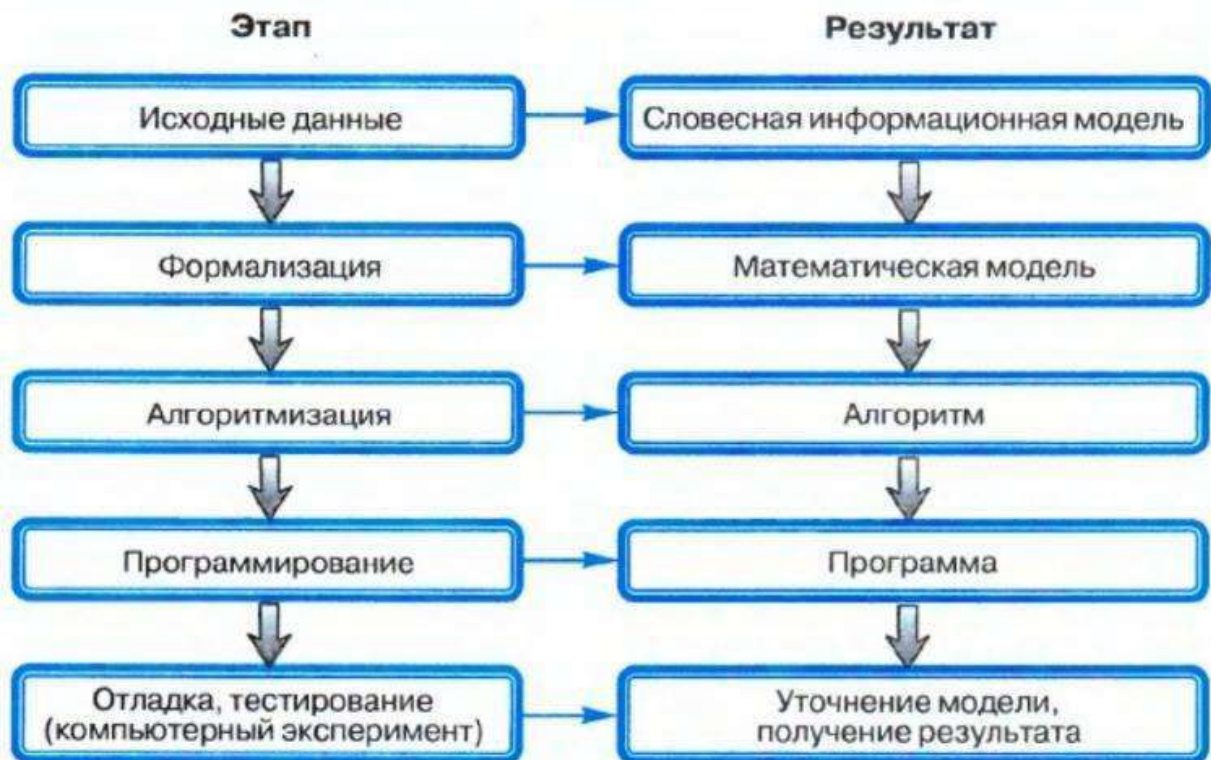
Псевдокод	Язык блок-схем
1. если-то	
если условие то действия все	
2. если-то-иначе	
если условие то действия 1 иначе действия 2 все	
3. выбор	
выбор при условии 1: действия 1 при условии 2: действия 2 . . . . . при условии N: действия N все	
4. выбор-иначе	
выбор при условии 1: действия 1 при условии 2: действия 2 . . . . . при условии N: действия N иначе действия N+1 все	

**3. Базовая структура "цикл" (повторение).** Обеспечивает многократное выполнение некоторой совокупности действий, которая называется телом цикла. Основные разновидности циклов представлены в таблице:

Псевдокод	Язык блок-схем
Цикл типа пока. Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова пока.	

<p>нц пока условие тело цикла (последовательность действий) кц</p>	
<p>Цикл типа для. Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне.</p>	
<p>нц для <math>i</math> от <math>i_1</math> до <math>i_2</math> тело цикла (последовательность действий) кц</p>	

## Этапы решения задачи на компьютере



### Расчет результатов выполнения алгоритма.

**Эффективность.** Рассматривая различные алгоритмы решения одной и той же задачи, полезно проанализировать, сколько вычислительных ресурсов они требуют (время работы, память), и выбрать наиболее эффективный.



**Под временем работы (running time) алгоритма** будем подразумевать число элементарных шагов, которые он выполняет. Положим, что одна строка псевдокода требует не более чем фиксированного числа операций (если только это не словесное описание каких-то сложных действий – типа «отсортировать все точки по  $x$ -координате»). Следует также различать *вызов (call)* процедуры (на который уходит фиксированное число операций) и её исполнение (*execution*), которое может быть долгим.

**Сложность алгоритма** – это величина, отражающая порядок величины требуемого ресурса (времени или дополнительной памяти) в зависимости от размерности задачи.

### **Анализ трудоёмкости алгоритмов.**

В качестве критерия оптимальности алгоритма выбирается трудоёмкость алгоритма, понимаемая как количество элементарных операций, которые необходимо выполнить для решения задачи с помощью данного алгоритма.

Функцией трудоёмкости называется отношение, связывающие входные данные алгоритма с количеством элементарных операций.

Одним из упрощенных видов анализа, используемых на практике, является асимптотический анализ алгоритмов. Целью асимптотического анализа является сравнение затрат времени и других ресурсов различными алгоритмами, предназначенными для решения одной и той же задачи, при больших объемах входных данных.

Используемая в асимптотическом анализе оценка функции трудоёмкости, называемая сложностью алгоритма, позволяет определить, как быстро растёт трудоёмкость алгоритма с увеличением объема данных. В асимптотическом анализе алгоритмов используются обозначения, принятые в математическом асимптотическом анализе

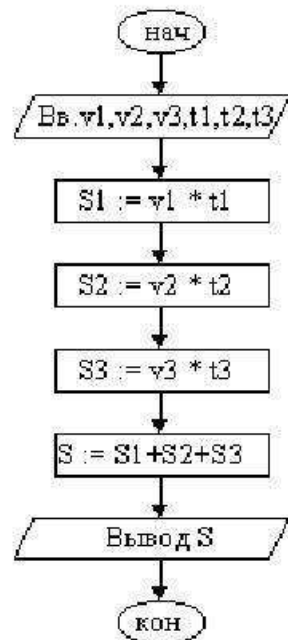
### **Заключительная часть.**

1. Закончить изложение материала.
2. Выдать задание на практическую работу.
3. Ответить на возникшие вопросы.
4. Принять защиту выполненных ранее практических работ.
5. Подвести итоги занятия.
6. Выдать задание на самоподготовку (домашнее задание).

**Третий вопрос: Применение на практике изученного материала (выполнение практического задания – составление алгоритма решения задачи (блок-схемы) по условию задачи).**

## Основные алгоритмические структуры

### Следование



Пешеход шел по пересеченной местности. Его скорость движения по равнине  $v_1$  км/ч, в гору —  $v_2$  км/ч и под гору —  $v_3$  км/ч. Время движения соответственно  $t_1$ ,  $t_2$  и  $t_3$  ч.

Какой путь прошел пешеход?

1. Ввести  $v_1, v_2, v_3, t_1, t_2, t_3$ .
2.  $S_1 := v_1 * t_1$ .
3.  $S_2 := v_2 * t_2$ .
4.  $S_3 := v_3 * t_3$ .
5.  $S := S_1 + S_2 + S_3$ .
6. Вывести значение  $S$ .
7. Конец.

## Задача на линейный алгоритм

**Вводятся объём и масса тела. Определить плотность материала этого тела.**

### Входные данные:

$V$  – объём тела (real);

$m$  – масса тела (real);

### Выходные данные:

$\rho$  – плотность тела (real).

### Паскаль-программа

```
Program z1;
```

```
uses crt;
```

```
Var V,m,p:real;
```

```
BEGIN
```

```
Clrscr;
```

```
Write('Введите значения объёма и массы тела: ');
```

```
ReadLn(v,m);
```

```
 $\rho := m/V$ ;
```

```
Write('Плотность тела
```

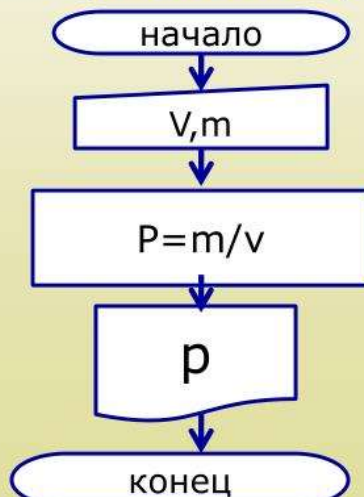
```
объёмом',V:7:1,' и массой' ,m:7:1,'
```

```
равна',p:8:2);
```

```
ReadLn;
```

```
END.
```

### БЛОК-СХЕМА

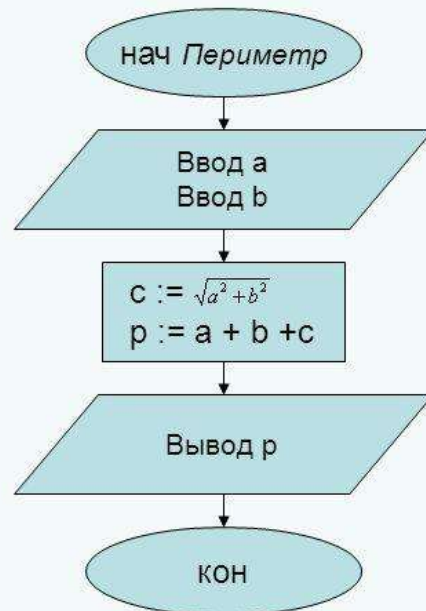


## Линейные алгоритмы

**Задача 2.** Написать программу нахождения периметра прямоугольного треугольника по известным катетам **a** и **b** (**a** и **b** – целые числа, вводимые с клавиатуры).

```

алг периметр
цел a, b
вещ c, p
нач
    ввод a
    ввод b
     $c := \sqrt{a^2 + b^2}$ 
     $p := a + b + c$ 
    вывод p
кон
  
```



MyShared

### Выполнение практического задания.

**Цель работы:** изучить основные сведения об этапах подготовки и обработки информации на вычислительной технике, об основных структурах алгоритмов, практическое применение полученных знаний – выполнение задания, включающего в себя составление алгоритма решения задачи (блок-схемы) по условию задачи.

**Задание: (исходные данные):** составить задачу линейного программирования (аналогично представленным выше), составить схему алгоритма ее решения (блок-схему). Указанные задачи должны быть различны у разных подгрупп учебной группы. Подготовить, защитить отчёт о выполнении работы.

### **Задание на самоподготовку (домашнее задание):**

1. Детально проработать, законспектировать материал занятия, размещенный в данном план-конспекте, в учебнике, указанном на с.2 текущего документа.
2. Подготовить отчёты о выполнении практической работы, подготовиться к защите данной работы.
3. Подготовиться к опросу по пройденному материалу.