

Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ ФГБОУ ВО «КемГУ»
Дата и время: 2024-02-21 00:00:00
471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Кемеровский государственный университет»
Новокузнецкий институт (филиал)

Факультет информатики, математики и экономики
Кафедра математики, физики и математического моделирования

Ю.С. Гаврилова

ВЫРАВНИВАЮЩИЙ КУРС ИНФОРМАТИКИ

*Методические рекомендации по изучению дисциплины
для обучающихся по направлению подготовки
02.03.03 Математическое обеспечение и администрирование информационных систем,
профиль «Программное и математическое обеспечение информационных технологий»*

Новокузнецк

2020

Гаврилова Ю.С.

Выравнивающий курс информатики: методические рекомендации по изучению дисциплины для студентов факультета информатики, математики и экономики, обучающихся по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем, профиль «Программное и математическое обеспечение информационных технологий»: / Ю.С. Гаврилова; Новокузнецкий ин-т (фил.) Кемеров. гос. ун-та. – Новокузнецк : НФИ КемГУ, 2020 – 23 с.

В работе представлены методические материалы по изучению дисциплины «Выравнивающий курс информатики»: основные теоретические сведения с примерами решения типовых задач; задания для контрольной работы, методические рекомендации по решению и оформлению, оценивание работ в балльно-рейтинговой системе; список основной и дополнительной литературы.

Методические рекомендации предназначены для наиболее рациональной организации внеаудиторной самостоятельной работы студентов при подготовке к выполнению контрольных работ и теста.

Рекомендовано на заседании
кафедры математики, физики и
математического моделирования
Протокол № 3 от 22 октября 2020г.
Заведующий кафедрой

 / Е.В. Решетникова

© Гаврилова Юлия Сергеевна
© Федеральное государственное бюджетное
образовательное учреждение высшего
образования «Кемеровский государственный
университет»,
Новокузнецкий институт (филиал), 2020
Текст представлен в авторской редакции

1. ПОСТРОЕНИЕ ПРОСТЫХ АЛГОРИТМОВ. ВЕТВЛЕНИЕ. ЦИКЛЫ

1.1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Алгоритм – это точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

Исполнитель алгоритма – это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

Алгоритм, записанный на некотором алгоритмическом, формальном языке, состоит из *заголовка алгоритма* (описания параметров) и *тела алгоритма* (последовательности команд исполнителя, преобразующих входные параметры в выходные).

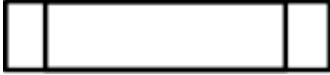
Наиболее распространенными формами представления алгоритмов являются: словесная, графическая, псевдокоды и программная.

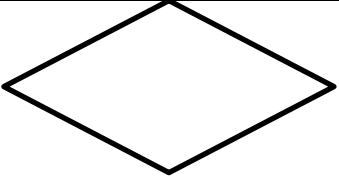
Словесная форма записи представляет собой описание последовательных этапов обработки данных на естественном языке (например, на русском).

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным. При графическом исполнении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного из действий. Такое графическое представление называется схемой алгоритма или блок-схемой. В блок-схеме каждому типу действий соответствует геометрическая фигура, называемая блочным символом. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий.

Для изображения блок-схем необходимо использовать общепринятый стандарт ГОСТ 19.701–90. Основные элементы, применяемые для построения блок-схем, представлены в таблице 1.

Таблица 1. Основные элементы блок-схем

Символ	Наименование	Описание
	Терминатор	Символ отображает выход во внешнюю среду и вход из внешней среды (<i>начало</i> или <i>конец</i> схемы программы, внешнее использование и источник или пункт назначения данных).
	Процесс	Символ отображает функцию обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации или к определению, по которому из нескольких направлений потока следует двигаться).
	Предопределенный процесс	Символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле).

	Решение	Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа. Соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути.
	Данные	Символ отображает данные, носитель данных не определен.
	Подготовка	Символ отображает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы).

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. Он занимает промежуточное место между естественным и формальным языками.

Программная форма представляет собой тексты программ, написанных на различных языках программирования.

Программа состоит из заголовка и блока. В заголовке указывается имя программы и список параметров, например

```
program n (input, output, x, y,...) ;
```

здесь n – имя программы; input – файл ввода; output – файл вывода; x, y – внешние файлы, используемые в программе. Заголовка может и не быть или он может быть без параметров. Блок программы состоит из шести разделов, следующих в строго определенном порядке:

1. раздел меток (label)
2. раздел констант (const)
3. раздел типов (type)
4. раздел переменных (var)
5. раздел процедур и функций
6. раздел действий (операторов).

Раздел действий должен присутствовать всегда, остальные разделы могут отсутствовать.

Каждый из первых четырех разделов начинается с соответствующего ключевого слова (label, const, type, var), которое записывается один раз в начале раздела и отделяется от последующей информации только пробелом, либо концом строки, либо комментарием.

Входные параметры – это данные, известные перед выполнением задачи, из ее условия.

Выходные параметры – это результат решения задачи.

Постоянной называется величина, значение которой не изменяется (поскольку такое изменение запрещено) в процессе исполнения алгоритма, а остается одним и тем же, указанным в тексте алгоритма. **Переменной** называется величина, значение которой меняется (в общем случае, может изменяться) в процессе исполнения алгоритма.

Переменные, которые не являются ни аргументом, ни результатом алгоритма, а используются только для обозначения вычисляемого промежуточного значения, называются **промежуточными**.

С понятием переменной связаны следующие характеристики (атрибуты):

- идентификатор – это ее обозначение и место в памяти;
- тип – множество допустимых значений и множество применимых операций к ней;
- значение – динамическая характеристика, может меняться многократно в ходе исполнения алгоритма. Во время выполнения программы в каждый конкретный момент величина имеет какое-то значение или не определена.

Различают переменные следующих простых типов: целые (Integer, Byte, ShortInt, Word, LongInt), вещественные (Real, Double, Single, Extended), логический (Boolean), символьный (Char), перечисляемый, диапазонный (интервальный).

Тип переменной определяет диапазон допустимых значений, принимаемых величинами этого типа; набор операций, допустимых над данной величиной; объем памяти, отводимой под эту переменную; способ представления величин в памяти компьютера (таблица 2).

Таблица 2. Основные типы переменных в языке Pascal

Идентификатор	Количество байт	Диапазон (множество) значений	Операции
Целые типы			
integer	2	-32768..32767	+, -, /, *, div, mod, <=, >= , < , > , <>
byte	1	0..255	
word	2	0..65535	
shortint	1	-128..127	
longint	4	-2147483648..2147483647	
Вещественные типы			
real	6	2,9x10 ⁻³⁹ - 1,7x10 ³⁸	+, -, /, *, <=, >= , < ,

single	4	$1,5 \times 10^{-45} - 3,4 \times 10^{38}$	>, <>, =
double	8	$5 \times 10^{-324} - 1,7 \times 10^{308}$	
extended	10	$3,4 \times 10^{-4932} - 1,1 \times 10^{4932}$	
Логический тип			
boolean	1	true, false	Not, And, Or, Xor, <=, >=, <, >, <>, =
Символьный тип			
char	1	все символы кода ASCII	+, >=, <=, =, <>, <, >

Перед тем, как начать использовать переменные в тексте программы, их необходимо описать (объявить). Все переменные описываются в разделе **var** программы:

```
var i, j, k : integer;
    LogPer : Boolean;
    A, B: Real;
    K: Char;
```

После того, как переменные были описаны, им можно присвоить какое-либо значение, например:

```
i:=1;
```

Значение для переменной можно получить от пользователя, используя процедуру ввода read(список переменных), например:

```
read(i);
```

Как только в программе встречается вызов процедуры Read, ЭВМ приостанавливает выполнение этой программы и ждет, пока пользователь введет с клавиатуры соответствующие значения, которые будут присваиваться переменным, перечисленным в списке ввода, в порядке перечисления. Значения вводимых данных одновременно отображаются на экране дисплея. После нажатия клавиши enter, когда все переменные примут свои значения из входного набора данных, определенного пользователем, выполнение программы продолжается с оператора, следующего за Read.

В списке ввода значения разделяются между собой пробелом (либо каждое вводится в отдельной строке). Присваивание значений из входного потока выполняется слева направо в соответствии с порядком следования переменных в процедуре Read. Процедура ReadLn похожа на Read. Разница лишь в том, что ReadLn реагирует на конец строки, и в случае его обнаружения происходит сразу переход к следующей строке.

В списке вывода может быть либо одно выражение, либо последовательность выражений, разделенных между собой запятыми.

Процедура Write осуществляет вывод значений выражений, приведенных в его списке, на текущую строку до ее заполнения. С помощью процедуры WriteLn реализуется вывод значений выражений, приведенных в его списке, на одну строку дисплея и переход к началу следующей строки.

Пример вывода данных:

```
write('Значение переменной i=' ,i);
```

Пример 1. Пешеход шел по пересеченной местности. Его скорость движения по полю 1 км/ч, по брусчатке - 2 км/ч и по асфальтированной дороге - 3 км/ч. Время движения соответственно 1, 2 и 3 ч. Составить алгоритм для определения пути, пройденного пешеходом, в словесной, графической и программной формах.

В *словесной* форме алгоритм решения данной задачи будет выглядеть следующим образом:

1. Запросить у пользователя ввод значений, характеризующих движение пешехода по разным видам поверхности (v_1 , v_2 , v_3 , t_1 , t_2 и t_3).
2. Вычислить путь, пройденный пешеходом по полю по формуле $S_1 = v_1 * t_1$.
3. Вычислить путь, пройденный пешеходом по брусчатке по формуле $S_2 = v_2 * t_2$.
4. Вычислить путь, пройденный пешеходом по асфальтированной дороге по формуле $S_2 = v_2 * t_2$.
5. Вычислить общий путь, пройденный пешеходом.

6. Вывести полученное значение на экран.

В программной форме на языке Pascal запись алгоритма будет выглядеть следующим образом:

```
program walk;  
var v1,v2,v3,t1,t2,t3,s1,s2,s3,s:real;  
begin  
    readln(v1);  
    readln(v2);  
    readln(v3);  
    readln(t1);  
    readln(t2);  
    readln(t3);  
    s1:=v1*t1;  
    s2:=v2*t2;  
    s3:=v3*t3;  
    s:=s1+s2+s3;  
    writeln(s);  
end.
```

Графическая форма представления алгоритма изображена на рисунке 1.

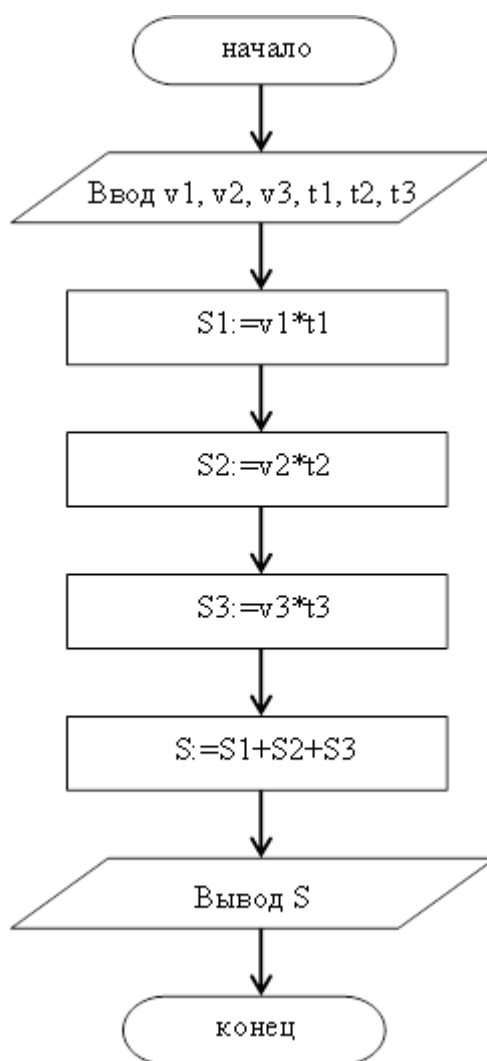


Рисунок 1. Блок-схема алгоритма для примера 1

Графическая форма представления алгоритма изображена на рисунке 1.

1.2 БАНК ЗАДАНИЙ К КОНТРОЛЬНОЙ РАБОТЕ № 1

1. Дана сторона квадрата, составить программу, которая определит его периметр.
2. Дан радиус окружности, составить программу, которая определит ее диаметр.
3. Считая, что Земля - идеальная сфера с радиусом $R=6350$ км, составить программу, которая определит расстояние до линии горизонта от точки с заданной высотой H над Землей.

4. Дана длина ребра куба, составить программу, которая определит объем куба и площадь его боковой поверхности.
5. Дан радиус окружности, составить программу, которая определит длину окружности и площадь круга.
6. Даны два целых числа, необходимо составить программу, которая определит их среднее арифметическое; их среднее геометрическое.
7. Известны объем и масса тела, составить программу, которая определит плотность материала этого тела.
8. Известны количество жителей в государстве и площадь его территории, составить программу, которая определит плотность населения в этом государстве.
9. Составить программу для решения линейного уравнения $ax+b=0$ ($a \neq 0$).
10. Даны катеты прямоугольного треугольника, необходимо составить программу, которая определит его гипотенузу.
11. Составить программу, которая определит площадь кольца по заданным внешнему и внутреннему радиусам.
12. Даны катеты прямоугольного треугольника, необходимо составить программу, которая определит его периметр.
13. Даны два числа. Составить программу, которая определит среднее арифметическое и среднее геометрическое их модулей.
14. Даны основания и высота равнобедренной трапеции, необходимо составить программу, которая определит ее периметр.
15. Даны стороны прямоугольника, необходимо составить программу, которая определит его периметр и длину диагонали.
16. Даны два числа, необходимо составить программу, которая определит их сумму, разность, произведение, а также частное от деления первого числа на второе.

17. Даны длины сторон прямоугольного параллелепипеда, необходимо составить программу, которая определит его объем и площадь боковой поверхности.

18. Известны координаты на плоскости двух точек. Составить программу вычисления расстояния между ними.

19. Даны основания и высота равнобедренной трапеции, необходимо составить программу, которая определит периметр трапеции.

20. Даны основания равнобедренной трапеции и угол при большем основании, необходимо составить программу, которая определит площадь трапеции.

21. Треугольник задан координатами своих вершин, необходимо составить программу, которая определит периметр и площадь треугольника.

22. Выпуклый четырехугольник задан координатами своих вершин, необходимо составить программу, которая определит площадь этого четырехугольника как сумму площадей треугольников.

23. Написать программу, которая запрашивает у пользователя два числа a и b , затем вычисляет путь, который проползла улитка со скоростью a км/ч за b секунд и выводит на экран результат с указанием единиц измерения.

24. Написать программу, которая запрашивает у пользователя три числа a , b и c , выводит на экран, какой процент от суммы данных чисел занимает каждое из них.

25. Известна стоимость 1 кг конфет, печенья и яблок, необходимо составить программу, которая определит стоимость всей покупки, если купили x кг конфет, y кг печенья и z кг яблок.

26. Известна стоимость монитора, системного блока, клавиатуры и мыши. Составить программу, которая определит сколько будут стоить 3 компьютера из этих элементов? N компьютеров?

27. Возраст Тани - X лет, а возраст Даши - Y лет. Составить программу, которая определит их средний возраст, а также на сколько

отличается возраст каждого ребенка от среднего значения.

28. Два автомобиля едут навстречу друг другу с постоянными скоростями V_1 и V_2 км/час. Составить программу, которая определит, через какое время автомобили встретятся, если расстояние между ними было S км.

29. Два автомобиля едут друг за другом с постоянными скоростями V_1 и V_2 км/час ($V_1 > V_2$), необходимо составить программу, которая определит какое расстояние будет между ними через 30 минут после того, как первый автомобиль опередил второй на S км.

30. Известно значение температуры по шкале Цельсия. Составить программу, которая определит соответствующее значение температуры по шкале: а) Фаренгейта; б) Кельвина. Для пересчета по шкале Фаренгейта необходимо исходное значение температуры умножить на 1,8 и к результату прибавить 32, а по шкале Кельвина абсолютное значение нуля соответствует 273,15 градуса по шкале Цельсия.

1.3 ОСОБЕННОСТИ ОЦЕНИВАНИЯ КОНТРОЛЬНОЙ РАБОТЫ В БАЛЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ

Контрольная работа по разделам «Построение простых алгоритмов» и «Циклы» является промежуточной формой контроля знаний студентов и представляет собой письменное выполнение определенных заданий. Она предназначена для проверки знаний студентов по учебной дисциплине «Выравнивающий курс информатики», а также служит для закрепления полученных знаний, умений и навыков. В контрольной работе студентам предлагаются задачи, сформулированные на основании материала, проработанного на практических занятиях или самостоятельно изученного студентами.

Перед тем как приступить к выполнению контрольной работы, студентам следует ознакомиться с теоретическим материалом и разобраться с разобранными в нем типовыми задачами.

Варианты контрольной работы состоят из 10 заданий и формируются случайным образом из представленных в разделе «Банк заданий к контрольной работе № 1».

Система оценивания заданий контрольной работы представлена в таблице 1.2.

Таблица 1.2. Оценивание контрольной работы № 1 в БРС

Критерии оценивания задания	Количество баллов
Логично и последовательно выполнены все шаги решения, рассуждения имеют четкое обоснование, получен верный ответ.	1
Ход решения задания верный, но аргументация неполная или из-за вычислительной ошибки получен неверный ответ.	0,5
Нарушена логическая цепочка рассуждений, решение неполное, ответ неверный.	0
Максимальное количество баллов за контрольную работу № 1	10

Оформление контрольной работы должно соответствовать Правилам оформления учебных работ студентов¹, принятым в НФИ КемГУ.

¹ См. раздел «Литература для оформления учебных работ» в списке рекомендуемой литературы

2 ОБРАЗЕЦ ТЕСТОВЫХ ЗАДАНИЙ ДЛЯ ИТОГОВОГО ТЕСТА ПО ДИСЦИПЛИНЕ «ВЫРАВНИВАЮЩИЙ КУРС ИНФОРМАТИКИ»

1. *Что будет выведено на экран в результате выполнения программы, если пользователь ввел число 34567890?*

```
program try_first;
var a,b,c,i:integer;
begin
  readln(a);
  writeln;
  b:=0;
  c:=1;
  while a>0 do
  begin
    if (a mod 2=0)
      then b:=b+(a mod 10)
      else c:=c*(a mod 10);
    a:=a div 10;
  end;
  writeln('b=',b,'; c=',c);
end.
```

2. *Какие из этих выражений верны?*

- а) процедура всегда возвращает хотя бы одно значение;
- б) объявление процедуры происходит до начала основного текста программы;
- в) для процедуры обязательно должны объявляться переменные;
- г) в процедуре могут объявляться константы.

3. *В программе задана функция randommatrix:*

```
function randommatrix(m,o:integer):matrix;
begin
  for i:=1 to m do
  for j:=1 to o do
  result[j][i]:=random(15);
  end;
end;
```

Можно ли ее вызвать с помощью одной строки randommatrix(2,8);?

- а) можно;
- б) нет, так как нет операции присваивания, обязательной для вызова функции;

- в) нет, так как при вызове функции передаются не все необходимые параметры;
- г) нет, так как при вызове функции необходимо указать ключевое слово function.

4. Что делает данная программа?

```
program try_first;
const n=10;
var a,b:array[1..n]of integer;
    i,j,k,max:integer;
begin
    randomize;
    for i:=1 to n do
    begin
        a[i]:=random(10);
        write(a[i], ' ');
    end;
    writeln;
    for i:=1 to 5 do
    begin
        k:=a[i];
        a[i]:=a[n-i+1];
        a[n-i+1]:=k;
    end;
    for i:=1 to n do
        write(a[i], ' ');
    end.
```

- а) сортирует массив по убыванию;
- б) сортирует массив по возрастанию;
- в) меняет местами некоторые элементы массива;
- г) меняет местами элементы с 1 по 5.

4. Как в программе можно задать константу с именем «var»?

- а) нужно создать специальную процедуру;
- б) нужно создать в разделе var переменную типа const;
- в) в разделе const;
- г) нельзя.

5. Для чего применяется оператор mod?

6. В программе задаются типы данных с помощью конструкции:

```
type vector=array[1..n]of integer;
matrix=array[1..n]of vector;
```

Можно ли с помощью этих типов задать матрицу размерностью 5x2??

- а) нельзя, потому что в типах данных задаются только одномерные массивы;
- б) нельзя, потому что количество строк матрицы не равно количеству столбцов;
- в) нельзя, потому что элементы матрицы должны быть типа real;
- г) можно.

3. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная учебная литература

1. Комарова, Е.С. Практикум по программированию на языке Паскаль: учебное пособие. Часть 1 / Е.С. Комарова - Москва - Берлин, 2015. – 85 с. - ISBN 978-5-4475-4914-5. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=426942. – (дата обращения: 22.03.2020). – Текст : электронный.

2. Комарова, Е.С. Практикум по программированию на языке Паскаль : учебно-методическое пособие. Часть 2 / Е.С. Комарова. - Москва-Берлин : Директ-медиа, 2015. – 123 с. - ISBN 978-5-4475-4915-2. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=426943. – (дата обращения: 22.03.2020). – Текст : электронный.

Дополнительная учебная литература

1. Прохорова, О.В. Информатика [Электронный ресурс] : учебник / О.В. Прохорова. – Электрон. текстовые дан. – Самара: СГАСУ, 2013. – 106 с. – ISBN 978-5-9585-0539-5. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=256147. – (дата обращения: 22.03.2020). – Текст : электронный.

Литература для оформления учебных работ

1. Правила оформления учебных работ студентов : учебно-методическое пособие / И.А. Жибинова, А.Е. Аракелян, О.В. Соколова, Ю.Н. Соина-Кутищева. – Новокузнецк : НФИ КемГУ, 2018. – 124 с. – Текст : непосредственный.

2. ГОСТ 19.701-90 (ИСО 5807-85) Единая система программной документации (ЕСПД). Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения : межгосударственный стандарт : издание официальное : введен впервые : дата введения 1992-01-01/ Москва Стандартиформ, 2010 – 158 с. – Текст: непосредственный.

4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ОБУЧАЮЩИМСЯ ПО ВЫПОЛНЕНИЮ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

4.1. Методические указания обучающимся по подготовке к промежуточному контролю

Готовиться к зачету/экзамену необходимо последовательно, с учетом контрольных вопросов, разработанных ведущим преподавателем кафедры. Сначала следует определить место каждого контрольного вопроса в соответствующем разделе темы учебной программы, а затем внимательно прочитать и осмыслить рекомендованные научные учебные работы, соответствующие разделы рекомендованных учебников. При этом полезно делать хотя бы самые краткие выписки и заметки. Деятельность над темой можно считать завершенной, если вы сможете ответить на все контрольные вопросы и дать определение понятий по изучаемой теме. Для обеспечения полноты ответа на контрольные вопросы и лучшего запоминания теоретического материала рекомендуется составлять план ответа на контрольный вопрос. Это позволит сэкономить время для подготовки непосредственно перед зачетом за счет обращения не к учебной литературе, а к своим записям. При подготовке необходимо выявлять наиболее сложные, дискуссионные вопросы, с тем, чтобы обсудить их с преподавателем на обзорных лекциях и консультациях. Нельзя ограничивать подготовку к зачету простым повторением изученного материала. Необходимо углубить и расширить ранее приобретенные знания за счет новых идей и положений. Результат по сдаче зачета/экзамена объявляется студентам, вносится в зачетную/экзаменационную ведомость. При получении отметки «не зачтено»/ «неудовлетворительно» повторная сдача осуществляется в другие дни, установленные деканатом.

4.2. Методические указания обучающимся по работе с учебной литературой

Работу с учебной литературой целесообразно начать с изучения общих работ по теме, а также учебников и учебных пособий. Далее рекомендуется перейти к анализу монографий и статей, рассматривающих отдельные аспекты проблем, изучаемых в рамках курса, а также официальных материалов и неопубликованных документов (научно-исследовательские работы, диссертации), в которых могут содержаться основные вопросы изучаемой проблемы.

Работу с источниками надо начинать с ознакомительного чтения, т.е. просмотреть текст, выделяя карандашом его структурные единицы. При

ознакомительном чтении закладками отмечаются те страницы, которые требуют более внимательного изучения.

В зависимости от результатов ознакомительного чтения выбирается дальнейший способ работы с источником. Если для разрешения поставленной задачи требуется изучение некоторых фрагментов текста, то используется метод выборочного чтения. Если в книге нет подробного оглавления, следует обратить внимание ученика на предметные и именные указатели.

Избранные фрагменты или весь текст (если он целиком имеет отношение к теме) требуют вдумчивого, неторопливого чтения с «мысленной проработкой» материала. Такое чтение предполагает выделение: 1) главного в тексте; 2) основных аргументов; 3) выводов. Особое внимание следует обратить на то, вытекает ли тезис из аргументов или нет.

Необходимо также проанализировать, какие из утверждений автора носят проблематичный, гипотетический характер.

Умение работать с текстом приходит постепенно. Наилучший способ научиться выделять главное в тексте, определять проблематичный характер утверждений, давать оценку авторской позиции – это сравнительное чтение, в ходе которого происходит знакомство с различными мнениями по одному и тому же вопросу, сравнивается весомость и доказательность аргументов сторон и делается вывод о наибольшей убедительности той или иной позиции.

Если в учебной литературе встречаются разные точки зрения по тому или иному вопросу из-за сложности прошедших событий и правовых явлений, нельзя их отвергать, не разобравшись. При наличии расхождений между авторами необходимо найти рациональное зерно у каждого из них, что позволит глубже усвоить предмет изучения и более критично оценивать изучаемые вопросы. Знакомясь с особыми позициями авторов, нужно определять их схожие суждения, аргументы, выводы, а затем сравнивать их между собой и применять из них ту, которая более убедительна.

Следующим этапом работы с научной и учебной литературой является создание записей. Форма записей может быть разнообразной: простой или развернутый план, тезисы, цитаты, конспект и др.

План – структура письменной работы, определяющая последовательность изложения материала. Он является наиболее краткой и потому самой доступной и распространенной формой записей содержания исходного источника информации; это перечень основных вопросов, рассматриваемых в источнике. План может быть простым и развернутым. Их отличие состоит в степени детализации содержания и, соответственно, в объеме. Преимущество плана состоит в том, что план позволяет наилучшим образом уяснить логику мысли автора, упрощает понимание главных моментов произведения. Кроме того, он позволяет быстро и глубоко проникнуть в сущность построения произведения и, следовательно, гораздо легче ориентироваться в его содержании и быстрее обычного вспомнить

прочитанное. С помощью плана гораздо удобнее отыскивать в источнике нужные места, факты, цитаты и т.д.

Выписки представляют собой небольшие фрагменты текста (неполные и полные предложения, отделы абзацы, а также дословные и близкие к дословным записи об излагаемых в нем фактах), содержащие в себе квинтэссенцию содержания прочитанного. Выписки представляют собой более сложную форму записи содержания исходного источника информации. Выписки позволяют в концентрированной форме и с максимальной точностью воспроизвести наиболее важные мысли автора, статистические и даталогические сведения. В отдельных случаях – когда это оправдано с точки зрения продолжения работы над текстом – вполне допустимо заменять цитирование изложением, близким дословному.

Тезисы – сжатое изложение содержания изученного материала в утвердительной (реже опровергающей) форме. Отличие тезисов от обычных выписок состоит в том, что тезисам присуща значительно более высокая степень концентрации материала. В тезисах отмечается преобладание выводов над общими рассуждениями. Записываются они близко к оригинальному тексту, т.е. без использования прямого цитирования.

Аннотация – краткое изложение основного содержания исходного источника информации, дающее о нем обобщенное представление. К написанию аннотаций прибегают в тех случаях, когда подлинная ценность и пригодность исходного источника информации исполнителю письменной работы окончательно неясна, но в то же время о нем необходимо оставить краткую запись с обобщающей характеристикой.

Резюме – краткая оценка изученного содержания исходного источника информации, полученная, прежде всего, на основе содержащихся в нем выводов. Резюме весьма сходно по своей сути с аннотацией. Однако, в отличие от последней, текст резюме концентрирует в себе данные не из основного содержания исходного источника информации, а из его заключительной части, прежде всего выводов. Но резюме излагается своими словами – выдержки из оригинального текста в нем практически не встречаются.

Конспект представляет собой сложную запись содержания исходного текста, включающая в себя заимствования (цитаты) наиболее примечательных мест в сочетании с планом источника, а также сжатый анализ записанного материала и выводы по нему. При выполнении конспекта требуется внимательно прочитать текст, уточнить в справочной литературе непонятные слова и вынести справочные данные на поля конспекта. Нужно выделить главное, составить план. Затем следует кратко сформулировать основные положения текста, отметить аргументацию автора. Записи материала следует проводить, четко следуя пунктам плана и выражая мысль своими словами. Цитаты должны быть записаны грамотно, учитывать лаконичность, значимость мысли. В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении

конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Конспекты пишутся на одной стороне листа, с полями и достаточным для исправления и ремарок межстрочным расстоянием (эти правила соблюдаются для удобства редактирования). Если в конспектах приводятся цитаты, то непременно должно быть дано указание на источник (автор, название, выходные данные, № страницы). Впоследствии эта информация может быть использована при написании текста реферата или другого задания.

Таким образом, при работе с научными источниками и учебной литературой важно уметь:

- сопоставлять, сравнивать, классифицировать, группировать, систематизировать информацию в соответствии с определенной учебной задачей;
- обобщать полученную информацию, оценивать прослушанное и прочитанное;
- фиксировать основное содержание сообщений; формулировать, устно и письменно, основную идею сообщения; составлять план, формулировать тезисы;
- готовить и презентовать развернутые сообщения типа доклада;
- работать в разных режимах (индивидуально, в паре, в группе), взаимодействуя друг с другом;
- пользоваться реферативными и справочными материалами;
- контролировать свои действия и действия своих товарищей, объективно оценивать свои действия;
- обращаться за помощью, дополнительными разъяснениями к преподавателю, другим студентам;
- пользоваться различными словарями, опорами в тексте (ключевые слова, структура текста, предваряющая информация и др.);
- использовать при говорении и письме синонимичные средства, слова-описания общих понятий, разъяснения, примеры, толкования и др.;
- повторять или перефразировать реплику собеседника в подтверждении понимания его высказывания или вопроса;
- обратиться за помощью к собеседнику (уточнить вопрос, переспросить и др.)