Государственное бюджетное профессиональное образовательное учреждение «Дзержинский педагогический колледж»

**Методические разработки практических занятий**

**по ОП.04 Основы алгоритмизации и программирования\_**

**Дзержинск – 2017-2021**

|  |  |
| --- | --- |
| Одобрено на заседании ПЦК преподавателей спец. ИнформатикаПротокол \_\_\_\_\_\_\_от\_\_\_\_\_\_\_\_Председатель ПЦК \_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_/ Составитель: Руденко Н.А., Пучкина Т.С. | Методические разработки практических занятий составлены в соответствии с требованиями федеральных государственных образовательных стандартов среднего профессионального образования по спец. 09.02.07. Информационные системы и программирование  |

Методические разработки практических занятий по учебной дисциплине ОП.04 Основы алгоритмизации и программирования специальность 09.02.07. Информационные системы в программировании содержат план практического занятия с указанием вида практической работы в соответствии с рабочей программой учебной дисциплины. В них имеется необходимый теоретический материал о способах подготовки наборов тестовых данных для функционального и структурного тестирования, приводятся примеры применения перечисленных методов и критериев тестирования, включены задачи для самостоятельной работы и вопросы для самоконтроля.

**Практическая работа по теме «Языки программирования. Основные типы алгоритмических конструкций»**

**Цель работы**: формирование знаний и умений по работе с блок-схемами алгоритмов, разработка ветвящихся алгоритмов.

**Краткие теоретические сведения**

**Основные этапы решения задач на компьютере**

*Процесс решения задач на компьютере* *-* это совместная деятельность человека и ЭВМ. Этот процесс можно представить в виде нескольких последовательных этапов. На долю человека приходятся этапы, связанные с творческой деятельностью — постановкой, алгоритмизацией, программированием задач и анализом результатов, а на долю компьютера — этапы обработки информации в соответствии с разработанным алгоритмом.

Рассмотрим эти этапы на следующем примере: пусть требуется вычислить сумму двух целых чисел и вывести на экран результат.

*Первый этап - постановка задачи.* На этом этапе участвует человек, хорошо представляющий предметную область задачи. Он должен четко определить цель задачи, дать словесное описание содержания задачи и предложить общий подход к ее решению. Для задачи вычисления суммы двух целых чисел человек, знающий, как складываются числа, может описать задачу следующим образом: ввести два целых числа, сложить их и вывести сумму в качестве результата решения задачи.

*Второй этап - математическое или информационное моделирование. Цель этого этапа - создать такую математическую модель решаемой задачи, которая может быть реализована в компьютере. Существует целый ряд задач, где математическая постановка сводится к простому перечислению формул и логических условий. Для вышеописанной задачи данный этап сведется к следующему: введенные в компьютер числа запомним в памяти под именами А и В, затем вычислим значение суммы этих чисел по формуле А+В, и результат запомним в памяти под именем Summa.*

*Третий этап - алгоритмизация задачи. На основе математического описания необходимо разработать алгоритм решения.*

*Алгоритмом называется точное предписание, определяющее последовательность действий исполнителя, направленных на решение поставленной задачи. В роли исполнителей алгоритмов могут выступать люди, роботы, компьютеры.*

*Используются различные способы записи алгоритмов. Широко распространен словесный способ записи: это записи рецептов приготовления различных блюд в кулинарной книге, инструкции по использованию технических устройств, правила правописания и многие другие. Наглядно представляется алгоритм языком блок-схем.*

*Четвертый этап — программирование. Программой называется план действий, подлежащих выполнению некоторым исполнителем, в качестве которого может выступать компьютер. Составление программы обеспечивает возможность выполнения алгоритма и соответственно поставленной задачи исполнителем-компьютером. Во многих задачах при программировании на алгоритмическом языке часто пользуются заменой блока алгоритма на один или несколько операторов, введением новых блоков, заменой одних блоков другими.*

*Пятый этап - ввод программы и исходных данных в ЭВМ. Программа и исходные данные вводятся в ЭВМ с клавиатуры с помощью редактора текстов, и для постоянного хранения осуществляется их запись на гибкий или жесткий магнитный диск.*

*Шестой этап - тестирование и отладка программы. На этом этапе происходят исполнение алгоритма с помощью ЭВМ, поиск и исключение ошибок. При этом программисту приходится выполнять работу по проверке работы программы, поиску и исключению ошибок, и поэтому для сложных программ этот этап часто требует гораздо больше времени и сил, чем написание первоначального текста программы. Отладка программы - сложный и нестандартный процесс. Исходный план отладки заключается в том, чтобы оттестировать программу на контрольных примерах.*

*Контрольные примеры стремятся выбрать так, чтобы при работе с ними программа прошла все основные пути блок-схемы алгоритма, поскольку на каждом из путей могут быть свои ошибки, а детализация плана зависит от того, как поведет себя программа на этих примерах: на одном она может зациклиться (т. е. бесконечно повторять одно и то же действие); на другом - дать явно неверный или бессмысленный результат и т. д. Сложные программы отлаживают отдельными фрагментами.*

*Седьмой этап - исполнение отлаженной программы и анализ результатов. На этом этапе программист запускает программу и задает исходные данные, требуемые по условию задачи.*

*Полученные в результате решения выходные данные анализируются постановщиком задачи, и на основании этого анализа вырабатываются соответствующие решения, рекомендации, выводы. Например, если при решении задачи на компьютере результат сложения двух чисел 2 и 3 будет 4, то следует сделать вывод о том, что надо изменить алгоритм и программу.*

*Возможно, что по итогам анализа результатов потребуются пересмотр самого подхода к решению задачи и возврат к первому этапу для повторного выполнения всех этапов с учетом приобретенного опыта. Таким образом, в процессе создания программы некоторые этапы будут повторяться до тех пор, пока мы получим алгоритм и программу, удовлетворяющие показанным выше свойствам.*

*Алгоритмизация.*

*Блок схемы как графическое представление алгоритмов.*

*Основные блоки, используемые в блок-схемах алгоритмов.*

*Основным в пр*оцессе программирования является разработка алгоритма. Это один из наиболее сложных этапов решения задачи с использованием ЭВМ. В начале обучения программированию целесообразно не привязываться сразу к какому-либо языку, разрабатывать алгоритмы без записи на ЯВУ, а, например, с помощью блок-схем. После такой "чистой" алгоритмизации проще перейти к записи того же алгоритма на определённом языке программирования. Далее продемонстрирован именно такой подход.

Основными алгоритмическими структурами (ОАС) являются следование, ветвление и цикл. В более сложных случаях используются вложения ОАС.

Ниже приведены графические обозначения (обозначения на блок-схемах) ОАС.
Структура “следование”


Полное ветвление


Неполное ветвление


Цикл с предусловие (цикл ПОКА)


Цикл с постусловием (цикл ДО)


Цикл с параметром

На схемах СЕРИЯ обозначает один или несколько любых операторов; УСЛОВИЕ есть логическое выражение (ЛВ) (если его значение ИСТИНА, переход происходит по ветви ДА, иначе — по НЕТ). На схеме цикла с параметром использованы обозначения: ПЦ — параметр цикла, НЗ — начальное значение параметра цикла, КЗ — конечное значение параметра цикла, Ш — шаг изменения параметра цикла.

Начало и конец алгоритма на блок-схемах обозначают овалом, вводимые и выводимые переменные записываются в параллелограмме.

В примерах мы будем использовать запись алгоритмов с помощью блок-схем и словесное описание.

**Примеры:**

**1. Линейные алгоритмы**

Простейшие задачи имеют линейный алгоритм решения. Это означает, что он не содержит проверок условий и повторений.

Пример 1. Пешеход шел по пересеченной местности. Его скорость движения по равнине v1 км/ч, в гору — v2 км/ч и под гору — v3 км/ч. Время движения соответственно t1, t2 и t3 ч. Какой путь прошел пешеход?

1. Ввести v1, v2, v3, t1, t2, t3.

2. S1 := v1 \* t1.

3. S2 := v2 \* t2.

4. S3 := v3 \* t3.

5. S := S1 + S2 + S3.

6. Вывести значение S.

7. Конец.

Для проверки работоспособности алгоритма необходимо задать значения входных переменных, вычислить конечный результат по алгоритму и сравнить с результатом ручного счета.

Пример 2. Дано натуральное трехзначное число n, в записи которого нет нулей. Составить алгоритм, который возвращает значение ИСТИНА, если верно утверждение: "число n кратно каждой своей цифре", и ЛОЖЬ — в противном случае.

1. Ввести число n

2. A := n mod 10

3. B := n div 100

4. C := n div 10 mod 10

5. L := (n mod A=0) and (n mod B=0) and (n mod C=0)

6. Вывод L

7. Конец

На приведенной выше схеме DIV и MOD соответственно операции деления нацело и получения остатка от целочисленного деления.

**2. Ветвление**

Достаточно часто то или иное действие должно быть выполнено в зависимости от значения логического выражения, выступающего в качестве условия. В таких случаях используется ветвление (развилка).

Пример 1. Вычислить значение функции



**3. Цикл**

Алгоритмическая конструкция используется для решения задач, в которых для достижения цели необходимо многократное повторение последовательности действий.

Пример 1

При решении задачи находится очередной член последовательно и, если он больше, добавляется к сумме.



1. Ввести e

2. S := 0

3. A := 1/4

4. n := 3

5. Сравнить А с e. Если A>=e, переход к п. 10

6. S := S + A

7. A := (n-1)/(n\*n)

8. n := n + 1

9. Переход к п. 5

10. Вывод S

11. Конец

В рассмотренных выше примерах количество повторений заранее неизвестно. В первом оно зависит от количества цифр в записи натурального числа, во втором — от числа e.

В тех же случая, когда количество шагов известно из условия задачи, проще и предпочтительней использовать цикл с параметром.

Пример 2. Найти произведение первых k натуральных чисел, кратных трём.

При составлении алгоритма учтем, что первое натуральное число, кратное 3, есть тройка, а все последующие больше предыдущего на 3.

1. Ввод k

2. P := 1 {здесь накапливаем произведение}

3. T := 0 {здесь будут числа, кратные 3}

4. I := 1

5. Если I > k, переход к п. 10

6. T := T + 3

7. P := P \* T

8. I := I + 1

9. Перейти к п. 5

10. Вывод P

11. Конец

Изучение программирования разумно начинать собственно с разработки алгоритмов, не акцентируя первоначально внимания на записи алгоритма на том или ином языке программирования.

Контрольные вопросы

1. Основные этапы решения задач на компьютере.
2. Свойства алгоритма. Типы вычислительных процессов.
3. Блок схемы. Понятие и правила построения.
4. Примеры построения блок-схем алгоритмов.

**Задания:**

1. Разработать блок-схему алгоритма, которая вычисляет частное двух чисел. Должна проверяться правильность исходных данных и, если они неверные, выдать сообщение об ошибке.

2. Разработать блок-схему алгоритма вычисления площади кольца по радиусам кольца и отверстия. Должна проверяться правильность исходных данных и, если они неверные, выдать сообщение об ошибке.

3. Разработать блок-схему алгоритма, переводящую время из минут и секунд в секунды. Должна проверяться правильность введенных данных и в случае, если данные неверные, должно выводиться соответствующее сообщение.

4. Разработать блок-схему алгоритма, которая проверяет, является ли год високосным.

5. Разработать блок-схему алгоритма решения квадратного уравнения. Должна проверяться правильность исходных данных. В случае, если коэффициент при второй степени неизвестного равен нулю, вывести соответствующее сообщение.

6. Разработать блок-схему алгоритма вычисления стоимости покупки с учетом скидки. Скидка в 10% предоставляется, если сумма покупки больше 1000 руб.

7. Разработать блок-схему алгоритма вычисления стоимости покупки с учетом скидки. Скидка в 3% предоставляется, если сумма покупки 500 руб, в 5% — если сумма больше 1000 руб.

8. Разработать блок-схему алгоритма, которая сравнивает два введенных числа. Блок-схема должна определить, какое число больше, или, если числа равны, вывести соответствующее сообщение.

9. Разработать блок-схему алгоритма, которая проверяет, является ли введенное целое число четным.

10. Разработать блок-схему алгоритма, которая проверяет, делится ли на три введенное целое число.

11. Разработать блок-схему алгоритма вычисления стоимости разговора по телефону с учетом 20% скидки, предоставляемой по субботам и воскресеньям.

Вводимыми данными являются длительность разговора и номер дня недели.

12. Разработать блок-схему алгоритма, которая вычисляет оптимальный вес человека, сравнивает его с реальным и выдает рекомендацию необходимости поправиться или похудеть. Оптимальный вес вычисляется по формуле: Рост(см)-100.

13. Разработать блок-схему алгоритма, которая запрашивает номер месяца и затем выводит соответствующее название времени года. В случае, если введено недопустимое число, должно выводиться сообщение "Ошибка вводимых данных".

14. Разработать блок-схему алгоритма, которая запрашивает номер дня недели и выводит одно из сообщений: "Рабочий день", "Суббота" или "Воскресенье".

15. Разработать блок-схему алгоритма, которая заменяет меньшее из двух чисел их удвоенным произведением, а большее – полусуммой этих чисел.

16. Разработать блок-схему алгоритма, которая вычисляет значение функции



17.Разработать блок-схему алгоритма, которая сортирует четыре введенных числа в порядке возрастания.

18. Разработать блок-схему алгоритма, которая вычисляет значение функции

**Практическая работа по теме «Линейные алгоритмы»**

**Цель работы**: формирование знаний и умений по разработке программ на линейные алгоритмы

**Задание.** Найти значение алгебраического выражения, соответствующего варианту задания. Вывести результаты на печать. Все результаты выводить в развернутом виде (например: «Сумма чисел А и B равна 3.7854»). Значения вводимых величин должны иметь не менее четырех значащих цифр и задаются студентом самостоятельно.

**Варианты**

Через a и b обозначены подлежащие вводу числа c плавающей точкой.



**Практическая работа по теме «Программирование ветвлений»**

**Цель работы**: формирование знаний и умений по разработке программ на разветвляющиеся алгоритмы

**Ход выполнения работы**

1. Изучить структуры алгоритмов разветвляющихся вычислительных процессов. Изучить отличия структурного алгоритма от неструктурного.
2. Изучить синтаксис и работу операторов ветвления if и switch.
3. Изучить пример выполнения задания. Подключите необходимые библиотеки. Выполнить структуризацию программного кода.
4. Разработать алгоритм решения индивидуального задания, удовлетворяющий требованиям.

**Требования к алгоритму**. Алгоритм должен соответствовать требованиям структурного подхода. Необходимо организовать ввод и вывод всех требуемых данных через диалог ЭВМ с человеком (интерфейс ввода-вывода), организовать проверку входных данных на допустимые значения. Запрещается использование неструктурного оператора goto. Возможно использование досрочного выхода из процедуры с помощью оператора return.

1. Разработать программу на языке С++ для алгоритма индивидуального задания.
2. Выполнить отладку и компиляцию программы, получить исполняемый файл.
3. Выполнить тестирование программы несколькими наборами входных данных.

**Пример выполнения программы**

Составить программу, выполняющую суммирование всех цифр от 0 до 9.

**Код программы:**

const int level = 2;

int main()

{ setlocale(LC\_ALL, "RUS");

//диалог и ввод данных

 int lm = 0; //для меню

cout << "Выберите желаемое действие: " << endl;

 cout << "1: Решить задачу: " << endl;

 cout << "2: Выйти из приложения (без решения задачи)" << endl;

 cout << "Введите желаемое действие: ";

 cin >> lm;

 switch(lm){ //switch

 case 1:

 float a, b, c, d;

 printf("Введите число a = ");

 scanf("%f", &a);

 printf("Введите число b = ");

 scanf("%f", &b);

 printf("Введите число c = ");

 scanf("%f", &c);

 printf("Введите число d = ");

 scanf("%f", &d);

//решение задачи

 if ((a <= b) & (b <= c) & (c <= d)) {

 a = d; //a = b = c = d;

 b = d;

 c = d;

 }

 else

 if ((a > b) & (b > c) & (c > d))

// если написать противоположное условие, то операторы будут в блоке "true"

 ; //пустой оператор

 else { // различные способы возведения в квадрат

 a \*= a;

 b = b\*b;

 c = pow(c, 2);

 d = pow(d, level);

 }

//вывод результатов

 printf("\nЧисло a = %3.2f", a);

 printf("\nЧисло b = %3.2f", b);

 printf("\nЧисло c = %3.2f", c);

 printf("\nЧисло d = %3.2f\n\n", d);

 break;

 default:

 break;

 }// конец switch

 std::system("pause");

 return 0;

}

**Задания**

 1: Ввести число. Если оно неотрицательно, вычесть из него 50, в противном случае прибавить к нему 100.

2  Вывести первое число, если оно меньше второго, и оба числа, если это не так

3 Ввести рост человека. Вывести на экран "высокий", если его рост превышает, 180 см и "не очень высокий" в противном случае.

4Составить программу, которая спрашивает возраст человека и, если ему 18 лет и больше, сообщает “Замечательно. Вы уже можете водить автомобиль”, а в противном случае – “К сожалению, водить автомобиль Вам рановато”

5 Найти максимальное из трех чисел паскаль

6. Психологи рекомендуют учитывать оптимальное соотношение возрастов при вступлении в брак. Оно таково: возраст невесты равен половине возраста жениха плюс. Составьте программу для определения, подходят ли жених и невеста друг другу по возрасту.

7 Увеличьте число х на 10, если оно не отрицательно, во всех остальных случаях уменьшите его на 10

8 Фирма осуществляет набор сотрудников. Условия приема требуют не менее 3 лет рабочего стажа и возраст более 20 лет. Написать программу, проверяющую, удовлетворяет ли кандидат этим требованиям

9 В баскетбольную секцию принимаются юноши в возрасте 13 лет и не ниже 165 см. Составить программу зачисления юношей в секцию.

10  Определить, является ли введенное число двузначным.

 11 Написать программу для подсчета суммы только положительных из трех данных чисел.

12 Даны три числа. Написать программу для подсчета количества чисел, равных нулю.

13 Даны три различных целых числа. Найти среднее из них. Среднем назовем число, которое больше наименьшего из данных чисел, но меньше наибольшего.

14 Составить программу, вычисляющую стоимость покупки с учетом скидки: куплено три товара и, если сумма покупки превышает 2000 р., то начисляется скидка 3%.

15 К финалу конкурса лучшего по профессии "Специалист электродорожник" были допущены трое: Иванов, Петров, Сидоров. Соревнования проходили в 3 тура. Иванов в первом туре набрал m1 баллов, во втором - n1, в третьем - p1. Петров соответственно: m2, n2, p2. Сидоров: m3, n3, p3. Составить программу, определяющую кто победитель и сколько набрал баллов. Решение

16 Валя и Вера на своем садовом участке собрали А кг клубники. Из них В кг собрала Вера. Кто из девочек собрал клубники больше и на сколько?

17 Напишите программу-модель анализа пожарного датчика в помещении, которая выводит сообщение «Пожарная ситуация», если температура (ее значение вводится с клавиатуры) в комнате превысила 60ºС.

18 Хватит ли в баке автомобиля бензина, чтобы доехать до ближайшей заправочной станции?

19 Составить программу упорядочивания трех чисел x, y, z по возрастанию.

20 Имеется билет с шестизначным номером. Определить, является ли билет счастливым.

21 Составить программу, которая запрашивает ввод температуры тела человека и определяет, здоров он или болен (здоров при 36 < t < 37)?

**22** Написать программу, которая спрашивает у пользователя возраст, и выдает информацию о том, является ли он совершеннолетним.

**23** Найти максимальное число из трех введенных с клавиатуры

**24** Даны два числа. Меньшее из этих чисел заменить суммой данных чисел, большее - произведением.

**25**Составить программу, вычисляющую стоимость покупки с учетом скидки: куплено три товара и, если сумма покупки превышает 1500 р., то начисляется скидка 2%

**Контрольные вопросы**

1. Какие управляющие структуры используются в языке C++ для организации разветвляющихся алгоритмов?
2. Поясните порядок выполнения оператора if…else и его сокращенной формы.
3. Поясните порядок выполнения оператора switch. Каково здесь назначение оператора break?
4. Сколько операторов можно написать после ключевых слов ifи else?
5. Что такое составной оператор?
6. Как разрешается неоднозначность, которая может возникнуть при использовании вложенных операторов if

**Практическая работа по теме «Программирование циклов»**

**Цель работы**: формирование знаний и умений по работе с блок-схемами алгоритмов и составление программ на циклические алгоритмы.

* **Контрольные вопросы**
1. Определение циклического алгоритма.
2. Цикл с параметром
* **Задания:**

**1.** Разработать блок-схему алгоритма и программу, которая выводит таблицу квадратов первых десяти целых положительных чисел.

Ниже приведен пример:

Таблица квадратов.

----------------

Число Квадрат

----------------

1 1

1. 4
2. 9
3. 16
4. 25

6 36

7 49

8 64

9 81

10 100

**2.** Разработать блок-схему алгоритма и программу, которая вводит 5 дробных чисел и вычисляет их среднее арифметическое.

Ниже приведен пример:

Вычисление среднего арифметического последовательности дробных чисел.

Введите 5 чисел:
***-> 5.4***

***-> 7.8***

***-> 3.0***

***-> 1.5***

***-> 2.3***

Среднее арифметическое введенной последовательности: 4.00

**3.** Разработать блок-схему алгоритма и программу, которая вычисляет сумму первых n положительных целых чисел. Количество суммируемых чисел должно вводиться.

Ниже приведен пример:

Вычисление суммы положительных чисел.

Введите количество суммируемых чисел -> ***20***

Сумма первых 20 положительных чисел равна 210

**4.** Разработать блок-схему алгоритма и программу, которая вычисляет сумму первых членов ряда: 1, 3, 5, 7 ... Количество суммируемых членов ряда вводится.

Ниже приведен пример:

Вычисление частичной суммы ряда: 1,3,5,7 ...

Введите количество суммируемых членов ряда -> ***15*** I

Сумма первых 15 членов ряда равна 330

**5.** Разработать блок-схему алгоритма и программу, которая вычисляет сумму первых n членов ряда: 1+1/2+1/3+1/4+… Количество суммируемых членов вводится.

Ниже приведен пример:

Вычисление частичной суммы ряда: 1+1/2+1/3+ ...

Введите кол-во суммируемых членов ряда -> ***15***

Сумма первых 15 членов ряда равна 3.3182

**6.** Разработать блок-схему алгоритма и программу, которая создает таблицу степеней двойки от нулевой до десятой.

Ниже приведен пример:

Таблица степеней двойки

-----------------------

1. 1
2. 2
3. 4
4. 8
5. 16
6. 32
7. 64
8. 128
9. 256
10. 512
11. 1024

**7.** Разработать блок-схему алгоритма и программу, которая вычисляет факториал введенного числа. (Факториалом числа n называется произведение целых чисел от 1 до n. Например, факториал 1 равен 1, 8 — 40320).

Ниже приведен пример:

Вычисление факториала

Введите число, факториал которого надо вычислить

-> ***7***

Факториал 7 равен 5040

**8.**Разработать блок-схему алгоритма и программу, которая выводит таблицу значений функции у = -2,4x2+5x-3 в диапазоне от -2 до 2, с шагом 0,5.

Ниже приведен пример:

----------------

х | у

----------------

-2 | -22.60

-1.5 | -15.90

-1 | -10.40

-0.5 | -6.10

1. | -3.00
0.5 | -1.10
2. | -0.40
1.5 | -0.90

2 | -2.60

**9.** Разработать блок-схему алгоритма и программу, которая вычисляет среднее арифметическое вводимой с клавиатуры последовательности дробных чисел. Количество чисел должно вводиться.

Ниже приведен пример:

Вычисление среднего арифметического последовательности дробных чисел.

Введите количество чисел последовательности ***-> 5***

Вводите последовательность.

***-> 5.4***

***-> 7.8***

***-> 3.0***

***-> 1.5***

***-> 2.3***

Среднее арифметическое введенной последовательности: 4.00

**10.** Разработать блок-схему алгоритма и программу, которая вводит последовательность из пяти чисел и после ввода каждого числа выводит среднее арифметическое введенной части последовательности.

Ниже приведен пример:

Обработка последовательности дробных чисел

После ввода каждого числа нажимайте <Enter>

***-> 12.3***

Введено чисел: 1 Сумма: 12.30 Сред. арифметическое: 12.30

***-> 15***

Введено чисел: 2 Сумма: 27.30 Сред. арифметическое: 13.65

***-> 10***

Введено чисел: 3 Сумма: 37.30 Сред. арифметическое: 12.43

***-> 5.6***

*Введено*чисел: 4 Сумма: 42.90 Сред. арифметическое: 10.73

***-> 11.5***

Введено чисел: 5 Сумма: 54.40 Сред. арифметическое: 10.88

**11.** Разработать блок-схему алгоритма и программу, которая вычисляет среднее арифметическое последовательности вводимых чисел. После ввода последнего числа выводится минимальное и максимальное число последовательности. Количество чисел последовательности задается.

Ниже приведен пример:

Обработка последовательности дробных чисел.

Введите количество чисел последовательности ***-> 5***

Вводите последовательность:

***-> 5.4***

***-> 7.8***

***-> 3.0***

***-> 1.5***

***-> 2.3***

Количество чисел: 5

Среднее арифметическое: 4.00

Минимальное число: 1.5

Максимальное число: 7.8

**12.** Разработать блок-схему алгоритма и программу, которая формирует таблицу стоимости, например, яблок в диапазоне от 100 г до 1 кг с шагом 100 г.

Ниже приведен пример:

Введите цену одного килограмма и нажмите <Enter>

(копейки от рублей отделяйте точкой)

***-> 16.50***

Вес Стоимость

(гр) (руб.)

----------

100 1.65

200 3.30

300 4.95

400 б.60

500 8.25

600 9.90

700 11.55
800 13.20

900 14.85

1000 16.50

**13.** Разработать блок-схему алгоритма и программу, которая выводит таблицу значений функции у = -х. Диапазон изменения аргумента от -4 до 4, шаг приращения аргумента 0,5.

**14.** Разработать блок-схему алгоритма и программу вычисления значения величины

N = (знак П - знак произведения k из диапазона [7;20])

**15.** Разработать блок-схему алгоритма и программу, которая вычисляет сумму первых n целых положительных четных целых чисел. Количество суммируемых чисел должно вводиться.

Ниже приведен пример:

Вычисление суммы четных положительных чисел.

Введите количество суммируемых чисел:

-> ***12***

Сумма первых 12 положительных четных чисел равна 156

**16.** Разработать блок-схему алгоритма и программу, которая выводит на экран таблицу умножения на 7 чисел от 2 до 9.

Ниже приведен пример:

Таблица умножения на 7:

7 x 2 = 14

7 x 3 = 21

7 x 4 = 28

7 x 5 = 35

7 х 6 = 42

7 x 7 = 49

7 x 8 = 56

7 x 9 = 63

**17.** Разработать блок-схему алгоритма и программу, которая выводит на экран квадрат Пифагора — таблицу умножения.

Ниже приведен пример:

1 2 3 4 5 6 7 8 9 10

1. 1 2 3 4 5 6 7 8 9 10
2. 2 4 6 8 10 12 14 16 18 20
3. 4 8 9 12 15 18 21 24 27 30
4. *5*8 12 16 20 24 23 32 36 40
5. 5 10 15 20 25 30 35 40 45 50
6. 6 12 18 24 30 36 42 48 54 60
7. 7 14 21 28 35 42 49 56 63 70
8. 8 16 24 32 40 48 56 64 72 80
9. 9 18 27 36 45 54 63 72 81 90

**18.** Разработать блок-схему алгоритма и программу, которая вычисляет значения величины

M = 7 \*

(знак ∑ - знак алгебраической суммы для с из диапазона [1;20])

**19.** Разработать блок-схему алгоритма и программу, которая выводит таблицу значений функции y=(x-2)+(x+l). Диапазон изменения аргумента от -4 до 4, шаг приращения аргумента 0,5.

**20.** Разработать блок-схему алгоритма, которая вычисляет значения величины

C =

(знак ∑ - знак алгебраической суммы для a из диапазона [4;12])

**21.** Разработать блок-схему алгоритма и программу, которая выводит таблицу квадратов первых пяти целых положительных нечетных чисел.

Ниже приведен пример:

----------------

Число Квадрат

----------------

1 1

3 9

5 25

7 49

9 81

**22.**Разработать блок-схему и программу, которая выводит на экран таблицу умножения на 9 чисел от 2 до 9.

Ниже приведен пример:

Таблица умножения на 9:

9 x 2 = 18

9 x 3 = 27

9 x 4 = 36

9 x 5 = 45

9 х 6 = 54

9 x 7 = 63

9 x 8 = 72

9 x 9 = 81

**23.** Разработать блок-схему алгоритма и программу, которая вычисляет произведение первых n положительных четных целых чисел. Количество чисел должно вводиться.

Ниже приведен пример:

Вычисление произведения четных положительных чисел.

Введите количество чисел:

-> ***4***

Произведение первых 4 положительных четных чисел равно 384

**24.** Разработать блок-схему алгоритма и программу, которая выводит двузначные числа, кратные десяти, и вычисляет их сумму.

Ниже приведен пример:

Вычисление суммы двузначных чисел, кратных 10.

Это числа:

10 20 30 40 50 60 70 80 90

Их сумма равна 450

**25.** Разработать блок-схему алгоритма и программу, которая выводит на экран таблицу деления на 7 чисел от 70 до 7 с шагом 7.

Ниже приведен пример:

Таблица деления на 7:

70 / 7 = 10

63 / 7 = 9

56 / 7 = 8

49 / 7 = 7

42 / 7 = 6

35 / 7 = 5

28 / 7 = 4

21 / 7 = 3

14 / 7 = 2

7 / 7 = 1

**26.**Разработать блок-схему алгоритма и программу, которая выводит на экран таблицу значений функции у= 5х - 2 на отрезке [1; 10] с шагом b=2.

Ниже приведен пример:

----------------

х | у

----------------

1 | 3

3 | 13

5 | 23

7 | 33

9 | 43
**27.** Разработать блок-схему алгоритма и программу, которая вычисляет значения величины D =

(знак П - знак произведения для к из диапазона [5;10] )

**28.** Разработать блок-схему алгоритма и программу, которая преобразует введенное десятичное число в двоичное.

Ниже приведен пример:

Преобразование десятичного числа в двоичное

Введите целое число от 0 до 255 и нажмите <Enter>

***-> 49***

Десятичному числу 49 соответствует двоичное 00110001

Для завершения нажмите <Enter>

**29.** Разработать блок-схему алгоритма, которая выводит все числа, кратные числу 7 из диапазона от 21 до 63, находит их среднее арифметическое.

**30.** Разработать блок-схему для проверки знания таблицы умножения. Она должна вывести 10 примеров и выставить оценку: за 10 правильных ответов — "отлично", за 9 и 8 — "хорошо", за 7 и 6 — "удовлетворительно", за 6 и менее — "плохо". Ниже приведен рекомендуемый вид экрана во время работы программы. Ответы пользователя выделены полужирным шрифтом:

\*\*\* Проверка знания таблицы умножения \*\*\*

После примера введите ответ и нажмите <Enter>.

5x3=**15**

7x7=**49**

1x4=**4**

4x3=**12**

9x4=**36**

8x8=**64**

7x8=**52**

Вы ошиблись! 7x8=56

4x7=**28**

3x5=**15**

2x5=**10**

Правильных ответов: 9

Оценка: Хорошо.

**Практическая работа по теме«Тестирование и оптимизация работы программ. Составление отчетной документации»**

**Цель работы**: формирование знаний и умений по работе с системой программирования, разработка программ, реализующих базовые алгоритмические конструкции (ветвления и циклы), тестирование программ и составление документации.

**Порядок выполнения работы при формировании отчета:**

1. Создать математическуюмодель решения задачи, содержащую описание переменных и связывающие их формулы, описать её в пункте **«Математическая модель».**
2. Создать графическую модель решения задачи – блок-схему в пункте**«Графическая модель»**
3. Разработать программу в среде программирования и описать её в пункте**«Программная модель решения задачи».**
4. Протестировать программу и результаты работы программы описать в пункте **«Тестирование».**
* **Задания:**

**1.**Ввести целое число и определить, верно ли, что в его записи есть две одинаковые цифры.

Пример:

Введите целое число: Введите целое число:

1234 1224

Нет. Да.

**2.**Найти произведение цифр заданного четырёхзначного числа.

Пример:

Введите четырёхзначное число: 1234

Произведение цифр числа равнo 24

**3.**Ввести целое число и определить число цифр в нем, вывести эти цифры. Найти произведение и сумму цифр числа.

**4.**Написать программу, находящую минимальное и максимальное число в последовательности поочередно вводимых с клавиатуры чисел. Количество чисел задается во время работы программы.

**5.**Разработать программу, выводящую квадраты и кубы 15 чисел следующей последовательности:

1, 2, 4, 7, 11, 16, …

**6.**Разработать программу решения следующей задачи:

Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 4 дня.

**7.**Разработать программу, которая определяет, верно ли, что в заданном числе все цифры стоят по возрастанию.

Пример:

Введите число: 258

Верно, цифры стоят по возрастанию.

**8.**Разработать программу, которая определяет наибольший общий делитель двух чисел , вводимых с клавиатуры.

**9.**Разработать программу, которая последовательно вводит с клавиатуры N произвольных чисел, подсчитывает а) количество целых чисел и их сумму, б) количество дробных чисел и их произведение.

**10.**Разработать программу, которая вводит с клавиатуры последовательность из M дробных чисел и после ввода каждого числа выводит сумму и среднее арифметическое введенной части последовательности. Например:

Обработка последовательности дробных чисел

После ввода каждого числа нажимайте <Enter>

***-> 12.3***

Введено чисел: 1 Сумма: 12.30 Сред. арифметическое: 12.30

***-> 15***

Введено чисел: 2 Сумма: 27.30 Сред. арифметическое: 13.65

**11.**Разработать программу, которая вычисляет среднее арифметическое последовательности N чисел, вводимых с клавиатуры. После ввода последнего числа программа должна вывести минимальное число последовательности. Например:

Введите количество чисел последовательности***-> 5***

Вводите последовательность.

***-> 5.4***

***-> 7.8***

***-> 3.0***

***-> 1.5***

***-> 2.3***

Среднее арифметическое: 4.00

Минимальное число: 1.5

**12.**Разработать программу, которая выводит на экран таблицу умножения на 9 чисел от 2 до 9.

Ниже приведен пример:

Таблица умножения на 9:

9 x 2 = 18

9 x 3 = 27

9 x 4 = 36

9 x 5 = 45

9 х 6 = 54

9 x 7 = 63

9 x 8 = 72

9 x 9 = 81

**13.**Разработать программу, которая выводит все делители положительного целого числа, введенного с клавиатуры.

**14.**Разработать программу, которая выводит таблицу значений функции y=(x-2)+(x+l). Диапазон изменения аргумента от -4 до 4, шаг приращения аргумента 0,5.

**15.**Разработать программу, которая преобразует введенное десятичное число в двоичное.

Ниже приведен пример:

Преобразование десятичного числа в двоичное

Введите целое число от 0 до 255 и нажмите <Enter>

***-> 49***

Десятичному числу 49 соответствует двоичное 00110001

Для завершения нажмите <Enter>

**16.**Ввести целое число и определить, верно ли, что в его записи есть две одинаковые цифры.

Пример:

Введите целое число: Введите целое число:

1234 1224

Нет. Да.

**17.**Найти произведение цифр заданного четырёхзначного числа.

Пример:

Введите четырёхзначное число: 1234

Произведение цифр числа равнo 24

**18.**Ввести целое число и определить число цифр в нем, вывести эти цифры. Найти произведение и сумму цифр числа.

**19.**Написать программу, находящую минимальное и максимальное число в последовательности поочередно вводимых с клавиатуры чисел. Количество чисел задается во время работы программы.

**20.**Разработать программу, выводящую квадраты и кубы 15 чисел следующей последовательности:

1, 2, 4, 7, 11, 16, …

**21.**Разработать программу решения следующей задачи:

Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 4 дня.

**22.**Разработать программу, которая определяет, верно ли, что в заданном числе все цифры стоят по возрастанию.

Пример:

Введите число: 258

Верно, цифры стоят по возрастанию.

**23.**Разработать программу, которая определяет наибольший общий делитель двух чисел , вводимых с клавиатуры.

**24.**Разработать программу, которая последовательно вводит с клавиатуры N произвольных чисел, подсчитывает а) количество целых чисел и их сумму, б) количество дробных чисел и их произведение.

**25.**Разработать программу, которая вводит с клавиатуры последовательность из M дробных чисел и после ввода каждого числа выводит сумму и среднее арифметическое введенной части последовательности. Например:

Обработка последовательности дробных чисел

После ввода каждого числа нажимайте <Enter>

***-> 12.3***

Введено чисел: 1 Сумма: 12.30 Сред. арифметическое: 12.30

***-> 15***

Введено чисел: 2 Сумма: 27.30 Сред. арифметическое: 13.65

**26.**Разработать программу, которая вычисляет среднее арифметическое последовательности N чисел, вводимых с клавиатуры. После ввода последнего числа программа должна вывести минимальное число последовательности. Например:

Введите количество чисел последовательности***-> 5***

Вводите последовательность.

***-> 5.4***

***-> 7.8***

***-> 3.0***

***-> 1.5***

***-> 2.3***

Среднее арифметическое: 4.00

Минимальное число: 1.5

**27.**Разработать программу, которая выводит на экран таблицу умножения на 9 чисел от 2 до 9.

Ниже приведен пример:

Таблица умножения на 9:

9 x 2 = 18

9 x 3 = 27

9 x 4 = 36

9 x 5 = 45

9 х 6 = 54

9 x 7 = 63

9 x 8 = 72

9 x 9 = 81

**28.**Разработать программу, которая выводит все делители положительного целого числа, введенного с клавиатуры.

**29.**Разработать программу, которая выводит таблицу значений функции y=(x-2)+(x+l). Диапазон изменения аргумента от -4 до 4, шаг приращения аргумента 0,5.

**30.**Разработать программу, которая преобразует введенное десятичное число в двоичное.

Ниже приведен пример:

Преобразование десятичного числа в двоичное

Введите целое число от 0 до 255 и нажмите <Enter>

***-> 49***

Десятичному числу 49 соответствует двоичное 00110001

Для завершения нажмите <Enter>

**Практическая работа «Программирование функций»**

**Цель работы**: формирование знаний и умений по работе с функциями

**Задания:**

1. Рассчитать значение *х*, определив и использовав функцию :

|  |  |
| --- | --- |
| а)   | г)  |
| б)   | д)   |
| в)  | е)  |

1. Определить значение

 а) z=max(a,2b)\*max(2a-b,b) б) z=min(a,3b)\*min(2a-b,b)

 где max(x,y), min(x,y) - максимум и минимум из чисел x,y

 При решении:1) не использовать функцию max, min.

 2) определить и использовать функцию max,min.

1. Даны основания и высоты двух равнобедренных трапеций. Найти сумму их периметров. (Определить функцию для расчета периметра равнобедренной трапеции по ее основаниям м высоте.)
2. Даны три квадратных уравнения ax2 + bx + c = 0, bx2 + ax + c = 0, cx2 + ax + + b = 0. Сколько из них имеют вещественные корни ? (Определить функцию, позволяющую распознавать наличие вещественных корней в квадратном уравнении.)
3. Найти периметр треугольника, заданного координатами своих вершин. (Определить функцию для расчета отрезка по координатам его вершин)
4. Даны вещественные числа a, b, c, d, e, f, g.

|  |  |
| --- | --- |
| Найти площадь пятиугольника, изображенного на рисунке. (Определить функцию для расчета площади треугольника по трем сторонам.) |  е  g d  a f  c b |

1. Даны вещественные числа x1, y1, x2, y2,... x5, y5.Найти площадь пятиугольника (см 10.6) вершины которого имеют координаты (x1, y1), (x2, y2), ...( x5, y5). (Определить функцию для расчета площади треугольника по координатам его вершин).
2. Даны две последовательности целых чисел : а1, а2,... а8, и b1, b2, ... b8. Найти количество четных чисел в первой из них и количество нечетных вол второй. (Определить функцию, позволяющую распознавать четные числа.)
3. Найти значение выражения (2\*5! + 3\*8!)/(6! + 4!), где n! функция.
4. Даны два натуральных числа. Выяснить, в каком из них сумма цифр больше.(сумма цифр - функция).
5. Даны n различных чисел. Определить максимальное из них. (Определить функцию, находящую максимум из двух различных чисел.)
6. Найти НОД(a,b,c)=НОД(НОД(a,b),c). (где НОД(a,b) - функция.)
7. Даны n натуральных чисел. Найти их НОД. (где НОД(a,b) - функция.)

Написать рекурсивную функцию :

 а) вычисления суммы цифр натурального числа;

 б) вычисления количества цифр натурального числа.

1. Даны первый член и разность арифметической прогрессии. Написать рекурсивную функцию для нахождения :

 а) n-го члена прогрессии; б) суммы n первых членов прогрессии.

1. Даны первый член и знаменатель геометрической прогрессии. Написать рекурсивную функцию для нахождения :

 а) n-го члена прогрессии;б) суммы n первых членов прогрессии.

**Практические работы по темам «Решение задач на заполнение и вывод линейного массива. Решение задач на нахождение суммы, произведения, количества элементов. . Решение задач поиск максимального и минимального элементов массива. Решение задач на сортировку массива.»**

**Цель работы**: формирование знаний и умений по работе с блок-схемами алгоритмов, реализующих обработку элементов одномерного массива. Их реализация на языке программирования.

**Порядок выполнения работы:**

1. Создать математическуюмодель решения задачи, содержащую описание переменных и связывающие их формулы, описать её в пункте **«Математическая модель».**
2. Создать графическую модель решения задачи – блок-схему в пункте**«Графическая модель»**
3. Разработать программу в среде программирования и описать её в пункте**«Программная модель решения задачи».**

**Краткие теоретические сведения**

Массив — это набор переменных одного типа, имеющих одно и то же имя. Доступ к конкретному элементу массива осуществляется с помощью индекса. В языке Паскаль все массивы располагаются в отдельной непрерывной области памяти. Первый элемент массива располагается по самому меньшему адресу, а последний — по самому большему. Массивы могут быть одномерными и многомерными.

Как и другие переменные, массив должен быть объявлен явно, чтобы компилятор выделил для него определенную область памяти (т.е. разместил массив). Здесь тип обозначает базовый тип массива, являющийся типом каждого элемента. Размер задает количество элементов массива.

Доступ к элементу массива осуществляется с помощью имени массива и индекса. Индекс элемента массива помещается в скобках после имени. Индекс первого элемента любого массива в СИ++ равен единице.

Объем памяти, необходимый для хранения массива, непосредственно определяется его типом и размером. Можно сказать, что одномерный массив — это список, хранящийся в непрерывной области памяти в порядке индексации.

К стандартным операциям при работе с массивом можно отнести: поиск максимального и минимального по значению элемента, обработку элементов по определенному алгоритму, сортировку по возрастанию или по убыванию значений элементов массива.

Сортировка массива методом парных перестановок («метод пузырька») - сравнение соседних элементов и, если нужно, их перестановка. Если за один проход сортировка не приносит нужного результата, приходится просматривать все пары элементов несколько раз (а именно, если в массиве K элементов, то алгоритм нужно повторить K-1 раз)

**ПРИМЕР:** Дан массив, содержащий элементы {7,8,-5,-1}. Отсортируем его по возрастанию:

Меняем пары элементов значениями 1-ый раз:

1) {7,8,-5,-1} – обмена 1-го и 2-го элементов значениями не будет;

2) {7,-5,8,-1} – обмен 2-го и 3-го элементов значениями;

3) {7,-5,-1,8} – обмен 3-го и 4-го элементов значениями;

За первый проход массив не был отсортирован, поэтом повторяем алгоритм 2-ой раз:

1) {-5,7,-1,8} – обмен 1-го и 2-го элементов значениями;

2) {-5,-1,7,8} – обмен 2-го и 3-го элементов значениями;

3) {-5,-1,7,8} – обмена 3-го и 4-го элементов значениями не будет;

Массив отсортирован.

* **Контрольные вопросы**
1. Что такое массив?
2. Что такое размерность массива?
3. Какого типа могут быть элементы массива?
4. Какого типа могут быть индексы элементов массива?
5. Какими способами может быть заполнен массив? Приведите примеры.
6. Какие действия выполняют обычно над элементами массива?

**Задания:**

1. Составить программу, заменяющую каждый элемент массива на противоположный по знаку. Вывести на экран исходный и полученный массивы (вывод массива в строку). Размерность массива произвольная.
2. Сформировать массив Y значений функции Y=BX2+D при x=11,12, …20. Значения B и D вводятся с клавиатуры. Вывести полученный массив в столбец.
3. Подсчитать количество отрицательных элементов массива с номерами позиций, кратных трем. Исходный массив вывести в строку. Размерность массива произвольная.
4. Составить программу, позволяющую найти минимальный и максимальный элементы массива. Вывести исходный массив в строку, а также значения максимального и минимального элементов. Размерность массива произвольная.
5. Написать программу, заменяющую каждый элемент, стоящий в четной позиции массива, его модулем. Вывести исходный и полученный массивы (в строку). Размерность массива произвольная.
6. Составить программу, вычисляющую среднее арифметическое всех элементов массива. Вывести исходный массив в строку и значение среднего арифметического. Размерность массива произвольная.
7. Составить программу, удваивающую четные по номеру элементы массива, а нечетные по номеру уменьшить вдвое. Вывести исходный и полученный массивы (в строку). Размерность массива произвольная.
8. Составить программу, позволяющую посчитать количество отрицательных и положительных элементов массива. Массив вывести в строку. Размерность массива произвольная.
9. Заменить элементы массива, превышающие число С, на сумму С+2\*B, гдеB – превышение данного элемента над числом С. Вывести исходный и полученный массивы (в строку). Размерность массива произвольная.
10. Составить программу нахождения произведения элементов массива, стоящих в четной позиции массива. Вывести массив в строку. Размерность массива произвольная.
11. Составить программу, заменяющую каждый элемент, стоящий в четной позиции массива, на элемент, в 3 раза больший. Вывести исходный и преобразованный массив (в строку). Элементы массива принадлежат диапазону от -5 до 5.
12. Подсчитать количество отрицательных, нулевых и положительных элементов массива с номерами позиций, кратных двум, начиная с первого. Размерность массива произвольная.
13. Написать программу, находящую во вводимом с клавиатуры массиве все элементы, равные нулю. На экран выдать номера их позиций и значения. Размерность массива произвольная. Диапазон значений от -20 до 20.
14. Составить программу, заменяющую каждый элемент массива на элемент, в 2 раза больший и противоположный по знаку. Диапазон значений от -4 до 10. Вывести исходный и полученный массивы (в строку).
15. Подсчитать количество положительных элементов массива с номерами позиций, кратных трем. Вывести исходный массив в строку, размерность массива произвольная.
16. Составить программу, позволяющую найти минимальный элемент массива. Подсчитать количество элементов в массиве, равных минимальному. Размерность массива
17. Составить программу, вычисляющую среднее арифметическое всех положительных элементов массива. Размерность массива и диапазон значений произвольные.
18. Составить программу, удваивающую четные по номеру элементы массива, а нечетные по номеру уменьшить вдвое. Размерность массива и диапазон значений произвольные.
19. Составить программу, позволяющую посчитать среднее арифметическое отрицательных и положительных элементов массива. Размерность массива 10 элементов, массив заполнить путем ввода элементов с клавиатуры.
20. Составить программу нахождения произведения четных по порядковому номеру элементов массива. Размерность массива 10 элементов, массив заполнить путем ввода элементов с клавиатуры.
21. Дан массив из 8 чисел. Написать программу подсчета количества элементов массива, равных наибольшему элементу в этом массиве. Массив заполнить путем ввода элементов с клавиатуры.
22. Дан массив из 25 целых чисел. Написать программу поиска элементов массива, равных наибольшему. Вывести на экран номера таких элементов. Заполнение с помощью датчика случайных чисел.
23. Дан массив из 30 элементов. Написать программу, которая находит и выводит количество элементов массива, значение которых больше среднего арифметического значения элементов этого массива. Размерность массива и диапазон значений произвольные.
24. Дан массив на 30 элементов. Написать программу, которая позволяет найти и вывести сумму элементов массива, значение которых больше среднего арифметического значения элементов этого массива. Размерность массива и диапазон значений произвольные.
25. Дан массив из 25 целых чисел. Написать программу подсчета суммы положительных элементов, предшествующих максимальному элементу массива.
26. Дан массив из 25 целых чисел. Написать программу вычисления суммы отрицательных элементов, предшествующих минимальному элементу.
27. Дан массив из 30 элементов. Написать программу, вычисляющую произведение положительных элементов, расположенных после максимального элемента массива.
28. Дан массив из 30 элементов. Написать программу, вычисляющую произведение отрицательных элементов, расположенных после минимального элемента массива.
29. В массиве хранится информация о стоимости каждого из 50 марок легковых автомобилей. Определить, сколько стоит самый дорогой автомобиль.
30. В массиве хранится информация о стоимости 1 килограмма 30 видов конфет. Определить номер самого дешевого вида конфет.
31. В массиве хранится информация о среднедневной температуре за каждый день июля. Определить даты самых теплых дней.
32. Дан массив целых чисел. Переписать его второй, четвертый, и т.д. элементы в другой массив такого же размера :

а) расположив элементы на тех же местах, что и в исходном массиве;

б) расположив элементы подряд с начала массива.

1. Дан массив из 20 вещественных чисел. Сформировать два массива размером 10, включив в первый их них элементы заданного массива с четными индексами, а во второй - с нечетными.
2. Дан массив целых чисел. Выяснить :

а) верно ли, что сумма элементов, которые больше 20 превышает 100;

б) верно ли, что сумма элементов, которые меньше 50, есть четное число.

1. Известна численность каждого из 40 групп училища. Верно ли, что в самой многочисленной группе учится на 10 учеников больше, чем в самой малочисленной.
2. Удалить из массива вещественных чисел :

а) минимальный (максимальный) элемент;

б) третий элемент;

в) упорядочить массив по возрастанию

1. Дан массив из 20 элементов. Переставить первые три и последние три элемента, сохранив порядок их следования.
2. Вставить в массив целых чисел :

а) число 10 после второго элемента;

б) заданное число после первого отрицательного (перед последним четным элементом)

**Практические работы по теме «Двумерные масивы»**

**Цель работы**: формирование знаний и умений по работе с блок-схемами алгоритмов, реализующих обработку элементов двумерного массива. Их реализация на языке программирования.

* **Краткие теоретические сведения**

Массив — это набор переменных одного типа, имеющих одно и то же имя. Доступ к конкретному элементу массива осуществляется с помощью индекса. В языке Паскаль все массивы располагаются в отдельной непрерывной области памяти. Первый элемент массива располагается по самому меньшему адресу, а последний — по самому большему. Массивы могут быть одномерными и многомерными.

Как и другие переменные, массив должен быть объявлен явно, чтобы компилятор выделил для него определенную область памяти (т.е. разместил массив). Здесь тип обозначает базовый тип массива, являющийся типом каждого элемента. Размер задает количество элементов массива.

Доступ к элементу двумерного массива осуществляется с помощью имени массива и индексов (индекс нахождения в строке и индекс нахождения в столбце). Индексы элемента массива помещаются в скобках после имени. Индекс первого элемента любого массива в языке Паскаль равен [0,0].

Объем памяти, необходимый для хранения массива, непосредственно определяется его типом и размером. Можно сказать, что массив — это список, хранящийся в непрерывной области памяти в порядке индексации.

К стандартным операциям при работе с массивом можно отнести: поиск максимального и минимального по значению элемента, обработку элементов по определенному алгоритму, сортировку по возрастанию или по убыванию значений элементов массива.

* **Контрольные вопросы**
1. Что такое двумерный массив?
2. Что такое размерность двумерного массива?
3. Какого типа могут быть элементы двумерного массива?
4. Какого типа могут быть индексы элементов двумерного массива?
5. Какими способами может быть заполнен двумерный массив? Приведите примеры.
6. Какие действия выполняют обычно над элементами двумерного массива?
* **задания**

**Задания по двумерным массивам**

1. Образовать матрицу R в три строки и четыре столбца. Подсчитать сумму элементов 2-ой строки матрицы.
2. Образовать двумерный массив ABC[5][5] и вывести его на экран. Подсчитать количество нулевых элементов массива.
3. Заменить положительные элементы двумерной матрицы их целыми частями, а отрицательные – их модулями. Подсчитать произведение элементов 2-ой строки матрицы.
4. Получить квадратную матрицу произвольного размера из элементов. Подсчитать среднее арифметическое элементов главной диагонали матрицы.
5. Найти количество элементов двумерной матрицы, превосходящих 10 и меньших 20. Суммировать такие элементы.
6. Получить квадратную матрицу произвольного размера, если каждый её элемент. Найти среднее арифметическое элементов, лежащих ниже главной диагонали.
7. Определить произведение не лежащих на главной диагонали элементов квадратной матрицы произвольного размера, каждый элемент которой R [ i , j ] = | i – j |
8. Вычислить среднее арифметическое и среднее геометрическое всех элементов двумерной матрицы произвольного размера. Найти сумму элементов квадратной матрицы произвольного размера

а) расположенных на главной диагонали

б) расположенных выше главной диагонали

в) расположенных ниже главной диагонали

1. Найти сумму кубов элементов главной диагонали квадратной матрицы произвольного размера. Подсчитать количество отрицательных элементов матрицы.
2. Подсчитать число отрицательных, нулевых и положительных элементов двумерного квадратного массива, лежащих:

а) выше главной диагонали

б) ниже главной диагонали.

1. Найти произведение элементов квадратной матрицы произвольного размера, расположенных а)выше главной диагонали, б) ниже главной диагонали, в) на главной диагонали.
2. Найти произведение элементов квадратной матрицы, лежащих выше главной диагонали, если каждый элемент такой матрицы A [ i, j] = | j –2i |
3. Найти среднее арифметическое элементов двумерной матрицы, каждый элемент которой B
4. Найти сумму квадратов элементов, расположенных ниже главной диагонали квадратной матрицы произвольного размера. Каждый элемент матрицы вычисляется по формуле С [ i, j] = 2i + 2j
5. В двумерном массиве произвольной размерности подсчитать количество элементов, равных наибольшему элементу в этом массиве. Вывести значение наибольшего элемента.
6. Дан двумерный квадратный массив из 16 элементов. Написать программу, которая находит и выводит количество элементов массива, значение которых больше среднего арифметического значения элементов этого массива.
7. Подсчитать в двумерном квадратном массиве целых чисел количество четных чисел и их сумму. Размерность массива произвольная.
8. Подсчитать в двумерном массиве произвольной размерности сумму элементов каждой строки.
9. Дан квадратный массив из 25 целых чисел. Написать программу поиска элементов массива, равных наибольшему. Вывести на экран номера позиций таких
10. Дан квадратный массив на 16 элементов. Написать программу, которая позволяет найти и вывести сумму элементов массива, значение которых больше среднего арифметического значения элементов этого массива.
11. Подсчитать в двумерном массиве целых чисел количество нечетных чисел и их произведение и сумму.
12. В двумерном массиве произвольной размерности найти максимальный из четных и минимальный из нечетных элементов массива.
13. Дан целочисленный прямоугольный массив А. Составьте программу, которая меняет местами столбцы, содержащие максимальный и минимальный элементы (предполагается, что такие элементы единственные)
14. Дан целочисленный прямоугольный массив А. Составьте программу, которая меняет местами строки, содержащие максимальный и минимальный элементы (предполагается, что такие элементы единственные)

**Практическая работа по теме «Работа со строками»**

**Цель работы**: формирование знаний и умений по работе с блок-схемами алгоритмов, реализующих обработку символьных и строковых данных. Реализация алгоритмов на языке программирования.

**Краткие теоретические сведения**

Язык С++ может обрабатывать не только числа, но и символы. Это даёт возможность представлять в программах тексты и выполнять над ними некоторые операции, обрабатывать различные ведомости, документы, справочники и т.д.

Значениями символьного типа является всё множество символов ПК. Каждому символу приписывается целое число в диапазоне 0...255. Это число служит кодом внутреннего представления символов. Для кодировки используется код ASCII (Американский стандартный код для обмена информацией). Символьная константа (литера) - это символ, заключенный в апострофы, например: 'A', 'R', '+', '7'. Внешние апострофы являются признаком символьной константы, которая занимает 1 байт памяти.

Символьная переменная принимает значение одного символа. В разделе переменных она обозначается описателем символьного типа.

**Строковые данные**

Строковые данные - это последовательность символов произвольной длины (до 255 символов). Размер строки указывают в скобках, но если размер не указан, то он считается равным 255. Для строковых данных применимы операции объединения (конкатенации) и сравнения.

* **Задания:**

**Уровень 1**

**ВАРИАНТ 1.**

Подсчитать сколько раз среди символов заданной строки встречается буква «F», определить длину строки.

**ВАРИАНТ 2.**

Составить программу, которая печатает заданное слово задом наперед, определить длину строки.

**ВАРИАНТ 3.**

Из заданной строки получить новую, выбросив из нее все точки. Вывести на экран
полученную строку.

**ВАРИАНТ 4.**

Подсчитать количество цифр в заданной строке. Если их нет, выдать соответствующее сообщение. Определить длину введенной строки.

**ВАРИАНТ 5.**

Определить количество слов во введенном с клавиатуры тексте, определить количество слогов во введенном с клавиатуры слове.

**ВАРИАНТ 6.**

Из заданной строки получить новую, удалив все звездочки («\*») и повторив
каждый символ, отличный от звездочки. Распечатать полученную строку.

**ВАРИАНТ 7.**

В заданной строке подсчитать отдельно количество символов «а», «b», «с», «d».

**ВАРИАНТ 8.**

Написать программу, которая проверяет, является ли введенная с клавиатуры строка двоичным числом. Определить количество разрядов числа.

**ВАРИАНТ 9.**

Из заданной строки получить новую, заменив в ней все символы после первой
звездочки («\*») знаком «-» (если в строке нет звездочек, оставить ее без
изменения). Распечатать полученную строку.

**ВАРИАНТ 10.**

Выдать номер позиции заданной строки, в которой впервые встречается буква
«G». Если такой буквы в строке нет, выдать сообщение об этом.

**ВАРИАНТ 11.**

В заданной строке подсчитать количество букв латинского алфавита и количество цифр.

**ВАРИАНТ 12.**

Из заданной строки получить новую, удалив все символы, не являющиеся
буквами латинского алфавита.

**ВАРИАНТ 13.**

Написать программу, которая проверяет, является ли введенная с клавиатуры строка шестнадцатеричным числом.

**ВАРИАНТ 14.**

Из заданной строки получить новую, заменив все запятые на точки. Распечатать
полученную строку. Определить количество символов строки.

**ВАРИАНТ 15.**

Выдать номер позиции заданной строки, в которой в пятый раз встречается буква
«W». Если такой позиции нет, выдать сообщение об этом.

**ВАРИАНТ 16.**

В заданной строке преобразовать все заглавные буквы латинского алфавита в
строчные, а строчные в заглавные.

**ВАРИАНТ 17.**

Проверить, является ли заданное слово палиндромом («перевертышем», как
слова КАЗАК, ШАЛАШ и т.д.)

**ВАРИАНТ 18.**

Произвести последовательную замену каждого 2-го символа строки на восклицательный знак.

**ВАРИАНТ 19.**

Преобразовать введенную вопросительную фразу в восклицательную (если исходная фраза не является вопросительной, повторить ввод)

**ВАРИАНТ 20.**

В заданной строке определить порядковые номера всех цифр. Если цифр нет, выдать соответствующее сообщение.

**ВАРИАНТ 21.**

Подсчитать сколько раз среди символов заданной строки встречается буква «S», выдать номера позиций встреченной буквы, определить длину строки.

**ВАРИАНТ 22.**

Составить программу, которая печатает заданное слово задом наперед, определить длину строки.

**ВАРИАНТ 23.**

Из заданной строки получить новую, удалив из нее все запятые. Если их нет, выдать соответствующее сообщение. Вывести на экран полученную строку.

**ВАРИАНТ 24.**

Подсчитать количество знаков, отличных от цифр в заданной строке. Если их нет, выдать соответствующее сообщение. Определить длину введенной строки и количество цифр в ней.

**ВАРИАНТ 25.**

Определить количество слов во введенном с клавиатуры тексте, определить количество слогов во введенном с клавиатуры слове.

**ВАРИАНТ 26.**

Из заданной строки получить новую строку, удалив все знаки препинания. Распечатать полученную новую строку.

**ВАРИАНТ 27.**

В заданной строке подсчитать количество всех символов, кроме «а», «b», «с», «d».

**ВАРИАНТ 28.**

На вход программе подаются сведения о номерах школ учащихся, участвующих в олимпиаде. Написать программу, которая будет выводить информацию, из какой школы было меньше всего участников (но из этой школы был хотя бы один участник).

Номер школы не более чем двузначный.

**Рекомендуемый вид экрана:**

*Введите количество участников олимпиады:****7***

*Ф.И. участника | № школы |*

*---------------------------------------------------*

***Иванов Петр 7***

***Петров Василий 67***

***Коробков Дмитрий 89***

***Дроздов Николай 17***

***Кузин Владимир 7***

***Степанов Кирилл 89***

***Алмазов Константин 67***

*Меньше всего участников было из школы № 17*

**Уровень 2**

1. Дан непустой текст из заглавных русских букв, за которым следует точка. Определить, упорядочены ли буквы по алфавиту.

2. Напечатать в алфавитном порядке все различные строчные русские буквы, входящие в заданный текст из 200 литер.

3. Известно, что в строке S находится не более 40 латинских букв. Напечатать эту строку, все вхождения "abc" заменив на "def".

4. Дана строка, состоящая из 10 слов. За последним словом точка. Напечатать эту же последовательность слов, но в обратном порядке.

5. Дана строка, состоящая из 10 слов. За последним словом точка. Напечатать эту же последовательность слов, но в алфавитном порядке.

6. Дана строка, состоящая из 10 слов. Напечатать те слова последовательности, у которых буквы упорядочены по алфавиту.

7. Дана строка, состоящая из 10 слов. Напечатать те слова последовательности, которые симметричны.

8. Дана последовательность, состоящая из 10 слов. Напечатать те слова последовательности, в которых первая буква слова входит в него ещё раз.

9. Дана последовательность, состоящая из 10 слов. Напечатать все слова последовательности, предварительно преобразовав каждое из них по следующему правилу: перенести первую букву в конец слова.

10.Дана последовательность, состоящая из 10 слов. Напечатать все слова последовательности, предварительно удалив из слов все последующие вхождения первой буквы.

11.Дана последовательность, состоящая из 10 слов. Напечатать все слова последовательности, удалив из каждого слова нечётные буквы, если слово имеет нечётное количество символов.

12.Дан текст из 60 символов. Напечатать этот текст, подчёркивая (ставя минусы в соответствующих позициях следующей строки) все входящие в него гласные.

13.Дана последовательность, состоящая из 10 слов. Напечатать все различные слова, указав для каждого из них число его вхождений в последовательность.

14.В строке S находится не более 80 латинских букв. Напечатать эту строку, удалив все вхождения th.

15.В строке S находится не более 80 латинских букв. Напечатать эту строку, добавляя после каждой буквы "q" букву "u".

16. Составить программу, заменяющую каждую встреченную в слове заданную букву на заданное сочетание букв (например, в слове ЭВМ заменить букву В сочетанием ТИ; получим слово ЭТИМ)

17. Составить программу, подсчитывающую, сколько раз в заданном слове встречается данное сочетание букв (например, в слове КУКУРУЗА сочетание КУ встречается 2 раза)

18. Составить программу, вычеркивающую из слова заданную букву. Если в слове такой буквы нет, то выдать соответствующее сообщение (например, из слова ДОЛИНА вычеркнуть букву О – получиться слово ДЛИНА).

19. Составить программу удаления из данного текста некоторой его части (например, из слова КЛУБОК вычеркиваем ЛУБ, получим КОК)

20. Составить программу, которая форматирует текст: если после знака препинания нет пробела, то программа добавляет его. Если предложение начинается со строчной буквы, то она заменяется на прописную.

21. Составить программу, заменяющую каждую встреченную в слове заданную букву на заданное сочетание букв (например, в слове ЭВМ заменить букву В сочетанием ТИ; получим слово ЭТИМ)

22.Дан текст, состоящий не менее чем из 5 слов. Напечатать все слова из текста, отличные от слова, введенного с клавиатуры.

23. Дано предложение, состоящее не менее чем из 5 слов. Напечатать текст, состоящий из последних букв всех слов предложения.

24.Дано предложение, состоящее не менее чем из 5 слов. Напечатать все слова из предложения, содержащие ровно две буквы "d".

25.Даны два предложения, состоящие не менее чем из 5 слов. Слова каждого предложения заменить на совпадающие с ними по длине слова из другого предложения (Заменять на соответствующие по счёту слова - например, второе слово первого предложения из 5 букв заменить на второе встретившееся слово из 5 букв во втором).

26.Даны два предложения. Напечатать те слова, которые есть и в первом предложении и во втором.

27.Дано предложение. Удалить из него пробелы и буквы "ъ", "ы", "ь". Напечатать.

28.Дано предложение. Перевернуть все его слова и вывести на экран.

29.Дана строка, содержащая 60 символов. После каждого пятого символа вставить символ пробела. Напечатать результат.

30.Дано предложение, состоящее из 10 слов. Напечатать все одинаковые слова, указав для каждого из них число его вхождений в предложение.

31.Дана последовательность, состоящая из 10 слов. Напечатать все слова последовательности, предварительно удалив из слов все последние буквы.

32.Даны две последовательности, состоящие из 5 слов. Составить из них одну строку включая в неё слова то из одной то из другой последовательности ставя между ними символ пробела. Вывести результат.

33.Дано предложение. В каждом слове удалить двойные буквы.

34.Дана строка содержащая русские и латинские буквы. Определить каких (русских или латинских) больше и насколько.

35.Дано предложение. Удалить из него те буквы, которые являются окончанием хотя бы одного слова.

36.Дана строка содержащая русские буквы. Произвести аллитерацию. Т.е. заменить все русские буквы на созвучные с ними латинские. Вывести исходную строку и результат.

37.По аналогии с заданием 32 заменить латинские буквы русскими.

38.Дано предложение, содержащее русские и латинские буквы. Удалить из него латинские буквы.

39.Дана строка символов. Удалить из неё все повторно встречающиеся символы.

40.Напечатать в алфавитном порядке все различные строчные латинские буквы, входящие в заданный текст из 100 символов.

41.Дана строка, состоящая из 9 слов. За последним словом точка. Напечатать последовательность тех же слов, но в порядке, обратном алфавитному.

42.Дана последовательность, состоящая из 10 слов. Перевернуть те слова последовательности, в которых последняя буква слова входит в него ещё раз. Вывести результат.

43.Дана последовательность, состоящая из 10 слов. Напечатать все слова последовательности, предварительно преобразовав каждое из них по следующему правилу: поменять местами первую и последнюю букву слова.

44.Дана последовательность, состоящая из 10 слов. Напечатать все слова последовательности, удалив из неё те слова, в которых есть двойные буквы.

45.Дана последовательность, состоящая из 10 слов. Напечатать те слова последовательности, которые имеют нечётное количество символов.

46.Дан текст из 60 символов. Напечатать этот текст, удалив все входящие в него гласные.

43.Дана последовательность, состоящая из 10 слов. Напечатать все повторяющиеся слова, указав для каждого из них число его вхождений в последовательность.

44.В строке S находится не более 70 латинских букв. Удалить из неё строчные буквы. Вывести результат.

45.В строке S находится не более 80 латинских букв. Напечатать эту строку, добавляя перед каждой буквой "x" букву "z".

46.Дан текст, состоящий не менее чем из 5 слов. Напечатать все слова из текста, содержащие только латинские буквы.

47.Дано предложение, состоящее не менее чем из 5 слов. Напечатать текст, состоящий из первых букв всех слов предложения.

48.Дано предложение, состоящее не менее чем из 5 слов. Напечатать все слова из предложения, содержащие ровно одну букву "s".

49.Даны два предложения. Напечатать те слова, которые есть в первом предложении, но нет во втором.

50.Дано предложение. Удалить из него пробелы и буквы "э", "ю", "я". Выдать результат.

51.Даны две строки. Напечатать те символы, которые есть в первой строке, но нет во второй.

52.Даны две строки. Напечатать в алфавитном порядке чётные символы этой строки.

53.Дана последовательность, состоящая из 10 слов. Напечатать все слова последовательности, предварительно удалив из слов все первые буквы.

54.Даны две последовательности, состоящие из 5 слов. Составить из них текст включая в неё слова, из первой и второй последовательности, предварительно удалив в каждом слове первой последовательности первые буквы, а во второй вторые. Вывести результат.

55.Дано предложение. В каждом нечётном слове удалить латинские буквы, в каждом чётном русские.

56.Дана строка содержащая русские и латинские буквы. Определить каких (русских или латинских) больше и если больше латинских удалить русские, если больше русских удалить латинские.

57.Дано предложение. Удалить из него те буквы, которые являются вторыми по счёту хотя бы одном слове предложения.

58.Дана строка содержащая русские и латинские буквы. Заменить все русские буквы на созвучные с ними латинские, если русских больше или наоборот, если больше латинских. Вывести исходную строку и результат.

59.Дано предложение. Удалить из каждого слова все чётные буквы и вывести результат.

60.Дана последовательность из 10 слов. Удалить из неё слова, которые содержат и русские, и латинские буквы.

**Практическая работа по теме «Указатели»**

**Цель**: закрепить знания и умения по решению задач с использованием указателей.

**Ход выполнения работы**

1. Изучить теоретические основы работы с указателями.
2. Выполнить задание из примера.
3. Оформить отчет в тетради для практических работ.
4. Ответить на контрольные вопросы.

**Пример выполнения программы**

Реализуйте указанный код, запустите программу, произведите трассировку, указывая в трассировочной таблице результат работы программы и результат ручной трассировки. В чем заключается смысл программы? Напишите комментарии к строкам операций над динамической памятью.

Выполните структуризацию программы, оформите ее в тетради для практических работ.

**Код программы:**

# include <iostream.h>

int main (void)

{

float x= 12.3, у;

float \*p;

р=&х;

у=\*p;

cout << "x=" << x << " y=" << y << endl;

 (\*р)++;

cout << "x=" << x << " y=" << y << endl;

у=1+(\*р)\*у;

cout << "x=" << x << " y=" << y << endl;

}

**Контрольные вопросы**

1. Какие операции применимы к динамической памяти?
2. Как происходит задание переменных в динамической памяти?
3. Какие операции работы с динамической памятью вам известны?

**Задания:**

* 1. Дан указатель: double \*\*p = 0; Выполните следующие задания (решения можно оформлять внутри функции main):
	\* создайте конструкцию, изображенную на рисунке;
	\* выведите число, указанное в квадратике, на экран;
	\* после этого удалите все динамические объекты.



Опять же – ищем самое короткое решение. Автор: [Olej](https://purecodecpp.com/archives/author/olej)

2. Объявите указатель на массив типа double и предложите пользователю выбрать его размер. Далее напишите четыре [функции](https://purecodecpp.com/archives/1078): первая должна выделить память для массива, вторая  – заполнить ячейки данными, третья – показать данные на экран, четвертая – освободить занимаемую память.  Программа должна предлагать пользователю продолжать работу (создавать новые динамические массивы ) или выйти из программы.

3. Объявите указатель на массив типа int и выделите память память для 12-ти элементов. Необходимо написать функцию, которая поменяет значения четных и нечетных ячеек массива. Например есть массив из 4-х элементов:





4.  Объявить и заполнить двумерный динамический массив случайными числами от 10 до 50. Показать его на экран. Для заполнения и показа на экран написать отдельные функции. (подсказка: функции должны принимать три параметра – указатель на динамический массив, количество строк, количество столбцов).  Количество строк и столбцов выбирает пользователь.

**Практическая работа по теме «Объектно-ориентированное программирование»**

**Часть 1 (до 5 баллов).** Реализуйте класс «Линейный однонаправленный список», как показано в заготовке, поясните принцип организации полей и методов в структуре «Узел», классе «Линейный список». В каком доступе находятся поля и методы структуры TNode? Как называется класс/структура, содержащие в качестве поля ссылку или указатель на самого себя? Какие методы класса TList могут быть сделаны кон- стантными? Дополните код класса TList деструктором, адекватно удаляющим список. Продемонстрируйте его работу. Вынесите реализацию всех методов класса за пределы класса. Убедитесь в работоспособности класса на примере приведенного кода в теле main().

**#include** <iostream>

**using namespace std;**

**struct** TNode //структура для описания узла в списке

{

**int** info; TNode \*next; TNode(int x){

info=x; next=0;}

};

**class** TList // сам класс «Линейный список»

{

**private:**

TNode \* head; //голова списка TNode \* tail; //хвост списка

**public:**

TList()

{

//ваш код

}

**bool** addToTail(int x){//добавляем элемент в конец списка

//ваш код

}

**bool** isEmpty(){ // пуст ли список

//ваш код

}

**void** print() // вывод списка на печать

{

//ваш код

}

**bool** delFromEnd()

{

//ваш код

}

**bool** addAfterNode(int valueToSearch, int valueToAdd)

//добавление после заданного узла

{

//ваш код

}

**bool** delNode(**int** valueToDelete) //удаление заданного узла

{

//ваш код

}

};

**int** main()

{

TList myList;

**if** (myList.isEmpty())

**for** (int i=0; i<5; i++) myList.addToTail(i);

**if** (!myList.isEmpty()) myList.print(); myList.addAfterNode(0,-1000);

**if** (!myList.isEmpty()) myList.print(); myList.addAfterNode(2,-222);

**if** (!myList.isEmpty()) myList.print(); myList.addAfterNode(4,-444);

**if** (!myList.isEmpty()) myList.print(); myList.delNode(-222);

**if** (!myList.isEmpty()) myList.print();

**while** (!myList.isEmpty())

{

myList.delFromEnd();

myList.print();

} **return** 0;

}

**Часть 2 (до 5 баллов).** В данной лабораторной работе все описания классов должны быть вынесены в отдельный заголовочный файл. Реализация функций-элементов класса должна быть написана в отдельном модуле, а основную программу, иллюстрирующую применение всех методов вашего класса, следует реали- зовать еще одним модулем. Во всех заданиях предусмотрите конструкторы с аргументами по умолчанию, а также дружественную перегруженную операцию вывода в поток и чтения из потока. Помните, что в каж- дом классе должны быть предусмотрены константные функции get.

## Варианты

1. Создайте класс с именем Complex для выполнения арифметических действий с комплексными числами. Напишите программу для проверки вашего класса. Комплексные числа должны быть представлены в форме RePart+ImPart\*i, где i\*i = –1. Используйте переменные с плавающей точкой для представления закрытых данных класса.

Создайте конструктор, деструктор и открытые функции-элементы для следующих действий:

* 1. сложение 2 комплексных чисел;
	2. вычитание 2 комплексных чисел;
	3. умножение 2 комплексных чисел.

Перегрузите операцию вывода в поток для печати комплексного числа в виде |A|exp{i\*z}.

1. Cоздайте класс c именем Rational для выполнения арифметических действий с дробями. Напишите программу для проверки вашего класса. Используйте целые переменные для представления закры- тых данных класса – числителя и знаменателя. Создайте функцию-конструктор класса, которая по- зволяет объекту этого класса принимать начальные значения при его объявлении. Конструктор должен содержать значения по умолчанию на случай отсутствия заданных начальных и должен хранить дроби в сокращенном виде. Создайте открытые функции-элементы для случаев:
	1. сложение 2 дробей (здесь и далее результат должен храниться в сокращенном виде);
	2. вычитание 2 дробей;
	3. перемножение 2 дробей;
	4. деление 2 дробей;
	5. печать дроби в десятичном виде.

Перегрузите операцию вывода в поток для печати дроби в виде а/b.

1. То же, что и в варианте 2, но для дроби, числитель и знаменатель которой – комплексные числа. Перегруженная операция печати должна выглядеть следующим образом: на экран должна выво- диться дробь в виде (ReA+i\*ImA)/(ReB+i\*ImB).
2. Модифицируйте класс myTime (о котором говорилось на лекциях) так, чтобы включить функцию- элемент tick, которая дает приращение времени, хранящегося в объекте myTime, равное одной се- кунде. Объект Time должен всегда находиться в непротиворечивом состоянии. Напишите про- грамму для проверки функции-элемента tick в цикле, которая печатала бы время в каком-либо стандартном формате на каждой итерации цикла и иллюстрировала правильную работу функции- элемента tick. Удостоверьтесь в правильности работы в следующих случаях:
	1. приращение с переходом в следующую минуту;
	2. приращение с переходом в другой час;
	3. приращение с переходом в другой день.

Предусмотрите перегруженную операцию вывода в поток, а также перегруженную операцию ин- кремента в префиксной и постфиксной формах.

1. Модифицируйте класс myDate (о котором говорилось в лекциях) так, чтобы в нем присутствовала функция-элемент nextDay, которая будет увеличивать дату на 1 день. Объект myDate должен все- гда находиться в непротиворечивом состоянии. Напишите программу, проверяющую функцию nextDay в цикле и печатающую время в стандартном формате на каждой итерации цикла. Проверь- те правильность работы функции в следующих случаях:
	1. приращение с переходом в следующий месяц;
	2. приращение с переходом в следующий год.

Предусмотрите перегруженную операцию вывода в поток, а также перегруженную операцию ин- кремента в префиксной и постфиксной формах.

1. Создайте класс параллелограмм, который хранит только декартовы координаты его четырех углов. Конструктор вызывает набор функций, которые принимают 4 группы координат и проверяют, что- бы каждая из координат х и у находилась в первом квадранте в диапазоне от 0,0 до 50,0. Это мно- жество функций также должно проверять, что переданные координаты определяют параллело- грамм. Должны быть предусмотрены функции-элементы, вычисляющие длины сторон параллело- грамма, периметр и площадь. Включите функцию, которая определяла бы, не является ли паралле- лограмм прямоугольником.

Перегрузите операцию вывода в поток для печати всей информации об объекте.

1. Создайте класс Rectangle (прямоугольник). Класс имеет атрибуты length (длина) и width (ширина), каждый из которых по умолчанию равен 1. Он имеет функции-элементы, которые вычисляют пе- риметр и площадь прямоугольника. Он имеет функции записи и чтения для длины и ширины. Функции записи должны проверять, что длина и ширина – числа с плавающей запятой, находя- щиеся в пределах от 0,0 до 20,0.

Перегрузите операцию вывод в поток для печати всей информации о прямоугольнике.

1. Создайте класс прямоугольник, который хранит только декартовы координаты четырех углов пря- моугольника. Конструктор вызывает набор функций, которые принимают 4 группы координат и проверяют, чтобы каждая из координат х и у находилась в первом квадранте в диапазоне от 0,0 до 20,0. Это множество функций также должно проверять, что переданные координаты определяют прямоугольник. Должны быть предусмотрены функции-элементы, вычисляющие длину и ширину прямоугольника, периметр и площадь. Включите функцию, которая определяла бы, не является ли прямоугольник квадратом.

Перегрузите операцию вывод в поток для печати всей информации о прямоугольнике.

1. Модифицируйте класс прямоугольник из варианта №8 так, чтобы включить в него функцию draw, которая изображает прямоугольник внутри окна 25 на 25. Включите функцию setFillCharacter, что- бы задавать символ, которым будет заполняться прямоугольник внутри. Включите функцию setPerimeterCharacter, чтобы задавать символ, которым будут печататься границы прямоугольника. Включите функцию поворота прямоугольника на 90 градусов вокруг одной из его вершин против и по часовой стрелке. Перегрузите операцию вывод в поток для печати всей информации о прямоугольнике и рисования прямоугольника.
2. Создайте класс треугольник, хранящий только декартовы координаты вершин. Конструктор вызы- вает набор функций, которые принимают 3 группы координат и проверяют, чтобы каждая из коор- динат х и у находилась в первом квадранте в диапазоне от 0,0 до 50,0. Функции также должны проверять, чтобы треугольник не «схлопывался» в прямую линию. Должны быть предусмотрены функции-элементы, вычисляющие длину сторон, периметр и площадь треугольника. Включите функцию, которая определяла бы, не является ли треугольник равнобедренным, равносторонним или прямоугольным.

Перегрузите операцию вывода в поток для печати всей информации о треугольнике.

1. Создайте класс треугольник, хранящий длины двух сторон и значение угла между ними. Должны быть предусмотрены функции-элементы, вычисляющие длину третьей стороны, значения 2 остав- шихся углов, периметр и площадь треугольника. Включите функцию, которая определяла бы, не является ли треугольник равнобедренным, равносторонним или прямоугольным.

Перегрузите операцию вывода в поток для печати всей информации о треугольнике.

1. Создайте класс прямая призма, хранящий только декартовы координаты вершин основания и вы- соту призмы. Конструктор вызывает набор функций, которые принимают группы координат и про- веряют, чтобы каждая из координат х и у находилась в первом квадранте в диапазоне от 0,0 до 250,0. Должны быть предусмотрены функции-элементы, вычисляющие длину ребер, периметр и площадь основания, а также площадь боковой поверхности, площадь поверхности и объем призмы. Перегрузите операцию вывода в поток так, чтобы она печатала, какая фигура лежит в основании, и ее основные характеристики.
2. Создайте класс пирамида, хранящий только декартовы координаты вершин основания и вершины пирамиды. Конструктор вызывает набор функций, которые принимают группы координаты. Долж- ны быть предусмотрены функции-элементы, вычисляющие длину ребер, периметр и площадь ос- нования, а также площадь боковой поверхности, площадь поверхности и объем пирамиды. Перегрузите операцию вывода в поток так, чтобы она печатала, какая фигура лежит в основании, и ее основные характеристики.
3. Создайте класс конус, хранящий только декартовы координаты центра основания, радиус основа- ния и высоту конуса. Должны быть предусмотрены функции-элементы, рассчитывающие периметр и площадь основания, а также площадь боковой поверхности, площадь поверхности и объем кону- са.

Перегрузите операцию вывода в поток для печати всей информации об объекте.

1. Создайте класс усеченный конус, хранящий только декартовы координаты центра основания, ра- диусы оснований и высоту конуса. Должны быть предусмотрены функции-элементы, рассчиты- вающие периметр и площадь оснований, а также площадь боковой поверхности, площадь поверх- ности и объем конуса. Перегрузите операцию вывода в поток для печати всей информации об объекте.
2. Создайте класс усеченная пирамида, хранящий только декартовы координаты вершин оснований. Конструктор вызывает набор функций, которые принимают группы координат одного основания. Высота пирамиды задается случайным образом, координаты второго основания вычисляются в со- ответствии с высотой. Должны быть предусмотрены функции-элементы, вычисляющие периметр иплощадь основания, а также площадь боковой поверхности, площадь поверхности и объем пира- миды.

Перегрузите операцию вывода в поток для печати всей информации об объекте (какая фигура ле- жит в основании, ее основные характеристики).

1. Создайте класс "Спортсмен", в котором будут храниться данные о ФИО атлета, представляемой стране, виде спорта, установленных рекордах, завоеванных местах на первенствах страны, конти- нента, мира и олимпиадные достижения. Предусмотрите методы "Установить рекорд", "Получить медаль" (помните, что медали бывают разные...). Предусмотрите также все необходимые методы установки и чтения данных-элементов.

Перегрузите операцию вывода в поток для вывода информации о спортсмене.

1. Создайте класс «Запись в адресной книге». В нем хранятся фамилия и имя человека, номера теле- фонов (нескольких, в т.ч. домашних, рабочих и сотовых), район и адрес проживания, e-mail (не- сколько). Конструктор должен вызывать функцию, считывающую эти данные из текстового файла. Напишите функции-элементы для установки и чтения данных. Предусмотрите метод поиска номе- ров сотовых телефонов, метод формирования текстового файла с заголовком и подписью (как заго- товки письма на электронный адрес).

Перегрузите операцию вывода в поток для печати информации об объекте вида «Запись в адресной книге».

1. Создайте класс матрица размерностью *n*  *n* , который хранит только размерность матрицы и мак- симальное по модулю значение элемента матрицы (и указатель на целое). Конструктор должен вы- зывать функцию заполнения матрицы случайными числами в заданном диапазоне. Напишите функции-элементы для:
	1. транспонирования матрицы;
	2. умножения матрицы на число;
	3. сложения матриц;
	4. умножения двух матриц.

Перегрузите операцию вывода в поток для печати всей информации об объекте. Перегрузите также оператор вычитания матриц.

1. Создайте класс правильный многоугольник, который хранит число вершин и их координаты. Кон- структор вызывает набор функций, которые проверяют, чтобы число вершин было не менее 3, что- бы многоугольник был правильным, и в случае ошибки устанавливают значения всех вершин в 0. Должны быть предусмотрены функции-элементы, вычисляющие периметр, площадь многоуголь- ника. Должна быть предусмотрена функция-элемент вывода на печать информации о числе сторон многоугольника. Напишите программу-драйвер, иллюстрирующую применение вашего класса. Перегрузите операцию вывода в поток для печати всей информации об объекте.