**Задание 1.**

В первое задание входят следующие этапы.

1. Модернизация программы решения квадратного уравнения с использованием вызова следующих подпрограмм: подпрограммы собственно решения квадратного уравнения, подпрограммы расчета невязок корней этого уравнения, подпрограммы расчета коэффициентов уравнения по заданным вами значениям корней этого уравнения.

Что такое подпрограмма? Почти тоже, что и программа, но начинается она не с заголовка

**PROGRAM** имя программы, а с заголока

**SUBROUTINE** имя подпрограммы (b1,b2,…),

где b1,b2,…- список **формальных** параметров, а далее идет тело подпрограммы из последовательности любых операторов, как и в

обычной, так называемой головной программе. Последний оператор

подпрограммы – это оператор **END**, перед которым лучше поставить

оператор **RETURN** ,чтобы было ясно, что это закрывается подпрограмма. Главным отличием является то, что в проекте может быть только одна программа и запускается на выполнение из проекта именно она, а подпрограмм может быть любое число, но запускаются (вызываются) они на выполнение операторами

**CALL**  имя подпрограммы (a1,a2,…),

где **CALL** – служебное слово,

a1,a2,… - список **фактических** параметров.

Эти операторы (операторы вызова подпрограмм) располагаются в любом месте среди выполняемых операторов, как головной программы, так и любой другой подпрограммы. После выполнения подпрограммы будет выполнен оператор, следующий за оператором вызова (**CALL**) этой подпрограммы (находящийся в другой подпрограмме или головной программе). Завершение работы проекта, таким образом, может произойти только в основной (головной) программе.

**Следует запомнить**, что имена переменных в данной подпрограмме не доступны в других подпрограммах и в головной программе!!! Это означает, что даже если какие -то из имен a1,a2,… (или даже все) совпадают по написанию с именами b1,b2,… то они всё равно будут считаться разными и в разных подпрограммах и в головной программе будут принимать разные значения. Передача информации из головной программы в подпрограммы и обратно может происходить

**только (**заисключениемспециально предусмотренных случаев) через параметры следующим образом. Значения фактических параметров передаются переменным, обозначенным как формальные параметры и наоборот только при выполнении оператора **CALL** !!! Запомните, что соответствие между фактическими и формальными параметрами определяется **только порядком их перечисления в списке параметров**. Поэтому описание свойств параметра *b*, стоящего на *n*-м месте в списке формальных параметров подпрограммы, должно совпадать с описанием свойств параметра *a*, стоящего на *n*-м месте в списке фактических параметров основной программы или другой подпрограммы.

Подключаться к проекту подпрограммы могут либо через добавление отдельных файлов, на которых находятся исходные тексты подпрограмм в произвольном порядке, либо нахождением их на файле с исходным текстом основной программы (порядок расположения исходных текстов основной программы и подпрограмм произволен).

Ниже приведен пример исходного файла для некоторого проекта, содержащий программу PR1 и подпрограмму p1**. Создайте новый проект с произвольным именем и подключите к нему этот исходный файл. Запустите проект на выполнение и внимательно проанализируйте результаты работы этой программы. После чего снимите знак комментария (!) в подпрограмме и вновь запустите программу на выполнение! Объясните ошибку при выполнении!**

**subroutine p1(a1,b1)**

**real a1**

**a=4.4; print**\*,'a1,b1 =',a1,b1; b1=b1+a1

**print**\*,'b1=',b1; **read**\*

! a1=5.5; read\*

**end**

**PROGRAM PR1**

**a=1.5; b=2.2; b1=3.3; call p1(1.7,b)**

**print**\*,'a,b= ',a,b; **read**\*

**end**

**Теперь преобразуйте свою программу решения квадратного уравнения следующим образом.**

В головной программе описать действительные переменные *a,b,c,x1,x2* и целую переменную *ip* (можно и по правилу умолчания). Запросить ввод переменных *a,b,c*. Подпрограмма решения квадратного уравнения должна иметь следующий заголовок

**SUBROUTINE** *kvad* (*a,b,c,ip,x1,x2*).

В тексте подпрограммы необходимо переменной *ip* присваивать значение 1 при действительных корнях, а переменным *x1,x2* присваивать значения этих корней. Если же корни комплексно сопряженные то переменной *ip* присваивается значение 2, а переменные *x1,x2* принимают значения их действительной и мнимой частей. Оператор вызова этой подпрограммы должен быть таким

***CALL*** *kvad* (*a,b,c,ip,x1,x2*).

Аналогичным образом надо оформить подпрограммы расчета невязок корней этого уравнения и подпрограммы расчета коэффициентов уравнения по заданным вами значениям корней этого уравнения. То же относится к операторам вызова этих подпрограмм.

1. Для вызова подпрограмм необходимо организовать выдачу на экран меню вида

**МЕНЮ**

1. Вызов подпрограммы решения квадратного уравнения
2. Вызов подпрограммы расчета невязок
3. Вызов подпрограммы расчета коэффициентов
4. Конец

Введите номер пункта меню, который вы хотите выполнить.

1. Для реализации работы с меню необходимо освоить конструкцию оператора множественного выбора **SELECT CASE** ,закрывающегося оператором **END SELECT** , кроме того необходимо изучить и использовать форматный оператор печати (вывода на экран).
2. Для организации многовариантного расчета без выхода из программы

необходимо освоить конструкцию простейшего оператора **DO** , закрывающегося оператором **END DO**.

1. Решение кубического уравнения вида

*ax3 + bx2 + cx + d* = 0 (1)

В головной программе описать действительные переменные *a,b,c,d,x1,x2* и целую переменную *ip* (можно и по правилу умолчания). Запросить ввод переменных *a,b,c,d*.

**При *a=*0** вызвать подпрограмму решения квадратного уравнения с заголовком:

**SUBROUTINE** *kvad* (*a,b,c,ip,x1,x2*),

в которой переменной *ip* присваивается значение 1 при действительных корнях, а переменные *x1,x2* принимают значения этих корней. Если же корни комплексно сопряженные то переменной *ip* присваивается значение 2, а переменные *x1,x2* принимают значения их действительной и мнимой частей. Далее вызвать эту подпрограмму следующим оператором

***CALL*** *kvad* (*b,c,d,ip,x1,x2*).

**При значении *a* не равном нулю** заменой переменных: *x = y – b/3a*  уравнение (1) сводится к виду:

*y3 + 3py + 2q = 0* ,(2)

где *p =*(*3ac – b2*)*/9a2*; *q =* (*2b3/27a3 – bc/3a2 + d/a*)/*2.*

Уравнение (2) при *q = 0* имеет корень: *y1 = 0,* а два других корня находятся оператором : ***CALL*** *kvad* (*1,0,3\*p,ip,y2,y3*), после чего необходимо вернуться к исходным переменным (*x1,x2,x3*) в соответствии с заменой: *x = y – b/3a* .

Уравнение (2) при *p = 0* имеет корень: *y1 =* *-* и два других корня:

*y2 =- y1*( *1+ i* )/2; *y3 =- y1*( *1- i* )/2. Напоминаем: *x = y – b/3a* .

В случае *p = q = 0* уравнение (1) имеет три равных корня : *x1=x2=x3=– b/3a.* В остальных случаях решение зависит от знаков величин *p* и *q* , а также от знака величины *D = q2 + p3* и определяется нижеприведенной таблицей, где величина *r* определяется формулой:

*r =* (*q/*

**Таблица**

|  |  |  |
| --- | --- | --- |
| *p* < 0 | | *p* > 0 |
| *D ≤ 0* | *D* > 0 |
| *cosφ = q/r3* | *chφ = q/r3* | *shφ = q/r3* |
| *y1= -2r* cos(φ*/3*) | *y1= -2r*ch(φ*/3*) | *y1= -2r*sh(φ*/3*) |
| *y2 = 2r* cos((φ)*/3*) | *y2= r*ch(φ*/3*)+i *r*sh(φ*/3*) | *y2= r*sh(φ*/3*)+i *r*ch(φ*/3*) |
| *y3 = 2r* cos((φ)*/3*) | *y3= r*ch(φ*/3*) *-*i *r*sh(φ*/3*) | *y3= r*sh(φ*/3*) *-*i *r*ch(φ*/3*) |