### **Семинар Модульное программирование 3.Средства создания универсальных подпрограмм. Указатель на функцию**

При проектировании сложной программы разработчик стремится создать как можно более универсальные подпрограммы, чтобы сократить их количество и общий объем программы. Для этого С/С++ представляют определенные возможности. Такие как применение:

1. Параметров сложных структурных типов.
2. Указателя на функцию (параметров-функций).

К параметрам сложных структурных типов относятся:

* Параметры-многомерные массивы.
* Параметры-строки.
* Параметры структуры.

#### Применение указателя на функцию как параметра.

Как уже отмечалось ранее, функция характеризуется типом возвращаемого значения, именем и сигнатурой.

Сигнатура определяется количеством, порядком следования и типами параметров.

При использовании имени функции без последующих скобок и параметров, имя функции выступает в качестве указателя на эту функцию, и его значением служит адрес размещения функции в памяти. То есть, имя функции – это указатель, хранящий ее адрес, который может быть присвоен другому указателю.

Однако при объявлении для нового указателя должен быть задан тот же тип, что и возвращаемое функцией значение, и тот же список параметров с точностью до типов формальных параметров (имена параметров могут различаться).

Указатель на функцию объявляется следующим образом:

**<Тип функции>(\* <Имя>)(<Спецификация параметров>);**

Например:

**int (\*ptrfun)(int, int);**

При объявлении указатель на функцию может быть инициализирован, но в качестве значения должен быть указан адрес функции, тип и сигнатура которой соответствуют определяемому указателю. При присваивании указателей на функции тоже надо следить за соответствием типов возвращаемых значений и сигнатур правой и левой частей операции присваивания.

Например:

*Описание функций:*

**char f1(char){…}**

**char f2(int){…}**

**void f3(float){…}**

**int f4(float){…}**

**int f5(int){…}**

*Объявление указателей на функции:*

void (\*ptr1)(float) = f3; // инициализированныйуказатель

int (\*ptr2)(int);

char (\*ptr3)(int);

*Присваивание указателей:*

ptr2 = f5; ptr3 = f2;// корректное

prt2 = f4; ptr3 = f1; // некорректное: несоответствие типов или спецификаций.

**Задание 1.** Написать программу нахождения корня функции одной переменной вещественного типа (y=f(x)) на заданном отрезке с заданной точностью методом половинного деления.

Использовать подпрограммы и указатель на функцию.

Для решения задачи используем материал семинара 2. Модифицируем подпрограмму нахождения корней root, добавив в качестве параметра в заголовке указатель на функцию:

**float (\*funuk)(float).**

В основной программе определим рабочий указатель на функцию того же типа, для организации обработки нескольких функций:

**float (\*ptr)(float).**

Программу проверим на нахождении корней трех функций

Y= X^2-1 на отрезке -2 0, или 0 2

Y=X-3 на отрезке 1 4

Y=X^2\*sin(x)-x+1 на отрезке -4 4

Отрезки и точность будем вводить с клавиатуры.

**#include <stdio.h>**

**#include <math.h>**

**// Список функций**

**float F1(float x)**

**{ return x\*x\*sin(x)-x+1;}**

**float F2(float x)**

**{ return x\*x-1;**

**}**

**float F3(float x)**

**{ return x-3;**

**}**

**// Функция нахождения корня на отрезке a – b с точностью eps**

**float root(float (\*funuk)(float),float a,float b, float eps)**

**{ float fx,fa,fb,x;**

**x=(a+b)/2;**

**fx=funuk(x);**

**while(fabs(fx)>=eps)**

**{ fa=funuk(a);**

**fb=funuk(b);**

**if (fx\*fa<0)**

**{ fb=fx;**

**b=x;}**

**else**

**{fa=fx;**

**a=x;}**

**x=(a+b)/2;**

**fx=funuk(x);**

**}**

**return x;**

**}**

**float (\*ptr)(float);**

**// Основная программа**

**int main()**

**{ float xn,xk,eps;**

**puts("Input Xn,Xk,eps for y=x^2\*sin(x)-x+1");**

**scanf("%f %f %f",&xn,&xk,&eps);**

**ptr=F1; // функция F1= x^2\*sin(x)-x+1**

**if (ptr(xn)\*ptr(xk)<0)**

**printf("Root F1 on %7.3f - %7.3f raven %8.6f\n",xn,xk,root(*F1*,xn,xk,eps));**

**else printf("Root F1 on %7.3f - %7.3f is epsent\n",xn,xk);**

**puts("Input Xn,Xk,eps for y=x^2-1");**

**scanf("%f %f %f",&xn,&xk,&eps);**

**ptr=F2; // функция F2= x^2-1**

**if (ptr(xn)\*ptr(xk)<0)**

**printf("Root F1 on %7.3f - %7.3f raven %8.6f\n",xn,xk,root(*F2*,xn,xk,eps));**

**else printf("Root F1 on %7.3f - %7.3f is epsent\n",xn,xk);**

**puts("Input Xn,Xk,eps for y=x-3");**

**scanf("%f %f %f",&xn,&xk,&eps);**

**ptr=F3; // функция F3= x-3**

**if (ptr(xn)\*ptr(xk)<0)**

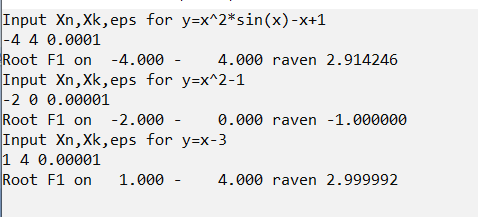
**printf("Root F1 on %7.3f - %7.3f raven %8.6f\n",xn,xk,root(*F3*,xn,xk,eps));**

**else printf("Root F1 on %7.3f - %7.3f is epsent\n",xn,xk);**

**return 0;**

**}**

**Пример работы программы**



**Задание 2.** Написать программу табуляции любой функции одной переменной на заданном отрезке с заданным шагом. Выбор функции оформить в виде модулей.

**Модуль – Список функций**

**#ifndef SPISFUN\_H**

**#define SPISFUN\_H**

**#include <math.h>**

**float F1(float x)**

**{ return x\*x\*sin(x)-x+1;}**

**float F2(float x)**

**{ return x\*x-1;**

**}**

**float F3(float x)**

**{ return x-3;**

**}**

**float Fsin(float x)**

**{ return sin(x);**

**}**

**float Fcos(float x)**

**{ return cos(x);**

**}**

**float Fexp(float x)**

**{ return exp(x);**

**}**

**#endif // SPISFUN\_H**

**Модуль Табуляции**

**#ifndef MODTABUL\_H**

**#define MODTABUL\_H**

**#include <stdio.h>**

**#include <stdlib.h>**

**void TablicaF(float (\*funuk)(float),float a,float b, float h,char s[])**

**{ float fx,x;**

**x=a;**

**system("cls");**

**puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");**

**puts("| Tablica means function |\n");**

**printf("| %16s |\n",s);**

**printf("| on diapazon %5.2f -%5.2f step is %5.3f |\n",a,b,h);**

**puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");**

**puts("| X | Y |");**

**puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");**

**while(x<=b)**

**{ fx=funuk(x);**

**printf("| %6.2f | %8.4f |\n",x,fx);**

**puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");**

**x=x+h;**

**}**

**puts("\n");**

**puts("\n");**

**}**

**#endif // MODTABUL\_H**

**Основная программа**

**#include <stdio.h>**

**#include <math.h>**

**#include <locale.h>**

**#include <stdlib.h>**

**#include "SpisFun.h"**

**#include "ModTabul.h"**

**float (\*ptr)(float);**

**int main()**

**{ float xn,xk,h;**

**setlocale(0,"russian");**

**int kod,key=1;**

**while(key==1)**

**{ system("cls");**

**puts(" Selected function for tabulation");**

**puts("1 - y=sin(x)");**

**puts("2 - y=cos(x)");**

**puts("3 - y=exp(x)");**

**puts("4 - y=x^2\*sin(x)-x+1");**

**puts("5 - y=y=x^2-1");**

**puts("6 - y=x-3");**

**puts("7 - Exit");**

**scanf("%d",&kod);**

**switch(kod)**

**{ case 1: { ptr=Fsin;**

**puts("Input Xn,Xk,h for y=sin(x)");**

**scanf("%f %f %f",&xn,&xk,&h);**

**TablicaF(*ptr*,xn,xk,h,"y=sin(x)");break;}**

**case 2: {ptr=Fcos;**

**puts("Input Xn,Xk,h for y=cos(x)");**

**scanf("%f %f %f",&xn,&xk,&h);**

**TablicaF(*ptr*,xn,xk,h,"y=cos(x)");break;}**

**case 3: {ptr=Fexp;**

**puts("Input Xn,Xk,h for y=exp(x)");**

**scanf("%f %f %f",&xn,&xk,&h);**

**TablicaF(*ptr*,xn,xk,h,"y=exp(x)"); break;}**

**case 4: { ptr=F1; ptr=F1;**

**puts("Input Xn,Xk,h for y=x^2\*sin(x)-x+1");**

**scanf("%f %f %f",&xn,&xk,&h);**

**TablicaF(*ptr*,xn,xk,h,"y=x^2\*sin(x)-x+1");break;}**

**case 5: { ptr=F2;**

**puts("Input Xn,Xk,h for y=x^2-1");**

**scanf("%f %f %f",&xn,&xk,&h);**

**TablicaF(*ptr*,xn,xk,h,"y=x^2-1");break;}**

**case 6: { ptr=F3;**

**puts("Input Xn,Xk,h for y=x-3");**

**scanf("%f %f %f",&xn,&xk,&h);**

**TablicaF(*ptr*,xn,xk,h,"y=x-3");break;}**

**case 7: key=0; break;**

**default: key=0;**

**}**

**system("pause");**

**}**

**Пример работы программы**

