

ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ

АХБОРОТ ТЕХНОЛОГИЯЛАРИ ФАКУЛЬТЕТИ

“Информатика” кафедраси

ИНФОРМАТИКА ва АТ ФАНИДАН

КУРС ИШИ

БАЖАРДИ: _____

КАБУЛ КИЛДИ: _____

ТОШКЕНТ 2013

Р Е Ж А :

- ❖ Кириш
- ❖ Назарий кисм
- ❖ Масаланинг куйилиши
- ❖ Масаланинг С / С++ тилидаги коди
- ❖ Дастурнинг ДОС ойнасидаги натижаси
- ❖ Хулоса
- ❖ Фойдаланилган адабиётлар руйхати

КИРИШ

Информатика термини лотинча *informatic* сўзидан келиб чиққан бўлиб, тушунтириш, хабар қилиш, баён этиш маъноларини англатади.

Ўзбекистон Республикаси информатика ва ҳисоблаш техникаси йўналишида жажқон даражасидаги илмий мактаблар яратгани, уларда тадқиқотлар муваффақиятли олиб борилаётганлиги билан шартли равишда фахрлана олади.

”Математика фанининг эҳтимоллар назарияси ва математик статистика, дифференциал тенгламалар ва математик физика, функционал таҳлил соҳасидаги ютуқлари республикадан анча узоқда ҳам машқур” деб ёзади Ўзбекистон Республикаси Призеденти И. А. Каримов.

Информатиканинг инсон фаолияти мустақил соҳаси сифатида ажралиб чиқиши биринчи навбатда компьютер техникаси ривожланиши билан боғлиқ. Бунда асосий хизмат микропроцессор техникасига тўғри келади.

Информатика ахборотни қайта ишлаш, уларни қўллаш ва ижтимоий амалиётнинг турли соҳаларига таъсирининг ЭҲМ тизимларига асосланган ҳолда ишлаб чиқиш, лойиҳалаш, яратиш, баҳолаш, ишлашнинг турли жиҳатларини ўрганувчи илмий фан соҳасидир.

Информатика бу жиҳатдан ахборот моделларини қуришнинг умумий методологик тамойилларини ишлаб чиқишга йўналтирилган. Шу боис ахборот услублари объект, ҳодиса, жараён ва ҳақозаларни ахборот моделлари ёрдамида баён этиш имкониятига эгадир.

Информатиканинг вазифалари, имкониятлари, восита ва услублари кўпқирралиги бўлиб, унинг кўплаб тушунчалари мавжуд. Уларни умумлаштириб қуйидагича талқинни тавсия этамиз.

И н ф о р м а т и к а - компьютерлар ёрдами ва уни қўллаш муҳити воситасида ахборотни янгилаш жараёнлари билан боғлиқ инсон фаолияти соҳасидир.

Информатика кенг маънода инсоният фаолиятининг барча соҳаларида

асосан компьютерлар ва телекоммуникация алоқа воситалари ёрдамида ахборотни қайта ишлаши билан боғлиқ фан, техника ва ишлаб чиқаришнинг хилма-хил тармоқлари бирлигини ўзида намоён этади.

Информатикани тоғ маънода уч ўзаро алоқадор қисмлар - *техник воситалар (hardware)*, *дастурий воситалар (software)* ва *алгоритмик воситалар (brainware)* сифатида тасаввур этиш мумкин. Ўз навбатида информатикани ҳам умуман, ҳам қисмлари бўйича турли жиҳатлардан: халқ хўжалиғи тармоғи, фундаментал фан, амалий фан соҳаси сифатида кўриб чиқиш мумкин.



НАЗАРИЙ КИСМ

Энди алгоритмларнинг асосий хоссаларини кўриб чиқамиз. Алгоритмларнинг асосий хоссалари сифатида куйидагиларни келтириш мумкин.

1. Алгоритмнинг узлуклилиги (дискретлиги) - бу алгоритмнинг ҳар бир қадами тўла тугаллангандан сўнг навбатдаги қадами бажарилишини англатади. Алгоритмнинг бу хоссаси жараенларнинг узлуклилиги ва узлуксизлигидан келиб чиқади. Узлуксиз жараени мисолида металл кўйилиши жараенини келтириш мумкин.

2. Алгоритмнинг аниқлилиги - бу унинг ҳар бир қадами бир хил маънода тушунилиб унинг ўзидан олдинги қадамлар билан тўлалигича аниқланиши демакдир. Алгоритмнинг бу хоссаси шундан иборатки, алгоритмнинг ҳар бир қадами унинг ўзидан олдинги қадамлар тўла тугаллангандан сўнг бажарилиши мумкинлигидир.

3. Алгоритмнинг натижавийлиги - бу алгоритмнинг қадамлар сони чекли бўлиб, ушбу қадамлари тўлиқ бажарилгандан сўнг аниқ бир натижа олинишидир. Бу хосса, одий қилиб айтганда, алгоритмнинг сўнги қадами бўлишини талаб этади.

4. Алгоритмнинг умумийлиги - бу маълум бир масалани ечиш учун ишлаб чиқилган алгоритмни шунга ўхшаш бир қанча масалаларга қўллаш мумкинлигидир. Бу хосса автоматлаштирилган бошқарув тизимлари лойиҳалаштирилатганда жуда муҳим аҳамиятга эга. Бундай алгоритмлар намунавий деб этилади ва улар асосида ҳар хил бошқарув объектлари учун бир хил масалаларни ягона дастурий таъминоти асосида ечишга имкон беради.

Ҳисоблаш жараенларининг график тасвири.

Алгоритмлар одатда тўрт хил усулда ифодаланади:

-формулалар ердамида;

- жадвал усулида;
- сўзлар билан;
- блок-схема кўринишида.

Алгоритмнинг блок-схема тарзида тасвирланиши жуда кенг қўлланилади ва дастур тузиш учун энг қулай келади.

Блок-схема тарзида тасвирланган алгоритм маълум бир қоида бўйича узаро боғланган геометрик шакллардан иборат бўлади.

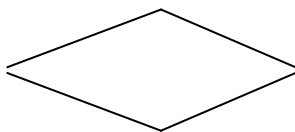
Алгоритмларнинг блок-схемасида қўлланиладиган асосий геометрик шакллари кўриб чиқамиз.



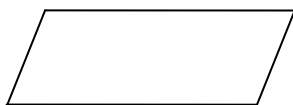
Бу шакл билан алгоритмнинг бошланиш ва тамом бўлиш нуқталари белгиланади.



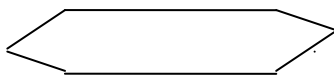
Тўғри тўртбурчак билан алгоритмнинг блок-схемасида бажариладиган ҳисоблаш амаллари кўрсатилади.



Ромб шаклида текширилиши керак бўладиган масаланинг шarti езилади. Ромб шартнинг бажарилиши еки бажарилмаслигига мос равишда «ха» еки «йук» иккита чиқиш йулига эга.



Параллелограмм шакли билан киритиладиган катталиклар ва чиқариладиган натижалар ифодаланади.



Модификация блоки. Бу шакл билан циклнинг бошқариш параметри ўзгариши берилади.



Стандарт дастур блоки. Бу шакл билан қўлланиладиган стандарт дастурлар кўрсатилади.

Ушбу мавзуда Информатика фанининг энг муҳим бўлимларидан бири - алгоритмлаштириш жараенларининг асослари баён этилган. Мавзуда алгоритмнинг тушунчаси ва хоссалари ёритилган. Шу билан бирга чизиқли, тармоқланган ва такрорланувчи ҳисоблаш жараенларининг алгоритмларини тузиш усуллари келтирилган. Ушбу мавзу бўйича амалий дарсларда тармоқли, такрорланувчи жараенларининг алгоритмларини блок-схема шаклида тузишни ўрганиш тавсия этилади.

НАЗАРИЙ КИСМ

С / С++ тилидаги дастурнинг таркиби

С++ тилидаги дастур қуйидаги таркибдан ташкил топади:

1. Директивалар – `# include <file.h>` директива – инструкция деган маънони беради. С++ тилида дастурнинг тузилишига, яъни эҳтиёжига қараб, керакли директивалар ишлатилади. Улар `< >` белгиси орасида келтирилади. Умуман олганда қуйидаги директивалар мавжуд (жами 32 та):

🔥 `# include <stdio.h>` - С да оддий киритиш/чиқариш дастури учун. Бу ерда `std` - стандарт, `i` – input, `o` - output дегани.

🔥 `# include <iostream.h>` - С++ да киритиш/чиқариш учун, оддий операциялар бажарилса.

🔥 `# include <math.h>` - стандарт функцияларни ишлатиш учун

- ❖ # include <conio.h> - дастурнинг ташки куринишини шакллантириш учун;
- ❖ # include <string.h> - катор типдаги узгарувчилар устида амаллар бажариш учун;
- ❖ # include <stdlib.h> - стандарт кутубхона файллари чакириш учун;
- ❖ # include <time.h> - компьютер ичидаги соат кийматларидан фойдаланиш учун;
- ❖ # include <graphics.h> - C++ тилининг график имкониятларидан фойдаланиш учун.

Бу файллар махсус кутубхона эълон файллари ҳисобланадилар ва улар алоҳида INCLUDE деб номланадиган папкада сақланадилар. Ҳозирда C++ кутубхонасини янгиланди ва ундаги файлларнинг номларидан .h (head – сарлавха маъносида) кенгайтмаси олиб ташланди ва олдида с харфи кушилди (C дан колган 18тасига). Бу файлларда функция прототиплари, типлари, узгарувчилар, узгармаслар таърифлари ёзилган булади.

Директивалар дастурни уни компиляция қилинишидан олдин текшириб чиқади.

2. Макрослар - # define макро қиймати. Масалан:

#define y sin(x+25) – $y = \sin(x+25)$ қиймати берилди;

#define pi 3.1415 - $\pi = 3.1415$

#define s(x) x*x - $s(x) = x*x$ (; белгиси қўйилмайди)

3. Глобал ўзгарувчиларни эълон қилиш. Асосий функция ичида эълон қилинган ўзгарувчилар локал, функциядан ташқарида эълон қилинганлари эса глобал ўзгарувчилар дейилади. Глобал ўзгарувчилар дастур давомида ишлайди ва хотирадан маълум жойни эгаллайди. Узгарувчини бевосита ишлатишдан олдин эълон қилса ҳам булади, у холда уз-чи локал булади. Глобал узгарувчилар номи локал узгарувчилар номи билан бир хил булиши ҳам мумкин. Бундай ҳолатда локал узгарувчининг қиймати жорий функция ичидагини қийматини узгартиради, функциядан чиқиши билан глобал узгарувчилар ишлайди.

4. Асосий функция - main () ҳисобланади. Бу функция дастурда бўлиши шарт. Умуман олганда C++даги дастур

функциялардан иборат деб каралади. `main ( )` функцияси { бошланади ва дастур охирида беркитилиши шарт } . `main` – асосий деган маънони беради. Бу функция олдида унинг тоифаси курсатилади. Агар `main ( )` функцияси берадиган (кайтарадиган) жавоб оддий суз ёки гаплардан иборат булса, ҳеч қандай натижа қайтармаса, `void` сузи келтирилади. `main ( )` функцияси дастур томонидан эмас, балки ОС томонидан чакирилади. ОСга қиймат қайтариш шарт эмас, чунки у бу қийматдан фойдаланмайди. Шунинг учун `main ( )` функциясининг тирини **void** деб курсатганимиз маъкул. Хар бир функциянинг уз аргументи булади, шунинг учун `main` функция ()лари ичига унинг параметри келтирилади. Баъзан у буш булиши ҳам мумкин. Бу функциядан чиқиш учун одатда **return** оператори ишлатилади. 0 (ноль) қийматининг қайтарилиши операцион системага ушбу дастур нормал бажарилиб тугаганини билдиради. **return** орқали қайтадиган қиймат типи функция эълонидаги қайтиш типи билан бир хил бўлиши керак. Масалан `int main ( )` ва 0 (ноль) қиймат бутун типлидир. Бу функциядан сўнг локал ўзгарувчилар, қисм дастурлар, уларнинг ҳақиқий параметрлар эълон қилинади. Сўнгра дастурнинг асосий операторлари (киритиш/чиқариш, ҳисоблаш ва х.к.) ёзилади. Агар бу операторлар мураккаб тоифали бўлсалар, уларни алоҳида {} қавсларга олинади. C++ тилида дастур кичик ҳарфларда ёзилади. Баъзи операторлар катта ҳарфлар билан келиши мумкин, бундай ҳолларда улар алоҳида айтиб ўтилади. Операторлар охирига ; белгиси қўйилади. Операторлар бир қаторга кетма-кет ёзилиши мумкин. Дастурда изоҳлар ҳам келиши мумкин, улар `/* ...*/` белгиси орасига олинади. Агар изоҳ бир қаторда тугаса, уни `//` белгисидан кейин ёзилади. Масалан:

```
main ( ) // c++ тилининг асосий функцияси
cout <<"1-функциянинг қиймати="<<f1; // f1 функциянинг
қийматини чиқади.
```

ИНТЕРПОЛЯЦИЯЛАШ МАСАЛАСИНИНГ КУЙИЛИШИ.

Аксарият хисоблаш усуллари масаланинг куйилишида катнашадиган функцияларни унга бирор муайян маънода якин ва тузилиши соддарок буулган функцияларга алмаштириш гоёсига асосланган. Шунинг учун функцияларни якинлаштириш масаласининг энг содда ва жуда кенг кулланиладиган кисми функцияларни интерполяциялаш масаласи куриб чикилади.

Интерполяциялаш масаласининг мохияти куйидагидан иборат:

Фараз килайлик $y = f(x)$ функция жадвал куринишида берилган булсин:

$$Y_0 = f(x_0), y_1 = f(x_1); y_2 = f(x_2); \dots\dots\dots y_n = f(x_n);$$

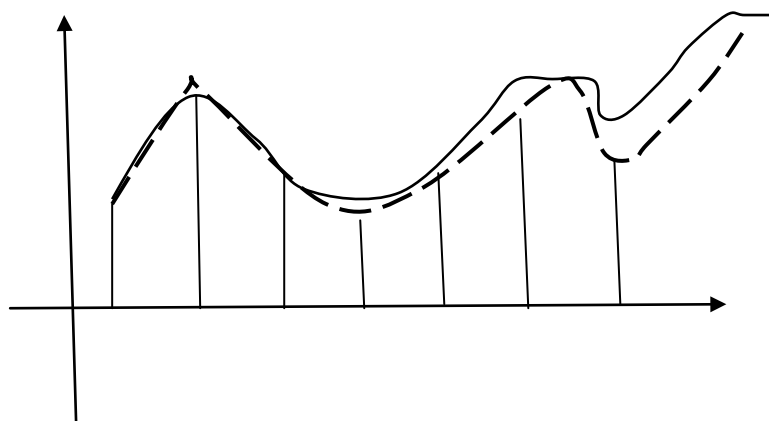
Одатда интерполяциялаш масаласи куйидагича куринишда куйилади:

Шундай n тартиблидан ошмаган $P(x) = P_n(x)$ купхад топиш керакки $P(x_i)$ берилган x_i ($i=0,1,2,\dots,n$) нукталарда $f(x)$ билан бир хил кийматларни кабул килсин, яъни $P(x_i) = y_i$.

Бу масаланинг геометрик маъноси куйидагидан иборат: даражаси n ортмайдиган шундай

$$Y = P_n(x) = a_0x^n + a_1x^{n-1} + \dots\dots\dots + a_n$$

Купхад курилсинки, унинг графиги берилган $M_i (x_i, y_i)$ ($i=0,1,\dots,n$) нукталардан утсин. Бу ердаги x_i ($i=0,1,\dots,n$) нукталар интерполяция тугун нукталари ёки тугунлар дейилади. $P(x)$ эса интерполяцияловчи функция дейилади.



Амалда топилган $P(x)$ интерполяцион формула $f(x)$ функциянинг берилган x аргументнинг интерполяцион тугунларидан фаркли, кийматларини хисоблаш учун кулланилади. Ушбу операция функцияни интерполяциялаш дейилади. Агар $x \in (a, b)$ булса, интерполяциялаш, акс холда экстраполяциялаш булади.

КУРС ИШИ МАСАЛАСИНИНГ БЕРИЛИШИ:

ИНТЕРПОЛЯЦИЯ МАСАЛАСИ

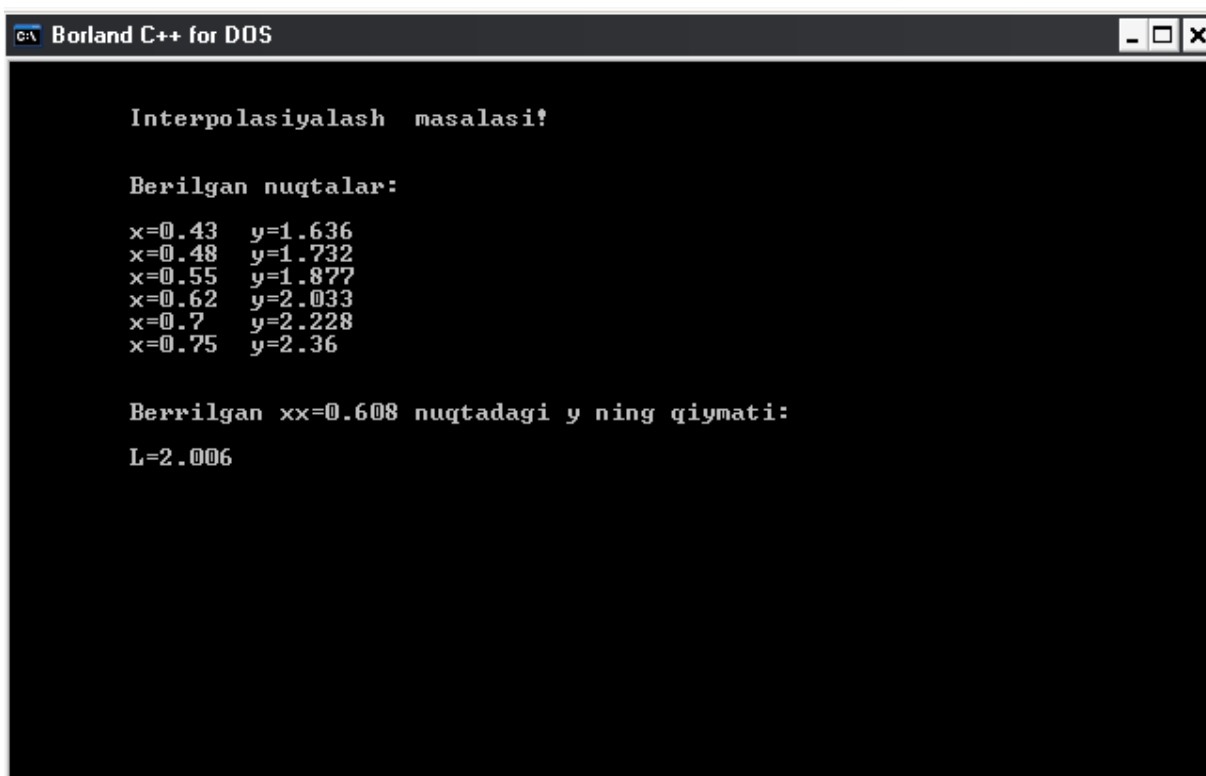
Берилган маълумотлар:

X	Y	№	X*
0,43	1,63697	27	0,436
0,48	1,73234		
0,55	1,87686		
0,62	2,03345		
0,70	2,22846		
0,75	2,35973		

МАСАЛАНИНГ С / С++ ТИЛИДАГИ КОДИ

```
#include <iostream.h>
# include <math.h>
# include <conio.h>
void main( )
{
float x[6]={0.43,0.48,0.55,0.62,0.70,0.75},
y[6]={1.63597,1.73234,1.87686,2.03345,2.22846,2.35973},
xx=0.436,
L=0, p;
int i, j, k;
clrscr( );
cout.precision(3);
for (i=0;i<6;i++)
{
p=1;
for (j=0;j<6;j++)
if (i != j)
p=p*(xx-x[j])/(x[i]-x[j]);
L=L+p*y[i];
}
cout<<"\n\n\tInterpolasiyalash masalasi!\n\n";
cout<<"\n\tBerilgan nuqtalar:\n\n";
for (i=0;i<6;i++)
cout<<"\tx="<<x[i]<<"\ty="<<y[i]<<endl;
cout<<"\n\n\tBerrilgan xx=0.608 nuqtadagi y ning qiymati:\n\n";
cout<<"\tL="<<L<<endl;
getch( ); }
```

НАТИЖАНИНГ ДОС ОЙНАДАГИ КУРИНИШИ:



The image shows a screenshot of a Borland C++ for DOS window. The window has a title bar that reads "C:\ Borland C++ for DOS" and standard Windows window controls (minimize, maximize, close). The main area of the window is black with white text. The text is as follows:

```
Interpolasiyalash masalasi!  
  
Berilgan nuqtalar:  
x=0.43  y=1.636  
x=0.48  y=1.732  
x=0.55  y=1.877  
x=0.62  y=2.033  
x=0.7   y=2.228  
x=0.75  y=2.36  
  
Berilgan xx=0.608 nuqtadagi y ning qiymati:  
L=2.006
```

КУРС ИШИНИ БАЖАРИШ УЧУН

ФОЙДАЛАНИЛГАН АДАБИЁТЛАР

1. Фаронов В.В. «Турбо Паскаль. Начальный курс», Москва, 1998
2. Б.В.Файсман. «Программирование на Турбо Паскале», Москва, 1992
3. Попов В.Б. «Турбо Паскаль для пользователей», Москва, 1998
4. Т.Х.Холматов ва бошқалар. «Информатика», Тошкент, 2002
5. Р.Каримов ва бошқалар. «Дастурлаш», Тошкент, 2003
6. Ш.Ш.Шоҳамидов. «Амалий математика элементлари», Тошкент 1997
7. Ашарина И.В. «Основы программирования на языках С и С++», Москва, 2002
8. Могилев А.В. «Информатика», Москва, 2004
9. Павловская Т.А. «С / С++ программирование на языке высокого уровня», С.Петербург, 2001
10. Б.П.Демидович, И.А.Марон. «Основы вычислительной математики», Москва, 1966