

ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ

**«Ахборот технологияларининг дастурий таъминоти»
кафедраси**

**«Тизимли дастурий таъминот» фани буйича
«Бейсик дастурлаш тили такрорлаш операторини тахлил
килувчи дастурни тузиш» мавзусида
КУРС ИШИ ХИСОБОТИ**

**Бажарди: 210-09-ИТ гурухи талабаси
Гулметов Ф.
Текширди: Каюмова М.Х.**

Тошкент - 2011

МУНДАРИЖА

МУНДАРИЖА	2
Кириш	3
1. Visual Basic дастурлаш тилидаги такрорлаш операторини тахлили масаласининг куйилиши	5
2. Назарий кичм	6
2.1. Visual Basicнинг мавжуд версиялари	6
2.2. Биринчи илова дастурни ёзиш	7
2.3. For ...Next цикллари	8
2.4. Do While ...Loop циклининг тахлили	9
3. Асосий кичм	10
3.1. For ...Next цикл операторининг лексик тахлили	10
3.2. Цикл операторининг тахлили алгоритми	11
3.3. Лексик тахлилнинг вазифалари	12
4. Курс ишида фойдаланилган дастурлаш тили хакида	16
5. Хулоса	18
Фойдаланилган адабиётлар	18
Иловалар	20

Кириш

Visual Basic- Windows асосида ишлайдиган дастурлар тузиш учун кулай восита хисобланади. Дастурларни мустиқил тузишдан мақсад, компьютерга мутиқак ҳокимлиқ қилиш, яъни иш давомида юзага қеладиған муаммоларни тезроқ ҳал этиш имқонини яратишдир. Компьютер дастурлари сермеҳнат ишларни автоматлаштиради, ҳатоларни қамайтиради ва меҳнат унумдорлигини оширади.

Basic (БЕЙСИҚ) сузи Beginner`s All-purpose Symbolic Instruction Code дан ҳосил булған (бошловчилар учун умум йуналишдаги дастурлаш тили). Бу тилнинг мазкур дастур версиясидаги Visual сузи дастур тузиш жараёнида қуплаб операциялар визуал воситалар ёрдамида, яъни бевосита кодларни ёзмасдан бажарилишини аңглатади.

Дастурлар тузиш сермашаққат жараён, лекин Visual Basic бу ишни сезиларли соддалаштиради ва сизга уни осонроқ бажаришга имқон беради. Visual Basicнинг яна бир ютуғи шундаки, у Microsoft Office пакети дастури ва Internet ресурслари билан ишлайди. Visual Basic бир неча версияда ишлаб чиқилади.

Бу дастурлаш тилини Microsoft Visual Basic нинг .NET.B.VB.NET платформасидаги ривожлантирилгани сифатида қараш мумкин ва бу ердаги узгартиришлар тилни Visual Basic нинг бошқа версиялари билан мос қелмаслиқларига олиб қелди. Бу эса VB.6.0 иловаларини автоматик равишда VB.NETга утқазилни муаммога айлантиради. Эски версия лойиҳаларини (*.vbr) уларни аввалдан VB.NET форматга маҳсус мастер (Migration Wizard) ёрдамида айлантирилгандан сунг очиш имқони булади. Булардан ташқари яна, бу жуда муҳим, ишланманинг мантиқ ва концепцияси сезиларли узгартирилған.

VB дастурчилар жамиятининг суровларини ҳисобга олған ҳолда VB.NET нинг ишлаб чиқарувчилари янада янгиланған, қудратлироқ ва эгилувчанроқ версияни тақлиф этдилар. Microsoft VB.NETга қатор янги, шу жумладан объектларга йуналтирилған дастурлашнинг тулиқ имқониятлари ва

web иловаларни ишлаб чиқаришнинг кенгайтирилган қурилмалари имкониятларини қўшди.

Visual Basic.NETни Visual Basicнинг янги версияси деб қараш керакми ёки умуман янги тил деб қараш керакми деган масала тортишувларга сабаб бўлмоқда. Тилнинг асосий синтаксисига жуда катта ўзгартиришлар киритилмаган, янги хусусиятларни амалга ошириш учун турли кенгайтиришлар қўшилган. Маълумотлар турларини икки муҳим ўзгартиришлари амалга оширилган. VB6 билан таққослаганда Integer тури 16 битдан 32 битгача, Long тури 32 битдан 64 битгача кенгайтирилган. Ушбу ўзгартиришлари VB.NET нинг барча версияларида мавжуд. Эски бутунсонли тур Short исми билан сақланган. Графика интерфейси тахрирчиси ўзгартиришсиз қолган. Тил семантикасига анчагина ўзгаришлар киритилган.

Ушбу курс ишида Visual Basic дастурлаш тилининг таққорлаш операторини таҳлил қилиш масаласи қўрилади.

1. Visual Basic дастурлаш тилидаги такрорлаш операторини тахлили масаласининг куйилиши

Ушбу курс ишида Visual Basic дастурлаш тилининг такрорлаш операторини тахлили масаласи кўрилади. If Then бошқариш структуралари кодни ташкилий қисмларга –тармоқларга ажратиш имконини беради. For ...Next – бу цикл бўлиб, бир мартадан ортиқ бажариладиган код блокинни тузиш имконини беради.

Демак, бу курс ишида ана шу такрорлаш оператори иштирокидаги масалаларни тахлили билан шугулланамиз.

2. Назарий кисм

ForNext цикллари узгарувчи-хисоблагичга эга булиб, унда бажарилган цикллар сони сақланади. Бу узгарувчига цикл ичида ҳеч қандай қиймат берилмайди, акс ҳолда у ишламайди.

Циклларнинг 3 хил тури мавжуд:

1. ForNext цикллари;
2. Do While....Loop цикллари;
3. Do UntilLoop цикллари.

2.1. Visual Basicнинг мавжуд версиялари

Visual Basic бир неча версияда ишлаб чиқарилади.

1. Learning Edition (Уқув таҳрири). Бу версия бошловчилар учун жуда мос булиб, Windows да ишловчи дастурлар тузиш учун зарур бўлган барча элементларни ўзида жамлайди.
2. Professional Edition (Профессионал –қасбий таҳрири). Компилятор дастурлари ишини тезлаштирувчи Active X қушимча компонентларини ва маълумотлар базаларини бошқаришнинг қенгайтирилган воситаларини ўз ичига олади. Бу версия жиддий иловаларни ишлаб чиқариш учун мулжалланган.
3. Enterprise Edition (Қорхоналар учун таҳрири) қўпчилик фойдаланувчили маълумотлар базаларига эга иловаларни ишлаб чиқишга имкон беради ва иловаларни дастурчилар гуруҳлари ҳамкорликда ишлаб чиқиш воситаларини ўз ичига олади.
4. Visual Basic for Applications (Иловалар учун Visual Basic). Visual Basicнинг мазкур версияси Microsoft Office пакети дастури таркибига қиради.

2.2. Биринчи илова дастурни ёзиш

Visual Basic билан ишлашга дархол киришиш учун куйида учта оддий кадам билан иловани яратиш кетма-кетлиги келтирилган. Visual Basicнинг сиз яратадиган шаклни (формани) ишлашга мажбур қиладиган харақтга келтирувчи механизми ходисалар, хоссалар ва услублардир. Visual Basicни ишга туширинг ва очилаётган диалогда «Стандарт exe» файл танланганлигини текширинг. Сунг «Очиш» тугмасини босинг. Сунгра «Хоссалар» номли кичик ойнани топинг ва «Сарлавха» хоссасидан унга жойлашган майдонда сичкончани босинг. Бу ерда «Менинг биринчи дастурим» деб ёзинг ва Return тугмасини босинг. Шакл сарлавхаси сатридаги матн кандай узгаришини кузатинг.

Агар хоссалар ойнасини ёки Visual Basic нинг бирор бошка асосий ойнасини топа олмасангиз, уни View менюсидан танлаб олинг. Шундан сунг танланган ойна экранда куринади.

Сизнинг биринчи илова дастурингиз факат шакл (ойна) хосил қилади. Сиз бу шаклнинг улчамларини, рангини узгартиришингиз, уни суришингиз ёки кичиклаштиришингиз, ойнани ёпиб, ундан чикишингиз мумкин. Аслида бу илова-дастур хеч кандай иш бажармайди, лекин бу Windows учун тузилган дастурдир.

2.3. ForNext цикллари

Куйидаги дастур фойдаланувчи томонидан киритилган пул маблагини учун йиллик даромадни ҳисоблайди.:

Юқорида айтилганидек, ForNext цикллари узгарувчи-ҳисоблагичга эга бўлиб, унда бажарилган цикллар сони сақланади. Бу узгарувчига цикл ичида ҳеч қандай қиймат берилмайди, акс ҳолда у ишламайди.

Шаклга ёзув, матн ойнаси ва руйхат ойнасини киритинг.

Тугмани икки марта босинг ва Босиш ходисасини учун куйидаги кодни ёзинг.

```
Dim curA: mount As Currency
Dim iCountvar As Integer
List 1.Clear
curAmount=Vai(text1.Text)
if curAmount<>0 Then
For iCoutvar=3 To 15
List 1. Additem(«да»+Str(iCoutvar)
+ «% йиллик фойиз S даромад келтиради»-
+Str/curAmount*iCountvar/100))
Next iCountvar
Else
MsgBox(«Сон ҳақиқий эмас»)
End if
```

Бу код ForNext циклини if Then оператори ичига қандай қилиб қўйиш мумкинлигини кўрсатади. Операторларни ичига қўйиб ишлатиш баъзи қулайликларни бергани билан, жуда кўп ичига қўйишлардан фойдалангани – дастур устида ишлашни қийинлаштиришга олиб келади.

2.4. Do While....Loop циклининг тахлили

Хамма цикллар ҳам ForNext дан фойдаланавермайди. Бошқа яна иккита имконият мавжуд. Do While....Loop ва Do UntilLoop цикллари. Do While....Loop цикли берилагн шартли ифода False кийматга эга булганда, Do UntilLoop эса берилган шартли ифода True кийматга эга булганда бажарилади.

1. Шаклга тугмани жойлаштиринг.
2. Тугмани икки марта босинг ва Босиш ходисаси учун куйидаги кодни киритинг:

```
Dim answer As Integer
```

```
Do Until answer=vbYes
```

```
Answer=MsgBox("Yes-Чиқиш, NO – давом эттириш", vbYesNO
```

```
Loop
```

3. Дастурни ишга туширинг ва тугмани босинг. Диалог панели пайдо булади. Агар сиз Yes тугмасини боссангиз, диалог ёпилади ва йикл тугайди, чунки answer= vbYes шартли бажарилади. Агар сиз NO тугмасини боссангиз, диалог ёпилади ва дархол яна пайдо булади.

3.Асосий кисм

3.1. ForNext цикл операторининг лексик тахлили

Компиляция жараёнининг биринчи фазаси бу –лексик тахлил боскичи булиб, унда идентификаторларни, константаларни ва тил сузларини ягона белгиларга (лексемаларга) ифодаловчи белгилар каторларини гурухлаш бажарилади.

Бу жараён компиляциянинг бошка фазалари билан параллел бориши (масалан, бир утишли компиляторларда) мумкин. Аммо, хар кандай холда хам компиляторни конструкциясини ифодалаш ва куришда лексик тахлилни мустакил фаза сифатида тассавур килиш мумкин.

Сканерлаш блоки хар бир лексемани у ёки бу синфга тегишли эканлигини аниклаган холда бериши керак. Синфларни танлови трансляция килинаётган тилнинг хусусиятларидан боглик булади. Купинча узгарувчилар исми, константалар, калит сузлар, арифметик амаллар ва мантикий амаллар, махсус белгиларни ажратиш курсатиш мумкин.

Англанаётган каторларни характери лексик тахлил жараёнини анчагина соддалаштириши мумкин.

Барча ушбу каторларни регуляр ифодалар ёрдамида генерациялаш мумкин.

Регуляр ифодалар 3 тур грамматикаларига эквивалентдир.

Юкорида айтиб утганимздек, лексик тахлил кирувчи дастурни белгилар кетма-кетлигига айлантиради. Ва бизнинг холимизда бу куйидаги курунишда булади:

- 1- Такрорлаш оператори;
- 2- Хисоблагич;
- 3- Узлаштириш белгиси;
- 4- Мантикий кийматлар;
- 5- Кавслар;
- 6- Ажраткич ишораси;

3.2. Цикл операторининг тахлили алгоритми

1. Бошланиши.
2. Бейсик тили такрорлаш оператори белгиларини киритиш.
3. Текшириш: агар кирувчи белги 0 сони булса, у холда дастур уз ишини тугатиш учун 10 кадамга утади, акс холда 4 кадамга утилади.
4. Текшириш :агар кирувчи файл элементи калит суз, яъни For , Do While, Loop, Until, To ёки Next булса, у холда «такрорлаш оператори» деган жавоб монитор экранда чикади ва яъна кейинги элементни укишга 2 кадамга утилади, акс холда 5 кадамга утилади.
5. Текшириш :агар кирувчи белги = яъни, узлаштириш белгиси булса , у холда «узлаштириш» деган жавоб монитор экранда чикади ва яъна кейинги элементни укишга 2 кадамга утилади, акс холда 6 кадамга утилади.
6. Текшириш :агар кирувчи белги i , j булса, яъни хисоблагичлардан бири булса у холда «хисоблагич» деган жавоб монитор экранда чикади ва яъна кейинги элементни укишга 2 кадамга утилади, акс холда 7 кадамга утилади.
7. Текшириш :агар кирувчи белги False, True ёки Not булса, мос равишда «ёлгон киймат», «рост киймат» ёки «инкор» деган жавоб монитор экранда чикади ва яъна кейинги элементни укишга 2 кадамга утилади, акс холда 8 кадамга утилади.
8. Текшириш :агар кирувчи белги очилган кавс, ёпилган кавс ёки ажраткич булса, мос равишда «очилган кавс», «ёпилган кавс» ёки «ажраткич» деган жавоб монитор экранда чикади ва яъна кейинги элементни укишга 2 кадамга утилади, акс холда 9 кадамга утилади.
9. Текшириш :агар кирувчи белги юкорида айтилганлардан биронтасига хам мос келмаса, у холда «тилнинг нотаниш белгиси» деган жавоб монитор экранда чикади ва яъна кейинги элементни укишга 2 кадамга утилади.
10. Алгоритм ишини тугатилиши.

3.3.Лексик таҳлилнинг вазифалари

Ҳар бир компилятор оддий командаларнинг чекланган тўпламини бажариш қобилиятига эга. Ихтиёрий мураккаб ҳаракатлар ушбу командаларнинг кетма-кетлиги кўринишида тасвирланади. Юқори даража дастурлаш тилида ёзилган дастурни бажариш учун, уни авваломбор машина кодларидаги командалар кетма-кетлигига ўтказилади.

Бошланғич берилган дастур (қандайдир дастурлаш тилида ёзилган) белгилар кетма-кетлигидан иборат бўлиб, улар компьютерга киритилади ва бажарилиш учун керак бўлган кўринишга айлантирилади.

Компилятор бу шундай, аниқ бир кўринишдаги белгилар қаторини (яъни берилган дастурлаш тилидаги дастур матнини) қабул қиладиган ва бошқа белгилар қаторини (машина тилидаги дастурни) чиқарадиган дастурдир. Компиляторларга бир қатор умумий хусусиятлар хоски, улар компиляция қилинаётган дастурларни ташкил этиш жараёнини соддалаштирадилар. Биз биламизки ихтиёрий компилятор таркибига қуйидаги учта асосий компонента киради:

- лексик таҳлилчи (сканирлаш блоки);
- синтаксис таҳлилчи;
- машина командалари кодлари генератори.

Таҳлилчиларнинг ҳаракат тамойилларини формал моделлар ёрдамида ифодалаш мумкин бўлган ҳолда, кодлар генератори учун умумий формал тасаввурлар мавжуд эмас. Лексик таҳлил фазасида дастурнинг бошланғич берилган матни лексемалар деб аталган бир бири билан боғланмаган занжирлар кўринишидаги белгиларга (бирликларга) ажратилади. Бундай матнли бирликлар калит сўзлар (IF, DO ва бошқалар), ўзгарувчилар исмлари, константалар ва амалларнинг ишоралари (+, - ёки *). Бундай сўнг бу сўзлар алоҳида белгилар гуруҳи эмас, бўлинмаслар сифатида қаралади. Дастурни лексемаларга бўлаклангандан сўнг, грамматик таҳлил деб аталувчи ва операторларнинг кетма-кетлигининг тўғрилигини текширадиган, синтаксис таҳлил фазаси келади. Мисол учун, «IF ифода THEN гап» кўринишига эга IF

гапи учун грамматик таҳлил қуйидагича: IF лексемасидан кейин тўғри ифода келади, ушбу ифодадан кейин THEN лексемаси келади, ундан сўнг яна тўғри ифода келади ва у «;» белгиси билан тугайди. Охирида кодни генерациялаш жараёни бажарилади ва синтаксис таҳлил натижаларидан фойдаланилиб бажаришга тайёр машина тилидаги дастур ташкил этилади. Ихтиёрий компилятор таркибига юқорида айтиб ўтилган учта компонента кирса ҳам, уларнинг ўзаро ҳаракати турли усуллар орқали амалга оширилади. Ушбу компоненталар орасидаги ўзаро ҳаракатларнинг кенг тарқалган вариантларини қараб чиқамиз.

Сканерлаш блоки бошланғич дастурни ўқийди ва лексемалар файли сифатида ифодалайди. Синтаксис таҳлилчи эса бу файлни ўқийди ва дастурни янги ифодасини чиқаради, масалан постфикс кўринишида. Ва ниҳоят, бу файл код генератори орқали ўқилади ва дастурни объект коди ташкил этилади.

Лексик таҳлилчи бу компиляторнинг бўлаги бўлиб, кировчи матнни ўқийди ва кировчи тилнинг лексемаларни ажратади. Лексема- бу тилнинг структура ўлчови бўлиб, тилнинг элементар белгиларидан ташкил топади. Лексема мажбурий бўлак эмас, лекин компиляторнинг таркибига қўшилади. Шунинг учун:

1.Таҳлил босқичида матнлар билан ишлашни соддалаштиради ва қайта ишланган маълумот ҳажмини қисқартиради.

2.Лексемаларни матнда ажратиш ва таҳлил учун содда ва фойдали таҳлил техникаси қўлланилади. Синтаксис таҳлил босқичида эса таҳлилнинг етарли мураккаб тизимларидан фойдаланилади.

3.Лексик таҳлил, конструкция бўйича кировчи дастур матни билан ишлаш жараёнида структураси тилнинг версиясидан боғлиқ бўлган, мураккаб синтаксис таҳлилчини аниқлайди.

Лексик синтаксис таҳлилчини ўзароҳаракатлари қуйидагича амалга оширилиши мумкин:

1.Кетма-кет. Лексик тахлилчи кирувчи дастур матнини бошидан охиригача қараб чиқади ва лексемаларнинг структураланган жадвалини ҳосил этади, бу жадвалда инедтификаторнинг калит сўзларини махсус келишилган кодларга ўзгартирилади ва бундан ташқари идентификаторлар жадвали билан алоқа ўрнатилади.

2.Параллел. Кирувчи дастур матнининг лексик тахлили босқичма – босқич бажариладики, синтаксис тахлилчи тилнинг навбатдаги конструкциясини тахлилини бажаришда лексик тахлилчига кейинги белгини ўқиш учун мурожаат қилади ва шу билан бирга ҳар бирини кутиб туришини кўрсатади.

Ушбу ҳолатда хатоликлар юз берганда «рад этиш» бўлиши мумкин ва лексемаларни тахлилини тўғри бажарилганида лексема лексемалар жадвалига ёзилади.

Лексик тахлил ўз ичига қуйидаги масалаларни ечишни олади:

дастурнинг берилиш матнини қараб чиқиш (сканирлаш);

изоҳлар ва кераксиз бўш белгиларни йўқотиш;

лексик ўлчовларни ажратиш (лексик ўлчовларни чегараларини топиш);

лексемларни ички тасвирлаш кўринишларига айлантириш;

лексик ўлчовларни англаш;

лексик тахлилчининг чикувчи жадвалларини ташкил этиш.

Кўпгина дастурлаш тилларида лексемларни чегаралари бўлиб бўш белгилар, ажраткичлар, ва амал ишоралари ҳисобланадилар.

Оддий ҳолда лексема – бу чегаралар орасида жойлашганлардир.

Константа ва исмлар каби лексемаларни англаш учун 3 синф регуляр грамматикасидан фойдаланилади ва англовчи тугалланган автоматни ташкил этади.

Скаринлаш натижасида дастур узун қатордан ташкил топади.

Лексик тахлилчини амалга ошириш учун икки йўналиш мавжуд.

Биринчи томон—лексик ва синтаксис тахлилчиларнинг ўзаро ҳаракатларини амалга оширишнинг икки йўналиши.

Иккинчи — лексик тахлилчининг қурилиши.

Биринчи йўналиш: лексик ва синтаксис тахлилчиларнинг ўзаро ҳаракатларини амалга оширишнинг охириги чизмаси. Лексик тахлилчи барча дастурни қараб чиқади ва барча лексемалар учун кодлар жадвалини тузади ва натижаларни синтаксис тахлилчига узатади. Бу ҳолатда лексик ва синтаксис тахлилчилар бўлакланган ҳисобланадилар ва компилятор босқичида бир биридан кейин бажариладилар.

4. Курс ишида фойдаланилган дастурлаш тили хакида

Си++ - бу ифодаларнинг ёзилишининг кулай усули, берилганлар структурасининг замонавий бошқарув механизми ва бой операторлар тўпламига эга бўлган дастурлашнинг универсал тилидир. Си++ дастурлаш тили „жуда ҳам юкори даража“ тили ҳам, „катта“, тил ҳам эмас, бирон бир аниқ сохага фойдаланишга махсуслаштирилган ҳам эмас. Лекин унга кўйилган чегараларнинг йўқлиги ва универсаллиги бу дастурлаш тилининг баъзи кучли тилларга кўра кўпгина масалаларни кулай ва фойдалироқ ечилишига имкон беради.

Тажрибаларимиздан келиб чиққан ҳолда шуни айтишимиз мумкинки, Си++ дастурлаш тили –кулай, ифодали ва кенг масалалар синфи учун керак бўладиган дастурлаш тилидир.

Хотира синфи типни спецификацияси олдида ўзгарувчини эълон қилаётганимизда аниқ кўрсатилиши мумкин. Агар хотира синфи кўрсатилмаган бўлса, у ҳолда у ўша ўзгарувчи эълон қилинган жойга боглиқ ҳолда аниқланади. Қандайдир функциянинг ичида эълон қилинган ўзгарувчилар хотирада жойлашадиган динамик ўзгарувчилардир. Ҳеч қандай функцияни ичида жойлашмаган, функциялар билан бир хилда эълон қилинган ўзгарувчилар статик ўзгарувчилар дейилади. Қандайдир функцияни ичида эълон қилинган ўзгарувчиларда, хотира синфидан катъий назар, фақат шу функциянинг ичида фойдаланиш мумкин. Функциялари билан бир хил даражада эълон қилинган ўзгарувчилар, яъни олдида static ёки extern деган ключ сўзи бўлган. ўзгарувчилар программада ихтиёрий ерда фойдаланилиши мумкин, агарда программа бир неча программани файллардан тузилган бўлса ҳам. Функциялар билан бир хил даражада деб эълон қилинган ўзгарувчилар, яъни олдида static ключ сўзи турган ўзгарувчилар ҳамма функцияларда ишлатилиши мумкин, лекин фақат ўзи эълон қилинган файл учун.

Си++ тилида ихтиёрий программа бу ташки объектлар тўплами – яъни ўзгарувчилар ёки функциялар тўпамидир. Ташки ўзгарувчилар ихтиёрий функциядан ташқарида аниқланган ва кўп функциялар учун фойдаланиши

мумкин. функцияларнинг ўзи ҳар доим ташкидир, чунки тилнинг ўзи функцияни бошқа функция ичида аниқлашга имкон бермайди.

Кўп алгоритмик тилларда маълумотлар бериладиган параметрлар, массивлар статистик объектлар деб аниқланарди, чунки уларга олдиндан канча микдорда кийматлар берилиши маълум хисобланарди. Шу сабаб ҳар бир программада маълумотлар хотиранинг қай қисмини эгаллашини аввалданок билиш мумкин эди.

Си тилида динамик объектлар тушунчаси киритилган. Бундай объектлар ўлчови программа ишлаш шароитига қараб ўзгаради ва программа динамик объектларга бажарилиш фаолиятида қерағича (сўрокка қараб) жой ажратади. Мисол учун тузилган программа учун 100, 200 ёки 500 та маълумот киритиш лозим бўлса, мос равишда хотирадан шунча ячейка бандланади.

5. Хулоса

Ушбу курс ишидан олинган асосий натижа ва хулосалар қуйидагича:

- 1) Замонавий Visual Basic дастурлаш тилида такрорлаш операторларининг таҳлили келтирилган.
- 2) Таҳлил қилинган имкониятлар бўйича назорат мисолларининг натижалари келтирилган.
- 3) Курс ишини бажаришда фойдаланилган объектларга мўлжалланган дастурлаш тили C++ ҳақида асосий маълумотлар берилган.

Фойдаланилган адабиётлар

- 1.Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. «Объектно-ориентированное программирование» Изд. МГТУ им.Н.Э.Баумана, 2003 г.
- 2.Каримов И.А., Жахон молиявий –иктисодий инкирози, Узбекистон шароитида уни бартараф этишнинг йуллари ва чоралари. Тошкент: Узбекистон нашриёти, 2009.
- 3.Каримов Р,Ирискулов С. «Дастурлаш», Тошкент, «Узбекистон», 2003.
- 4.Бек Л. «Введение в системное программирование», Москва, «Мир» , 1988 г. «Наука» 1988г.

Иловалар

```
#include<stdio.h>
#include<iostream.h>
#include<string.h>
#include<ctype.h>
#define MAXLINE 1000
int getline(char line[], int max);
int strindex(char source[], char searchfor[]);
void copy(char to[], char from[]);
char pattern[]="For";
char pattern1[]="Next";
char pattern2[]="Do While";
char pattern3[]="Loop";
char pattern4[]="To";
char pattern5[]="=";
char pattern6[]="i";
char pattern7[]="j";
char pattern8[]="Until";
char pattern9[]="False";
char pattern10[]="True";
char pattern11[]="Not";
char pattern12[]="(";
char pattern13[]=")";
char pattern14[]=")";

main()
{
    int found=0;
    char line[MAXLINE];
    printf("*BEYSIK tili takrorlash operatorini taxlili dasturi**\n");
    printf("\nBEYSIK tili takrorlash operatori belgilarini kiriting -->\n");
    printf("\n Tugatish uchun 0 ni kiriting -->\n");
    while(getline(line, MAXLINE)>0)
    if(strindex(line, pattern)>=0)
    {
        printf("%s -takrorlash operatori\n", line);
        found++;
    }
    else
    if(strindex(line, pattern1)>=0)
        printf("%s -takrorlash operatori\n", line);
    else
    if(strindex(line, pattern2)>=0)
        printf("%s -takrorlash operatori\n", line);
    else
    if(strindex(line, pattern3)>=0)
        printf("%s -takrorlash operatori\n", line);
    else
    if(strindex(line, pattern4)>=0)
        printf("%s -takrorlash operatori\n", line);
    else
```

```

    if(strindex(line, pattern5)>=0)
printf("%s -o`zlashtirish\n", line);
else
    if(strindex(line, pattern6)>=0)
printf("%s -xisoblagich\n", line);
else
    if(strindex(line, pattern7)>=0)
printf("%s -xisoblagich\n", line);
else
    if(strindex(line, pattern8)>=0)
printf("%s -takrorlash operatori\n", line);
else
    if(strindex(line, pattern9)>=0)
printf("%s -yolg`on qiymat\n", line);
else
    if(strindex(line, pattern10)>=0)
printf("%s -rost qiymat\n", line);
else
    if(strindex(line, pattern11)>=0)
printf("%s -inkor etish\n", line);
else
    if(strindex(line, pattern12)>=0)
printf("%s -ochilgan qavs\n", line);
else
    if(strindex(line, pattern13)>=0)
printf("%s -ajratkich\n", line);
else
    if(strindex(line, pattern14)>=0)
printf("%s -yopilgan qavs\n", line);
else
printf("%s -notanish belgi\n", line);
return found;
}

int getline(char s[], int lim)
{
    int c,i;
    i=0;
for(i=0;i<lim-1 &&(c=getchar())!='0' && c!='\n'; ++i)

        s[i]=c;
    if(c=='\n') {
        s[i]=c;
        ++i;
    }
    s[i]='\0';
    return i;
}

int strindex(char s[], char t[])
{
    int i,j,k;

```

```
for(i=0; s[i]!='\0'; i++){
    for(j=i, k=0; t[k]!='\0' && s[j]==t[k]; j++, k++)
        ;
    if(k>0 && t[k]=='\0')
        return i;
}
return -1;
}
```