

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ**  
**ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ**  
**ТЕРМИЗ ДАВЛАТ УНИВЕРСИТЕТИ**  
**ФИЗИКА - МАТЕМАТИКА ФАКУЛЬТЕТИ**  
**«АХБОРОТ ТЕХНОЛОГИЯЛАРИ»**  
**КАФЕДРАСИ**

**Ҳайитов Отабек Чори ўғли**  
**Макрослар ёрдамида ўқув эффектлари ва трюкларини яратиш**

5110700- “Информатика ўқитиш методикаси” таълим йўналиши бўйича  
бакалавр даражасини олиш учун ёзилган

**БИТИРУВ МАЛАКАВИЙ ИШИ**

**Илмий раҳбар: и.ф.д.О.Қ.Хатамов**

**Термиз – 2017 й.**



## **М у н д а р и ж а**

<b>Кириш.....</b>	
<b>I.БОБ. VISUAL BASIC FOR APPLICATION ДАСТУРЛАШ МУҲИТИ ВА ИМКОНияТЛАРИ.....</b>	
<b>1.1. Microsoft Office дастурларида макрослар билан ишлаш.....</b>	
<b>1.2. Microsoft Excel билан ишлашда VBA нинг функционал имкониётлари.....</b>	
<b>1.3. Макрос ёзишнинг дастлабки қадамлари ва массивлар билан ишлаш.....</b>	
<b>II.БОБ. VBA ОБЪЕКТЛАР ИЕРАРХИЯСИ, ФАЙЛЛАР БИЛАН ИШЛАШ ВАЎҚУВ ЭФФЕКТЛАРИ ҲАМДА ТРЮКЛАРИ ЯРАТИШ АМАЛИЁТИ.....</b>	
<b>2.1. VBA да дастурий код ёзишнинг объектлар иерархияси.....</b>	
<b>2.2. VBA муҳитида файлларга ахборотни киритиш ва чиқариш.....</b>	
<b>2.3. Макрослар ёрдамида ўқув эффектлари ва трюкларини яратиш амалиёти.....</b>	
<b>Хулоса ва таклифлар.....</b>	
<b>Фойдаланилган адабиётлар.....</b>	

## КИРИШ

**Мавзунинг долзарблиги.** 1992 йил 8 декабрда Ўзбекистон Республикаси Вазирлар Маҳкамасининг қарори билан Фан ва Техника Давлат қўмитаси қошида Ахборотлаштириш бўйича бош Бошқарма тузилди. Қўмита томонидан 1993 йил май ойида ахборотлаштириш ҳақида, 1994 йилда ЭҲМ ва маълумотлар базаси учун дастурларни ҳуқуқий муҳофазалаш тўғрисида қонунларнинг қабул қилиниши бу масалада туб бурилиш ясади. Ушбу қонунда мамлакатимизда замонавий ахборот технологияларини ривожлантириш, ахборот хизматлари соҳасини кенгайтиришни асосий йўналишлари аниқ-равшан белгилаб берилди.

Вазирлар Маҳкамасининг 1994 йил 2 декабрдаги фармойиши билан Ўзбекистон Республикасини Ахборотлаштириш концепцияси маъқулланди. Ушбу концепцияда ахборотлаштириш жамият тараққиётининг объектив жараёни эканлиги кўрсатиб ўтилди. Ахборотлаштириш концепцияси 3 та асосий мақсадга йўналтирилган:

1. Замонавий ахборот технологияларини ривожлантириш, давлатнинг барча субъектлари учун ахборот хизматларини кенгайтириш;
2. Иқтисодиёт ва ижтимоий соҳаларда ахборот тизимлари шаклланишига кўмаклашиш;
3. Мамлакатни жаҳон ахборот тизимлари ва тармоқларига улаш.

Концепциянинг асосий қоидалари, вазифалари ҳисобга олинган ҳолда «Ўзбекистон Республикасининг ахборотлаштириш дастури» ишлаб чиқилди. Дастур 3 қисмдан иборат:

- миллий ахборот-ҳисоблаш тармоғини тузиш;
- ЭҲМни математик ва дастурий таъминлаш;
- шахсий компьютер билан таъминлаш;

Мазкур дастурда Вазирлик ва Маҳкамаларнинг ахборот тармоқлари, Миллий ахборотни ҳисоблаш тармоғини яратиш, компьютерлар ва ҳисоблаш техникаси воситаларини ишлаб чиқаришни ташкил этиш, янги

ахборот технологиялари соҳасида кадрлар тайёрлашни такомиллаштириш, хужжатлаштиришнинг меъёрий-услубий ва ҳуқуқий тизимини яратиш ва бошқалар ўрин олган. Юқоридаги кўрсатилган вазифаларни бажариш бир неча босқичда амалга ошириш кўзда тутилган.

Дастурни амалга ошириш борасида ҳукуматимиз томонидан шу кунгача бир қанча қарорлар қабул қилинди. Жумладан, 2002 йил 6 июнда «Компьютерлаштиришни янада ривожантириш ва ахборот – коммуникация технологияларини жорий этиш тўғрисида»ги қарори қабул қилинди. Қарорда белгиланган чора-тадбирларнинг амалга оширилиши ахборотлаштиришнинг миллий тизимлари барпо этилишини, иқтисодиётга ва жамиятнинг ҳар бир аъзоси ҳаётига компьютер техникаси ва ахборот технологиялари оммавий жорий этилиши учун шарт-шароитларни таъминлайди, жаҳон бозорида мамлакатимиз иқтисодиётининг рақобатбардошлилигини оширади.

2002 йил 12 декабрда «Ахборот эркинлиги принциплари ва кафолатлари тўғрисида»ги қонун қабул қилинди. Ушбу қонуннинг асосий вазифалари ахборот эркинлиги, принциплари ва кафолатларига риоя этилишини, ҳар кимнинг ахборотни эркин ва монеликсиз излаш, олиш, текшириш, тарқатиш, фойдаланиш ва сақлаш ҳуқуқлари рўёбга чиқарилишини, шунингдек ахборотнинг муҳофаза қилинишини ҳамда шахс, жамият ва давлатнинг ахборот борасидаги хавфсизлигини таъминлашдан иборат.

2004 йил февралда «Ахборотлаштириш тўғрисида»ги, «Электрон рақамли имзо тўғрисида»ги қонунлар қабул қилинди. «Ахборотлаштириш тўғрисида»ги Қонуннинг мақсади ахборотлаштириш, ахборот ресурслари ва ахборот тизимларидан фойдаланиш, «Электрон рақамли имзо тўғрисида»ги қонуннинг мақсади эса, электрон рақамли имзодан фойдаланиш соҳасидаги муносабатларни тартибга солишдан иборат.

2004 йил 29 апрелда “Электрон ҳужжат айланиши тўғрисида”ги қонун қабул қилинди. Электрон ҳужжат айланишида электрон ҳужжатларни муҳофаза қилиш, электрон ҳужжат айланишининг иштирокчиларига ёки

бошқа юридик ва жисмоний шахсларга зарар етказилишининг олдини олиш мақсадида конун ҳужжатларида белгиланган тартибда амалга оширилади.

Кўрсатиб ўтилган чора-тадбирлар республиканинг жаҳон жараёнларига муваффақиятли интеграциялашуви имкон бериб, мамлакат иқтисодиёти самарадорлигини ўсишида телекоммуникациялар, компьютер ва ахборот технологияларининг фаол роли ошишини, инсонларнинг фаолият ва турмуши техник қурилмалар ва хизматларнинг энг замонавий турлари билан жиҳозланишини таъминлаш, яратилган дастурлар имкониятларини янада кенгайтириш учун шароитлар яратади. Шу нуқтаи назардан мазкур битирув малакавий иш долзарб масалага қаратилганлиги билан эътиборга моликдир.

**Ишнинг мақсади ва вазифалари:** Бизга маълумки, дастурлаш яратилаётган лойиҳани ёзишнинг махсус усули. Бир қарашда дастурни қайта ишлаш учун компьютер хотирасига киритиш етарли-ю, аммо унинг услуби ва айниқса, ўзига хос ёзилишининг аҳамияти йўқдек бўлиб кўринади. Аслида эса дастур тушунарли ва ўқиш учун осон бўлиши керак. Зарурат туғилганда уни тушуна олиш ёки унга керакли ўзгартиришлар киритиш мумкин бўлиши керак. Ҳам табиий, ҳам математик тилларда тасвирланган ифодалар воситаларидан оқилона фойдаланиш сифатли дастурлар яратишнинг етарли шартларидан ҳисобланади. Шуларни эътиборга олган ҳолда, ушбу битирув малакавий **ишнинг мақсади** қилиб, Microsoft Office билан ишлашда макрослардан фойдаланиб, ўқув эффектлари ва трюклари яратиш услубияти билан танишишдир. Мазкур мақсаддан келиб чиқиб, қуйидаги вазифалар

белгиланган:

- Microsoft Office дастурларида макрослар билан ишлаш;
- Microsoft Excel билан ишлашда VBA нинг функционал имкониятлари билан танишиш;
- VBA да дастурий код ёзишнинг объектлар иерархиясини таҳлил этиш;
- VBA муҳитида файлларга ахборотни киритиш ва чиқариш амалиётини ўрганиш;
- макрослар ёрдамида ўқув эффектлари ва трюкларини яратиш.

**Ишдаги илмий янгиликлар:** Ишнинг илмий янгилиги VBA (Visual Basic for Application) дастурлаш муҳити, унинг компонентлари ва иловаларидан фойдаланиб, фойдаланувчилар учун қулай интерфейсга эга бўлган ўқув эффектлари ва трюклари яратиш технологиясининг илмий-назарий ва амалий жиҳатларини очиқ бериш ҳисобланади.

**Ишнинг амалий аҳамияти:** Бугунги кунда компьютер фойдаланувчилари унда ишлаш жараёнида жуда кўплаб амалий дастурлардан фойдаланишади. Жумладан, шулар қаторида офис дастурлари пакети энг кўп қўлланиладиган дастурлар тоифасига киради, табиийки офис дастурларида қўшимча функциялар ҳосил қилиш учун қўлланиладиган дастурий иловаларга ҳам бугунги кунда эҳтиёж ортмоқда. Бундай дастурларни ишлаб чиқиш, уларни фойдаланувчиларга тақдим қилиш дастурчилар олдида доимгидек муҳим бўлиб қолмоқда. Чунки, дастурлар сони кўпайгани сари фойдаланувчиларда ўзига қулай дастурни танлаш имконияти ортиб боради. Ушбу битирув малакавий ишида яратилган дастурлар ҳам ана шундай дастурлардан бири ҳисобланади. Битирув малакавий ишида бундай дастурий воситани яратиш технологияси амалий жиҳатдан атрофлича очиқ берилган.

Дастур тузиш ижодий иш ҳисобланади, чунки ҳатто аниқ ифодаланган мақсадга эришиш учун усулларни излаш ҳам умумий ҳолда янги билим ва қўникмаларни ишлаб чиқиш ва қўллашни талаб қилади. Баъзи хусусий ҳолларда тизимли ва формал дастурлаш процедураларини топиш мумкин. Агар дастурлаш учун қўйилган масала алгоритм кўринишида тасвирланган бўлса, у ҳолда дастурлаш алгоритмнинг ёзилиш тилидан бажарувчи томонидан тушунарли бўлган тилга кўчиришдан иборат бўлади. Алгоритмларнинг ёзувларини ўтказувчи тизимли процедураларни излаш ёки дастурни масаланинг шартларидан ва қўшимча маълумотлар асосида яратиш синтез дастури ва унинг дастурни трансляция қилиш хусусий ҳолининг мавзуси (предмети) ҳисобланади.

Дастурлаш методикаси дастурнинг зарур ўзига ҳослигига асосий эътиборни қаратади, чунончи ўзига ҳосликдаги маълумотлардан унумли

фойдаланиш дастурнинг ишончилигини оширади. Дастурнинг тўғрилигини текширишни осонлаштирувчи унинг тузилишини яратиш дастурлашнинг асосий жиҳати ҳисобланади.

Ўзбекистон Республикаси Кадрлар тайёрлаш Миллий дастурида ҳам ёш авлодни янги компьютер технологияларини эгаллаши, уларнинг билимларини ривожланган хорижий талабалари савиясига етказиш долзарб масала килиб қўйилган.

Ушбу битирув малакавий ишида VBA (Visual Basic for Application) дастурлаш муҳитининг кучли имкониятларидан фойдаланиб, Microsoft Office дастури таркибида бажарилувчи, улар билан ишлашни осонлаштирувчи дастурий кодлар яратиш жараёни кўриб чиқилган.

Ушбу **битирув малакавий иш тузилиши** мантиқан боғланган кириш, икки боб, олти параграф, хулоса ва фойдаланилган адабиётлар рўйхатидан иборат.



# I. БОБ. VISUAL BASIC FOR APPLICATION ДАСТУРЛАШ МУҲИТИ ВА ИМКОНИЯТЛАРИ

## 1.1. Microsoft Office дастурларида макрослар билан ишлаш.

VBA(Visual Basic for Application) - бу ном Microsoft компанияси томонидан Office дастурларининг дастурлаш тилига берилган номдир. VBA, шунингдек, бошқа ишлаб чиқарувчилар учун ҳам яроқли бўлиб, улар бу тилдан ўз дастурларида фойдаланишлари мумкин. VBA ва Visual Basic нинг автоном ишловчи версиялари орасидаги фарқни билиш учун икки ташкил этувчини фарқлаш керак[3,6]:

### 1. Visual Basic тилининг негизи.

Объектлар, яъни тугма, рўйхатлар дарчаси ва **Active X** бошқарув элементларидир. VBA да ҳам, Visual Basic нинг автоном ишловчи версиясида ҳам тил негизи (асоси) ўзгармайди. Шу билан бир вақтда бу тил- бу объектлар тўплами бўлиб, уларнинг таркиби ўзгариб туради. VBA билан ишлаш имкони бўлган дастурда ўзига хос дастур объектлари бўлади ва улардан фойдаланувчи ўз кодида фойдаланиши мумкин.

Мисол учун, Wordда **Dosument** (Хужжат) ва **Paragraph** (Абзац) объектлари бор. Бу объектлар билан ишлашни ўрганиб олган бўлсангиз, VBA коди ёрдамида бу объектлар билан ишлаш имконига эга бўласиз.

Шуниси эътиборга лойиқки, системада Office дастури ўрнатилган бўлса, Office дастурларининг исталган элементи учун VBA кодини ёзиш мумкин. Масалан, Excelda ёзилган VBA кодини Word учун ёки Visual Basic автоном дастурида ёзилган коддан Office дастурининг исталган элементида фойдаланиш мумкин.

VBA да Visual Basicнинг автоном версиясидагига ўхшаш матн муҳаррири ва созлагич (отладчик) бор, лекин бу элементлар ўз имкониятлари бўйича бир нечта чеклашларга эга. VBA да кодни автоном EXE – файлгача компиляция қилиш имконияти йўқ, бундан ташқари, VBA (native code) процессорнинг бевосита йўриқномасигача қадар компиляция қила олмайди.

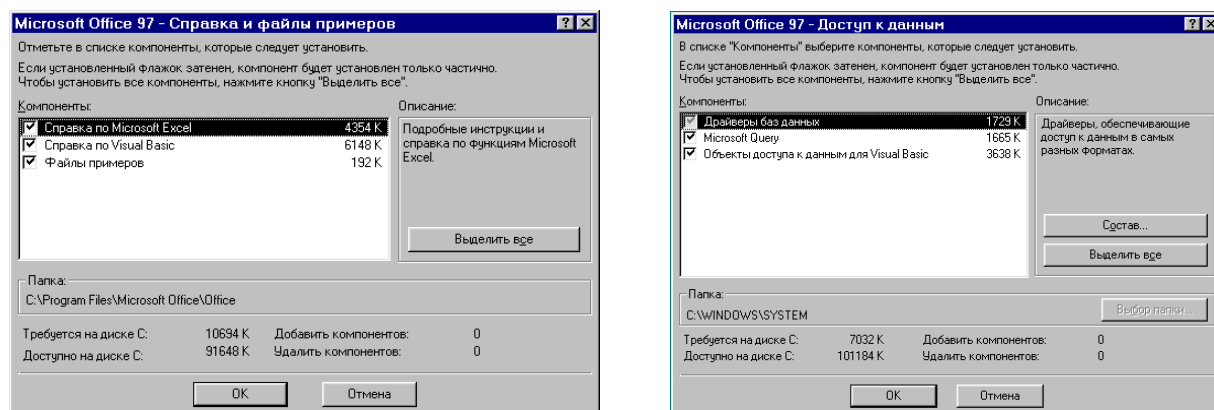
Шу билан бирга VBA да **Data** бошқариш элементи ҳам йўқ. Лекин маълумот базалари орқали “соф” VBA коди билан ишлаш имкони бор.

VBA нинг ноёб имконияти ҳам бор. Дастур тузишни макрослар ёзишдан бошлашингиз мумкин, яъни Office сиз учун код ёзади.

VBA Excel, Word, Access ва Power Point дастурларининг бир қисми ҳисобланади. Outlook дастури эса Visual Basic нинг бошқа, имконияти камроқ **Vbcript** версиясидан фойдаланади[16].

VBA ни ўрнатиш.VBA Office дастурларини ўрнатишганда автоматик тарзда ўрнатилади. Лекин VBAнинг маълумотномаси ўрнатилмайдди. VBA да дастур тузиш жараёнида маълумотнома зарур бўлиши мумкин. Шу билан бирга VBA билан ишлашда ахборотлар базаси билан ишлаш имконини берувчи объектларни ўрнатиш зарурияти ҳам юзага келиши мумкин.

**VBA компонентларини ўрнатиш.** VBA маълумотномаси файлларини ўрнатиш учун Office ни ўрнатиш дастурини ишга тушуриш лозим. “Добавить (Удалит компоненты)” (Add/ Remove components) Word, Excel ва PowerPoint буйруғини танлаб, “Справка и файлы примеров” (Help files and samples) сатрини белгиланг. Access учун “Справка по программированию” (Programming Help) сатрини белгилаш лозим(1.1-расм). Ҳар бир дастур ўз объектлари учун маълумотнома ўрнатади, шу билан бирга маълумотнома файллари турличадир.



1.1 -расм

Агар сизда ахборотлар базалари билан ишловчи код ёзиш зарурати юзага келса, Visual Basic учун ахборотлар базаси билан ишлаш имконини

берувчи объектларни (Data Access Objects For Visual Basic) ўрнатиш зарур. Бунинг учун Office ни ўрнатиш дастурининг “Доступ к данным” Data Access бўлимини белгилаш зарур.

VBA дастурлаш воситалари. VBA ўзининг дастурлаш муҳитига эга. У Office нинг барча дастурлари учун бир хил.

VBA маълумотномаси катта ҳажмни эгаллайди. Word ва Excel сизга маълумотноманинг фақат энг зарур элементларини ўрнатиш учун зарур. Мисол учун, Word ва Excel га бу маълумотнома зарур, Power Point учун эса иккиламчи восита ҳисобланади.

VBA маълумотномаси файллари катта ҳажмни эгаллайди. Word ва Excel сизга маълумотноманинг фақат энг зарур элементларини ўрнатиш учун зарур. Мисол учун, Word ва Excel га бу маълумотнома зарур, Power Point учун эса иккиламчи восита ҳисобланади.

VBA ни ишга туширишнинг бошқа бир усули менюдан Вид>Макрос>Макросы (Tools>Macro>Macros) ни танлаш ва “Edit” (Таҳрир) тугмасини босиш ёки Create (“Яратиш”) тугмачасини босиб, янги макрос ёзишни бошлашдир.

VBA дастурлар муҳити мустақил (автоном) дастур каби ишлайди ва автоном Visual Basic га кўп жиҳатдан ўхшаш. Visual Basic ва VBA да эквивалент тугмалар бир-бирига тўғри келади. Масалан, дастурни ишга тушириш учун F5 тугмасини босиш етарли.

## **1.2. Microsoft Excel билан ишлашда VBA нинг функционал имкониятлари.**

Microsoft Excel жадвал муҳаррири қуйидаги масалаларни ҳал этишга мўлжалланган[3]:

- ўрнатилган формулалар, функциялар, макрослар ва бошқалар ёрдамида жадвал катталикларни киритиш ва улар устида амаллар бажариш;
- маълумотни бошқариш ва таҳлил этиш (якуний ва оралик маълумотларни автоматик ҳисоб -китоб қилиш, маълумотларни

тузилмалаштириш ва бирлаштириш, якуний жадвал ва ҳисоботлардан фойдаланиш ва ҳаказо);

- турли манбалардан (хаттоки OLAP маълумотлар базасидан) зарурий маълумотларни импорт қилиш ва кейинчалик ишлов бериш. XML – шаклда қўллаб-қувватлаш;
- график объект ва диаграммалар билан ишлаш;
- Lotus Notes дастури билан ҳамкорликда ишлаш ва маълумот алмашиш. Бундан ташқари бошқа дастурий маҳсулотлар билан интеграциялашув (“Галактика”, “1С ва бошқалар);
- интернетда ишлаш (веб-саҳифадаги маълумотни ўзгартириш, Microsoft Excel маълумотларини интернетга жойлаштириш, веб-файлларни қўллаб-қувватлаш ва гипермуружаат ва бошқалар);
- бошқа дастурлар (Word, Power Point, Access ва бошқа) билан ҳамкорликда маълумотларга муружаатни амалга ошириш;
- турли кўринишдаги: аналитик, якуний, график, диаграмма кўринишидаги ҳисоботларни шакллантириш;
- Microsoft Office стандарт функцияларни бажариш: ҳужжатни чоп этиш, маълумотни излаш ва уни алмаштириш, хатоликлар мавжудлигини текшириш, маълумотларни ҳимоялаш ва бошқалар;
- VBA дастурлаш тилини қўллаш ёрдамида иловалар яратиш.

Юқорида санаб ўтилган бандларнинг иккинчисига эътибор берилса, макрос деган термин ишлатилмоқда.

Олдинги параграфларда таъкидланганидек, макрос – бу фойдаланувчи томонидан ёзилган ишлар кетма-кетлигини ўз ичига олган макробўйруқлардир. Ёзиш Microsoft Office пакетига ўрнатилган VBA (Visual Basic for Application)- дастурлаш тилида амалга оширилади[4].

Маълумки, бир хил ишни такрор-такрор бажариш инсонларнинг асабига тегади. Шунинг Excel электрон жадвал билан ишлаш вақтида тез-тез такрорланувчи буйруқлар кетма-кетлиги бўладиган бўлса, уни дарров макрос кўринишида ёзиб қўйиш мақсадга мувофиқ. Яна ҳам яхшироқ бўлган бўлар

эди, агар фойдаланувчи қандайдир янги буйруқлар кетма-кетлигини кейинчалик ҳам бир неча марта ишлатиш мумкинлиги билса, биринчи учраганда макрос сифатида ёзиб қўйилиши ишнинг самарадорлигини оширади. Кейинги сафар макросдан фойдаланиб кетилаверади.

Одатда, макрослардан қўлланиш вазифасига нисбатан фойдаланиш самарали, аммо ёзилган макрос дастурий кодини VBA да дастурлашни ўрганишни бошлаш учун мисол сифатида фойдаланиш тўғри эмас. Сабаби бир неча макрослар ёзгандан сўнг, фойдаланувчида уларни янада такомиллаштириш ва янада эгилувчан қилиш истаги пайдо бўлади. Натижада VBA ёрдамида макросларга озроқ “интеллект” қўшиб янада такомиллаштириш мумкин ва уларни ҳолатга боғлиқ ҳолда турли ишларни бажаришга мажбур қилиш мумкин.

Энди эса VBA имкониятлари билан танишиб чиқайлик.

Иловаларга юқори даражада интеграциялашуви ҳисобига VBA тили дастурчига ўзига хос масалаларни ечишда Microsoft Officeнинг мавжуд имкониятларидан фойдаланиш учун шароит яратиб беради. Шу билан бирга VBA ушбу пакет иловаларига янги имкониятлар қўшиш орқали улар билан ишлашда уларнинг функционалиги ва қулайлигини оширади.

Excelда қўллаш жиҳатидан дастурига VBA да қуйидаги вазифаларни бажариш мумкин:

- ўзига хос ҳисоб-китобларни бажарувчи функцияларни қўшади;
- жадвалга маълумотларни киритишни тезлатади;
- фойдаланувчи тез-тез бажарадиган типик ишларни автоматлаштиради;
- мавжуд инструментлар панели асосида янги вазифаларни бажарувчи янги буйруқлар менюсини яратади;
- маълумотларнинг кўргазмалигини оширади (мисол учун катакларни турли рангларга бўяш орқали);
- катта ҳажмдаги маълумотларни ишлашни автоматлаштиради;

- катта ҳажмдаги бирламчи маълумотлардан берилган ахборотни танлаш билан боғлиқ бўлган турли кўринишдаги ҳисоботларни, бланкалар ва бошқа ишларни яратишни автоматлаштиради.

Юқоридаги ишларни амалга ошириш учун VBAда кўп бўлмаган етарлича содда дастурий кодларни ёзиш зарур бўлади[5,13].

VBA лойиҳа муҳарририда учта дарча очилади: юқори чап дарча – лойиҳа тузиш дарчаси (Project), пастки чап дарча - хоссалар дарчаси (Properties) ва ўнг томондаги дарча - дастурий модулар матни дарчаси.

Project (Лойиҳа) дарчаси ёрдамида лойиҳа тузилишини кўриб чиқиш, лойиҳа элементларини қўшиш ва йўқотиш, модуль тартибини таҳрирлаш учун (мас модуль нишонига икки марта чертиш орқали) очиш мумкин.

Properties (хоссалар) дарчаси лойиҳанинг ажратилган элементлари хоссасини бериш учун ишлатилади.

Умуман олганда, Microsoft Excelда VBA лойиҳа қуйидаги элементларни ўз ичига олади:

- VBAнинг стандарт модуллари;
- синф модуллари;
- ишчи китоблар модуллари;
- ишчи варақлар модуллари;
- диаграммалар модуллари;
- формалар.

VBAнинг стандарт модуллари – улар лойиҳа элементлари бўлиб, бевосита лойиҳанинг бошқа элементлари билан ишлатилувчи дастурий кодни ўз ичига олади(глобал функциялар, ўзгарувчилар, ўзгармаслар ва ҳакоза). Лойиҳа тузилмалари дарчасида стандарт модулар Modules йиғма жилдида гуруҳланади.

Одатда стандарт модуларда ишчи варақ ва ишчи китоб каби конкрет объектларга боғланмаган дастурлар ёзилади. Айнан ушбу модулда кўпгина мисоллар ёзилади.

Синф модуллари –бу фойдаланувчи (дастурчи томонидан ишлаб чиқилган) синф ишини амалга оширувчи дастурий код ёзиладиган модуллар.

Ишчи китоб модули (Microsoft Excel Objects папкасидаги китоб) - бу ишчи китобни бошқариш(манипуляция қилиш) бўйича қўшимча имкониятлар амалга ошириладиган синфлар модули. Мазкур модулга ёзилган дастур бевосита ишчи китоб объекти хосса ва усулларига мурожаат қилади.

Ишчи китоб модулининг одатий синф модулидан фарқи шундаки , VBA даги дастурдан ишчи китоб объекти экземплярлари яратиб бўлмайди – у автоматик яратилади.

Ишчи варақ модули (Microsoft Excel Objects папкаси) – бу китобнинг берилган ишчи варақларини бошқариш бўйича қўшимча имкониятларни амалга оширувчи синф модулидир. Ушбу модулларда ёзилган дастурлар ишчи варақ объектига хосса ва усуллари бевосита мурожаат қилади. Бу ерда ҳам VBA даги дастур ёрдамида ишчи варақ объекти экземплярлари яратиш керак эмас – фақат бўш модулли янги ишчи варақ яратиш мумкин.

Диаграммалар модули (Microsoft Excel Objects)- бу ишчи китобнинг айрим варақларига чиқарилган диаграммаларни бошқариш (манипуляция қилиш) бўйича қўшимча имкониятларни амалга оширувчи синф модулидир. Диаграммалар модулининг ўзига хослиги ишчи варақ модули ўзига хослиги билан бир хил.

Форма (Forms папкаси) – бу VBA лойиҳа элементи бўлиб, у ёрдамида фойдаланувчи билан биргаликда кўрсатиш учун мулоқот дарчаси яратиш мумкин. Форма икки қисмдан иборат: форма модули ва форманинг ўзи (мулоқот дарчаси).

Форма модули - бу форма ҳолати амалга ошириладиган синф модулидир. Ушбу модулни ишлатиш одатий синф модули кабилар. Форма модулини очиш учун Project (Лойиҳа) дарчаси контекст менюсининг View Code (кодни кўриб чиқиш) банди ишлатилади.

Очилган форманинг хоссасини ўзгартириш учун Properties(Хоссалар) дарчасидан фойдаланилади. Toolbox (Инструментлар панели) дарчасида

формага қўшиладиган бошқариш элементларини танлаш мумкин. Бошқаришнинг ихтиёрий элементини қўшгандан сўнг, унга икки марта чертиш билан форма модулидаги мос кодни таҳрирлашга ўтиш мумкин.

### **1.3. Макрос ёзишнинг дастлабки қадамлари ва массивлар билан ишлаш.**

**Excel да макрос ёзиш.** Айрим VBA-иловалар –улар ичида Microsoft Word, Excel ва Power Point лар макросларни ёзиш воситасига эга бўлиб, унинг ишлаши оддий магнитофон ишлашини эслатади. Макросни ёзиш воситаси улагандан сўнг у барча буйруқларни то ёзишни тўхтатиш тугмачасини чертмагунча ёзиб боради. Макрос ёзилгандан сўнг унда ёзилган буйруқлардан, керак бўлиб қолганда қайта фойдаланишингиз мумкин.

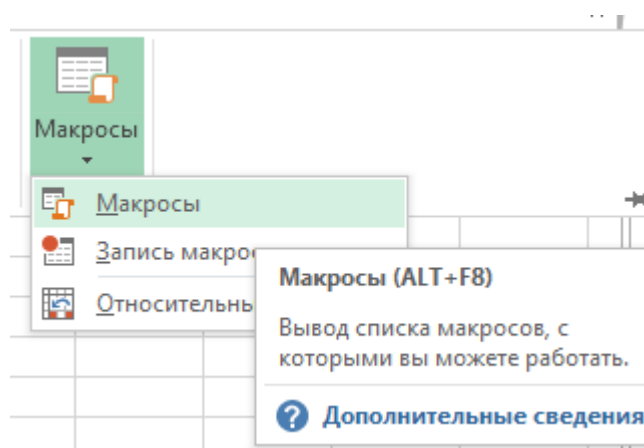
Макрос сўзи катта деган маънони англатади. Яъни унча катта бўлмаган буйруқларни битта катта буйруққа бирлаштириш деган маънони англатади. Макрос – VBA-дастурлар учун яна бир номдир. Макросларни ёзиш воситаси ўзининг иш вақтида фойдаланувчи фойдаланадиган ҳар бир буйруқни VBA нинг дастурий кодига ўтаради. Тугалланган макрос VBA процедураси сифатида сақлайди. Бизга маълумки, процедура – бажарилиши учун номи билан чақириладиган VBA дастурий кодининг бир бирлиги.

Макрос Accessда ҳам, VBAда ҳам мавжуд бўлиб, Accessда улар VBA-дастури эмас.

Excelда макрос ёзиш учун Excel ишга туширилгач, янги иш китоби очилганлигини текширинг. Сўнггра менюдан Вид>Макросы ни танласак, жадвалнинг ўнг юқори бўрчагида қуйидаги - расмдаги тасвир пайдо бўлади.

Агар янги макроснинг кодини ёзмакчи бўлсак, Макросы дарчасига кириш зарур ва Имя макроса сатрига макрос номини киритиш керак. Шундан сўнг мазкур дарчанинг ўнг томонидаги Создать тугмаси фаоллашади. Ушбу ишни ALT+F8 клавишлари ёрдамида ҳам бажариш мумкин.





**1.2-расм.**

2. Макросга ном қўйишда қуйидаги қоидаларга риоя қилиш лозим:

Макрос номи рақамлар билан эмас, балки ҳарф билан бошланиши зарур. Биринчи ҳарфдан сўнг рақам кетиши мумкин.

Макрос номи очик жой (пробел) га эга бўлмаслиги зарур. Сўзларни ажратиши учун сўзнинг биринчи ҳарфини катта ҳарф билан бошлаш лозим.

Макрос номида тиниш белгиларни ишлатиш ҳам ман этилади. Агар улар ишлатилган бўлса VBA Invalid procedure name маълумотини чиқаради.

Запис макроса мулоқот даражаси турли опцияларга эга. Қуйидаги айримлари билан танишамиз

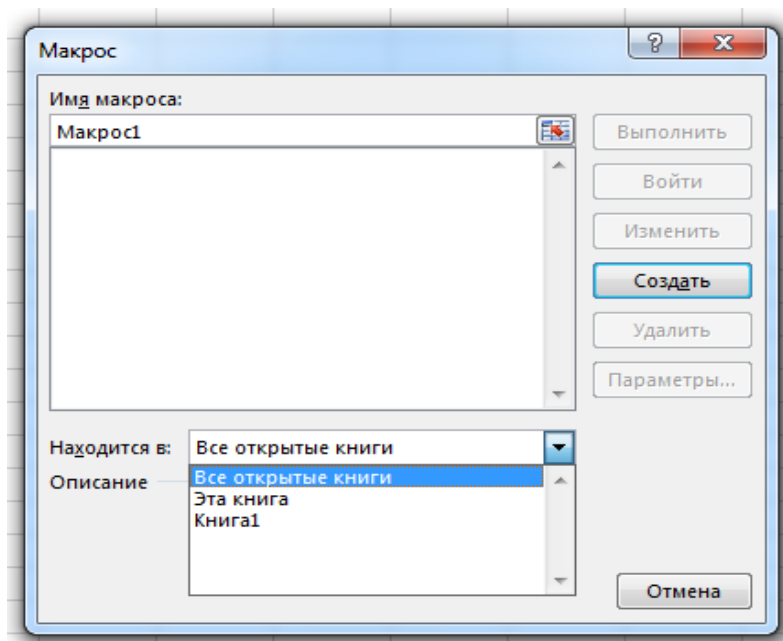
Макрос ёзувини тўлиғича киритиш учун жой бўлиши мумкин.

Макросни сақлаш жойини аниқлаш учун вариантлар бўлади. Одатда макросни мазкур ҳужжатнинг ўзида ёки ҳужжат шаблонида сақлаш таклиф этилади.

Ускуналар панелида тугмачага макрос бириктириш ёки унга тугмалар комбинациясини белгилаш таклиф этилади. Агар фойдаланувчи иловаси ушбу опцияларни кўзда тутса, иккиланмасдан ундан фойдаланиш мақсадга мувофиқ.

Пайдо бўлган мулоқот дарчасида макрос номини ёзинг, масалан, “Макрос”(1.3-расм). “Находится в” рўйхатида “Эта книга” сатри

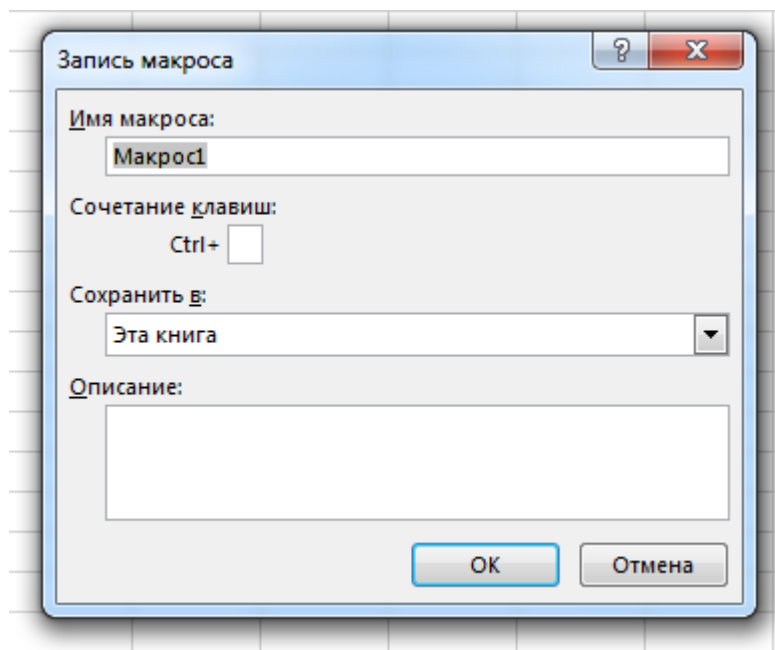
танланганлигини текшириб кўринг ва ОК ни босинг. VBA да дастурлашни бошлашнинг энг қулай йўли макрос ёзиш, яъни Visual Basic кодини ёзишдир.



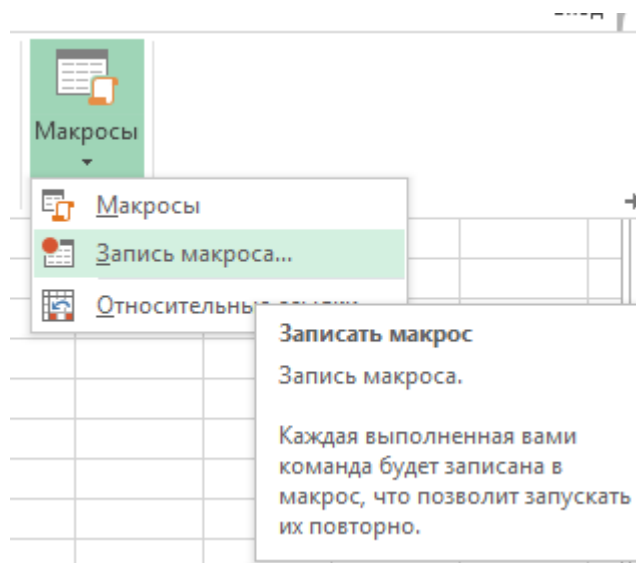
**1.3-расм.**

Агар ёзилган макрос барча очилган китобларга таъсир этишини хоҳласангиз, “Находится в” рўйхатидан “Все открытые книги” сатрини, агар фақат биринчи китобга таъсир этишини хоҳласангиз “Книга1” сатрини танлаш мақсадга мувофиқ.

Агар Excel жадвалида бажариладиган амалларни макрос сифатида ёзиб қўйиш учун “Запись макроса” дарчасини очиш талаб этилади ( -расм). Бу дарча фаол ҳолатда бўлса, сизнинг Excel жадвалида бажарган барча ҳаракатингиз макрос сифатида ёзилади ( 1.4-расм).



**1.4-рasm.**



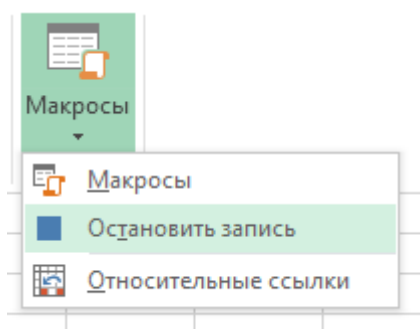
**1.5-рasm**

Масалан, куйидаги жадвални (1.6 -рasm)ни тузишни бошлайлик.

	A	B	C	D	E	F	G
1		1995	2000	2005	2010	2015	
2	1-dukon	456	568	789	256	894	
3	2-dukon	256	561	789	236	852	
4	3-dukon	567	741	852	258	693	
5							

**1.6-расм**

Жадвалда бажарилган барча ҳаракатлар макрос сифатида ёзилиб боради. Агар ушбу макросни кўрмоқчи бўлсак, Остановить запись сатрини танлаб макрос ёзишни тўхтатиш лозим (1.7 -расм). Шундан сўнг макрос тузилишини кўриш мумкин.( 1.8- расм)



**1.7 -расм.**

```

Sub Макрос1 ()
|
' Макрос1 Макрос
'
' Сочетание клавиш: Ctrl+r
'

Range ("A2").Select
ActiveCell.FormulaR1C1 = "1-dukon"
Range ("A3").Select
ActiveCell.FormulaR1C1 = "2-dukon"
Range ("A4").Select
ActiveCell.FormulaR1C1 = "3-dukon"
Range ("B1").Select
ActiveCell.FormulaR1C1 = "1995"
Range ("C1").Select
ActiveCell.FormulaR1C1 = "2000"
Range ("D1").Select
ActiveCell.FormulaR1C1 = "2005"
Range ("E1").Select
ActiveCell.FormulaR1C1 = "2010"
Range ("F1").Select
ActiveCell.FormulaR1C1 = "2015"
Range ("B2").Select
ActiveCell.FormulaR1C1 = "456"
Range ("C2").Select
ActiveCell.FormulaR1C1 = "568"
Range ("D2").Select
ActiveCell.FormulaR1C1 = "789"

```

### 1.8-расм.

Ушбу макрос бажарилишини текшириб кўрмоқчи бўлсангиз, очилган ишчи китобингизни ёпиб, Вид > Макросы менюсига киринг “Макрос открыть” номини танланг ва “Выполнить” тугмасини босинг. Юборидаги жадвални тузиш яна бошқатдан бошланади. Макрос ёзиш жараёнида макроснинг каерда сақланаётганлигига эътибор берилиши лозим. Макрос ишга тушиши учун макрос ёзилган китоб ишга туширилган бўлиши зарур. Макросни исталган вақтда ишга тушириш учун эса уни “Личная книга макросов”. Макрослар шахсий китобида сақлаш лозим.

Ўзингиз яратган макросни кўриш учун, макрос ёзилган китобнинг очиклигини текширинг ва сўнгра макрослар рўйхатни очиш учун Вид Макросы ни танланг ва зарурий макросни танлаб, уни ўзгартириш учун “Изменить” тугмасини босиш етарли.

Visual Basic энди очик ҳолда, унинг ойнасида эса сизнинг “Макрос1» номли макросингиз турибди. Макрос кодининг юқори қисмига эътибор берсак, ушбу макросни автоматик тарзда бажариш учун CTRL+r тугмаларини

Фойдаланувчи макрос кодини ўзгартириши ёки янги йўриқномалар ёзиб, макросни кенгайтириши мумкин.

Масалан, макросга қуйидаги кодни ёзсангиз, Excel биринчи катакка бугунги санани ёзади:

Active Sheet. Cells (1,1). Value = Str\$ (Now)

Бошқа ишчи китобни очиш учун кодни осонликча ўзгартиришингиз мумкин. Кодни бу тарзда, яъни қўлда тахрир қилиш имкони билан ёзиш катта имкониятларга эга.

**Автоматик тарзда бажарилувчи макрослар.** Бирорта хужжат очилганда ёки маълум бир ҳаракатга жавоб тариқасида бажарилиши мумкин бўлган макрос яратишингиз мумкин. Қуйида ана шундай макрос яратишга мисол келтирилган .

Excel ишчи китобини ишга туширишга жавоб тариқасида бажарилувчи макросни тузиш учун қуйидаги ишларни кетма-кет амалга ошириш зарур.

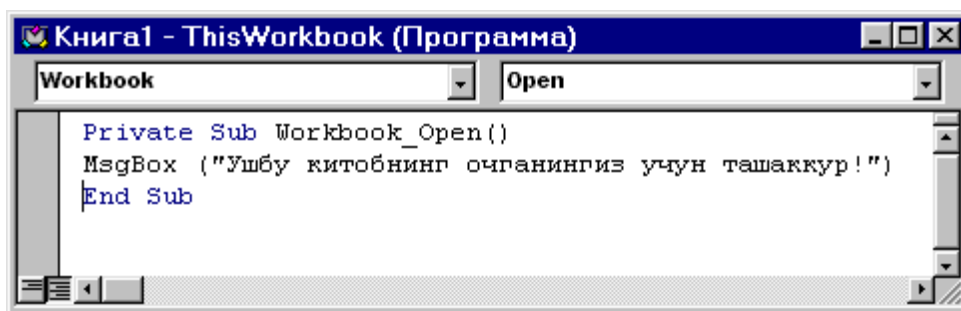
1. Excel ишчи китобларидан бирортасини очинг ёки янгисини яратинг. Visual Basic муҳарририни очинг. Агар лойиҳа ойнаси очилмаган бўлса “Вид” (View) менюсидан “Окно проекта” (Project Explorer) сатрини танланг. Лойиҳа ойнасида This Workbook сатрини топинг ва ўнг босинг. қалқиб чиқувчи менюдан “Программа” (View code) сатрини танланг.

2. Чап рўйхатдан Workbook сатрини танланг. Ўнг томондаги рўйхатда Open (Открыть) ҳодисаси кўрсатилганлигини текшириб кўринг, кейин эса исталган кодни, мисол учун, ташаккур эълон қилувчи панель кодини ёзишингиз мумкин.

Private SuB WorkBook – Open

MsgBook (“Ушбу китобни очганингиз учун ташакур”)

End SuB



**1.9-расм.**

Word хужжатлари ҳам Open ходисасига эга. Рўйхатда кўрсатилган бошқа кўпгина ходисалар учун ҳам код ёзишингиз мумкин.

3. Visual Basic муҳарририни ёпинг ва ишчи китобни сақланг. Сўнгра уни очинг, китоб очилиши билан код ишга тушади ва ташаккур эълон қилинган панел пайдо бўлади. Бундан ташқари, вақтни, санани кўрсатувчи ёки глобал ўзгарувчиларнинг бошланғич қийматларини кўрсатувчи код ёзиш мумкин.

Сиз макросга эга хужжатни очаётганингизда огоҳлантирувчи панель пайдо бўлади. Бу мулоқот дарчасини макровируслардан ҳимояланишга имкон беради. Агар сиз ишчи китобни ва макросни ўзингиз яратган бўлсангиз ёки улар олинган манбага ишончигиз комил бўлса, бу огоҳлантирувчи мулоқот дарчасини бекор қилишингиз мумкин. Лойиҳа дарчаси ҳақида батафсилроқ тўхталсак, Visual Basic For Application нинг лойиҳалар дарчаси (Project Explorer) сиз яратган шакл ва кодлар қаерда сақланаётганлигини кўрсатади ҳамда уларни таҳрирлаш учун очиш имконини беради. Office матнидаги Visual Basic коди ва шакллари хужжатларда ёки андозаларда сақланади. Яъни Office дастурларида яратилган хужжатларни VBA маҳсули деб аташ мумкин. Сиз Office хужжатларини ёки андозаларини сақлаш жараёнида улардан Visual Basic элементларини ҳам сақлайсиз.

Лойиҳалар дарчасининг қай тарзда ишлашини билиб олиш учун уни амалда синаб кўрган маъкул.

“Объект” (Object) тугмасини объектни кўриш учун босинг

Танланган объект кодини очиш учун “Программа” (View Code) тугмасини босиш кифоя. Танланган ишчи варақ ёки китобни икки марта босинг ёки Open (очиш) ёки Change (Алмаштириш) ҳодисаларига жавоб тариқасида бажарилувчи код ёзиш учун “Программа” (View Code) тугмани босинг. Папкалар шаклида ташкил қилинган лойиҳаларни кўриш учун “Папка” (Folder) тугмасини босинг.

Excelда очиладиган ҳар бир ҳужжат лойиҳа ҳисобланади. Макрослар шахсий китоби ҳам шу ерда кўрсатилган.

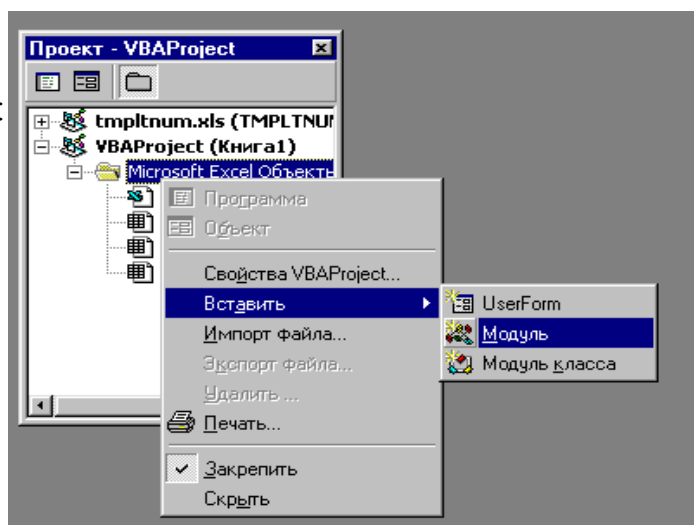
Модулар функциялар, кетма-кетликлар ёки глобал ўзгарувчилар қийматлари кодига эга бўлиши мумкин

**Wordда лойиҳа ойнаси.** Word лойиҳа ойнаси Excel лойиҳа ойнасига ўхшашдир. Лекин ўзига хос маълум фарқлари ҳам бор. Масалан, Word лойиҳасида Normal андозаси бор. Ундан ташқари, бу ерда мурожаат ҳисоби бошқа усулда ташкил этилган. Normal андозаси коди ва шаклини ҳар қандай ҳужжатга қўллаш мумкин. VBA га мурожаатлар Visual Basic автоном версиясидаги каби ишлайди.

**VBA да катаклар қийматларини ёзиш .** Excel да Visual Basic билан ишлаш жараёнида кўпинча катакларга қиймат беришингизга тўғри келади. Қуйида келтирилган мисолда ишчи китобга қиймат ва формула ёзиш тартиби келтирилган.

қуйидаги  
бажариш зарур:

Бунинг учун  
ишларни

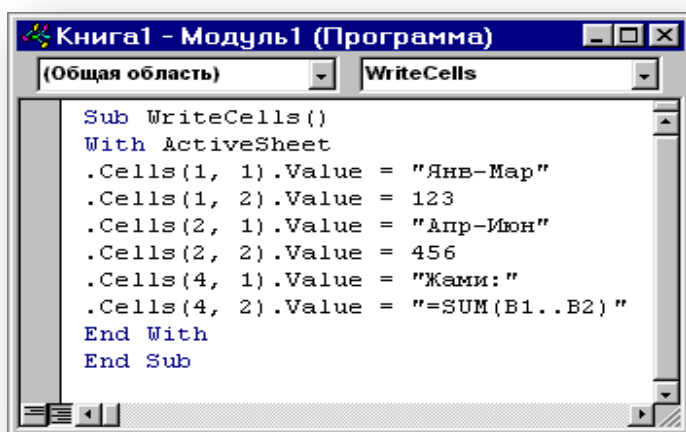




## 1.10-расм

1. Янги китоб яратинг ва Visual Basic муҳарририни очинг. “Вид” (View) менюсидан “Окно проекта” (Project Explorer) сатрини танланг. Лойиҳани Сичқонча ўнг тугмаси ёрдамида босинг ва контекст менюдан Вставить > Модуль (Insert>Module) сатрини танланг. Бу модуль лойиҳа дарчасида модуль 1 (Module 1) номи билан чиқади.

2. Код муҳаррири автомат тарзда очилади. Янги тартиб (процедура) яратиш учун Insert менюсида Procedure (Процедура) сатрини танлашингиз мумкин эди, лекин энг қулай йўли Sub WriteCells ни муҳаррирга ёзишдир. Enter тугмасини боссангиз,



## 1.11-расм

VBA End Sub операторини автоматик тарзда ёзиб қўяди. Ҳозиргина WriteCells номи билан бўш макрос яратдингиз. Унинг кодини ёзамиз:

With Active Sheet

.Cells (1,1) . Value = “Янв-Мар”

.Cells (1,2) . Value = 123

.Cells (2,1) . Value = “Апр-Июн”

.Cells (2,2) . Value = 456

.Cells (4,1) . Value = “Жами”

.Cells (4,2) . Value = “ = Sum (B1...B2)”

End With

Номланган оралиқ – VBA кодида ячейкаларни белгилашнинг қулай усулидир. Мисол учун, агар сизни “ Totalval ” оралиғи кизиқтирса, кодда бу оралиққа Range (“Totalval”) номли мурожаатларини бериш мумкин. Excel да формула ёзишда, формулани оддий матндан фарқлаш учун “=” дан фойдаланиш зарур.

3. Макрос ишга туширилгач, автоматик тарзда Excel да код ичига ёзилган қийматлар пайдо бўлади.

4. Макросни ишга туширинг ва Word ҳужжатини текшириб кўринг. Матн автоматик киритилган эканлигини кўриш мумкин.

Фойдаланувчи макросни ёки Visual Basic муҳарриридан, ёки бевосита Word ҳужжатидан ишга туширишингиз мумкин, бунинг учун Word нинг Вид>Макросы менюсидан фойдаланиш зарур.

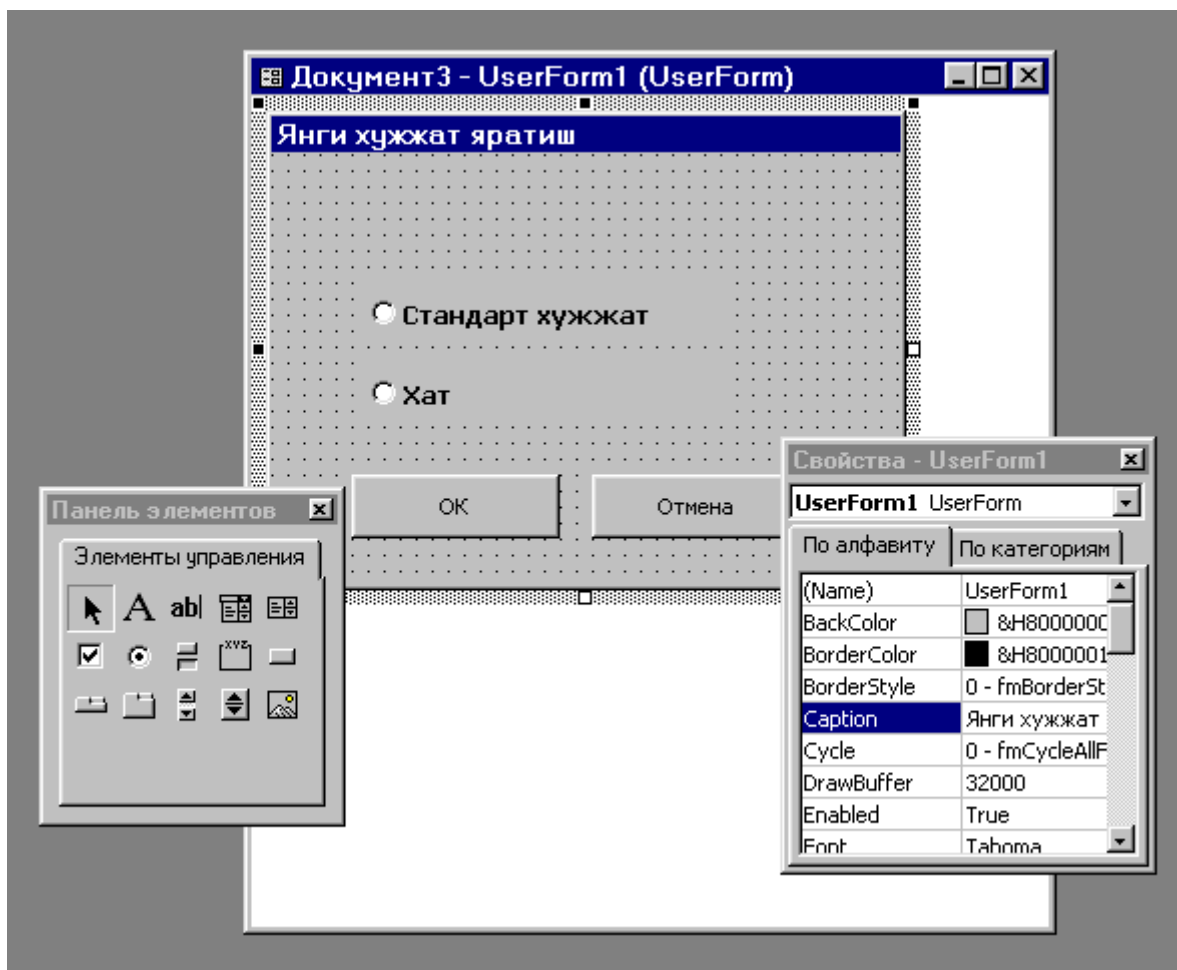
Word да мулоқот дарчаси яратиш ва ундан фойдаланиш.

Мазкур лойиҳадан мақсад, макросни тузиш бўлиб, у иккита андозадан бири асосида янги ҳужжат яратади.

1. Word да янги ҳужжат яратинг ва Visual Basic муҳарририни очинг. Янги ҳужжат билан боғлиқ лойиҳа дарчасига келиб сичқончасанинг ўнг тугмасини босинг ва менюдан Insert>UserForm ни танланг.

Жавоб тариқасида янги шакл очилади ва асбоблар панели пайдо бўлади.

Агар сиз макросдан мунтазам фойдалансангиз, уни ҳужжатда эмас, балки Normal.dot андозасида сақлаган маъкул. Тажриба тариқасида яратилган макросларни эса ҳужжатнинг сўзлари сифатида сақлаш мақсадга мувофиқ.



1.12 -расм.

Шаклни Visual Basic нинг автоном версиясида ишлайдиган қилиб яратиш учун - расмда кўрсатилган тарзда шаклга қайта улагич тугма ва иккита сарлавҳали тугма жойлаштириш лозим. Юқорида қайта улагич тугманинг Value хусусиятини True ҳолатига ўтказиш зарур. “OK” тугмачаси учун Default, “Отмена” тугмаси учун эса юқорида кўрсаткичини True ҳолатига ўтказиш керак. Enter ёки Escape клавишлари босилганда мулоқотнинг қандай ишлашини ҳодисани Default ва Cancel тугмалари белгилайди. Шакл код муҳаририни очиш учун икки марта босинг ва чап томондаги рўйхатдан “Умумий соҳа” (General) сатрини танлаш мақсадга мувофиқ. Ўнг томондаги рўйхатдан “Тавсиф” (Declarations) сатрини танлаб, код муҳарририни очиб қуйидаги кодни ёзинг:

```
Public DocOption As Integer
```

Бу ўзгарувчи фойдаланувчи томонидан мулоқотда танлаш натижаси қийматини сақлайди.

“ОК” тугмасининг Click (Босиш) ҳодисаси учун код муҳарририни очинг ва қуйидаги кодни ёзинг:

```
If Option Button 1. Value = True Then  
    User Form 1. DocOption = 1  
Elseif OptionButton 2. Value = True Then  
    UserForm 1. DocOption = 2  
Else  
    UserForm 1. DocOption = 0  
End If  
Me. Hide
```

“Отмена” тугмачасини икки марта босинг ва ягона сатрни киритинг:

```
Me. Hide.
```

Сўнгра лойиҳага модуль қуйинг ва агар бор бўлса, уни очинг.

Код муҳарририни очинг ва қуйидаги кодни ёзинг:

```
SuB DocStart ()  
    UserFopm 1. Show  
    Select, Case UserForm 2. Doc Option  
        Case1 стандарт, документ  
        Document Add Normal. DOT фойдаланади  
        Case 2 Документ письмо
```

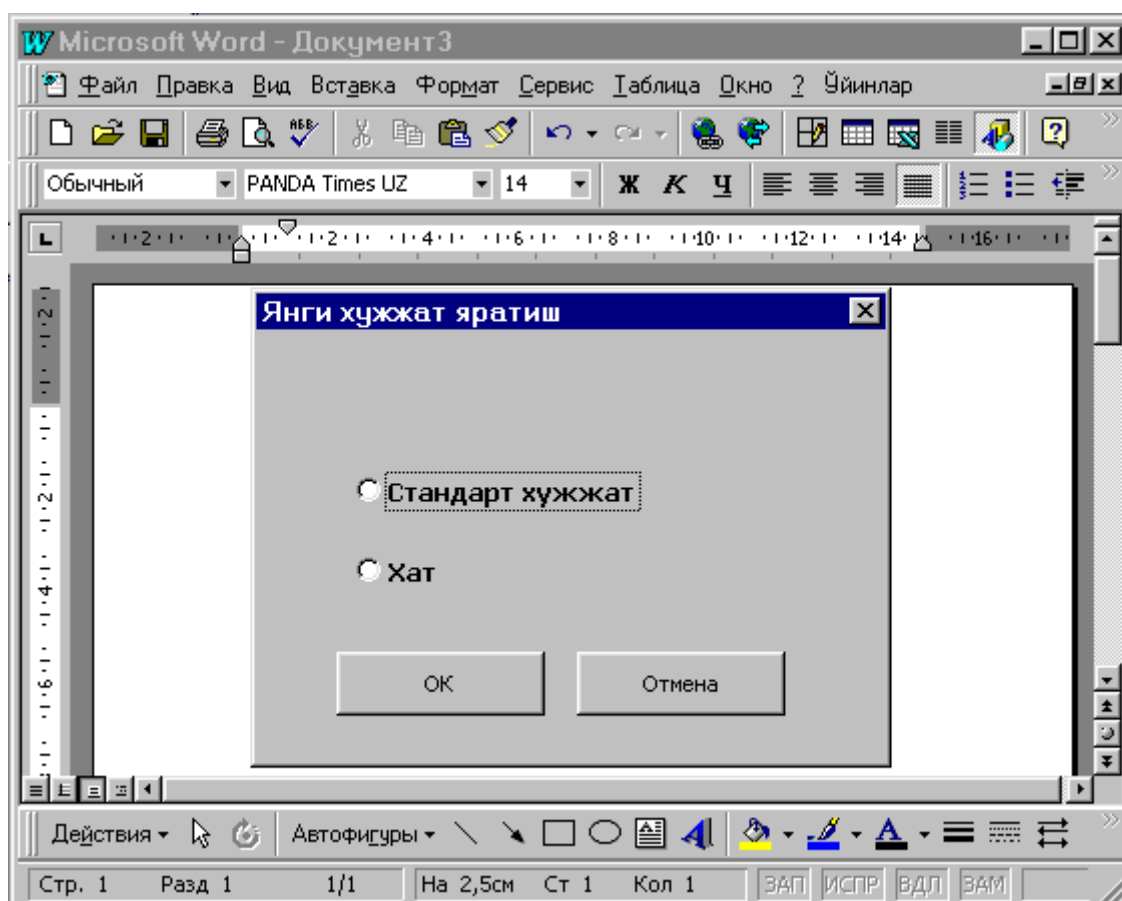
Documents. Add Template:= “C:\Programm Files-\Microsoft Office\Шаблоны \ письма и факсы\ Стандартное письмо. dot\

End Select

End SuB

Тизимда Wordнинг ўрнатилган жойига караб, андоза йўлини ва номини тўғри кўрсатишингиз керак

Visual Basic ни ёпинг ва Wordга кайтинг. Кейин DocStart макросини ишга туширинг, у янги танланган андоза асосида хужжат яратади.



1.13-расм.

Дастурлаш жараёнида кўпинча бир хил мазмун ва типга эга бўлган катта миқдордаги маълумотлар билан ишлашга тўғри келади. Бундай маълумотларни сақлаш учун одатда массивлардан фойдаланилади. Массив –

бир хил типда қийматлар тўплами бўлиб, улар битта ўзгарувчига бирлаштирилган[7-9].

VBA да массивлардан фойдаланиш учун имконият етарли даражада. Массивларни эълон қилиш учун қуйидаги кўрсатмадан фойдаланилади:

**Public | Private** массив номи ([Ўлчами]) [**As** тип номи] [,Массив номи ([Ўлчами]) [**As** тип номи] ...

ёки

**Dim | Static** Массив номи ([Ўлчами]) [**As** тип номи] [,Массив номи ([Ўлчами]) [**As** тип номи] ...

Биринчи кўрсатма массивни модул даражасида, иккинчи кўрсатма эса массивни процедура ёки функцияда эълон қилишида ишлатилади.

Массивни эълон қилишнинг оддий ўзгарувчини эълон қилишдан фарқи шундаки, бу ерда ўзгарувчи идентификатордан кейин массив ўлчами кўрсатилиши лозим. Агар ўлчами кўрсатилмаган бўлса, динамик массив яратилиб, ўлчами дастур бажарилиши давомида ўзгартирилиши мумкин. Динамик массивлар учун эълон қилиш кўрсатмаси формал кўринишда бўлади, яъни массивни ишлатиш учун **ReDim** кўрсатмаси орқали қайта эълон қилиш зарур бўлади.

Ўлчами кўрсатиш учун унинг қуйидаги шаклларида фойдаланиши лозим.

*Қуйи чегара* **To** *Юқори чегара* | Элементлар сони -|,

*Қуйи чегара* **To** *Юқори чегара* | Элементлари сони

VBA да ўлчамлари сони 60 дан ошмаган кўп ўлчовли массивлар яратишга рухсат этилган. Ўлчов ўлчами вергул билан ажратилади. *Қуйи чегара* **To** *Юқори чегара* кўринишида массив ўлчами берилишида қуйи ва юқори чегара аниқ кўрсатилиши лозим. (мисол учун 20 **To** 100). Массив ўлчамини беришда массивнинг берилган ўлчовида элементлар сони кўрсатилиши ҳам мумкин. Бундай берилишда қуйи чегара сифатида жимжитликдаги қиймат ишлатилади. Қуйида массивни эълон қилишга мисол келтирилган.

**Dim** avarValues ()

**Dim** astrValues (1 **To** 10) **As** String, astrValues2(10) **As** String.

**Dim** aintValues (1 **To** 10) **As** Integer, aintValues (10,3) **As** Integer

Жимжитликда массив ўлчамининг қуйи чегарасини эълон қилиш учун **Option Base** директиваси ишлатилади.

### **Option Base 0**

ёки

### **Option Base 1**

VBA да динамик массив ўлчами дастур ишлаш вақтида ўзгартириш мумкин. Бунинг учун **ReDim** кўрсатмасидан фойдаланилади.

**ReDim [Preserve]** Массив номи ([Ўлчами]) [**As** тип номи] [,Массив номи ([Ўлчами]) [**As** тип номи] ...

Ушбу кўрсатма элементлари вазифаси **Dim** кўрсатмаси элементлари вазифаси билан бир хил. Массив элементлари типини, агар массив номи – **Variant** типигаги ўзгарувчи идентификатори бўлганда ишлатилиши мумкин. Мисоллар

**ReDim** astrValues (1 **To** 10) aintValues (10,20)

**ReDim** varArray (2 **To** 4) **As** Boolean

**Preserve** да массив элементлари қийматини йўқотмасдан масств ўлчови ўзгартиришда фойдаланиш мумкин. Аммо ушбу калит сўздан фойдаланишда эҳтиёт бўлиш талаб этилади.

Қуйида **ReDim** кўрсатмаси **Preserve** калит сўзи билан биргаликда ишлашига доир мисоллар келтирамиз.

**ReDim** astrValues (1 **To** 5, 1 **To** 10)

Массив ўлчами камайтириш

**ReDim Preserve** astrValues (1 **To** 5, 1 **To** 15)

VBA да массивларнинг қуйи чегарасини ихтиёрий қилиб белгилаш мумкин. Чунки дастур ишлаш вақтида массивнинг қуйи чегарасини билиш мақсадга мувофиқ. Бунинг учун VBA да икки функция киритилган бўлиб, уларнинг формати қуйидагича:

**LBound** (Массив номи [, ўлчам рақами])

**RBound** (Массив номи [, ўлчам рақами])

Массив элементларига мурожаат учун VBA да ушбу элемент рақами ўзгарувчи массив идентификаторидан кейин кавс ичида бериш лозим. Агар бир нечта элементларига мурожаат қилинса, рақамлар вергул билан ажратилади.

**Мисол.**

```
Int Num = aint|Values (16)
```

```
intNum= aintValurs (3,5,17)
```

VBA тилидаги маълумотларнинг универсал типи учун ишлатиладиган **Variant** дан массивлар билан ишлашда ҳам фойдаланиш мумкин/

**Мисол.**

```
Dim aintValues (1 To 5) As integer
```

```
Dim varArray
```

```
varArray=aintValues
```

```
varArray (1)=1
```

```
varArray (2)=20
```

```
var Array (3) =17
```

VBA да массив қийматини тез тўлдириш имконияти мавжуд. Ушбу имконият **Array** функцияси ёрдамида амалга оширилади. Унинг формати қуйидагича:

**Array** (элементлар рўйхати)

Функция аргументи сифатида вергул билан ажратилган қийматлар рўйхатини қабул қилади. Мазкур функция **Variant** типдаги ўзгарувчида сақланувчи берилган қийматлар билан тўлдирилган массивни қайтаради. Қуйида **Array** функциясидан фойдаланишга доир мисол келтирилган.

```
varArray= Array(5,6,14,58,63)
```

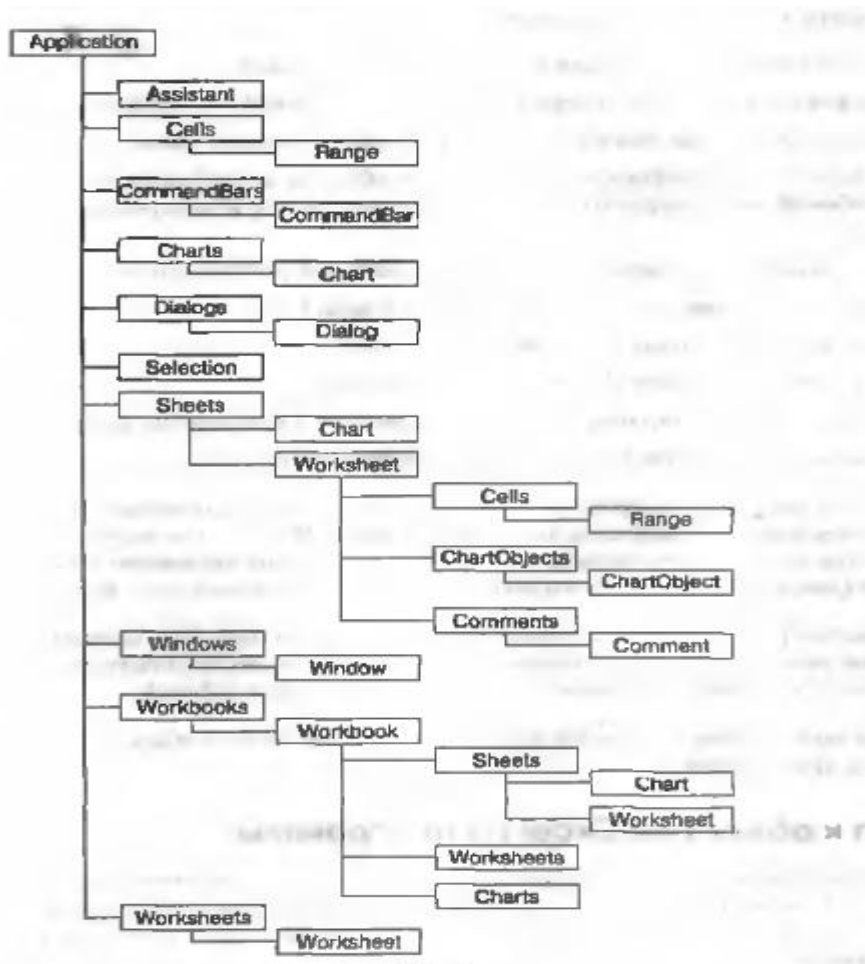


## П.БОБ. VBA ОБЪЕКТЛАР ИЕРАРХИЯСИ, ФАЙЛЛАР БИЛАН ИШЛАШ ВАЎҚУВ ЭФФЕКТЛАРИ ҲАМДА ТРЮКЛАРИ ЯРАТИШ АМАЛИЁТИ

### 2.1. VBA да дастурий код ёзишнинг объектлар иерархияси

Microsoft Office дастурлари учун VBA код ёзиш учун объектлари иерархияси тўғрисида маълумотга эга бўлиш керак. Қуйида биз Excel электрон жадвали объектлар иерархияси билан танишиб чиқамиз[10,21,22,23].

Excel объектлари иерархиясининг энг юқорида Application (илова) объекти бўлиб, у барча Excel дастурлари иш фаолияти учун жавоб беради(2.1-расм).



2.1-расм. Microsoft Excel объектлар тузилмаси

Кўпгина VBA дастурлари иш фаолиятини ошириш учун экранни янгилаш ўчириш лозим. Жимжитликда Excel VBA –коди бажарилиши жараёнида бажарилган ишчи китобдаги ҳар бир ўзгаришни тасвирлайди.

Бу эса ишлар бажарилиши тезлигини анчагина кескинлаштиради. Excelнинг ушбу иш режасини ўчириш учун Application Screen Updating (Экранни янгилаш) функциясидан фойдаланиш лозим.

#### Application Screen Updating-False

Процедура охирида Screen Updating хоссаси учун True қийматини тиклашни унутмаслик керак. Акс ҳолда фойдаланувчи олинган натижани экранда кўрсатмайди. Screen Updating хоссасидан фойдаланишда Application объектини аниқ кўрсатиш лозим. Аммо кўпгина ҳолларда Application объекти хоссасини тўғридан-тўғри ишлатиш мумкин. Мисол учун, Application объекти бўлган Activesheet хоссаси жорий моментда фаол бўлиб ишчи китоб ёки диаграмма тегишли бўлади. Кодда бундай вараққа мурожаат қилиш учун Application Activesheet ўрнига фақатгина Activesheet ишланмаси бўлади. Application объекти бўлган Workbooks коллекциясида берилган моментда очик бўлган барча китобларни ўз ичига оладиган маълум бир ишчи китобни идентификациялаш учун унинг номини қўштирноқ ичида келтириш керак. Мисол учун, Workbooks (“Шартнома тўлови.xls”) Activate.

Албатта ишчи варақлар ҳам тенг ҳуқуқли объектлардир. Улар Worksheets коллекциясига тегишли бўлиб, улар ҳам ишчи китобни идентификациялаш каби идентификацияланади.

Worksheets (“Лист3”) ифода Лист3 номли варақни кўрсатмоқда. Бўлак варақда жойлашган диаграмма учун айрим олинган Chart объекти Charts коллекциясига тегишли. Унга мурожаат Charts (“Parts Chart”) ифода ёрдамида амалга оширилади. Ишчи варақда фойдаланилган диаграммалар билан ишлаш бироз фарқ қилади.

Ишчи китобларни тасвирлашни Window (Дарча) объекти ёрдамида ўзгартириш мумкин. Айрим ҳолларда ишчи китоб тузилишини ўзгартиришда фойдаланиладиган Window объектлари Application объекти бўлган Windows

коллекцияси таркибига киради. Дарчага мурожаат қилиш уни ўзига олган ишчи китоб файл номини кўрсатиш орқали амалга оширилад.

Windows (“Тўлов эгри чизиғи. xls”)

Агар битта ишчи китоб учун бир неча дарча очик турган бўлса, ишчи китоб номидан кейин “:” билан ажратилган дарча номери кўрсатилиши керак. Масалан, Windows (“Тўлов эгри чизиғи. xls: 2”).

Excelда Cell (катак) объекти йўқ. Шунинг учун код ёзиш куйидагича амалга оширилади: VBA кодида катакни кўрсатиш учун Range объектидан фойдаланилади. Excelда Range объекти бир ёки варақнинг узлуксиз соҳасининг бир қанча катакларни ўз ичига олиши мумкин. Excelда ёзилган дастурий код таъсир этувчи усуллари мавжуд. Ушбу мақсадга эришиш учун куйидагилар ишлатилади.

**Катакка стандарт мурожаат.** Катакка A1 –усулида мурожаат қилиш Range объектлари билан ишлаш усулининг энг содда усулидир. Диапазонни аниқлаш учун Range калит сўзидан кейин мурожаат қилинадиган диапазон кўштирноқ ичига ёзилади.

Мисол учун: Active Sheet. Range (“B3”)

WorkSheets (“Sheet2”). Range (“M5:S20”)

**Номланган диапазонлар.** Агар ишчи варақлар номланган диапазонларга эга бўлса Range нинг VBA – объектлари уларга таяниши мумкин, масалан: WorkSheets (“Молиявий ҳисобот”) Range (“Фоиш тўлови”). Дастурий коднинг ўзида диапазонни номлаш учун Name (номи) функциядан фойдаланилади. Range(“A3:B4”).Name = “Прайс-лист”

**Қисқа ёзув.** Range объектларни тез-тез ишлатишда Excel диапазонни аниқлаш учун A1 усулида ёзишда ёки номни ишлатган ҳолда ёзишда Range калит сўзини ташлаб кетиш мумкин. Ушбу усулни амалга оширишда мурожаат қилинадиган катак ёки катаклар соҳаси квадрат қавс ичига олиши лозим.

Activsheet [“A1:F26”]

[“Чораклик ҳисоботи”]

**WorkSheets** объектининг **Cells** хоссаси. WorkSheets объектига Cells хоссасида координаталар кўрсатилмасдан ишлатиш ушбу ишчи варақнинг барча катакларини ўз ичига олади. Application. Cells берилган вақтда фаол бўлиб турган варақнинг барча катакларига мурожаат қилинади. Агар бирмунча локал диапазондаги катакларга мурожаат қилиш керак бўлса Cells хоссаси учун сатр ва устунлар координаталарининг сонли қийматларини кўрсатиш талаб этилади. Мисол учун, E3 катакни ўз ичига олган диапазонни кўрсатиш учун қуйидаги кодни киритиш зарур.

WorkSheets(“Тўлов варағи”). Cells (3,5)

**Selection (Ажратилган) хоссаси** . Агар дастурий код фойдаланувчи ажратган соҳагагина таъсир қилиши талаб қилинса Selection хоссасидан фойдаланилади.

**ActiveCells ( фаол катаклар) хоссаси.** Ушбу хосса берилган даражада фаол бўлган катаклар соҳасига мурожаат қилишда ишланади. Объект спецификаторисиз ишлатишда ActiveCells хоссаси фаол даражага мурожаат қилади. Value StorageBin= ActiveCells. Value

**Worksheet объектининг Rows (сатрлар) ёки Columns (устунлар)** хоссаси номери кўрсатилган устун ёки сатрнинг барча элементларига мурожаатни амалга оширади. Мисол учун.

Workbooks (“IOU. xls”).Worksheets (“Sheet Shootout”). Column(5).

Фойдаланувчи томонидан аниқланган объектларга мурожаат қилишда, диапазон объект бўлганлиги учун дастлаб унга ном қўйиб, кейинчалик ўша ном билан мурожаат қилиш мумкин. Мисол учун.

DIM RanGer As Range

Set RanGer = Worksheets (“Лист1”). Range (“B12:H13”)

Диапазон (соҳа) хоссасидан фойдаланиб, бир ҳаракат билан бутун бир соҳа характеристикасини ўзгартириш мумкин. Қуйидаги сатр билан берилган диапазондаги катакларда жойлашган матнни шрифтини ўзгартириш мумкин.

Worksheets(“Лист1”). Range (“B12:H13”). Font. Sife=14

Албатта , бу ерда With тузилмани ишлатиш мақсадга мувофиқ . Чунки ушбу тузилма билан бир неча хоссалар ёки бутун диапазон билан ишлаганда мақсадга мувофиқ

With SomeRange

.Value=20

.Font.Name = "Garamond"

.Font.Italic=True

.Locked =True

SomeRange =.Name

End With.

Диапазоннинг айрим каттаклари билан ишлашда Excelда бир ҳаракат билан кўп каттакли соҳада мавжуд бўлган қийматларни ўзгартириш усули йўқ. Шу сабабли соҳадаги барча каттакларни цикл оператори билан кўриб чиқиш мумкин, яъни For Each..., Next ушбу техникани қўллашда соҳадан каттаклар сонини билиш ҳам шарт эмас. Масалан.

For Each aCell In Selection

aCell.Value= aCell.Value+10

Next

Кўпинча айрим олинган катакда амал бажариш ёки умуман бажариладиган амални танлаш тўғрисида қарор қабул қилишдан олдин катак тузилишини текшириш керак бўлади. Катакнинг жорий қиймати асосланиб, дастурий код катакни ўзгартириш ёки бошқа амаллар бажариш бўйича қарор қабул қилиш мумкин. Бундай ҳолда ҳам For Each... Next циклдан фойдаланиш мумкин.

For Each In Worksheets ("Лист"). Range("A5:B10")

If IsNumeric(aCell)Then

Select Case aCell

Case 5 to 10

aCell.Font.Underline=x1UnderlineStyleSingle

Case 10 to 20

```

aCell.Font.Italic=True
Case Is>20
aCell.Font.Bold=True
End Select
End If
Next

```

**Ажратилганлар билан ишлаш.** Агар катак ва бир неча катаклар қиймати ёки форматини ўзгартириш учун фойдаланувчи авваломбор уларни ажратиб олиш зарур. Аммо VBAда катакларни ажратиш шарт эмас, чунки амаллар бажариладиган катакларни идентификациялаш учун Range объектидан фойдаланиш мумкин. Аммо VBAда ажратилганларга боғлиқ шундай воситалар борки, улар икки имкониятни амалга ошириш учун ишлатилади: дастурий код фойдаланувчи томонидан ажратилган катакларни ва код фойдаланувчига қайси катакларда муҳим амаллар бажарилаётганлигини аниқлаш мумкин.

Кўп ҳолларда фойдаланувчи коди ажратилган катакларга таъсир ўтказди, чунки Excelда бу ишларни ўрнатилган буйруқлар амалга оширади. Фойдаланувчи томонидан ажратилган соҳага мурожаат қилиш учун Application ёки Window объекти бўлган Selection (ажратиш) хоссаси ишлатилади. Application объекти бўлган Selection хоссаси ҳозирга вақтда фаол бўлган ишчи варақда ажратилган соҳани қайтаради.

```
Application.Selection.Value =20
```

```
Selection.Value =20
```

Агар мурожаат доимо маълум бир дарчага тегишли бўлишлик (жорий моментда қайси дарча фаол бўлишидан қатъий назар) шартни қаноатлантириш учун дарча Selection хоссасини ишлатиш зарур. Қуйидаги мисолда ушбу техника намоиш этилган ҳамда бу ерда жорий ажратилган соҳага кейинчалик қайта мурожаат қилиш учун қандай қилиб соҳага объектли мурожаат ўргатиш кўрсатилган.

```
DIM SelRange As Range
```

```

Set SelRange=Windows (“Ўйинчоқлар инвентиразацияси. xls”) Selection
With SelRange
    .CheckSpelling
    .AutoFit
    .Copy
End With

```

VBA ишчи вараққа ўзгаришлар киритар экан, бу ишни фойдаланувчи сезиши учун Select (ажратмоқ) усулидан ажратилган бўлакни керакли соҳа, диаграмма ёки бошқа бир жойга силжитишда фойдаланилади. Select усули Excelда мавжуд ҳар қандай объектга қўлланилади. Жумладан, Chart (Диаграмма) объектга ва унинг барча компонентларига Shape (шакл) объектига ва албатта Range объектига ҳам қўлланилади.

Диапазон (соҳа)ни ажратиш учун дастлаб у жойлашган ишчи варақни фаоллаштириш зарур. Кейин Range объекти учун Select усулини қўллаш лозим. Уни қуйидаги мисолда кўриш мумкин.

```

With Worksheets (“чизиқлар статистикаси”)
    .Activate
    .Range (“Кирим бўл.маган чиқимлар”). Select
End with

```

Select усули Worksheet объекtlари учун, кўриниб турибдики, кўрсатилган ишчи варақни фаоллаштиришдан бошқа ҳеч қандай иш бажарилмайди ва мавжуд ажратилган бўлакка ўзгартиришлар киритилмайди. Бошқача айтсак, у Activate ишчи варағи усулига эквивалентдир. Худди шунингдек Activate ва Select усулларида диаграмма жойлашган варақни фаоллаштиради, аммо бирор бир усул диаграмманинг ўзини фаоллаштирамайди.

Мисол учун

```

Charts (“Мижоз яшаш манзили”). Select

```

Диаграмма ёки киритилган диаграмма компонентини ажратиб олиш учун, сизни қизиқтирувчи объект учун Select усулини қўллаш мумкин.

Катакни фаоллаштириш учун ушбу катакка Range объектининг Activate (фаоллаштирмоқ) усулини қўллаш кифоя.

Агар фаоллаштирилган катак жорий ажратилган соҳа ичида бўлса, бутун бир диапазон ажратилган бўлиб қолади. Қуйидаги мисол худди шундай тамойилда ишлатилади.

```
Worksheets (" Мен кўрсатма бермагунча ҳеч нарса қилманг").Activate  
Range ("A1:E7").Select  
Range ("C4").Activate
```

Агар фаоллаштирилган катак ажратилган соҳадан ташқари бўлса, у яъни соҳа фаоллаштирилган катакка силжитиб олиб келинади.

**Ажратишнинг типини аниқлаш.** Ажратилган катаклар билан ишловчи дастурий код агар уни диаграммага қўлласса хатоликка олиб келиши мумкин. Шунинг учун ҳам ажратилган катаклар билан бирор бир амал бажаришдан олдин, ажратилган катаклар код қўшаётган объект типини ўз ичига олганлигига ишонч ҳосил қилиш лозим. VBA –функция бўлган TypeName ажратилган объект типини аниқлайди

If ... Then ёки Case ... Select тузилмалари ёрдамида ажратилган объектлар бўйича бажариш мумкин бўлган амаллар аниқланади.

```
With Selection
```

```
Select Case TypeName (Selection)
```

```
Case "Range"
```

```
.Value=2001
```

```
Case "ChartArea"
```

```
.Interior. ColorIndex=3'3=bright red
```

```
Case "Nothing"
```

```
MsgBox "Ҳеч нарса ажратилмаган"
```

```
Case Else
```

```
MsgBox "Ажратилган катаклар типини аниқлаш қийин  
бўлмоқда!"
```

```
End Select
```



End with,

**Фойдаланувчи функцияларни дастурлаш.** Маълумотларни таҳлил қилувчи Excelга ўрнатилган ускуналар барча зарурий амалларни бажаришга кодирдек кўринсада, VBA билан ишлаш бир мунча қулайдир.

VBAни ишлатиш ишчи варақлар учун фойдаланувчи функцияларни тузиш имконини бериб, улар катакларга ёзиладиган мавжуд формуладан анча афзалдир. Чунки фойдаланувчи функциялари Excelга ўрнатилган функциялар ҳисоблашнинг имкони бўлмайдиган ҳисоблашларни бажариши мумкин[11,12].

Excel фойдаланувчи функцияси - VBA функциянинг одатдаги процедураси. Функция процедураси унинг номини эълон қилиш билан билан бошланиб End Function оператори билан тугайди. Айрим ҳолларда функция номига қиймат берувчи оператор ҳам учраши мумкин бўлиб, ушбу қиймат функцияси ўзгартирилади. Қуйидаги содда мисол юқоридаги ишни тасвирлайди:

```
Function MamoryAvailable= Application. MemoryFree
```

```
End Function
```

Ушбу функция ҳозирги вақтга Excelга тегишли бўлган хотирани байтларда аниқлайди. Мазкур функция компьютер тизимида бўш бўлган хотира ҳажмини олар экан аргументга эга эмас. Қуйида эса бир мунча мураккаб функция билан танишамиз.

```
Function CheckForValue (aRange, Value)
```

```
For Each objCell In aRange
```

```
Check ForValue=False ‘ жимжитликда False қиймат олади.
```

```
    If objCell Value=Value Then
```

```
        Check ForValue=True
```

```
    Exit For
```

```
End If
```

```
Next objCell
```

```
End Function
```

Ушбу функция катаклар соҳасида берилган қийматнинг мавжуд ёки мавжуд эмаслигини текширади. Агар ўша қиймат соҳанинг бирор бир жойида бор бўлса, функция True (чин) қийматни, акс ҳолда False қиймат қабул қилади.

Функцияни аниқлашдан олдин уни икки аргумент –диапазон ва изланаётган қиймат билан таъминлаш зарур.

Фойдаланувчи функцияларини ишга тушириш усулларида бири –бу, албатта, VBAда стандарт процедурани ишга тушуриш кабидир. Яъни Sub процедурасида чиқаришдир. Ишчи вараққа функция қабул қиладиган қийматни жойлаштириш учун ўрнатилган, функцияни қандай ишлатса шундай ишлатиш зарур: тенглик белгисидан кейин катакка функция номини киритиш лозим. Функция номидан кейин кавс ичида ихтиёрий аргумент қўйилади.

Мисол учун:

=Memory Available()

Худди ўрнатилган функциялар каби фойдаланувчи функциялар ҳам катакдаги бирмунча мураккаб формуланинг қисми бўлиши мумкин.

Масалан:

=Memory Available() “ҳозир байтга мурожаат қилиш мумкин”

=If(CheckForValue(B8:B18,C8)) “Қиймат топилди”, “Қиймат топилмади”)

Фойдаланувчи функциялари формулалар панелини ишлатишда функцияни ёзишда “Формулалар панели” функция учун керакли барча нарсани кўрсатади. Функцияни танлашда “Вставка функции” мулоқот даражасида пайдо бўлувчи функцияни ёзишга қўшиб қўйиш етарли. Функцияни ёзиш учун Сервис> Макрос>Макросы буйруқларини танлагандан сўнг “Имя макроса” майдончасига функция номини киритиш лозим. Шундан сўнг “Параметры” тугмачасига мос майдончага функцияни ёзиш учун чертиш кифоя. Бошқа ишчи китобдаги фойдаланувчи функцияларига мурожаат қилиш учун, яъни бошқа бир олинган китобда сақланаётган

функцияларни ишлатиш учун функциядан олдин ишчи китоб номи ундов белгиси билан ажратилган бўлиши лозим.

Масалан:

=Содда функциялар xls! Ақлли функция (C4:D6,M9).

Жорий вақтда очик бўлмаган ёки кейинчалик мурожаат қилиш вақтида ёпиқ бўлган китобларда сақланаётган функцияларга мурожаат қилиш учун улар VBA ихтибосини ростлаш зарур. Бунинг учун Tools⇒Reference буйруғи танланади. Агар ишчи китоб мулоқот дарчасидаги рўйхатда йўқ бўлса уни маҳаллийлаштириш ва рўйхатга қўшиб қўйиш учун Browse тугмачасини босиш кифоя.

Агар китоб олдига байроқча ўрнатилмаган бўлса, мурожаатни фаоллаштириш учун байроқни ўрнатиш лозим. Энди ушбу китобдаги ихтиёрий функцияни, китоб номини кўрсатмай ишлатиш мумкин.

**Функцияни профессионал тарзда ёзиш.** Агар натижа биз хохлагандек бўладиган бўлса, функцияни ёзишдан воз кечмаслик керак. Функция бажарадиган ишни формулага топширишнинг ҳожати йўқ. Check For Value функциясининг келтирилган модификацияси тушунарсиз True ва False қийматли ўрнига изоҳ берувчи матн сатрини ҳам қабул қилиш мумкин.

Function CheckForValue 2 (aRange, Value)

For Each objCell In aRange

CheckForValue2="Изланётган қиймат" 6 Value "топилмади"

If objCell. Value=Value Then

Check ForValue2= "Изланётган қиймат" Value

objCell. Address "катакда жойлашган"

Exit For

End If

Next objCell

End Function

**Excelда ҳолатларни дастурлаш.** Фойдаланувчи томонидан киритилган ўзгаришларга Excel электрон жадвалининг реакцияси унинг иш

фаолиятининг ажралмас қисмидир. Шунинг учун Excelдаги жараён учун дастурий код ёзиш Office пакетлари иловаларига қараганда бирмунча муҳим аҳамият касб этади. Формуладаги ҳисоблашлар ёки диаграммаларда ишлатилувчи битта катак қийматини ўзгартириш бутун бир ишга китоб учун сезиларли оқибатларга олиб келиши мумкин. Ушбу жараёни бошқарувчи юқоридаги ҳолатни назоратга олиш имкони, Excelда ўрнатилган жавоблар тизими иш фаолиятини кенгайтириш ва яхшилаш VBA-дастури қўлидадир.

Дастурий код ёзишдан олдин қайси объект ҳолатга жавоб беради деган муаммони ҳал этиш зарур. Excelда ҳолатлар тўрт объектда: диаграмма, айрим олинган ишчи варақлар, ишчи китоблар ва Excel иловалар бир бутун ҳолатлар учун дастурий код ёзиш талаб қилинса, бундай код диаграмма ҳолати процедурасига мос келади. Аммо ишчи варақдаги ўзгаришларга жавоб сифатидаги ҳолатлар учун, дастурчи бир неча вариантни танлаш имконига эга бўлади.

Excelнинг айрим ҳолатлари фақатгина маълум бир объектларни таниса, улардан кўпчилиги Worksheet объектидан бошлаб Workbook ва Application объектларига бўлган иерархияни ташкил этади. Мисол учун, ишчи вараққа киритилган ўзгариш Workbook ва Application объектларининг Sheetchange (Варақ ўзгариши) ҳолатини тасвирлайди. Юқорида айтилганларни эътиборга олиб, ишчи варақ, ишчи китоб ёки илова ҳолатлари процедураси учун дастурий код мос келиши ёки келмаслигини аниқлаш лозим. Танлаш муаммоси кўп меҳнат талаб этади.

-агар код ягона ишчи варақдаги ўзгаришгагина жавоб берса, у ушбу ишчи варақ ҳолатлари процедурасига мос келади;

-агар процедура маълум бир ишчи китобнинг ихтиёрий ишчи варағидаги ўзгаришлар киритишда бажарилса, кодни ишчи китоб ҳолатлари процедурасига қўйиш лозим;

-агар код ихтиёрий очиқ ишчи китобни фаоллаштириши зарур бўлса, у сўссиз илова ҳолатлари процедурасига мос келади.

Excel да ихтиёрий объекти учун ҳолат процедурасини ёзиш технологияси VBA бошқариш элементи ёки шакли учун ҳолат кодлари ёзиш технологияси билан таққосланганда ҳеч қандай принципиал фарқга эга эмас.

Асосий жиҳатларини қуйидагича тасвирлаш мумкин.

- 1.Объект учун кодни таҳрирлаш даражасини очинг.
- 2.Дарчанинг юқори чап бурчагида берилган объектлар рўйхатидан объектни номи бўйича танланг.
- 3.Дарчанинг юқори чап бурчагида очилган процедуралар рўйхатидан код ёзилиши лозим бўлган ҳолатни танланг.
- 4.Бу вақтда дарчада танланган маълумот процедураси каркаси пайдо бўлади.

Коднинг ўзини ёзишдан ташқари, кодни таҳрирлаш дарчасини биринчи ўринга чиқариш ягона мураккаб момент бўлиши мумкин. Worksheet, Workbook ва Chart ларнинг айрим варақни банд этувчи объектлари учун ҳеч қандай ўзига хослик йўқ – шарҳловчи дарчасида объектни танлаш керак ва View Code тугмасини босиш лозим. Объектлар алфавит таркибида тасвирланади, уларнинг номини Options дарчасида ўзгартирилади. Ишчи вараққа қўлланган диаграмма ҳамда Excel иловалари объектлари учун ҳолат бирмунча мураккаб. Дарчадан мурожаат қилиш мумкин бўлган бундай объектларни яратиш учун синфларнинг махсус модуллари ёзиш керак. Агар ишчи варақда фойдаланувчи томонидан бажарилган ҳаракатларга дастурий код ўз муносабатини билдириши учун Change, Calculate ва SelectionChange (Worksheet объектлари учун) хоссалар ва уларга мос Sheet Change, Sheet SelectionChange (Workbook ва Application объектлари учун) ҳолатлар каби ускуналар зарур бўлади.

Фойдаланувчи процедураларини ишчи варақлар ёки диаграммалар фаоллаштирилган ёки фаоллаштирилмаган вақтда ишга тушиши учун Activate ва Deactivate ҳолатлари ишлатилади.

Change ва SheetChange ҳолатлари ҳар доим, фойдаланувчи ҳаракати ёки ихтибосни янгилаш натижасида ихтиёрий катак ёки бир қанча катаклардаги

қийматлар ўзгарганлигини тасвирлайди. Аммо ҳисобланадиган қийматлардаги ўзгаришлар ҳолатларини тасвирламайди. Қуйидаги мисолда Target деб номланган соҳа ичидаги ўзгартирилган қийматлар берилган оралиққа тушиш ёки тушмаслиги текширилади ва шундай қийматлар топилса уларнинг шрифти 16 кегел билан, ярим йўғон ва яшил ранг билан ажратилади.

```
Private Sub Worksheet Change (ByVal Target As Range)
```

```
    For Each oCell In Target
```

```
        If oCell >4 And oCell <11 Then
```

```
            With oCell.Font
```

```
                Bold=True
```

```
                Size=16
```

```
                Color=PGE (0, 255, 0) яшил
```

```
            End With
```

```
        End If
```

```
    Next oCell
```

```
End Sub
```

Бу ерда Target битта катак эмас, балки соҳа бўлиши мумкин. Шу сабабли рангга бўйаш, четлаштириш ва ўрнатиш амаллари бир вақтнинг ўзида бир неча катаклар устида бажарилиши мумкин. Бунинг учун Change маълумотлар процедураси ўз таркибига For Each... Next тузилмасини олиш зарур. Шундай вариантда процедура айрим олинган катакка, бир неча катаклардан ташкил топган соҳага қўлланиши мумкин.

Calculate ҳолати аниқланувчи Worksheet ва Chart объекти сифатида ишчи варақ ёки диаграмма Excel дастури ёрдамида доимий равишда янгиланишда юз беради.

Sheet Calculate ҳолатида Workbook ва Application объектлари учун жавоб кўринишида амалга ошади. Агар автоматик ҳисоблаш воситаси уланган бўлса, ушбу ҳолат ихтиёрий катак қиймати ўзгариши билан фаоллашади. Агар қайта ҳисоблаш воситаси қўлда бажарилишида уланган

бўлса, Calculate ҳолати фойдаланувчи <F9> клавишини босгандан сўнг амалга ошади.

Ҳолатлар процедураси Calculate ва SheetCalculate ҳолатлари учун ишчи варақ ўзгаришларида ҳисоблаш натижасига мос равишда ишлатилади. Мисол учун, агар қайта ҳисоблаш тартибланган рўйхат элементларини ўзгартириши аниқ бўлса, ҳисоб-китоб қилингандан сўнг рўйхатни тартибга келтириш учун WorkSheep Calculate ҳолатлар процедурасини ишлатиш мақсадга мувофиқ. Ушбу процедуралар ҳисоблаш натижасида қайси катак ўзгарганлиги ҳақида маълумот бермаганлиги учун, дастурий кодда ўзгарадиган катак манзили кўрсатилиши лозим.

Фаол катакларни силжитиш, ажратилган соҳани кенгайтириш ёки сиқишда Excel Worksheet учун Selektion Change ҳолати ишлатилади. Workbook ва Application объектлари учун Sheet Selektion Change ҳолати ҳам параллел амалга ошади. Ушбу ҳолатлар учун ҳолатлар процедурасини жорий ажратилган соҳа билан тескари алоқа учун ҳам ишлатиш мумкин. Қуйида берилган мисолда SelectionChange ҳолати жорий варақнинг юқори чап катагида фаол катак манзилини тасвирлаш ҳамда ишчи варақ номи ва ажратилган соҳа манзилини ҳолатлар сатрига жойлаштириш учун ишлатилган.

```
Private Sub Workbook SheetSelectionChange (ByVal Sh As Object, ByVal Target As Excel. Range
```

```
Sh. Range ("A1") =ActiveCell. Address
```

```
Application. StatusBar=Sh.Name & ":"& Target Address
```

```
End Sub
```

Дастурий кодни бирмунча мураккаблаштириш, оддий ажратиш ўрнига қандай жавобни чақириш мумкин: Мисол учун маълум бир катак ёки диапазон янги ажратилган соҳага тушса, фойдаланувчининг мулоқот дарчаси тасвирлансин.

```
Private Sub WorkSheet SelectionChange (By Val Target As Excel. Range)
```

```
If Target. Address= "$3$2" Then
```

MsgBox “Сиз зарурий катакни топдингиз”

End If

End Sub

**Динамик диаграммаларни дастурлаш.** Excel диаграммалари объектлари кўплари сичқонча билан боғлиқ бўлган кўп сонли ҳолатларни таниш имконига эга. Уларни ишчи китобда жойлашган Active X бошқарувчининг кўпгина элементлари деб ҳисоблаш мумкин. Диаграммалар Activate Deactivate ва Calculate ҳолатларини танийди[14,15,16].

Қуйидаги айрим ҳолатлар келтирилганки, улар учун код ёзиш лозим:

- DragOver ва DragPlot. Катаклар киритилган диаграммага олиб борилганда ёки уларга мос қўйиб юборилганда амалга оширилади.
- Mouse Down, Mouse Up ва Mouse Move. Сичқонча ҳаракатига жавоб берилади.
- Select диаграмманинг қандайдир қисми ажратилганда юз беради. Ёзилган код ажратилган элемент типи ва характеристикасини аниқлаш мумкин ва мос ҳаракатларни бажаради.
- Series Change фойдаланувчи диаграммадаги қийматни шу диаграммани ўзи орқали ўзгартирганда амалга ошади.

## 2.2. VBA муҳитида файлларга ахборотни киритиш ва чиқариш.

VBA файлларга ахборотларни киритиш ва чиқариш бўйича қатор имкониятларга эга. VBA да файлларни очиши учун махсус кўрсатма **Open** дан фойдаланиб, унинг формати қуйидагича:

**Open** *Файл номи* **For** *Мурожаат тури* [**Access** *Мурожаат тартиби*]  
[Блокировка] **As** [#] *Дескриптор* [**Len** = *Ёзув узунлиги*]

Қуйидаги жадвалда Open кўрсатмасининг элементлари кўрсатилган.

№	Элемент	Вазифалари
1	<i>Файл номи</i>	Очиладиган файлгача бўлган тўлиқ ёки нисбий йўлни кўрсатувчи сатр
2	<i>Мурожаат тури</i>	Очиладиган файлнинг VBA ишлайдиган иш тартиби берилади. <b>Append</b> - файл охирига маълумот қўшиш, <b>Binary</b> –файлга бинар тартибда <b>Input</b> -матнли тартибда



		файлдан ёзувларни кетма-кет ўқиш. <b>Output</b> -файлга матнли тартибда ёзувларни кетма-кет ёзиш, <b>Random</b> -файлларга ихтиёрий тартибда мурожаат.
3	<i>Мурожаат тартиби</i>	Файллар билан ишлашда бажариладиган амаллар берилади. <b>Read</b> - фақат ўқиш; <b>Write</b> - фақат ёзиш; <b>Read Write</b> - ўқиш ва ёзиш; <b>Read</b> ни Output билан, <b>Write</b> ни Input билан, <b>Read Write</b> ни эса на Input, на Output билан биргаликда ишлатиш мумкин эмас.
4	<i>Блокировка</i>	Очилган файлга бошқа иловалар мурожаатини тақиқлайди. <b>Lock Read</b> –файлга ёзишга рухсат, аммо ўқиш тақиқланади; <b>Lock Write</b> -ёзиш тақиқланади, аммо ўқиш мумкин; <b>Shared</b> -жимжитликда ўқиш ва ёзишга рухсат.
5	<i>Дескриптор</i>	Очилган файлни айнан тенглаштирувчи бутун қийматлар (1 дан 511 гача) кўринишда берилади.
6	<i>Ёзув узунлиги</i>	<b>Random</b> мурожаат турини ишлатишда байтларда берилган ёзув ўлчамини кўрсатади. <b>Output, Input, Append</b> мурожаат турлари учун буферда сақланади байтлар миқдорини кўрсатади.

### Мисоллар:

1. Open “D: \ MyTextFile.txt” For Random Access Read Write

As 1 Len=100

2. Open “D: \ MyTextFile.txt” For Input As 2

3. Open “D: \ MyTextFile.txt” For Output As 3

VBA дастурлаш муҳотида файллар очилиши рақамлар-дескрипторлар ёрдамида аниқланади. Ҳар бир очик файл учун дескриптор ягонадир. Кўпгина мисолларда кўриш мумкинки, очиладиган файлларга дескрипторни дастурчининг ўзи белгилаши ҳам мумкин. Унча катта бўлмаган ва содда дастурлар учун дескрипторни қўлда белгилаш қулай. Аммо катта лойиҳаларда, яъни бирқанча файллар билан ишлашда дескрипторларни қўлда белгилаш мураккаб бўлиб, ушбу жараёни назорат қилиш қийинлашади. Дастурчи дескрипторларни белгилашни тўғри олиб бораётганини назоратга олиш мақсадида VBA махсус функция **FreeFile** мавжуд бўлиб унинг фаолиятини қуйидагича

FreeFile ([Оралик])

Ушбу функция **Long** турдаги қийматни қабул қилиб, уни **Open** кўрсатмасида очиладиган файл дескриптори сифатида ишлатиш мукин. Ушбу функция унчалик мажбурий бўлмаган [*Оралик*] параметрга эга бўлиб, у 0 ёки 1 қиймат қабул қилади. Агар параметр 0 (жимжитликда) қиймат қабул қилса, функция 1-255 ораликдаги дескриптор қийматини, агар 1 га тенг бўлса 256-511 ораликдаги дескриптор қийматини олади. Агар ораликда бўш дескриптор бўлмаса, функция 0 қийматни қабул қилади.

Мисол:

Dim hFile As Long

hFile=FreeFile ‘ файл учун дескриптор олиш

‘Файл очилиши

Open “D:\MyTextFile. Txt” For Output As hFile

Файл очилиб, у билан барча амаллар бажарилгандан сўнг уни ёпиш зарур. Умуман айтганда операцион тизимнинг барча объектлари учун ёпиш амали мажбурийдир. Файл ёпилишида унинг дескриптори озод бўлади ва уни бошқа файл (агар очилиш вақтида блокировка қилинган бўлса) билан ишлатиш имконияти юзага келади.

VBAда файлни ёпиш учун **Reset** ва **Close** кўрсатмаларидан фойдаланилади. Ушбу кўрсатмаларнинг форматини қуйидагича:

Reset

Close [[#] *Дескриптор* [,[#] *Дескриптор*]...]

**Reset** кўрсатмаси олдин **Open** кўрсатмаси билан очилган барча файлларни ёпади. **Close** кўрсатмаси эса фақат кўрсатилган дескрипторли файлларнигина ёпади.

Close 1, # hFile

Агар **Close** кўрсатмасини ишлатишда ёпилувчи файллар дескрипторлари кўрсатилмаса, у барча очилган файлларни ёпади. VBA дастурларига файллар билан ишлашда маълумотларни ёзиш ва ўқиш учун кўпгина кўрсатмалар тақдим этилган. Ушбу кўрсатмаларни файлларга мурожаати бўйича уч гуруҳга ажратиш мумкин.

- кетма-кет мурожаат кўрсатмалари;
- ихтиёрий;
- бинар

### Кетма-кет мурожаат кўрсатмалари.

№	Кўрсатма	Мурожаат тури	Вазифаси
1	Write #Дескриптор [қиймат]	Output, Append	Файлга берилган параметр билан ёзади. Дескриптор - параметр [қиймат] билан берилган ўзгарувчи ёки ифода қиймати. Агар [қиймат] бир неча ўзгарувчи ёки ифодани ўз ичига олса, файлга ёзиш вақтида уларнинг қийматлари вергул билан ажратилади. Унда сатрли қийматлар қўштирноққа олинади. Файлга барча қийматлар ёзилгандан сўнг, унга кейинги сатрга ўтиш белгиси қўшиб ёзиб қўйилади.
2	Print #Дескриптор [қиймат]	Output, Append	Файлга берилган параметр билан ёзади. Дескриптор – параметр [қиймат] билан берилган ўзгарувчи ёки ифода қиймати файлнинг бир сатрига ёзилади ва табуляция белгилари билан ажратилади. Файлга барча қийматлар ёзилгандан сўнг, унга кейинги сатрга ўтиш белгиси қўшиб ёзиб қўйилади.
3	Input #Дескриптор ўзгарувчилар	Input	Файлга Write кўрсатмаси билан ёзилган ёзувни ўқиш учун ишлатилади. Файлдан ўқиладиган қийматлар сони билан аниқланади.
4	Line Input #Дескриптор ўзгарувчилар	Input	Файлдан сатрли ёзувни ўқийди. Кўпинча маълумотлар файлидан <b>PRINT</b> кўрсатмаси билан ёзилган маълумотларни ўқишда ишлатилади.

#### 1-мисол:

Sub Write To File ()

Open “D:\ MyText File.txt” For Output As 1

```

        'Файлга маълумотларни ёзиш.
Write #1, "қиймат" "Value" 154,32
Print #1, "Сўз 1" , "Сўз 2", 14.28464
Close 1
End Sub

```

## 2-мисол

```

Sub Read From File ()
    Dim str Val1, strVal 2, dblNumber
    Dim strString
    Open "D:\ MyTextFile. txt" For Input As 1
        'Файлдан маълумотларни ўқиш
    Input #1, str Val, strVal 2, dblNumber
    Line Input #1, strString
    Close 1
End Sub

```

VBАда **Input** ўрнатилган функция бўлиб, у ёрдамида файлдан берилган сондаги белгиларни ўқиш мумкин.

**Input** (*Белгилар сони, [#] Дескриптор*)

## 3-мисол.

```

Sub Test Input ()
    Dim strText As String
    Open "D:\ MyTextFile. txt" For Input As 1
        'Файлдан биринчи 10 та белгини ўқиш
    strText= input (10,1)
    Close 1
End Sub

```

Ихтиёрий тартибдаги мурожаат кўрсатмалари. Ихтиёрий тартибдаги мурожаатда (**Random**) файл ўзгармас узунликдаги ёзувлар мажмуаси сифатида таасаввур этилади.

Фақат ушбу мурожатда ёзувга файлдан ўқилувчи ёки ёзилувчи элементар ахборот бирлиги сифатида қаралади. Ҳар бир ёзув ўз рақамига эга (рақамлари 1дан бошланган) . VBАда ихтиёрий тартибда мурожаат қилиб файллар билан ишлаш учун ахборотларни ёзиш ва ўқиш учун Put ва Get кўрсатмалаидан фойдаланилади.

**Put** [*#*] *Дескриптор, [Ёзув рақами], ўзгарувчилар*

**Get** [#] *Дескриптор*, [*Ёзув рақами*], *ўзгарувчилар*

**Put** кўрсатмаси бажарилганда ўзгарувчи қиймати файлга ёзув рақами билан жойлашади. Агар ёзув рақами кўрсатилмаган бўлса, маълумот файлнинг жорий ёзувига жойлашади.

**Get** кўрсатмаси файлдан берилган рақамли ёзув қийматини ўқийди. Агар ёзув рақами кўрсатилмаган бўлса, файлнинг жорий ёзуви ўқилади.

**Мисол.**

```
Type Record
intVal As Integer
StrName As String *100
End Type
Sub Test Random Access ()
    Dim rec 1 As Record, rec 2 As Record
    'rec 1 ва rec 2 ларни қиймат билан тўлдириш.
    Open "D:\ MyRandom AccessFile. Txt" For Random
        Access Read Write As 1 Len=Len [rec 1]
        'Файлга маълумот ёзиш

    Put 1, rec 1
    Put 1, rec 2

    'Файлдан маълумотларни ўқиш

    Get 1, 2, rec 2
    Get 1, 1, rec 1
    Close 1
End Sub
```

Файлга бинар (Binary) мурожаат ўз моҳияти бўйича ихтиёрий тартибдаги мурожаат билан бир хил бўлиб, фақат битта тафовути бор. Бунда файл ёзув узунлиги 1 байтни ташкил этади. Файлга бинар мурожаатда ҳам **Put** ва **Get** кўрсатмалари ишлатилади. Бундан ташқари файлга бинар мурожаатда ҳам **Input** функцияси ишлатилади.

Амалиётда ўлчами номаълум бўлган файлдан маълумотларни ўқиш зарурияти юзага келади. Агар файл охирига етиб, яна ундаги маълумотларни ўқишга ҳаракат қилинса, кўпинча хатолик бўлаётгани кўрсатади. Бу ҳолатларнинг олдини олиш мақсадида **EOF** ва **LOF** функциялардан фойдаланилади.

**EOF** (*Дескриптор*)

**LOF** (*Дескриптор*)

**EOF** функция берилган дескрипторли файл охирига етганда **True** қиймат қайтаради, акс ҳолда- **False** қийматни. Агар **EOF** функция **False** қиймат қайтарса, файлдан ёзув ўқиш тугалланиши лозим. **Output** таркибида очилган файллар учун **EOF** функцияси ҳар доим **True** қиймат қабул қилади.

**LOF** функцияси берилган дескриптор параметри файлнинг узунлигини аниқлайди. Бу функция очиқ файл узунлигини байтларда ифодоловчи **Long** қийматини қабул қилади.

**LOC** (*Дескриптор*)

**Seek** (*Дескриптор*)

Икала функция ҳам берилган дескриптор параметрли файлнинг жорий позициясини аниқлайди. Аммо ушбу функцияларнинг ўзига хос жиҳатлари мавжуд.

**Loc** функция **Random** тартибида очилган файл учун охирги ўқилган ёки ёзилган ёзув рақамини аниқлайди. **Binary** тартибида очилган файллар учун охирги ўқилган ёки ёзилган байт рақамини аниқлайди. Кетма-кет мурожаат тартибида очилган файллар учун 128 га бўлинган байтларда берилган жорий позициясини аниқлайди.

**Seek** функция **Random** тартибида очилган файллар учун охирги ўқиш/ёзиш амалида файлдан ўқилган ёки файлга ёзилган ёзув рақамини аниқлайди. Бошқа файллар учун бу функция ўқиш/ёзиш амали бошланадиган кейинги ёзув рақами байтини аниқлайди.

Ахборотларни фойдаланувчига чиқариш учун VBAда қулай **MsgBox** функцияси мавжуд. У маълумот билан стандарт дарчани тасвирлаш имкониятига эга. У қуйидаги форматга эга:

**MsgBox** (*Маълумот матни* [,*услуг*] [,*сарлавҳа*] [*маълумот файли, мавзу индекси*])

Бу ерда, *Маълумот матни* – маълумот матни сатри, *сарлавҳа* – дарча сарлавҳаси сатрида тасвирланувчи матнли сатр, *маълумот файли* – маълумотнома файли номи, *мавзу индекси* – берилган маълумотнома файлидаги мавзуга бағишланган мулоқот дарчасига чиқариладиган матнли

сатр, *услуг* - маълумот берувчи дарчада чиқариладиган тугма ва бошқа фойдали маълумотларни тасвирловчи нишонни беради.

**Дарча услубини шакллантириш учун ишлатиладиган қийматлар.**

<b>VBA ўзгармаси</b>	<b>Қиймат</b>	<b>Тушунтириш</b>
vbOkOnly	0	Фақат Ок тугмасини тасвирлаш
vbOKCancel	1	Ок ва Cancel тугмаларини тасвирлаш
vbAbortRetryIgnore	2	Abort (Стоп), Retry (Қайтариш) ва Ignore (Ўтказиб юбориш) тугмаларини тасвирлаш
vbYesNoCancel	3	Yes (Ҳа), No (Йўқ) ва Cancel (Бекор қилиш) тугмаларини тасвирлаш
vbYesNo	4	Yes (Ҳа) ва No (Йўқ) тугмаларини тасвирлаш
vbRetryCancel	5	Retry (Қайтариш) ва Cancel (Бекор қилиш) тугмаларини тасвирлаш
vbCritical	16	Критик хатолик нишонларини тасвирлаш
vbQuestion	32	Сўроқ белгиси нишонини тасвирлаш
vbExclamation	48	Ундов белгиси нишонини тасвирлаш
vbInformation	64	Ахборот нишонини тасвирлаш
vbDefaultButton 1	0	Биринчи тугма ажратилади (чапдан ўнгга)
vbDefaultButton 2	256	Иккинчи тугма ажратилади
vbDefaultButton 3	512	Учинчи тугма ажратилади
vbDefaultButton 4	768	Тўртинчи тугма ажратилади
vbApplicationModal	0	Фойдаланувчи дарчаси ёпилмагунча иловалар билан ишлаб бўлмайди
vbSystemModal	4096	Фойдаланувчи дарчаси ёпилмагунча барча иловалар билан ишлаб бўлмайди
vbMsgBoxHelpButton	16384	Help (Маълумотнома) тугмасини тасвирлаш
vbMsgBoxSetForeground	65536	Маълумотлар дарчаси қолган бошқа дарчалар устидан тасвирламаслик
vbMsgBoxRight	524288	Матни ўнг чекка бўйича текслаш
vbMsgBoxRtl Reading	1048576	Матни ўнгдан чапга чиқариш тартибини улаш

**MsgBox функцияси қабул қиладиган қийматлар**

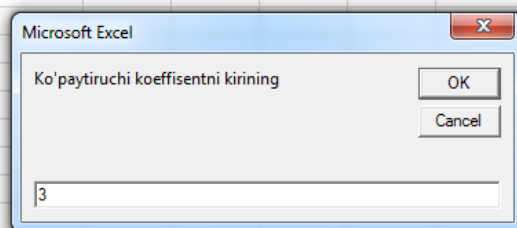
<b>VBA ўзгармаси</b>	<b>Қиймати</b>	<b>Тугмани босиш</b>
vbOk	1	Ок
vbCancel	2	Cancel (Бекор қилиш)
vbAbort	3	Abort (Стоп)
vbRetry	4	Retry (Қайтариш)
vbIgnore	5	Ignore (Ўтказиб юбориш)
vbYes	6	Yes (Ҳа)
vbNo	7	No (Йўқ)

## 2.3. Макрослар ёрдамида ўқув эффектлари ва трюкларини яратиш амалиёти.

Бирор бир соҳадаги барча сонларни бирор бир бошқа сонга кўпайтириш зарур бўлиб қолса қуйидаги макросдан фойдаланиш мумкин.

```
Sub макрос1()  
    Dim x1 As Double  
    Dim cell As Range  
    x1 = InputBox("Ko'paytiruchi koeffisientni kirining")  
    For Each cell In Selection  
        If IsNumeric(cell.Value) And cell.Value <> "" Then  
            cell.Value = cell.Value * x1  
        Else  
            MsgBox cell.Address & " katakda conli qiymat emas"  
        End If  
    Next  
End Sub
```

B	C	D	E	F	G	H	I	J	K	L	M	N	O
192	280	176											
220	280	120											
208	280	144											
220	280	120											
228	280	104											
212	280	136											
204	280	152											
188	280	184											
200	280	160											
180	280	200											
180	280	200											
204	280	152											
192	280	176											
232	280	96											
232	280	96											
192	280	176											
204	280	152											
54	70	16											
51	70	19											





	A	B	C	D
1				
2		576	840	528
3		660	840	360
4		624	840	432
5		660	840	360
6		684	840	312
7		636	840	408
8		612	840	456
9		564	840	552
10		600	840	480
11		540	840	600
12		540	840	600
13		612	840	456
14		576	840	528
15		696	840	288
16		696	840	288
17		576	840	528
18		612	840	456
19		54	70	16
20		51	70	19

**2.2-расм.**

Маълумотларни мулоқот даражаси ёрдамида киритиш.

Иш вақтида маълумотларни киритиш учун бўлак мулоқот дарчаси чақиришга тўғри келади. Ушбу маълумотни киритиш учун қуйидаги макросдан фойдаланамиз.

```
Sub InputDialog ()
```

```
Dim strInput As String
```

```
strInput=InputBox (Ma'lumotlarni kiriting", Ma'lumot kiritish"
```

```
End Sub
```

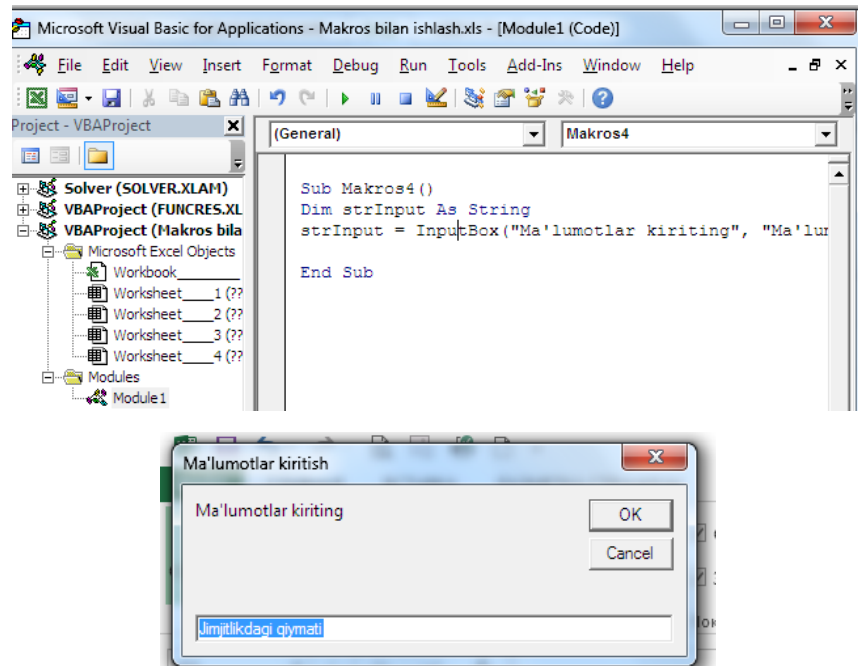
Зарурат бўлганда жимжитликда мулоқот дарчасининг маълумотлар киритиш майдонида тасвирланадиган матнни бериш ҳамда дарчанинг экранда жойлашиш позициясини (XPos ва YPos) ҳам бериш мумкин (2.3.-расм).

```
Sub New Input Dialog()
```

```
DIM strInput As string
```

```
strInput =InputBox ("Ma'lumotlar kiriting", "Ma'lumotlar kiritish",  
"Jimjitlikdagi qiymati", 200, 200)
```

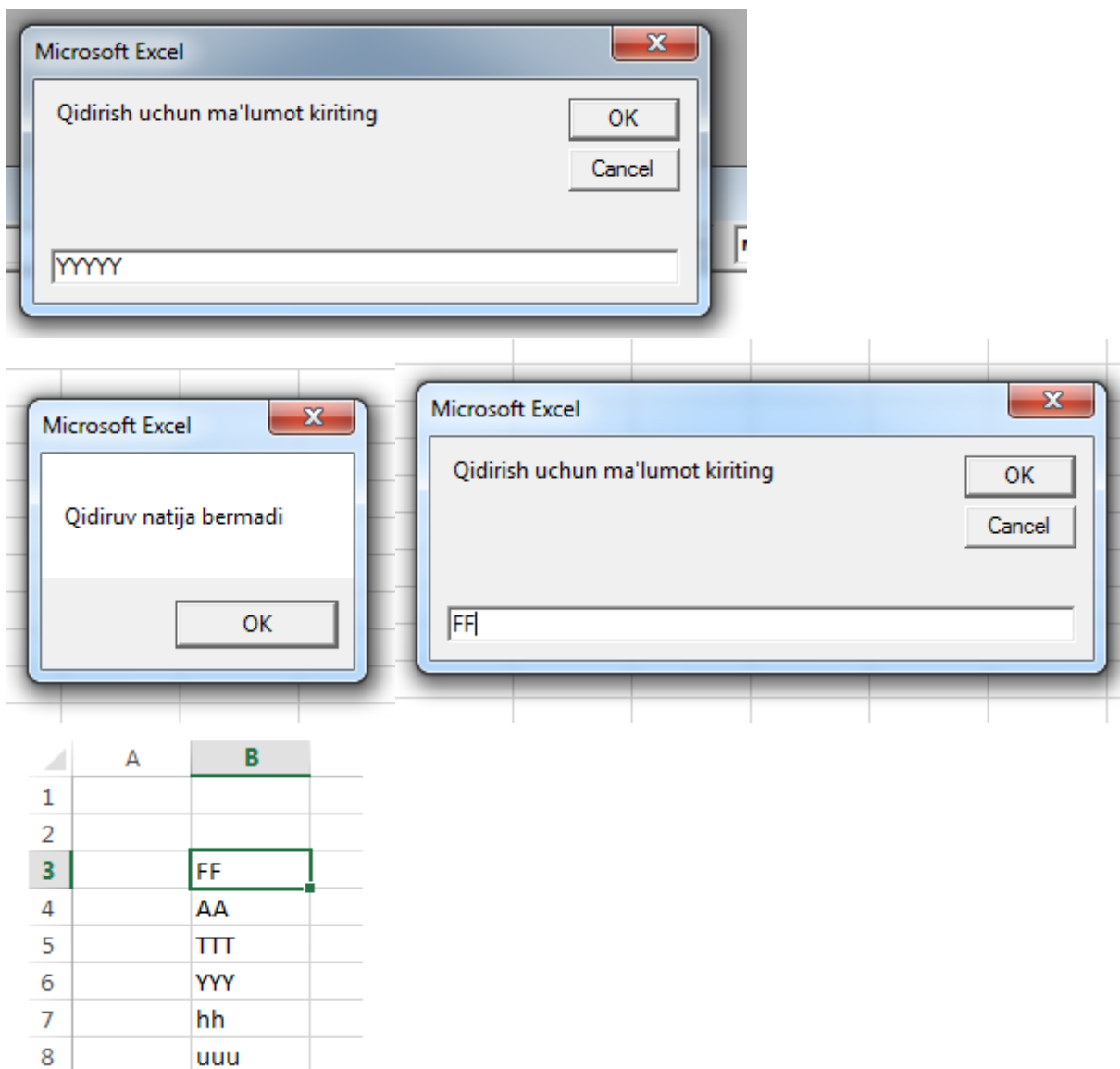
End Sub



2.3.-rasm

Берилган сўзни жадвалда мавжуд бўлган матндан кидириш учун қуйидаги макросни ишлатиш мумкин. Макроснинг ишлаш натижалари 2.4-расмда келтирилган.

```
Sub макрос3()  
Dim StrFindData As String  
Dim rgFound As Range  
Dim I As Integer  
'Izlash uchun satr kiritish  
StrFindData = InputBox(" Qidirish uchun ma'lumot kiritish")  
For I = 1 To Worksheets.Count  
    With Worksheets(I).Cells  
        'i-varaqdan izlash  
        Set rgFound = .Find(StrFindData, LookIn:=xlValues)  
        If Not rgFound Is Nothing Then  
            'Berilgan qiymat katakda topildi-u ajratilib ko'rsatilgan  
            Sheets(I).Select  
            rgFound.Select  
            Exit Sub  
        End If  
    End With  
Next  
'Qidiruv tugadi  
MsgBox ("Qidiruv natija bermadi")  
End Sub
```



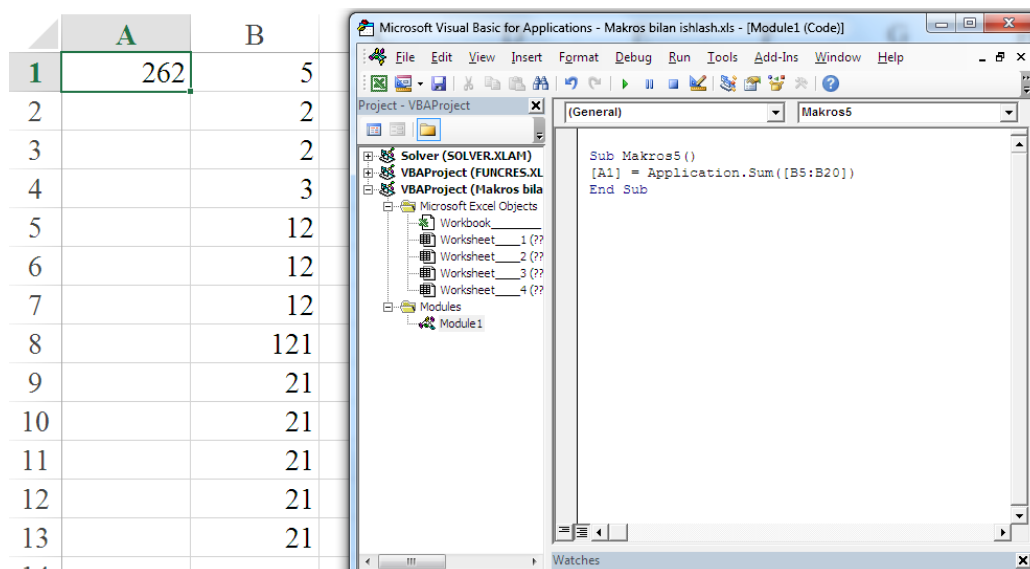
#### 2.4-расм

Зарурат юзасидан Excel стандарт функцияларини катакка киритмасдан фойдаланиш мумкин. Бунинг қуйидаги макросни киритиш кифоя(2.5.-расм).

```
Sub Func()
```

```
    [A1]=Application.Sum([B5:B20])
```

```
End Sub
```



2.5.-расм

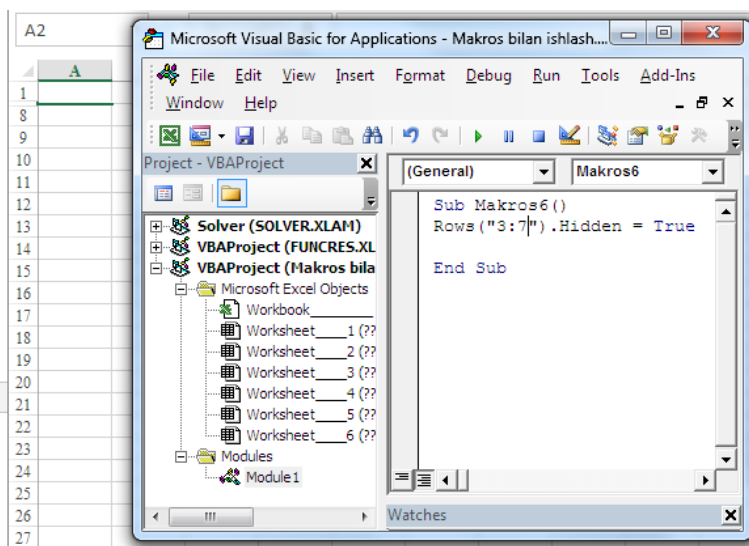
Бир қанча макрослар борки, улар ёрдамида сатр ва устунларни яшириш мумкин. Қуйида айримлари билан танишиб чиқамиз:

Сатрни яшириш учун қуйидаги макрос ишга туширилади:

```
Sub HideString()
    Rows(2).Hidden=True
End Sub
```

Бир неча сатрни яширмоқчи бўлсак қуйидаги макросни ишга тушуриш ЛОЗИМ.

```
Sub HideString()
    Rows("3:7").Hidden=True
End Sub
```



2.6.-расм

Ушбу макрос бажарилгандан сўнг 2,3,4,5,6,7-сатрлар яширинади (2.6.-расм).

Устунларни яширишда қуйидаги макрослардан фойдаланиш мумкин.

Sub HideColumns()

Columns("B:F").Hidden=True

End Sub

Юқорида яширинча устун ва сатрларни яна тасвирлаш учун макрослардан True қийматини False қийматга алмаштириш кифоя ва макросни яна бир бор ишга тушириш лозим.

Sub HideCell()

Range ("Maxfiy"). EntireRow.Hidden=True

End Sub

Юқоридаги макрос бажарилгандан сўнг "Maxfiy" номли катак жойлашган сатр яширин ҳолатга келади. Катаклар манзили бўйича бир неча сатрларни яшириш учун қуйидаги макросдан фойдаланса бўлади.

Sub HideCell()

Range ("B3:D5").EntireRow.Hidden=True

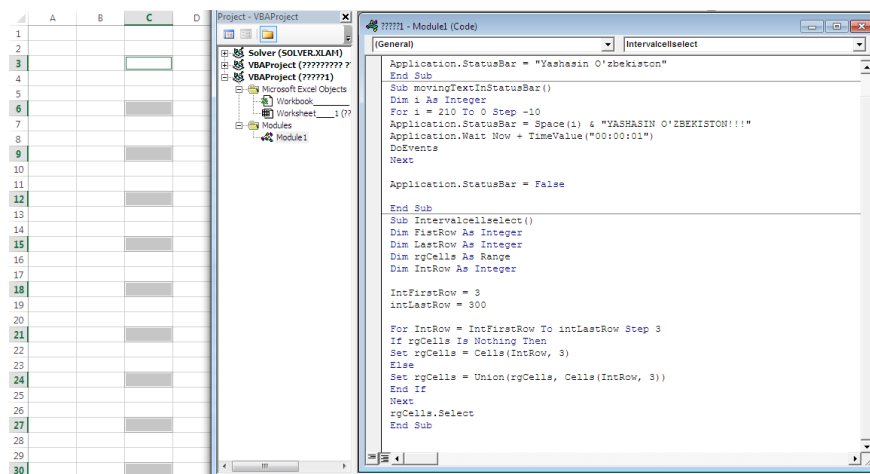
End Sub

Ушбу макрос бажарилганда 3-,4- ва 5- сатрлар яширин ҳолатга ўтади.

Устунларни яшириш ҳолатига ўтишда ҳам юқоридаги процедураларни бажариш кифоя, фақат EntireRow ўрнига EntireColumn операторини қўйиш керак.

**Макрослар ёрдамида катакларни ажратиш.**Трюклар ёрдамида Excelда катакларни тезда ажратиш имкони мавжуд бўлиб, қуйидаги айрим макрослар билан танишиб чиқамиз.

```
Sub IntervalCell Select()  
    DIM intFirstRow As Integer  
    DIM intLastRow As Integer  
    DIM rgCells As Range  
    DIM intRow As Integer  
        intFirstRow =3  
        intLastRow =300  
            For intRow=intFirstRow To intLastRow step 3  
                if rgCells Is Nothing Then  
                    Set rgCells=Cells(intRow, 3)  
                else  
                    Set rgCells=Union(rgCells, Cells(intRow,3))  
                End if  
            Next  
            rgCells.Select  
    End Sub
```



2.7.-расм

Ушбу макрос ишга туширилгандан сўнг С устуннинг учинчи катагидан бошлаб уч юзинчи катагигача ҳар уч катакнинг бири ажратиб кўрсатилади (2.7.-расм).

Худди шу натижани қуйидаги макрос ҳам бажариши мумкин.

```
Sub IntervalCell Select()
    DIM intFirstRow As Integer
    DIM intLastRow As Integer
    DIM rgCells As Range
    DIM Cell As Range
    DIM intRow As Integer

    intFirstRow =3
    intLastRow =300

    For intRow=intFirstRow To intLastRow step 3
        Set Cell=Cells(intRow,1)
        Set rgCells=Union(cell,
            If (intRow=intFirstRow,Cell, rgCells))
        End if
    Next
    rgCells.Select
End Sub
```

Турли соҳа ва катакларни ажратиш учун қуйидаги макросдан фойдалани мумкин.

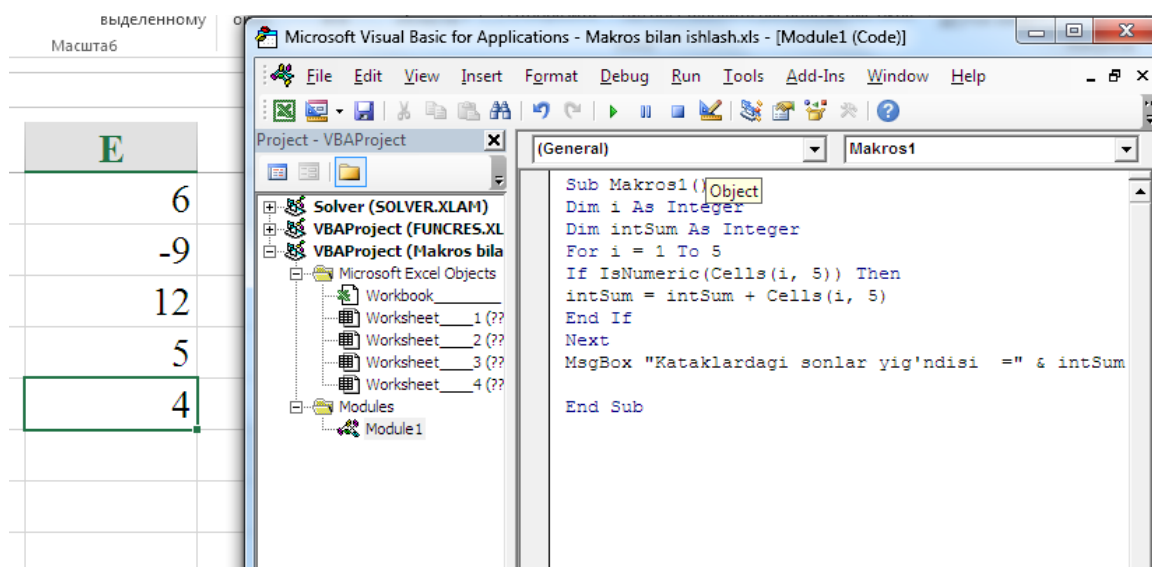
```
Sub SelectRange()  
Range("D3:D10, A3:A10, F3").Select  
End Sub
```

Ушбу макрос натижасида D устундан D3:D10 соҳа, A устундан A3:A10 соҳа ва F3 катак ажратилади.

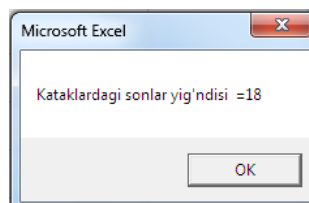
Унча мураккаб бўлмаган трюк ёрдамида кўрсатилган соҳадаги катакларда жойлашган сонлар устида амаллар бажариш мумкин. Масалан, қуйидаги макрос ёрдамида берилган соҳадаги катаклардаги сонлар йиғиндисини топиш мумкин.

```
Sub Calculate Sum ()  
    DIM i As Integer  
    DIM intSum As Integer  
    For i=1 To 5  
        intSum=intSum+Cells(i,1)  
    Next  
    MsgBox "Katakdaqi sonlar yig'indisi =" & intSum  
End Sub
```

Натижани 2.8.- расмда кўриш мумкин.







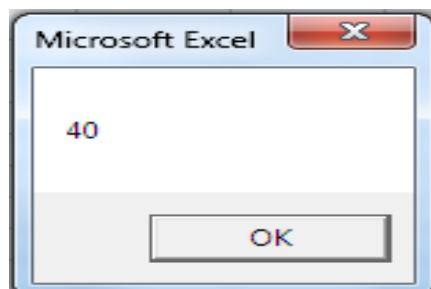
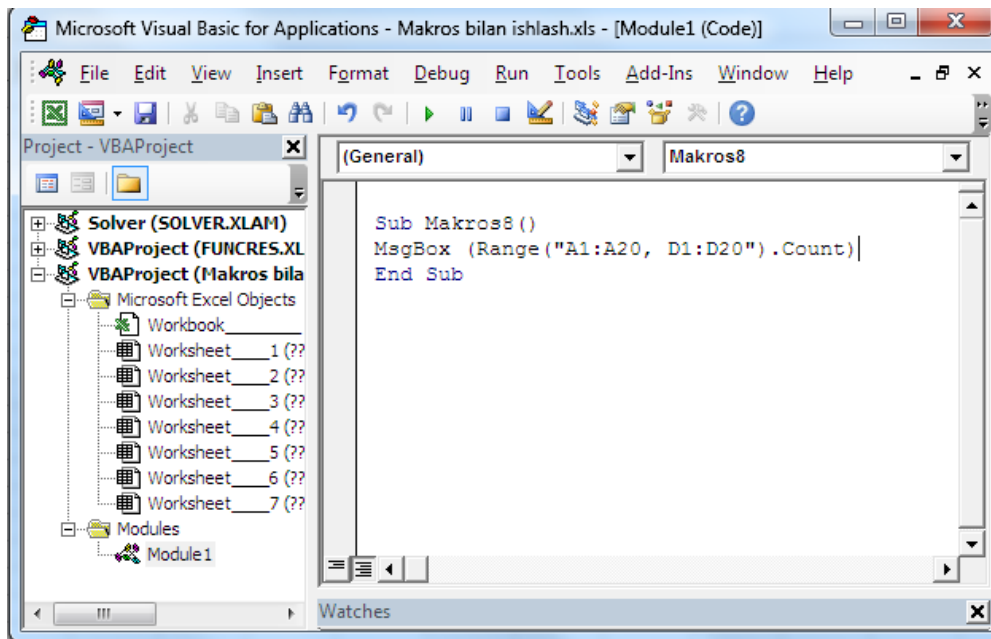
2.8.-расм

Қуйидаги макрос ёрдамида сонли қиймат мавжуд бўлган катаклар йиғиндиси ҳисобланади.

```
Sub Calculate Sum()
    DIM i As Integer
    DIM intSum As Integer
    For i=1 To 5
        If IsNumeric(Cells(i,1)) then
            intSum=intSum+Cells(i,1)
        End if
    Next
    MsgBox "Katakldagi sonlar yig'ndisi =" & intSum
End Sub
```

Қуйидаги макрос ёрдамида берилган соҳадаги катаклар сони аниқланади(2.9.-расм):

```
Sub CountOfCells()
    MsgBox(Range ("A1:A20, D1:D20").Count)
End Sub
```



2.9.-расм

Компьютер ҳақида маълумот олишда қуйидаги макросдан фойдаланилади. Макрос иш натижаси 2.10 -расмда берилган.

Sub макрос4() 'Kompyuter diskleri haqida axborot

Dim objFileSysObject As Object

Dim objDrive As Object

Dim IntRow As Integer

'Faylli tizim uchun ob'ekt yaratish

Set objFileSysObject = CreateObject("Scripting.FileSystemObject")

'varaqli tozalash

Cells.Clear

'1-varaqdan yozish

IntRow = 1

'Varaqa kompyuter diskleri haqida axborot yozish

On Error Resume Next

For Each objDrive In objFileSysObject.Drives

```

'Disk harfi
Cells(IntRow, 1) = objDrive.DriveLetter
'Tayoyrgarlik
Cells(IntRow, 2) = objDrive.IsReady
'Disk tipi
Select Case objDrive.DriveType
    Case 0
        Cells(IntRow, 3) = "Noma'lum"
    Case 1
        Cells(IntRow, 3) = "Olinadigan"
    Case 2
        Cells(IntRow, 3) = "Qattiq"
    Case 3
        Cells(IntRow, 3) = "Tarmoqli"
    Case 4
        Cells(IntRow, 3) = " CD-Row"
    Case 5
        Cells(IntRow, 3) = "RAM"
End Select
'Disk nishoni
Cells(IntRow, 4) = objDrive.VolumeName
'Umumiy o'lcham
Cells(IntRow, 5) = objDrive.TotalSize
'Bo'sh joy
Cells(IntRow, 6) = objDrive.AvailableSpace
IntRow = IntRow + 1
Next
End Sub

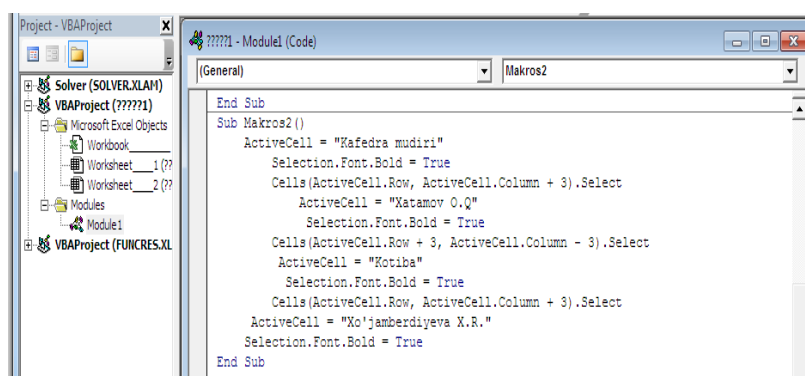
```

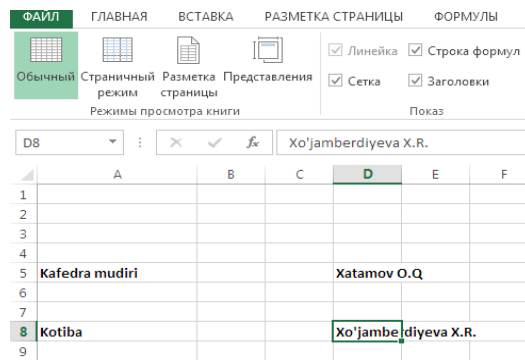
	A	B	C	D	E	F	G
1	C	ИСТИНА	Qattiq		1,05E+11	8,27E+10	
2	D	ИСТИНА	Qattiq		3,95E+11	3,57E+11	
3	E	ЛОЖЬ	CD-Row				
4	F	ИСТИНА	Olinadigan		4E+09	3,77E+09	
5							

## 2.10-расм

Қуйида берилган макрос ёрдамида олдиндан берилган матнни хужжатнинг исталган жойига қўйиш мумкин. Масалан, “Кафедра мудир: О.Хатамов” ва “Котиба: Хўжамбердиева Х.” каби икки сатрдан иборат жумлаларни ҳар бир хужжатнинг охирига қўйишга тўғри келади. Шунинг уни қайта-қайта ёзмасдан бирор бир тугмача макрос ёрдамида бириктириб қўйиш мумкин(2.11.-расм)

```
Sub Makros2()  
    ActiveCell = "Kafedra mudiri"  
    Selection.Font.Bold = True  
    Cells(ActiveCell.Row, ActiveCell.Column + 3).Select  
    ActiveCell = "Xatamov O.Q"  
    Selection.Font.Bold = True  
    Cells(ActiveCell.Row + 3, ActiveCell.Column - 3).Select  
    ActiveCell = "Kotiba"  
    Selection.Font.Bold = True  
    Cells(ActiveCell.Row, ActiveCell.Column + 3).Select  
    ActiveCell = "Xo'jamberdiyeva X.R."  
    Selection.Font.Bold = True  
End Sub
```





2.11.-расм.

Оддий диаграммаларни макрослар ёрдамида чиқариш учун қуйидаги дастурий коддан фойдаланса бўлади.

```
Sub макрос5()
```

```
Charts.Add
```

```
ActiveChart.ChartType = ColumnClustered
```

```
ActiveChart.SetSourceData Range("B1:E1"), xlRows
```

```
ActiveChart.Location xlLocationAsObject, Name
```

```
With ActiveChart
```

```
.HasTitle = True
```

```
.ChartTitle.Characters.Text = Name
```

```
.Axes(xlCategory, xlPrimary).HasTitle = True
```

```
.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =
```

```
Sheets(Name).Range("A1").Value
```

```
.Axes(xlValue, xlPrimary).HasTitle = True
```

```
.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text =
```

```
Sheets(Name).Range("A2").Value
```

```
.HasLegend = False
```

```
.HasDataTable = True
```

```
.DataTable.ShowLegendKey = True
```

```
With .Axes(xlCategory)
```

```
.HasMajorGrindlines = True
```

```
.HasMinorGrindlines = False
```

```
End With
```

With .Axes(xlValue)

.HasMajorGrindlines = True

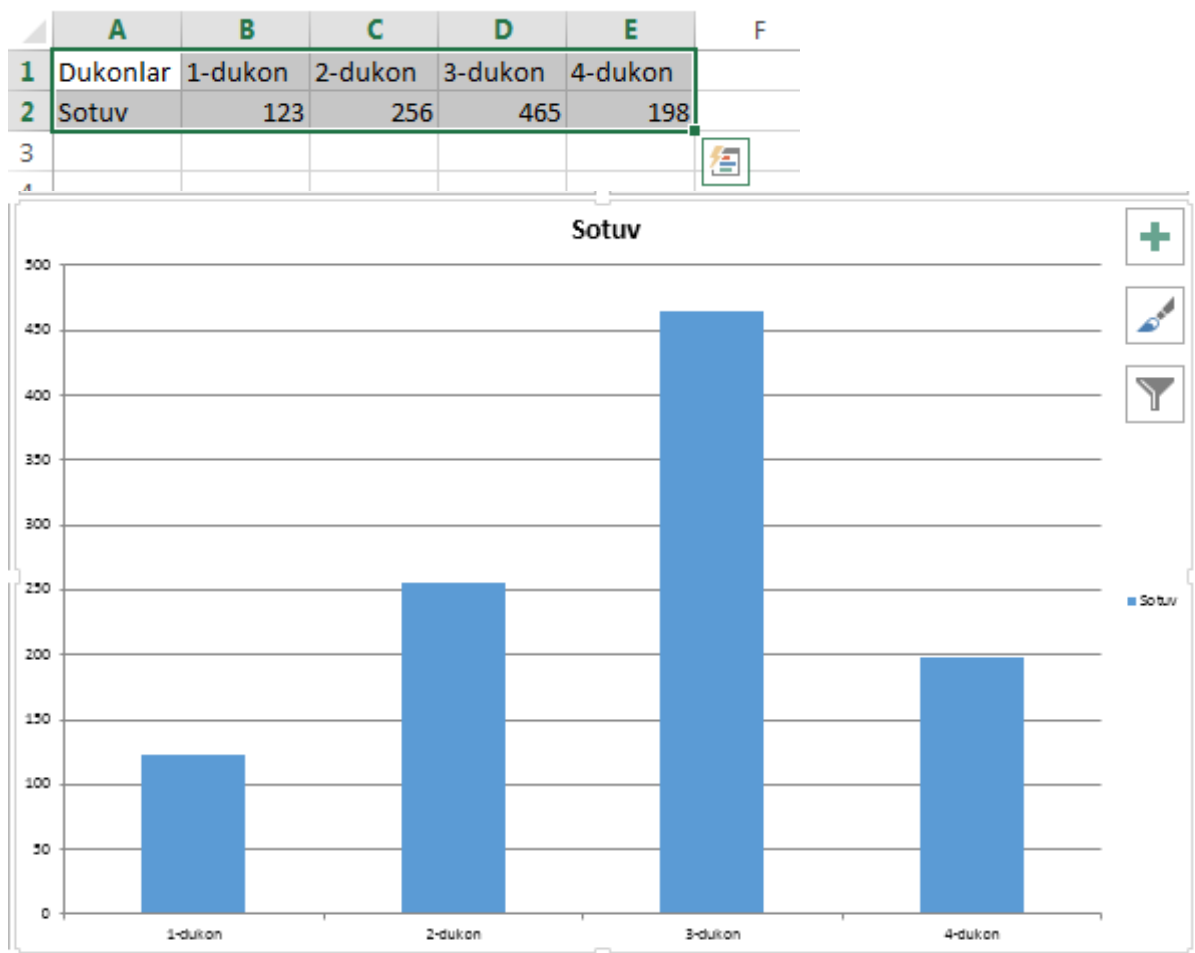
.HasMinorGrindlines = False

End With

End With

End Sub

Натижада 2.12-расмдаги диаграммани ҳосил қилиш мумкин.



**2.12-расм**

Макрослар ёрдамида арифметик амалларни бажариш учун содда калькулятор яратиш имкони ҳам мавжуд. Унинг коди қуйидача:

Sub Makros4()

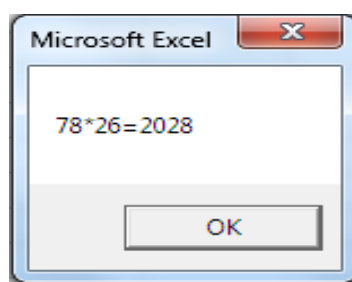
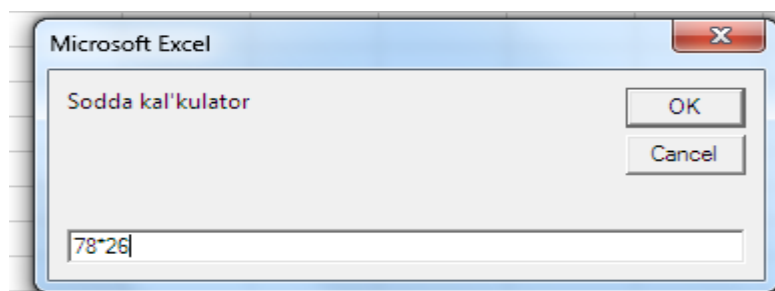
Dim i As String

i = InputBox("Sodda kal'kulyator")

MsgBox i & "=" & Application.Evaluate(i)

End Sub

Ушбу код ишга тушурилгач экранда мулоқот дарчаси пайдо бўлади. Ифодани ёзиб ОК тугмачаси босилса натижа кейинги дарчада ҳосил бўлади(2.13.-расм)



**2.13-расм**

Қуйида келтирилган макрос ҳам чизиқли диаграмма чизиш учун ёзилган. Ушбу макроснинг фарқи шундаки, диаграмма чиқариш ва ўчириш иккита тугмага бириктирилган(2.14.-расм)

```
Sub Makros6()
```

```
Charts.Add
```

```
ActiveChart.ChartType = xlLineMarkers
```

```
ActiveChart.SetSourceData Range ("b1:E3"), xlRows
```

```
ActiveChart.Location xlLocationAsObject, Name
```

```
With ActiveChart
```

```
.HasTitle = True
```

```
.ChartTitle.Characters.Text = Name
```

```
.Axes(xlCategory, xlPrimary).HasTitle = True
```

```
.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =  
Sheets(Name).Range("a1").Value
```

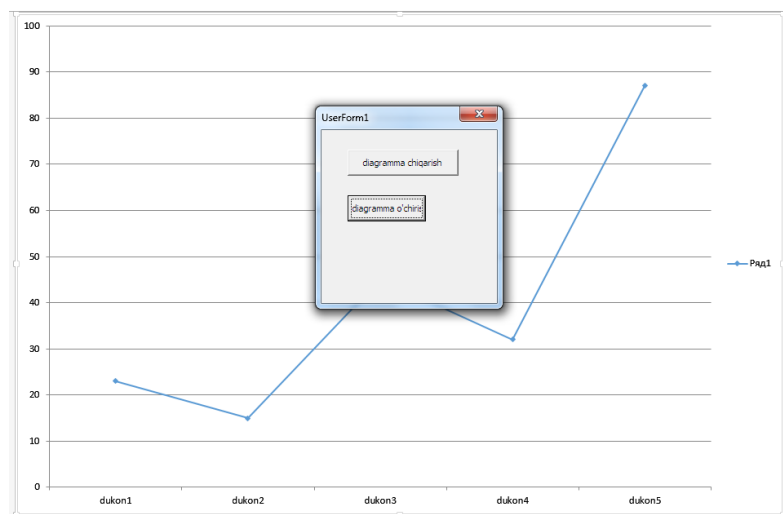
```
.Axes(xlValue, xlPrimary).HasTitle = True
```

```

        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text =
Sheets(Name).Range("a2").Value
        .HasLegend = False
        .HasDataTable = True
        .DataTable.ShowLegendKey = True
    With .Axes(xlCategory)
        .hasmajorgridlines = True
        .hasmajorgridlines = False
    End With
End With
End Sub

Sub deletechart()
    ActiveSheet.ChartObjects.Delete
End Sub

```



**2.14-расм**

Жадвалга элементларни автоматик киритишда қуйидаги макросдан фойдаланамиз.

Sub макрос6()

Dim lngRows As Long, intCols As Integer

Dim lngRow As Long, intCol As Integer

Dim alngValues() As Long

Dim lngStep As Long, lngVal As Long



```

lngRows = Val(InputBox("Ustun balandligi"))
intCols = Val(InputBox("Satr uzunligi"))
ReDim alngValues(1 To lngRows, 1 To intCols)
Set rgRange = ActiveCell.Range(Cells(1, 1), Cells(lngRows, intCols))
For lngRow = 1 To lngRows
    For intCol = 1 To intCols
        alngValues(lngRow, intCol) = Val(InputBox("Elementlarni kiriting"))
    Next intCol
Next lngRow
rgRange.Value = alngValues
End Sub

```

	A	B	C	D	E	F
1						
2						
3						
4						
5			1	2	3	
6			4	5	6	
7			7	8	9	
8						
9						

**2.15-расм**

Электрон жадвалда фаол диапазонни аниқлаш учун қуйидаги макрос ишлатилади.

```

Sub макрос7()
    Dim x1, x2 As String
    x2 = ActiveCell.Address
    x1 = Cells(1, ActiveCell.Column).Address
    Range(x1, ":", x2).Select
End Sub

```

Файлга маълумотлар ёзиш макроси кўриниши қуйидагича

```

Sub makros1()
    Dim h, d As Integer
    Dim i, j As Integer
    Dim A() As String

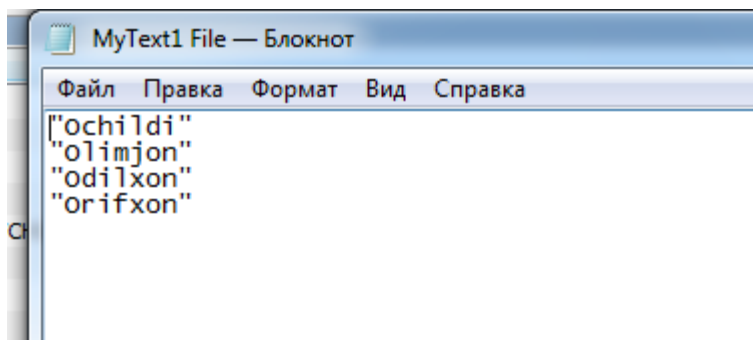
```

```

h = Val(InputBox("Ustun balandligi"))
d = Val(InputBox("Satr uzunligi"))
ReDim A(1 To h, 1 To d) As String
Open "D:\ MyText1 File.txt" For Output As 1
For i = 1 To h
    For j = 1 To d
        A(i, j) = InputBox("Elementlarni kiriting")
        Write #1, A(i, j)
    Next j
Next i
Close 1
End Sub

```

Юқоридаги макросни бажариш вақтида D дискда MyText1 File.txt файли очилиб киритилган маълумотлар ёзилади. Агар ушбу файл таркибини қарасак қуйидаги ёзувларни кўриш мумкин( -расм).



**2.16 -расм.**

## Хулоса ва таклифлар

Хулоса ўрнида шуни таъкидлаш лозимки VBA (Visual basic for Application) имкониятларини фойдаланувчиларнинг офис дастурларида ишлаш жараёнида тез-тез учраб турадиган турли масалаларини ҳал қилишда қўллаш бир қанча қулайликлар яратади. Мазкур масала юзасидан турли дастурий ишланмалар яратилган, бироқ ҳар бир дастурнинг ўзига хос камчиликлари мавжуд. Бу камчиликларни бартараф қилиш учун масала ечимини офис дастуридан чиқмаган ҳолда топиш мақсадга мувофиқ.

VBA кодларидан фойдаланиб ҳужжатни қайта ишлаш учун аввало бирор бир макрос ёзиб, шундан сўнг уни таҳрирлаш қулай.

Офис дастурларида макрос яратиш менюлар каторида “Вид” менюсининг “макросы” командасининг “записать макрос” бандини танлаш орқали бошланади.

Офис дастурларида ўқув эффектлари ва трюкларини яратувчи дастурий иловалар гарчи мустақил ишловчи дастур бўлмаса ҳам уни яратиш дастур яратишнинг анъанавий босқичларига бўйсунган ҳолда амалга оширилади.

Бунда аввало дастур учун математик модель ва алгоритм ишлаб чиқилади ва шу алгоритмга мос командалар дастурлаш тили операторлари кўринишида код муҳаррири ойнасига киритилади.

Яратилган дастур макрос кўринишида актив шаблонга сақланади. Ушбу шаблон файли (унинг кенгайтмаси \*.dot кўринишида бўлади) яратилган дастуирни бошқа компьютерларга ўтказиш учун хизмат қилади.

Бу файлни компьютернинг офис дастурлари(Word, Excel ва бошқа дастурлар) ўрнатилган директориясига мос каталог ичига нусхаланади, бунда каталогдаги эски шаблон файли янгиси билан алмаштирилади. Шуни таъкидлаш жоизки, бу яратилган дастурий иловаларнинг бошқа дастурлардан фарқли томони офис дастурлари билан ишлашни осонлаштиради. Бу дастур компьютер фойдаланувчилари ишларини енгиллаштирувчи яна бир дастурий восита бўлиб ҳисобланади.



### **Фойдаланилган адабиётлар**

1. Ўзбекистон Республикаси Конституцияси Т.: 2012-йил.
2. Ўзбекистон Республикасининг “Ахборот эркинлиги принципи ва кафолатлари тўғрисида”ги қонуни. 2003 йил.
3. Гладкий А.А., Чиртик А.А. Excel. Трюки и эффекты.- СПб.:Питер, 2008.-368 с.
4. Nazirov Sh.A., Qobulov R.V. Obyektga mo‘ljallangan dasturlash. G‘. G‘ulom nomidagi nashriyot-matbaa ijodiy uyi, Toshkent.-2007.
5. Симонович С.В., Евсеев Г.А. Занимательное программирование: Delphi. – М.: АСТ –ПРЕСС КНИГА: Инфорком – Пресс, 2001.
6. Каммингс С. VBA для “чайников”, Пер. С англ.-М.: “Вильямс”, 2001.- 448 с.
7. Культин Н.Б. Delphi в задачах и примерах. – СПб.: БХВ – Петербург, 2005.
8. Фаронов В.В. Программирование на языке высокого уровня: Учебник для вузов. –СПб.: Питер, 2003.
9. Бобровский С.И. Delphi 7. Учебный курс. – СПб.: Питер, 2005.
- 10.Кетков Ю.Л., Кетков А.Ю. Практика программирования: Visual Basic, C++ Builder, , Delphi. - СПб.: БХВ – Петербург, 2005.
- 11.Флёнов М. “Delphi в шутку и всерьез: что умеют хакеры” – Издательский дом «Питер», 2006
- 12.Кузнецова И.А. Практикум по Delphi для решения прикладных задач Нижний Новгород Литература 2005.
- 13.Симонович С.В., Евсеев Г.А. Занимательное программирование: Delphi. – М.: АСТ – ПРЕСС КНИГА: Инфорком – Пресс, 2001.
- 14.Культин Н.Б. Delphi в задачах и примерах. – СПб.: БХВ – Петербург, 2005.
- 15.Фаронов В.В. Программирование на языке высокого уровня: Учебник для вузов. СПб.: Питер, 2003.
- 16.Бобровский С.И. Delphi 7. Учебный курс. – СПб.: Питер, 2005.

### **Интернет сайтлари:**

17. [www.ziyonet.uz](http://www.ziyonet.uz)
18. [www.technoidea.boom.ru](http://www.technoidea.boom.ru)
19. [www.uzinfo.info](http://www.uzinfo.info)
20. [www.intuit.ru](http://www.intuit.ru).
21. [www. Officevba. Com](http://www.Officevba.Com) – Microsoft Office &Visual Basic for Applications Developer
22. [www. windx. Com](http://www.windx.Com) -Visual Basic Programmer's Journal
23. [www.elementkjournals. Com/ivb/](http://www.elementkjournals.Com/ivb/) - Inside Visual Basic
24. [www.vbapro.com/](http://www.vbapro.com/)
25. [www.mvps.org/vbnet/](http://www.mvps.org/vbnet/)
26. [www.vbcity.com](http://www.vbcity.com) /
27. [www.download.cnet.com](http://www.download.cnet.com)
28. [www.geocities.com/WallStreet/9245/](http://www.geocities.com/WallStreet/9245/)