

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ АЛОҚА, АХБОРОТЛАШТИРИШ
ВА ТЕЛЕКОММУНИКАЦИЯ ТЕХНОЛОГИЯЛАРИ ДАВЛАТ
ҚЎМИТАСИ
ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ**

Ҳимояга рухсат

Кафедра мудири : _____

« __ » _____ 2014 й

«Ўзбек-қозоқ икки тилли электрон луғат» мавзусида
Бакалаврнинг битирув малакавий иши

Битирувчи	_____	А.М. Убайдуллаев
	(имзо)	(фамилия)
Раҳбар	_____	Қ.С. Раҳмонов
	(имзо)	(фамилия)
Тақризчи	_____	А.Хайдаров
	(имзо)	(фамилия)
ҲФХ бўйича маслаҳатчи	_____	Е.А.Борисова
	(имзо)	(фамилия)

ТОЛКЕHT

МУНДАРИЖА

КИРИШ	9
1. МАВЖУД ЭЛЕКТРОН ЛУҒАТЛАР ТАҲЛИЛИ	12
1.1. Туркий тилларнинг ривожланиши ва шаклланиши	12
1.2. Туркий халқлар ва туркий тиллар рўйхати	15
1.3. Луғат ва унинг турлари	21
2. ДАСТУРИЙ ТАЪМИНОТНИ ЯРАТИШДА ФОЙДАЛАНИЛГАН ДАСТУРИЙ ВОСИТАЛАР	26
2.1. Java дастурлаш тили	26
2.2. H2 маълумотлар базаси	40
2.3. SQL тили	43
2.4. Objective C дастурлаш тили.....	59

3. ДАСТУРИЙ ТАЪМИНОТ ИШЛАБ ЧИҚИШДАГИ МОДУЛЛАР ВА УНДАН ФОЙДАЛАНИШ.....	65
3.1. Дастурий таъминот модуллари ва уларни яратиш	65
3.2. Фойдаланувчига йўриқнома.....	71
4. ҲАЁТ ФАОЛИЯТИ ХАВФСИЗЛИГИ.....	78
4.1. Корхоналарда меҳнат муҳофазасига оид ишларни ташкил қилиш	78
4.2. Техноген характердаги фавқулодда ҳолатларнинг содир бўлиши. Техноген ҳалокатлар.....	83
ХУЛОСА	86
ФОЙДАЛАНИЛГАН АДАБИЁТЛАР.....	87
ИЛОВА.....	89

КИРИШ

Ҳозирги даврда халқлар орасида сиёсий, маданий, илмий-техникавий, иқтисодий ва бошқа алоқаларни тезкор ривожланиши албатта глобаллашув жараёни, ахборот коммуникацион технологиялари(АКТ)ни юқори тезликда ўсиши ва пировард натижада бутун дунёда ахборотлаштирилган жамиятни куриш билан чамбарчас боғланган. Фан – техника ва технологияларнинг тезкор суръатда ўсиши пировард натижада турли тилларда мос адабиётлар яратилиши билан боғлиқ. Албатта мазкур адабиётлар билан танишиш учун уларни керакли (истеъмолчи) тилларга оператив ҳолда таржима қилиш муҳим аҳамиятга эга. Умуман олганда бир тилдан иккинчи тилга таржима қилиш бу ижодий иш ҳисобланади, чунки маълум бир манбани ҳар хил одам (мухассис) турлича таржима қилади. Таржима қилиш жараёнида маълум сўзларни таржимасини топиш учун оддий луғатларда озмунча вақт талаб этилади. Электрон луғатлар ёрдамида эса бу анча осонлашиш мумкин.

Албатта республикамизда АКТ ни барча соҳага тадбиқ этиш борасида бир неча қонунлар, фармонлар ва қарорлар қабул қилинган [1-4].

Бугунги кунда ахборот коммуникация технологиялари соҳасини жадал суръатлар билан тараққий эттириш давлатимиз иқтисодиётида амалга оширилаётган таркибий ўзгаришлар ҳамда иқтисодий ислохотларнинг бош йўналишларидан бири ҳисобланади. Чунки бу йўналиш нафақат республикани ахборотлашган жамиятга айлантириш учун хизмат қилади, балки, мамлакатимиз иқтисодиётини жадал суръатлар билан ривожлантиришда ўзига хос етакчи ролини ўйнайди. Дастурий вазифалардан келиб чиқиб, мамлакатимизда компьютер ва ахборот технологиялари, телекоммуникация ва маълумот узатиш тармоқларини, интернет хизматларини ривожлантириш ва замонавийлаштириш, уларни дунё стандартлари даражасига етказиш мақсадида кенг кўламли ислохотлар босқичма-босқич амалга оширилмоқда.

Мамлакатимизда бугунга келиб 9765 та мактаб фаолият олиб бормоқда. Шулардан 377 таси Қорақалпоқ тилида, 417 таси Қозоқ тилида, 60 таси

Қирғиз тилида ва 43 таси Туркман тилидаги мактаблардир. Булар қаторида ўрта-махсус касб-хунар ва олий таълим муассасаларини ҳам алоҳида таъкидлаш жоиз.

Фан – техника ва технологияларнинг тезкор суръатда ўсиши натижасида ўқув адабиётларига бўлган талаблар ҳам ортади. Албатта мазкур адабиётлар билан танишиш учун уларни керакли (истеъмолчи) тилларга таржима қилиш муҳим аҳамиятга эга. Шунда таржима қилишда электрон луғат дастури кўмакчи ҳисобланади.

Ушбу замон талаби ва долзарбликдан келиб чиққан ҳолда ушбу битирув малакавий иши ўзбек-қозоқ икки тилли луғат яратишга бағишланган.

Битирув малакавий ишини бажариш учун қуйидаги масалаларни кўриб чиқиш талаб этилади:

- Электрон луғат билан танишиш;
- Ҳозирги вақтдаги мавжуд ўзбек-қозоқ луғатларини таҳлил қилиш;
- Ўзбекча-русча ва қозоқча-русча луғатлар асосида ўзбекча-қозоқча луғатни ишлаб чиқиш;
- Ўзбек-қозоқ луғатни маълумотлар базасини шакллантириш;
- Маълумотлар базаси билан ишловчи дастурий воситаларни ўрганиб чиқиш;
- H2 маълумотлар базаси бошқариш тизимини ўрганиш;
- SQL тилини ўрганиш;
- Java технологияларини ўрганиш;
- JavaFX технологияси ва H2 маълумотлар базаси ёрдамида ўзбек-қозоқ луғатини яратиш;
- Электрон луғатдан фойдаланиш қўлланмасини ишлаб чиқиш;

Битирув малакавий иши кириш, тўртта боб, хулоса, фойдаланилган адабиётлар ва иловадан иборат.

Кириш қисмида ишнинг долзарблиги, мақсади ва боблар бўйича

қисқача маълумотлар берилган.

Биринчи бобда туркий тилларнинг ривожланиши ва шаклланиши, туркий халқлар ҳақида маълумот берилган. Бундан ташқари луғатлар ва уларнинг турлари келтирилган.

Иккинчи боб эса электрон луғатни яратишда фойдаланилган дастурий воситалар ва дастурлаш тиллар ҳақида қисқача маълумотлар берилган.

Учинчи боб эса дастурий таъминотни яратиш, асосий модуллар ва электрон луғатдан фойдаланиш интерфейсини яратишга бағишланган.

Тўртинчи боб эса техника ва ҳаёт фаолияти хавфсизлигига бағишланган.

Хулоса қисмида битирув малакавий ишини бажариш давомида олинган асосий натижалар келтирилган.

Фойдаланилган адабиётлар қисмида БМИ ни ёзиш давомида фойдаланилган адабиётлар ва интернет ресурс манзиллари келтирилган.

Иловада яратилган электрон луғатнинг дастурий коди келтирилган.

1. МАВЖУД ЭЛЕКТРОН ЛУҒАТЛАР ТАҲЛИЛИ

1.1. Туркий тилларнинг ривожланиши ва шаклланиши

Тилларнинг ривожланиши ва такомиллашуви халқ тарихи билан чамбарчас боғлиқ. Тиллар жамият ривожини билан биргаликда тараққий қилиб боради.

Қабила иттифоқининг юзага келиш жараёни ва турли уруғ-қабилаларнинг бирлашиши, уларнинг парчаланиши ва бирикиши, шунингдек, тилларининг бир-бири билан алоқаси натижасида лаҳжалар, шевалар битта қабила тилига бирлашган. Бу шева, лаҳжалар умумий луғат бойлигига ва грамматик қурилишга эга бўлди. Оқибатда бу ўзгаришлар асосий лаҳжа элат тили ёки миллий тилни юзага келтирди.

Уруғ ва қабила тилларининг, шунингдек, миллий тилларнинг ривожланиши туркий тиллар учун ҳам характерли. Туркий уруғ ва қабилаларининг бирлашиши ва парчаланиши III асрдан, ва аниқ маълумотларга таянадиган бўлсак, VI асрдан XV-XVI асрларгача давом этди, ниҳоят, миллий тиллар вужудга келди.

Шундай қилиб, муайян халқнинг юзага келиши, биринчидан, уруғ-қабила тузилишининг, иккинчидан, шу уруғ ёки қабила тилининг мерос бўлиб қолиши билан боғлиқ. Бироқ муайян қабила ва элатнинг аралашиб ёки чатишиб кетишига, унинг тилида субстратнинг мавжуд бўлиши унчалик катта таъсир кўрсатмайди.

Ҳар бир тил учун асос бўлган тил ҳар хил бўлиши мумкин. Масалан, қарақалпоқ тили учун асос (ўзак) тил қипчоқ-нўғай тили бирлиги бўлган, XIV-XVI асрдан бошлаб асос тил қипчоқ тилига айланган, XIV-XV асрларда эса Ғарбий Хунн тили асос бўлган, бу тилни синчиклаб ўрганиш, унда мўғул, тунгус-манчжур, ҳатто, олтой тилининг белгилари борлигини ҳам кўрсатади.

Шуниси характерлики, муайян қабила иттифоқи ёки элат тили билан ишлаб чиқариш орасида узвий алоқадорлик мавжуд. Масалан, кўчманчи турк халқлари чорвачилик билан шуғулланишган ва улар тилида шевалар

орасидаги фарқлар анча кам. Ўтроқ туркий халқлар эса қишлоқ хўжалиги билан шуғулланишган ва уларнинг тилида ички фарқ жуда катта. Масалан, қозоқ тилида ички фарқ анча кам, ўзбек тилида эса бу хусусият жуда катта. Қозоқ халқи ХХ аср бошларигача уруғ-қабилачилик хусусиятларини яхши сақлаган эди. Ўзбек ва уйғур халқлари эса уруғ-қабилачиликни унутишган. Шунинг учун ҳам уйғурлар одамларни яшаш жойи билан аташади, яъни, кашқарликлар, турфонликлар, ёрқентликлар каби.

Шу билан бирга, баъзи бир туркий тиллар гуруҳи ривожини учун уларнинг диалектик қурилиши муҳим омил бўлган. Уларнинг диалектик асосининг шаклланишида ички имкониятлардан ташқари, ташқи факторлар ҳам таъсир этган.

1. Бир-бирига яқин лаҳжали тиллар тизими. Биринчидан, туркий халқларнинг айрим гуруҳлари ўз тилларида, қабилалар уруғ тиллари хусусиятларини феодалистик тузумнинг охиригача ё унутишган ёки сақлаб қолишган. Туркий тилларнинг энг қадимги диалектик қурилишини ижтимоий ўзгаришлар юзага келтирди.

Иккинчидан, шу халқ диалектик қурилиши унинг уруғ-қабилалар таркибининг бир-бирининг ўхшаш ёки ўхшамаслигига боғлиқ. Масалан, туркманлар уруғ-қабилалар тузимига кўра бир-бирининг яқин, улар келиб чиқишига кўра ўғузлардир. Ўзбеклар эса учта катта қабилалар гуруҳидан келиб чиққан: қарлук, ўғуз, қипчоқ.

Учинчидан, айрим халқларнинг диалектик қурилишида уруғчилик тузимининг қолдиқлари сақланмаган ёки маъмурий халқлар шеваси шу ҳудудда яшовчи бошқа шевалар билан тўғри келмайди. Бироқ ривожланиш жиҳатидан шу ҳудудда яшовчи бошқа халқлар тиллари билан ҳамкорликда, алоқадорликда олдинга силжишди. Масалан, уйғур халқлари тили феодализм давридаёқ элат тили сифатида ривожланган ва уруғчилик қолдиқларини унутган. Қорақалпоқ тили уруғ-қабилалар тузилиши жиҳатдан бир-бирига яқин таркибида бир қанча шевалар бор, бироқ бошқа қўшни тиллар билан биргаликда ривожланган.

Тўртинчидан, ҳар бир тилнинг диалектик тузилиш турли хил субстратларнинг таъсири билан характерланади.

а) халқнинг уруғ-қабилачилик хусусиятларини яхши сақлаган лаҳжалар. Бундай лаҳжалар иккига бўлинади:

а) хакас тилининг шевалари (бу лаҳжа ўзида соғай (балтир), қочин, қизил, шор лаҳжа ва шевалари билан) уйғур уруғлари ва қабилаларининг бир-бирига яқин диалектлари асосида 1917 йилдан кейин ягона хакас элат тилини ташкил қилди.

б) туркман тилининг шевалар тизими. Бу тизим ўғуз лаҳжасининг бир-бирига яқин йомуд, така, гўклан, салир, сариқ, эрсари каби шевалари асосида туркман халқ тилини шакллантирган. Бироқ бу тилнинг шаклланишида ташқи факторларнинг, хусусан, Эрон диалектларидан нохурли, анаули, хасарлиларнинг таъсири ҳам мавжуд.

2. Бир-бирига яқин бўлмаган қabila диалектларидан тузилган тиллар тизими.

а) тоғли олтой тили, бу бир-бирига яқин бўлмаган икки лаҳжани ўз ичига олади:

1) шимолий гуруҳ шевалари (қадимги уйғур қабилалари асосида шаклланган туба, куманди, гуклан шевалари);

2) жанубий гуруҳ шевалари (келиб чиқиш жиҳатидан қадимги қирғиз қабилалари шевалари бўлмиш олтой (уйрот), телеут, теленгит шевалари).

б) ўзбек тилининг диалектлар тизими:

1) ўғуз;

2) қарлуқ;

3) қипчоқ.

Айрим халқнинг этник тузилишида уруғ-қabila излари сақланган бўлса-да, шева ва лаҳжалари худудий белгиларига қараб белгиланади. Бу ҳам турлича бўлади.

1. Шева ва лаҳжалари бир-бирига яқин қabila хусусиятлари асосида тузилган, асосий фарқлари худудий белгилари деб белгиланадиган тиллар.

а) қирғиз тили тарихан қадимги қирғиз тили икки асосий диалект билан боғлиқ: 1) қабила тилининг айрим хусусиятларини сақлаб қолган шимолий лаҳжа; 2) ўзбек тили билан алоқадорликда вужудга келган жанубий лаҳжа;

б) ўзида уруғ-қабила белгиларини яхши сақлаб қолган, аммо дифференцияси уруғ-қабила белгилари асосида эмас, балки ҳудудий белгилари асосида ривожланган шевалар тизими: бундай ривожланишни козоқ, қорақалпоқ ёки нўғай тиллари мисолида кўриш мумкин. Масалан, козоқ тили уч асосий ҳудудий диалектга эга: 1) шимолий-шарқий; 2) жанубий; 3) ғарбий;

2. Халқ тилларининг уруғ-қабила белгиларини сақлаган бўлса ҳам, бир хил бўлмаган уруғ-қабила белгиларига асосланадиган шевалар тизими.

3. Уруғ-қабила белгиларини сақлаган бўлса ҳам, шева ва лаҳжалари бир-бирига яқин бўлмаган халқлар тилининг диалектик тизими. Бундай тиллар тизимига ҳудудий белгилар асос қилиниб олинади.

1.2. Туркий халқлар ва туркий тиллар рўйхати

Мазкур бўлим [1] адабиёт асосида тайёрланган. Бу адабиётда илмий адабиётларда учрайдиган деярли барча туркий тил ва туркий халқлар киритилган. Рўйхатни тартиблашда биринчи навбатда миллат ва элатларнинг гуруҳлар таркибидаги номланиши келтирилди. Бундан ташқари, гуруҳ ва кичик гуруҳлар таркибига кирувчи энг кичик этник гуруҳлар ҳам рўйхатдан ўрин олган.

Туркий миллат ва элатларнинг номларини келтиришда ҳар бир халқлар номининг илмий адабиётларда учрайдиган вариантларининг деярли барчасини киритишга ҳаракат қилинди (қиёс. мас. олтойликлар, ўйротлар, оқ қалмиқлар каби), ҳозирги замон туркий тиллари учун эскирган атамалар қавс ичида (эск.) тарзида қайд қилинган [1].

Бундан ташқари, рўйхатда туркий тилларнинг ривожланиш ва шаклланиш жараёнида бир неча номлар билан аталганлиги ҳисобга олинди

ва улар маълум тартибда қайд қилинган.

Барча номлар алифбо тартибида келтирилди: аввал ўзбекча аталиши, катта қавс ичида халқ номининг транскрипцияда кўрсатилиши, айлана қавсда тилнинг гуруҳ ва кичик гуруҳлардаги ўрни жойлаштирилди. Барча халқ номлари (хоҳ у кўп сонли, хоҳ кам сонли бўлсин) мустақил тилга эга ҳисобланади, шунинг учун улар * билан, адабий тилга эга бўлган халқлар эса ** билан белгиланган.

Тенг аталадиган бир хил номларнинг (= тенглик аломати орқали) иккаласи ҳам бир хил қўлланади, шу сабабли улар махсус белги билан кўрсатилди. У ёки бу гуруҳга кирувчи этник қатламларни бир-бирига солиштириш усун қиёс. (қиёсланг), бошқа атама билан чоғиштириш эҳтиёжи туғилган ҳолларда қар. (қаранг) ҳаволаси қўлланган.

Рўйхатдан қадимги халқлар ҳамда ўлик тилларнинг номлари ҳам жой олган, улар эса қадим. (қадимги) сўзи орқали ажратиб кўрсатилган.

Абаларлар [аба-кижи], бошқа номлари: абинликлар — қар. шорликлар.

Абинлилар=абаларлар — қар. шорликлар.

Аварлар эск. — қар. булғорлар.

Адербижанлар эск. =озарбайжонлар.

**Озарбайжонлар [озарлар, азэрбайжан] (ўғуз гуруҳининг ўғуз-салжук кичик гуруҳига киради) - қиёс. муғаллар, қозоқ татарлари, қашқайлар, қизилбошлар, терекемлар, қорапапахлар (бошқача аталиши: озарбайжонлар, адербиджанлар, турклар, озарбажон турклари, озарбажон татарлари, кавказ орти татарлари).

**Озарлар қадим. [озарлар] (ўғуз-салжук кичик гуруҳига тегишли эски озарбайжон тили) - қар. озарбайжонлар.

Оқ нўғайлар [ақ ноғай] — қар. нўғайлар.

Оқсуйлар — қар. уйғурлар.

Олтойлар [алтай кижиди] (қирғиз-қипчоқ гуруҳига киради) - қиёс. маймалар, теленгетлар (чуйлилар, чуй қалмиқлари, телеслар); телеутлар, кумандинлар, лебеденлар (оққушликлар), тубалар (бошқача аталиши:

ўйротлар, олтойлар, тоғли олтойликлар, оқ қалмиқлар, тоғли қалмиқлар, олтой қалмиқлари).

Тоғли олтойлар эск. = олтойлар.

Аринлар = арин татарлари — қар. татарлар.

Балқарлар [балқар//малқар] (қипчоқ-половец кичик гуруҳига кирувчи қорачой-балқар тили).

Бараба = барабин татарлари — қар. татарлар.

Барабинликлар = барабин татарлари — қар. татарлар.

**Бошқирдлар [башқурт] (қипчоқ гуруҳининг қипчоқ-булғор кичик

Қирғиз-қозоқлар эск. = қар. қозоқлар.

**Қозоқлар [қазақ] (қипчоқ гуруҳининг қипчоқ-нўғай кичик гуруҳи тиллари), бошқача аталиши: қирғизлар, қирғиз-қайсоқлар, қайсоқ-қирғизлар, қозоқ-қирғизлар.

Олтой қалмиқлари эск. — олтойликлар.

Оқ қалмиқлар эск. — олтойликлар.

Тоғли қалмиқлар эск. — олтойликлар.

Чуй қалмиқлари = теленгитлар — қар. олтойликлар.

Қамасиликлар [қаңмажилар] (уйғур тилларининг хакас кичик гуруҳига киради), бошқача аталиши — кангаласлар.

Қарагаслар эск. — тофаларлар.

Қараимлар [қарай] (қипчоқ гуруҳи тилларининг қипчоқ-половец кичик гуруҳига киради) — қиёс. литва қараимлари, луцкликлар, галицкликлар, қримликлар.

**Қарақалпоқлар [қарақалпоқ, қаралпақ] (қипчоқ гуруҳи тилларининг қипчоқ-нўғай кичик гуруҳига киради) — қиёс. хива қарақалпоқлари, хоразм қарақалпоқлари, тошкент қарақалпоқлари, фарғона қарақалпоқлари, қўқон қарақалпоқлари, астрахан қарақалпоқлари.

Қора қирғизлар [қара қырғыз] эск. = қирғизлар.

Қараманлийликлар — қар. болқон турклари.

Қора нўғайликлар [қара ноғай] қар. нўғайликлар.

****Қирғизлар [қырғыз]** (қирғиз-қипчоқ гуруҳи тиллари), бошқача аталиши: қора қирғизлар [қара қырғыз], бурутлар [бурут], ёввойи тош қирғизлар.

****Қадимги қирғизлар [қырғыз]** (уйғур-ўғуз гуруҳининг уйғур-тукю кичик гуруҳига кирувчи тиллар), бошқача аталиши: енисей қирғизлари, қадимги турклар.

Қовулар қадим. — қар. узлар.

Қойбаллар, қойбалликлар = [хойбал] — қар. хакаслар.

Кондомликлар = [шор] — қар. шорликлар.

Конярлар=юрүклар — қар. болқон турклари.

Қримчоқлар — қар. қрим татарлари (бошқача аталиши: қрим жухудлари).

Қряшёнлар [крэшон] эск. — қар. татарлар.

Қумандинликлар [кубанды//қуманды//қуванды-кижи] (уйғур-ўғуз гуруҳи тилларининг хакас кичик гуруҳига киради) — қар. олтойликлар.

****Нўғайликлар [ноғай]** (қипчоқ гуруҳи тилларининг қипчоқ-нўғай кичик гуруҳига киради) — қар. қоранўғайликлар, окнўғайликлар, (бошқача аталиши: кубан нўғайлари, доғистон нўғайлари).

***Ғарб ўғузлари [оғуз]** (ўғуз гуруҳининг ўғуз-туркман кичик гуруҳи тиллари) — қиёс. ўн ўғуз, тўккуз ўғуз.

****Қадимги шарқ ўғузлари [оғуз]** (уйғур-ўғуз гуруҳининг уйғур-тукю кичик гуруҳи тиллари), бошқача аталиши: тукю ва тукюэ, кўктурклар, қадимги турклар, урхун турклари.

Ўйротлар [ойрот] эск. = олтойликлар.

****Усманийлар I қадим. [османли]** (ўғуз гуруҳининг ўғуз-салжүк кичик гуруҳига кирувчи эски усманийлар тили), бошқача аталиши: оттоманлар, оттоманликлар — қар. турклар.

Усмонлилар II [османлы] эск. — турклар.

****Татарлар [татар]** (қипчоқ гуруҳи тилларининг қипчоқ-булғор кичик гуруҳига киради), қиёс. волга бўйи татарлари: қозон татарлари ёки

қозонликлар, астрахан татарлари, қосимов татарлари ёки қосимовликлар, мишарлар ёки мишарликлар, қряшёнлар, нуғайбоқлар, типтарлар; сибир татарлари ёки ғарбий сибирликлар: турин татарлари, тюмен татарлари ёки тюменликлар, ишим татарлари ёки ишимликлар, ялутор татарлари ёки ялуторликлар, иртиш татарлари, тобол татарлари ёки тоболликлар, тар татарлари ёки тарликлар, сибир бухороликлари, чац татарлари, арин татарлари, барабин татарлари ёки барабинликлар, томск татарлари.

Абакан татарлари эск. = хакаслар.

Озарбайжон татарлари эск. — озарбайжонлар.

Белорус татарлари — қар. қрим татарлари.

Енисей татарлари эск. = хакаслар.

Кавказ орти татарлари эск. озарбайжонлар.

Ғарбий сибир татарлари = сибир татарлари — қар. татарлар.

Қозоқ татарлари — қар. озарбайжон.

**Қрим татарлари [ноғай] (қипчоқ гурухининг қипчоқ-ўғуз кичик гуруҳига тегишли) — қиёс. литва татарлари, белоруссия татарлари, қримчоқлар, бошқача аталиши: таврия татарлари, қрим татарлари, шимолликлар ёки чўлликлар.

Ялутаровск татарлари, бошқача номланиши — ялутаровликлар — қар. татарлар.

Такаликлар [тэкэ] — қар. туркманлар.

Теленгитлар [тэлэңит], бошқача номланиши: чуйликлар, қалмиқлар,, двоеданликлар, телеслар — қар. олтойликлар.

Телеслар [төлөс] = теленгетлар — қар. олтойликлар.

Телеутлар [тэлэңут, теленғут] — қар. олтойликлар.

Типтарлар I [типтэр] — қар. бошқирдлар.

Типтарлар II [типтэр] — қар. татарлар.

Таракамалар [тэрэкэмэ] — қар. озарбайжонликлар.

Тиргешлар = тубаларлар — қар. олтойликлар.

Тоболликлар = тоболлик татарлар — қар. татарлар.

Тўккуз ўғузлар қадим. [тоққуз-оғуз] — қар. ғарб ўғузлари.

Торқлар қадим. — қар. узлар.

**Турқлар [түрк] (ўғуз гуруҳи тилларининг ўғуз-салжук кичик гуруҳига киради), бошқача номланиши: анатолий турқлари, усмонли турқлар, оттоман турқлари, оттоманлар.

Анатолий турқлари эск. = турқлар.

Усмонли турқлар эск. = турқлар.

Оттоман турқлари эск. = турқлар.

Тозлук турқлари — қар. болқон турқлари.

**Туркманлар [түркмэн] (ўғуз гуруҳи тилларининг ўғуз-туркман кичик гуруҳига киради) — қиёс. такаликлар [тэкэ], йомудлар [йомут], гўкланлар [гөклэн], чаудорлар [чавдур], салирлар [салыр], сариклар [сарық], эрсариликлар [эрсары]; нурота туркманлари.

Шимолий кавказ туркманлари = трукманлар.

Ставропол туркманлари = трукманлар.

Турпелар қадим. — қар. узлар.

Турфонликлар [турпанлық] — қар. уйғурлар.

Тюменликлар = тюмен татарлари.

Турқлар [түрк] эск. — озарбайжонликлар.

Озарбайжон турқлари эск. = озарбайжонликлар.

**Уйғурлар [уйғур] (қарлук гуруҳи тилларининг қарлук-хоразм кичик гуруҳига киради) — қиёс. қашқарликлар, турфонликлар, ёркентликлар, хўтанликлар, хамийликлар, таранчлар, оқсуйликлар, саларлар, сариуйғурлар, лобнорликлар, хотонликлар ёки хотонлар.

Қадимги уйғурлар [уйғур] (уйғур-ўғуз гуруҳи тилларининг уйғур-тукю кичик гуруҳига тегишли), бошқача номланиши — қадимги турқлар.

Сари уйғурлар — қар. уйғурлар.

Урумлар [урум] (ўғуз гуруҳи тилларининг ўғуз-салжук кичик гуруҳига киради).

Урянхайлилар эск. = туваликлар.

**Хазарлар қадим. [булғор гуруҳи тилларига киради].

**Чувашлар [чұваш] (булғор гуруҳи тилларига киради) — қиёс. қуйи чувашлар [анатри, анатри чұвашсэм] — қар. чувашлар.

Юқори чувашлар [вирийал чұвашсэм] қар. чувашлар.

Қуйи чувашлар [анатри, анатри чұвашсэм] қар. чувашлар.

Чуйликлар [чуй-кижи] — теленгитлар — қар. чувашлар.

Чулимликлар = чулим татарлари.

Шахшилиглар [шахшылығ] = лебеденликлар — қар. олтойликлар.

*Шорликлар I [уйғур-ўғуз гуруҳи тилларининг хакас кичик гуруҳига киради] — қиёс. абаларлар ёки абинликлар, кондомликлар, матурликлар, бошқача номланиши: шорлар, томск-кузнецк татарлари, кузнецк татарлари.

Шорликлар II [шор] — қиёс. хакаслар.

Шорлар = шорликлар.

Эрсаликлар [эрсары] — қар. туркманлар.

Юруклар — қар. болқон турклари.

**Ёқутлар [саха, сақа] (уйғур-ўғуз гуруҳи тилларининг ёқут кичик гуруҳига киради) — қиёс. долғанлар.

Ялутарлар = ялутар татарлари = қар. татарлар.

Ёркентликлар [йә(р)кентлик] — қар. Уйғурлар [1].

1.3. Луғат ва унинг турлари

Луғат – маълум бир тамойилларга асосан жойлаштирилган ва уларнинг қиймати ҳақида маълумот берувчи, муомалада бўлган, яратилган ва бошқа тилга таржима қилиш мумкин бўлган сўзлар тўпламидан ташкил топган китоб (ёки морфема, сўз иборалари, идеома ва бошқалар). Лингвистик луғатлар ёки предмет ва унинг тўғрисидаги маълумотлар, фан ва маданият соҳасининг қайсидир фаолиятини англаувчи ва бошқалар [1].

Француз ёзувчиси Анатол Франсанинг фикрига кўра, луғат – бу алфавит тартибида жойлашган олам. Луғатлар нафақат бирор бир сўзни

маъносини билмай қолганда ишлатилади, балки у миллий маданиятнинг муҳим бўғини саналади [1].

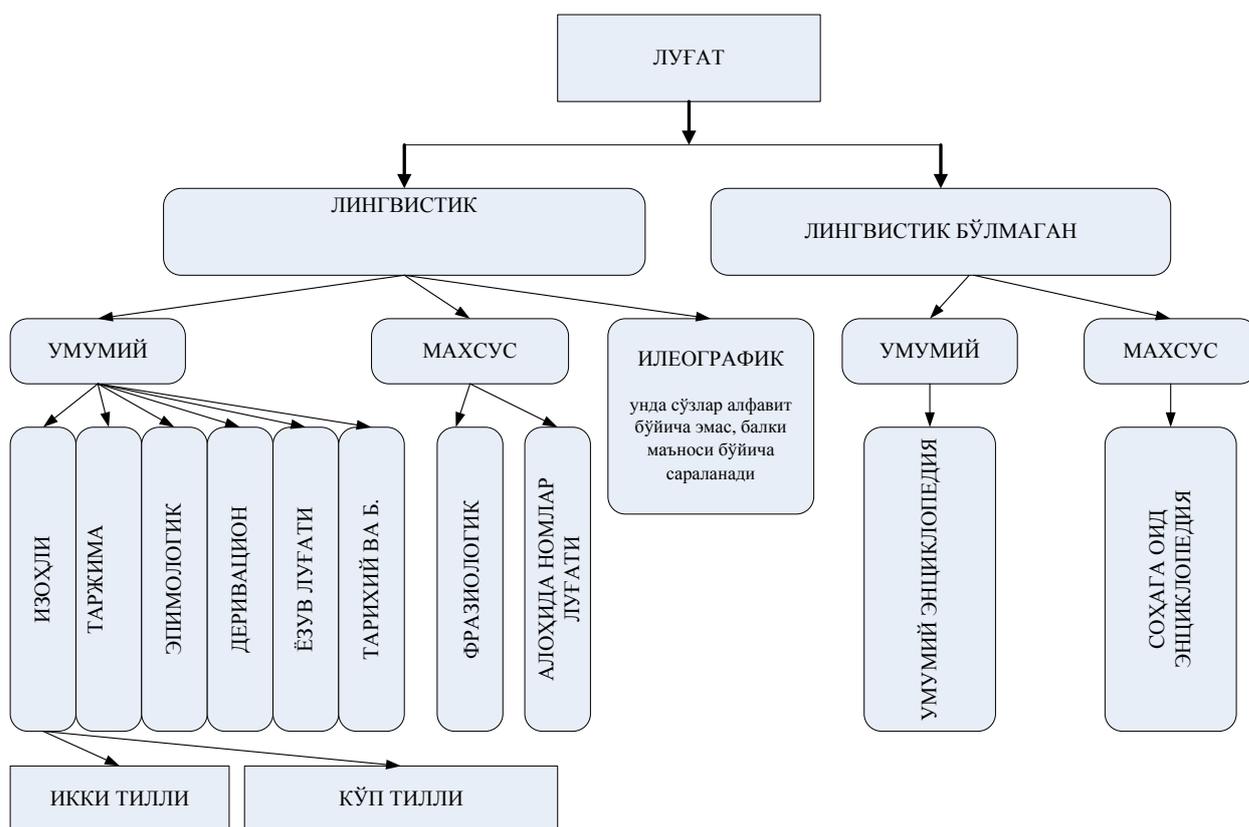
Wikipedia маълумотлар базасида электрон луғатнинг таърифларидан бири шундайки, электрон луғат бу компьютер ёки бошқа электрон қурилмадаги луғат деб таъриф берилган [2].

Бундай луғатлар керакли сўзни морфологияга асосланган ҳолда ва сўз ибораларини (қўлланишига кўра) излаш имкониятидан келиб чиқиб тез топиш имконини беради. Шунингдек, уни бошқа тилга таржима қилиш мумкин.

Техник нуқтаи назардан қараганда, электрон луғат – компьютердаги маълумотлар базаси, яъни луғатга оид мақолаларни кодланган тартибдаги кўриниши бўлиб, керакли сўзларни (фразалар, сўз бирикмалари) тез излаш имконини беради.

Ҳозирги вақтда электрон луғатлар бўйича жуда кенг кўламдаги танлов мавжуд (одийликдан тортиб, токи битта дастурий қобикқа жамланган, бир нечта махсус мавзуга оид луғатлар мавжуд, масалан, *lingvo*). Фойдаланувчи у ёки бу дастурий маҳсулотни ўз имконият даражаси, молиявий ҳолати, бажарадиган иш кўлами ва ўз кўрсаткичларидан келиб чиққан ҳолда танлаш имкониятига эга. Улардан талабалар, ўқувчилар, тадбиркорлар ва турли соҳа фойдаланувчилари фойдаланишлари мумкин.

Луғатларнинг турлари. Луғатлар электрон ва қоғоз кўринишида бўлиши мумкин. Булардан келиб чиққан ҳолда, луғатлар ҳар хил вазифага эга. Улар типларига кўра қуйидагича бўлинади (1.1-расм).



1.1-расм. Луғат ва унинг турлари

Электрон луғатлар ўз соҳасига доир фақат битта вазифани (масалан, изоҳли луғат) ва бошқалари эса кўп функцияларни бажаради (масалан, *abby lingvo*).

Компьютер дастурларидан фойдаланган ҳолда луғатларни тўлдириш. Ҳозирги вақтда бу муаммони ечишнинг маълум бўлган бир нечта усуллари бор:

- луғатларни тўлдириш учун кўп сонли ишчиларни жалб қилиш;
- луғатларни таҳрирлаш ва фойдаланувчиларга уни тўлдириш имкониятини тақдим этиш;
- мавжуд луғатлардан маълумотларни импорт қилиш (жумладан, тасвирларни қайта ишлаш воситаларидан фойдаланган ҳолда, сканер орқали нусхаланган луғатларни киритиш).

Масалан, *google online*-таржимони ва луғати сўз ёки сўз бирикмасини таржима қилишни имкониятини топа олмаса ўз вариантини таклиф этади ва уни сақлаб қўяди. *Wikipedia* энциклопедияси ҳам худди шундай схемада

ишлайди.

Электрон луғатларни асосий имкониятлари. Мисол сифатида, *abby lingvo* х3 луғатини оламиз. Бу электрон луғат ўз ичига “китоблар жавонини” олган. Унда нафақат луғат таржималари жойлаштирилган, балки изоҳли ва энциклопедик сўзлар ҳам киритилган.

Электрон луғатларни аналогик қоғозли нашрлари билан солиштирилгандаги афзалликлари. Электрон луғатларни анъанавий луғатлар билан солиштирганда бир қатор афзалликларга эга. Замонавий электрон луғатлар нафақат китобнинг ҳажмига нисбатан устунликка эга, балки унда сўз ёки сўз бирикмасини қидириш жудаям тез амалга ошади. Қидиришни ихтиёрий формада амалга ошириш мумкин. Масалан, *abby lingvo* дастурини олайлик, барча офис иловалари ва белгиланган сўзларни бир нечта клавишларни босиш ёрдамида амалга ошириш мумкин. Электрон луғатлар нафақат сўзларни транскрипциясини балки, уларни талаффузини ҳам ўз ичига олади. Бу ерда иккита ёндашув бор. Мультилексада товуш синтезатори ўрнатилган ва у барча сўзларни талаффуз қилади. Одатда, ҳар доим бу ёндашувга ишониб қолиш керак эмас. Синтезатор сўз талаффузини умуман бошқача талқин қилиши ёки урғуни нотўғри қўйиши мумкин. *abby lingvo* да асосий лексикани диктор оксфорд талаффузида ўқийди. Албатта, яхши электрон луғатларни энг асосий афзаллиги бир вақтнинг ўзида нафақат сўз бўйича қидиради, балки бутун бошли катта ҳажмдаги луғатдан маълумотларни қидиради. Бу ишни қоғоз вариантда ишлашнинг умуман имкони йўқ.

Электрон луғатни камчиликлари. Электрон луғатлар оддий луғатларни бир нечта камчиликларини ўзига мерос қилиб олган. Асосий муаммо бу луғатнинг тўлиқ бўлмаганлигидир. Шунингдек, луғатни долзарб ҳолатда ҳар доим қўллаб-қувватлаб туриш ҳам долзарб муаммолардан саналади. Бундай муаммо унчалик яхши кўринмаслиги мумкин, лекин бундай муаммолар ўз ўрнига эга. Камчиликларга дастурий воситалар тўпламининг аниқ бўлганларини талабидан келиб чиққан ҳолда кўпгина луғатларни келтириш

мумкин.

Бундай камчиликлар йиллар ўтган компьютерлаштиришни юқори темпда ривожланиши сайин йўқолиб бормокда. Жумладан, кўчиб юрувчи компьютерлар, ноутбуклар, нетбуклар, смартфонлар ва бошқаларнинг ривожланиши билан бу муаммоларни йўқолишига олиб келмокда. Бундай муаммоларни ечиш учун аниқ бир дастурий воситалар ишлаб чиқарувчилар томонидан ҳар хил платформалар (кроссплатформали) учун ишлаб чиқармокдалар. Электрон луғатларнинг яна бир камчилиги улар қоғоз кўринишидаги луғатлардан қиммат туришидир.

2. ДАСТУРИЙ ТАЪМИНОТНИ ЯРАТИШДА ФОЙДАЛАНИЛГАН ДАСТУРИЙ ВОСИТАЛАР

2.1. Java дастурлаш тили

Ахборот ва коммуникацион технологияларнинг таракқиётида дастурлаш ва дастурий воситалар яратилишининг ўзига хос ўрни бор. Техник база ривожланиши ва унинг имкониятларининг кенгайиши ушбу соҳадаги мутахассис муҳандислардан яратилган замонавий техник имкониятлардан самарали фойдаланиш муаммосини кўяди. Техник имкониятлар эса дастурий воситалар ёрдамида очиб берилади, яъни техник имкониятлар кенгайганлиги муаммоларнинг янги янги дастурий ечимлари ҳосил қилинишига йўл очади. Демак замонавий дастурчи локал компьютер, ягона операцион тизим ва ягона техник платформа асосида иш кўриб, бирор бир дастурлаш тили ёрдамида дастурий воситалар яратиш орқали ўзининг олдига қўйилган муаммоларни ҳал қила олмайди.

Дастурчи энди бир неча дастурлаш тилларини билиши, бир неча хил операцион тизимларнинг ўзаро ахборот алмашинувини бошқариши ва бир неча хил техник платформаларнинг ўзаро ҳамкорликда ишлай олишини таъминлаши керак бўлади. Яъни шундай дастурий воситалар яратилиши керакки, улар на операцион тизимларга ва на техник платформаларга боғлиқ бўлмаслиги керак. Айниқса, Веб лойиҳалар яратилаётганда бу муаммолар яққол сезилади. Веб лойиҳалар интернет технологияси асосида ишлаб, интернетга уланган компьютерларнинг техник платформаси ва операцион тизимлари ҳар хил бўлишини кўзда тутаяди. Яъни, бир жойда ИБМ PC архитектурасидаги компьютер Windows операцион тизими билан, бошқа жойда эса RISK архитектурали компьютер OS/2 операцион тизими билан, яна бир жойда эса Main Frame архитектурали ҳисоблаш машинаси Unix операцион тизими билан интернет тармоғига уланган бўлиши мумкин. Яратилган дастур комплекси биринчидан, уларнинг барчасида ишлай олиши, иккинчидан, улар ўзаро ахборот алмашганда, ушбу хараён ҳеч бир

муаммосиз амалга ошиши керак. Дастурчи дастур комплексини бир неча қисмларга ажратиб маълум қисмини бир техник платформага мослаштириб бир дастурлаш тилида, иккинчи қисмини бошқа техник платформага мослаштириб бошқа дастурлаш тилида, худди шунингдек операцион тизимларнинг ҳам ҳар ҳиллигини эътиборга олиб, яна бошқа дастурлаш тилларин қўллаши орқали яратиши керак бўлади. Масаланинг бундай қўйилишида дастурчи имкониятлари чекланганлиги туфайли дастур комплексини яратиш ишлари охиригача етмаслиги мумкин. Нима қилиш керак? Яратиладиган дастур комплекси барча техник платформаларда ва барча операцион тизимларда муаммосиз ишлаши, бундан ташқари эса у кам харажатли бўлиши керак.

Юқоридаги муаммолар Java дастурлаш технологияси ёрдамида анча осон ҳал қилиниши мумкин. Java дастурлаш технологияси Java дастурлаш тилига асосланган бўлиб, Java намунавий дастурлаш тили сифатида қабул қилинади. Java дастурлаш тили базасида Java иловалар, JavaScript функциялари, Java applet лар, Java Beans, маълумотлар базаси билан ишлаш (JDBC), GUI, servletлар, JSP, интерактив динамик Web саҳифалар ва мижоз сервер иловалар яратилиши мумкин. Java содда, этарли даражада турлаштирилган дастурлаш тили бўлиб, унинг ёрдамида объектга йўналтирилган конструксияларни қийинчиликсиз ва мамнуният билан яратиш мумкин. Бундан ташқари Java интернет билан ҳамкор бўлиб, кўплаб Web лойиҳалар унинг улушига тўғри келади. Java appletлар тўғрисида кўпчилик эшитган бўлсада, Java ёрдамида тўла қонли иловалар, компоненталар ёки сервлетлар яратилиши мумкинлигини кўпчилик билишмайди..

Appletларни яратиш. Java бу классларга асосланган объектга йўналтирилган дастурлаш тилидир. Класс (class) бу маълумотлар ва маълумотларни ишлатувчи услубларни бир бутунга инкапсуляция қилувчи (жамловчи) объектга йўналтирилган дастурлаш тилининг асосий парадигмаси (иш кўрадиган тушунчаси) ҳисобланади. Дастурчи ўз appletини

яратаётганда янги класс ларни аниқлаши ёки мавжуд пакетларда жойлашган классларни ишлатиши мумкин. Пакет бу Java класслардан ташкил топган махсус файлдир. Дастурчи мураккаб дастурни яратаётганда ўз классларини яратади ва уларни кейинчалик ишлатиш учун пакетларга жойлаштиради. Javaда стандарт номланган пакетларга жойлаштирилган ички классларнинг кўплаб мажмуалари мавжуд.

Appletлар яратишдаги биринчи пакет Java.applet дир. Демак интерпретаторга бериладиган биринчи кўрсатма:

```
import Java.applet.*;  
кўринишида бўлади.
```

Import операторидаги * белги Java.applet пакетидаги барча класслардан фойдаланиш мумкинлигини билдиради. Лекин биринчи appletда барча классларни ишлатишга эҳтиёж йўқ. Java.applet пакетидаги Applet классини ишлатишнинг ўзи етарли. Шунинг учун юқоридаги операторни

```
import Java.applet.Applet;
```

кўринишда ишлатиш мумкин бўлади. Юқоридагиларни эътиборга олиб биринчи appletни SalomJava деб номлаймиз ва кўйидаги кўринишда ёзамиз.

```
import Java.applet.*;  
import java.awt.*;  
public class SalomIvat extends Applet  
{  
    public void paint(Graphics g)  
    {  
        g.drawString("Javadan katta Salom!", 20, 50);  
    }  
}
```

ёки import операторини ишлатмасдан

```
public class SalomJava extends Java.applet.Applet  
{  
    public void paint(java.awt.Graphics g)
```

```

{
    g.drawString("Javadan katta Salom!", 20, 50);
}
}

```

`public void paint(Graphics g)` операторида `paint()` методи матн сатрини экранда акс эттириш учун чақирилаётганини билдиради. Паинт методи 1 расмда кўрсатилган Компонент классига тегишли бўлади. Унда ишлатилаётган `g` параметр `Graphics` классининг объекти ҳисобланади ва маълумотларни график қурилмага узатиш кераклигини билдиради.

Бирор матн тахриричи ёрдамида ҳосил қилинган 1 листингдаги матнни сақлаётганимизда файл номи танлашга эътибор беришимиз керак. Ҳосил қилинган класс номи (`SalomJava`) файл номи билан устма уст тушиши керак. Акс ҳолда компилятор хатога йўл қўйилганлигини билдиради. Файл кенгайтмаси эса Java кўринишида бўлиши керак. Файлинг тўлиқ номи `SalomJava`.

2 листинг. Класс файлини ишга туширувчи HTML коддан намуна (`SalomJava.html` файл). Appletни компиляция қилиш ва ишга тушириш Appletни компиляция қилиш ва ишга туширишга тайёрлаш учун компьютерга JDK (Java Development Kit) бошқача қилиб айтганда SDK (Software Development Kit) ўрнатилган бўлиши керак. Бу дастур Java файлларни компиляция қилиш имконини беради. Бунинг учун ўрнатиладиган файл, масалан, `jdk-6u3-windows-i586-p.exe` файл ишга туширилади. `jdk-6u3-windows-i586-p.exe` файл архив файл бўлиб, ўзи очилади тегишли компоненталарни компьютерга ўрнатади. Ўрнатилиш жараёнида дастур томонидан берилган саволларни ўқиб, уларга мос тўғри жавобларни бериш талаб этилади. JDK ўрнатиб бўлингач операцион тизимнинг `AUTOEXEC.BAT` ва `CONFIG.SYS` файлларида кўйидаги ишларни амалга ошириш керак бўлади:

1. PATH ўзгарувчига JDK пакетининг `bin` каталогида жойлашган `Java`, `Javas` файлларини операцион тизим топа олиши учун йўл кўрсатилиши керак.

Масалан, AUTOEXEC.BAT да PATH=c:\JAVA\bin; C:\JAVA\jre\bin кўринишида.

2. CLASSPATH ўзгарувчига жорий яъни Java бошланғич кодлари ёзилган каталогга ва фойдаланишимиз мумкин бўлган бошқа каталогларга муурожаат қилиш имконини беришимиз керак. Масалан, D дискда Java файлларни сақлаш каталогини JavaFiles деб номлайлик. Операцион тизимга буни таништириш учун CONFIG.SYS да SET CLASSPATH=.;c:\JAVA\lib;d:\JavaFiles; ёзувни амалга оширамыз.

1 Жадвал. <APPLET> теги атрибутлари.

Атрибут. Атрибут бажарадиган вазифа align Applet ичидаги матнни ёки бошқа объектларни applet ойнасига нисбатан жойлашишини аниқлайди:

- left - барча матн ва объектлар applet ойнасининг чап томонига сурилиб ёритилади;

- center - барча матн ва объектлар applet ойнасининг чап ва унғ томонларига нисбатан ўртасига суриб ёритилади;

- right - барча матн ва объектлар applet ойнасининг ўнғ томонига сурилиб ёритилади;

- top - барча матн ва объектлар applet ойнасининг юқори томонига сурилиб ёритилади;

- middle - барча матн ва объектлар applet ойнасининг вертикал томонлари бўйича марказлаштирилади;

- bottom - барча матн ва объектлар applet ойнасининг қўйи томонига сурилиб ёритилади;

alt Ушбу атрибут ёрдамида бирорта матн берилса, берилган матн броузер Java файл билан ишлай олмаган холда ёритилади.

code Applet байт кодини ўз ичига олувчи иккилик файл номи кўрсатилади. Ўз ўзидан ушбу файл HTML файл билан бир каталогда жойлашган деб тушинилади. Файл бошқа каталогда бўлса codebase атрибути ёрдамида тўғриланади.

codebase Appletнинг асосий URL манзили (адреси) ёки applet

жойлашган каталогга йўл.

height Applet ойнасининг пикселлардаги баландлиги.

width Applet ойнасининг пикселлардаги эни.

hspase Applet ойнасидан ўнгда ва чапда қолдириладиган бушлиқ.

vspace Applet ойнасининг тепасида ва пастида қолдириладиган бушлиқ.

name Applet идентификатори. Бу ном шу HTML хужжат ичида жойлашган бошқа appletлар томонидан ёки JavaScript сценарийларда ишлатилиши мумкин.

titlr Сарлавҳа сатри.

Класслар, методлар ва конструкторлар. Хамма Java дастурлар тўртта асосий блокдан ташкил топган бўлади, булар:

Класслар, методлар, ўзгарувчилар ва пакетлар. Бошқа дастурлаш тилларида методларнинг функциялардан ёки қисмдастурлардан, маълумотларни сақловчи ўзгарувчилардан иборат эканлигини биласиз. Класснинг ўзгарувчлари одатда майдонлар деб аталади. Класслар объектга йўналтирилган програмалаш асосида ташкил қилинади. Классларда инкапсуляция орқали ўзгарувчилар майдонлар ўзаро биргаликда ишлайдиган тизимни ҳосил қилади ва булар биргаликда программада объектни ҳосил қилади. Объектлар махсус метод классни конструктор ёрдамида ҳосил қилинади. Ва ниҳоят фойдаланувчи программаси ишлаши керак бўлган ҳамма класслар тўплами жойлашган пакетлар яратилади.

Классларнинг умумий тузилишини қуйидагича ёзишимиз мумкин:

```
class class_name
{
    //классдаги ўзгарувчиларни аниқлаш
    //класс конструкторлари
    //класс методлари
}
```

Пастда келтирилган программामизда иккита класс яратилган улар:

ComplexNumber, TestComplexNumber лардир.

TestComplexNumber да ComplexNumber классиде асосида яратилган объект ишга туширилади. ComplexNumber классиде комплекс сонлар ифодаланadi. Бу классда иккита ўзгарувчи real ва image орқали комплекс сонларнинг хақиқий ва каср қийматлари топилади. Бундан ташқари иккита конструктор ва иккита метод ишлатилган .

```
class class_name
{
private double real;
private double image;
//konstruktor
public ComplexNumber(double r,double i)
{real = r;
image = i;
}
// Konstruktor
public ComplexNumber(double r)
{this(r,0.0);
}
//metod
public ComplexNumber add(ComplexNumber cn)
{return new ComplexNumber(real + cn.real,imag + cn.imag);
}
//metod
public String toString()
{String msg;
msg = "("+real+", "+imag+")";
return msg;
}}
public class TestComplexNumber
```

```

{public static void main(String [] s)
{ComplexNumber cn1=new ComplexNumber(1.0,7.0);
ComplexNumber cn2=new ComplexNumber(5.0);
ComplexNumber res=cn1.add(cn2);
System.out.println(cn1.toString()+" "+
cn2.toString()+"="+
res.toString());
try{System.in.read();}
catch(java.io.IOException e){}
}
} class class_name
{private double real;
private double image;
//konstruktor
public ComplexNumber(double r,double i)
{real = r;
image = i;
}
// Konstruktor
public ComplexNumber(double r)
{this(r,0.0);
}
//metod
public ComplexNumber add(ComplexNumber cn)
{return new ComplexNumber(real + cn.real,imag + cn.imag);
}
//metod
public String toString()
{String msg;
msg = "("+real+", "+imag+"");
}
}

```

```

return msg;
}}
public class TestComplexNumber
{public static void main(String [] s)
{ComplexNumber cn1=new ComplexNumber(1.0,7.0);
ComplexNumber cn2=new ComplexNumber(5.0);
ComplexNumber res=cn1.add(cn2);
System.out.println(cn1.toString()+" "+
cn2.toString()+"="+
res.toString());
try{System.in.read();}
catch(java.io.IOException e){}
} }

```

Классларни яратишда класс иерархияси, хосса ва методлардан насл олиш муҳим рол ўйнайди. Насл олиш орқали олдин яратилиб қўйилган класслардан оталик класс сифатида ишлатиш имконини беради, бунда янги класс яратилади ва оталик класснинг хосса ва методларидан фойдаланиш имконини беради. Мурожат қилишни ва насл олишни бошқариш Класс коди жорий классдаги барча майдон ва методларга мурожат қила оладими деган савол бўлиши мумкин. Классларга бошқа класслардан мурожат қилишни бошқариш ва насл олаётганда унга мурожатни бошқариш учта модификатор орқали ифодалашимиз мумкин булар:

Очиқ(public) - хоҳлаган жойдан мурожат қилиш имконияти бор.

Ёпиқ(private) – фақатгина шу классга тегишли бўлган класслардан.

Химояланган(protected) – фақатгина шу пакетга тегишли барча методлар ва қисмкласслардан мурожат қилиш имконини беради.

Агар модификатор кўрсатилмаган бўлса мантиқий тарзда олади ва пакетли мурожат дейилади(private to package). Ва шу пакетнинг ичидангина мурожат қилиш имконини беради.

Юқорида ҳар хил пакетларга тегишли бўлган класслардан фойдаландик

, масалан Java.applet.Applet. Хақиқатдан савол туғилади “Пакет нима ўзи?”. Бир қанча оталик классларининг тўпламига пакетлар дейилади. Пакетлар ўз исмига эга бўлади, масалан туракс. Хар бир пакетга кирувчи файл, класс ва интерфейслар туракс пакетига тегишли бўлиши учун package орқали аниқланган бўлиши керак яъни:

```
package туракс;
```

package сўзи классларда ва интерфейсларда аниқланганда одатда уларнинг бошида ёзилиши шарт ҳисобланади. Фақат файл бошида бир мартагина ёзиб қўйсақ этарли ҳисобланади. Биринчи appletингиз каби бу ҳам сизга қийинчилик тўғдирмайди. Хўш бунинг учун нима қилишимиз керак бўлади, буинг учун Фраме классини ишлатишимиз керак бўлади. Фраме классини айнан windowс – дастуримизни тузишимиз учун керак бўлган классдир. Бунда Main() методиинг ёки шу каби бошқа методлардан фойдаланган ҳолда фраме классининг объектини ҳосил қилиб оламиз.

Конструкторда frame классининг параметрларининг қийматларин келтиришимиз керак бўлади. Бундан ташқари ойна яратишимиз учун унинг размерларини беришимиз керак бўлади , ундан кейин у бизга керакли бўлган ойнани экранга ҳосил қилиб беради. Экранга чиқариш show() методи орқали амалга оширилади.

Сизнинг биринчи дастурингиз. Файл HelloWorld.java.

```
import java.awt.*;  
public class HelloWin  
{public static void main(String[] args)  
{Frame frame = new Frame("HelloWin!");  
frame.setSize(250,100);  
frame.show();  
}}
```

Энди эса MS-DOS нинг командалар сатрига қуйидагини киритамиз:

```
Javac HelloWin.java
```

бу самарали тугаллангач

Java HelloWorld ни командалар сатрига киритамиз, ундан кейин экранимизда қуйдаги ойна ҳосил бўлади.

Java дастурлаш тилида маълумотларнинг асосан саккиз хил тури аниқланган. Хар бир ажратилган тур хотирадан маълум бир жойни эгаллайди. Маълумотларнинг тури Хотирадан эгаллаган жойи Автоматик тарзда оладиган қиймати

Boolean	8	False
byte	8	0
char	16	'x0'
short	16	0
int	32	0
long	64	0
float	32	0.0F
Double	64	0.0D

Ўзгарувчилар. Ўзгарувчилар – бу қийматни сақлаш учун ажратилган хотирадаги катакча. Бундай қийматлар сон, қатор, сана ва шу каби бошқалар бўлиши мумкин. Програма кодида ишлатилган хар бир ўзгарувчи аниқланган бўлиши керак. Қайта юкланиш вақтида маълумотларнинг қандай тури ўзгарувчилар аниқлайди.

Ўзгарувчиларни аниқлаш қуйдагича бўлади:

<modifier> <type> <variable name> [<,variable name>]

масалан double типдаги pi ўзгарувчисининг қийматини аниқлаш коди келтирилган, i ва j эса int туридаги ўзгарувчидир.

Модификаторлар. Ўзгарувчиларни аниқлишда модификаторлардан фойдиланишимиз мумкин, яъни ўзгарувчиларга мурожатни бошқариш учун. Модификаторларнинг 3 тури мавжуддир, булар қуйдагиларадир: public, private, protected.

Ўзгарувчиларни аниқлашда public модификаторини ишлатсак, бунда бу ўзгарувчини бошқа класс ёки пограмманинг хоҳлаган ички кодидан мурожат қилиш ва ўзгартириш имконига эга бўламиз. Масалан харидорнинг

исми ва фамилиясини аниқлаш учун public модификаторидан фойдаланамиз ва программамизнинг хоҳлаган жойидан мурожат қилиш имконига эга бўламиз.

```
class Employee
{
    public String lastName;
    public String firstName;
}
public class Test
{
    public static void main(String [] args)
    {
        Employee empl = new Employee();
        empl.firstName = "Subxon";
        empl.lastName = "Ziyadullayev";
        System.out.println(empl.firstName + " " + empl.lastName);
    }
}
```

Private модификатори эса фақат шу класимизнинг ичидагина ўзгартирувчимизни ишлатишимиз мумкин бўлиб қолади холос. Масалан юқоридаги программда public модификаторининг ўрнига private модификаторини ишлацак программамиз хатолик беради. private модифкаторига мисол кўрамиз.

```
class Employee
{
    public String lastName;
    public String firstName;
}
public class Test
{
```

```

public static void main(String [] args)
{
Employee empl = new Employee();
empl.firstName = "Subxon";
empl.lastName = "Ziyadullayev";
System.out.println(empl.firstName + " " + empl.lastName);
}
}
class Employee
{
private String lastName;
private String firstName;
public void setFirstName(String firstName)
{
this.firstName = firstName;
}
public void getFirstName(String firstName)
{
return firstName;
}
public void setLastName(String latsName)
{
this.lastName = lastName;
}
public void getLastName(String latsName)
{
return lastName;
}
}
public class Test

```

```

{
public static void main(String [] args)
{
Employee empl = new Employee();
empl.setFirstName ("Subxon");
empl.setLastName = ("Ziyadullayev");
System.out.println(empl.firstName + " " + empl.lastName);
}
}

```

Константаларни номлаш. Константалар деб шундай қийматга айтиладики бу қиймат программа ишлаши давомида ўзгармайди. Масалан: 12, 45, ёки “Sure thing”.

Java да коментлариларнинг учта тури мавжуддир.

Комментарининг биринчи тури бу иккитали слешдир (//). Бу турдаги коментарини ёзиш қуйидагича бўлади:

```
String firstName, lastName;
```

```
// Бу ўзгарувчилар харидорнинг исм ва фамилиясини аниқлаш
учун ишлатилган.
```

```
Иккинчи тури /* символлари билан бошланади ва */ символлари
билан тугалланади.
```

Масалан:

```
String firstName, lastName;
```

```
/* Бу ўзгарувчилар харидорнинг исм ва фамилиясини аниқлаш
учун ишлатилган. */
```

```
Комментарийнинг учинчи тури /** символлари билан бошланиб
*/ символлари билан тулланади.
```

Java да операторлар орасини ажратиш учун нуқтали вергул қўйилади (;). Масалан :

```
int x,y;
```

```
x = 3;
```

```
y = 4;
```

```
int z = x + y;
```

Java да код фигурали кавслардан иборат блоклар орасига ёзилган бўлади. Барча ўзгарувчилар кавслар ичида эълон қилинади. Масалан:

```
{  
int i = 10;  
double i = 10.5;  
}
```

Калит сўзлар.

```
abstract continue for new switch boolean          default          goto null  
synchronized  
break      do      if package this byte      double      implements private  
threadsafe byvalue      else      import protected throw case      extends  
instanceof public transient catch false int return true char final interface short try  
class finally long static void const float native super while
```

Массивларни аниқлаш учун квадрат кавслар ишлатилади яъни массив номининг эки турининг чап тамонига қўйилади. Масалан:

```
int number[];
```

```
String [] myString;
```

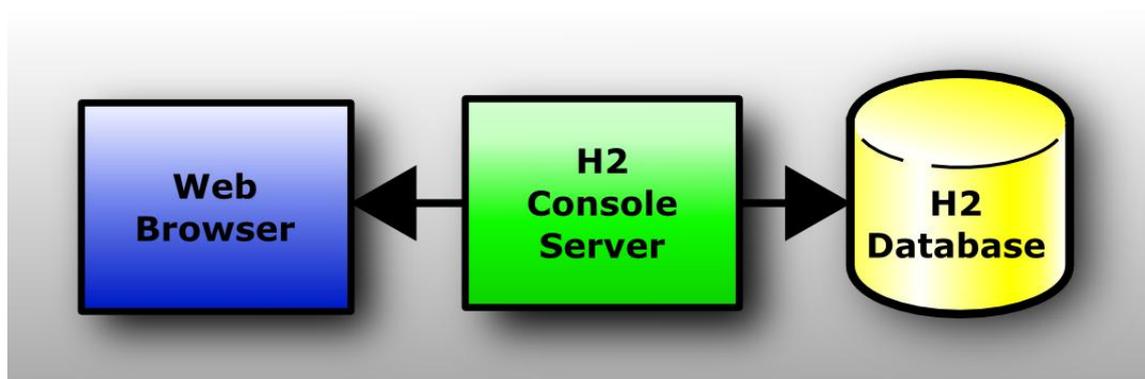
2.2. H2 маълумотлар базаси



Маълумотлар базасини бошқариш тизимига боғлиқ бўлган дастур ҳисобланади. У java тилида яратилган. Серверда ва java дастурларида мустақил равишда ишлай олади (яъни серверга алоқаси бўлмаган ҳолда). Дастурнинг ҳажми тахминан 1.5 МБ ни ташкил қилади. H2 очик кодли дастурлардан ҳисобланади. SQL стандартларига асосланган. Ҳам Тезкор хотирада ҳам доимий хотирада жадвал(table) лар яратиш имкони мавжуд. Жадваллар доимий ва вақтинчалик бўлиши мумкин. Маълумотлар базасидаги паролларни шифрлашда юқори

даражадаги шифрлаш алгоритмлари ишлатилади, масалан: ША-256, Тинй шифрлаш алгоритми, ХТЕА (Ехтендед ТЕА – узайтирилган Тiny алгоритми). Н2 маълумотлар базаси 2004 йил май ойида ривожланишни бошлаган, лекин 2005-йил декабр ойида нашрга чиқарилган. Маълумотлар базаси механизми Томас Мюллер (Thomas Mueller) томонидан яратилган. У шунингдек Java маълумотлар базаси механизми бўлган Гиперсоник (Hypersonic) SQL ни ҳам яратган. Кейинчалик ишлатилмай қўйилгач у дастурни қайта ишлаб чиқди ва кейинги версиясини ишлаб чиқди. Кейинги версиясини номи Hypersonic 2 бўлган яъни Н2.

Н2 console мухитида ишга туширилиши. Н2 сонсоле дастури маълумотлар базасига браузер орқали фойдаланиш имконини беради. У Н2 маълумотлар базаси бўлсин ёки бошқа java маълумотлар базасини бошқариш механизми (JDBC) бўлсин.



2.1 – расм. Н2 маълумотлар базаси структураси.

Н2 console ҳам клиентда ҳам серверда ишга тушириш имконини беради. Платформаларга боғлиқ равишда уни ишга туширишнинг бир неча йўллари мавжуд(2.1 жадвал):

2.1 жадвал

ОТ	Ишга тушириш
----	--------------

Windows	[Start], [All programs], [H2] ва [H2 Console (Command line)] Агар ҳеч қандай ойна очилмаса, демак Java дастури ўрнатилмаган. Яна бир бошқа усули браузер орқали http://localhost:8082 ни киритиш керак.
Windows	Браузерни очиб H2/bin url га кириб, H2.bat га икки марта босиш кифоя. Console ойнаси кўринади. Агар бирор муаммо дуч келса, ойна бу тўғрисида хабар беради.
Бошқа	H2*.jar файли устига икки марта босинг
Ўқи	Сонсоле ойнасини очиб қуйидагини киритинг: <pre>java -jar H2*.jar</pre>

FireWall. FireWall бу – тизим хавфсизлигини таъминловчи восита ҳисобланади. Агар сервер ни ишга туширсангиз *security warning* юз беради (агар сизда firewall дастури ўрнатилган бўлса). Агар бошқа компьютерлар сизни компютерингиздаги маълумотлардан фойдаланишларини хохламасангиз, firewall буларни блокка қўйишига имкон яратиб беришингиз керак. Локал компьютерларда боғланишлар ишлайверади. Агар фақат айнан шу компьютерда маълумотлар базасига боғланиш бўлишини хохласангиз, унда firewall дастурини масофадан туриб босқаришга рухсат беришингиз керак.

Кичик firewall дастури стандарт ҳолатда серверга ўрнатилган бўлади.

JDBC дан фойдаланган ҳолда маълумотлар базасига боғланиш.

Маълумотлар базасига боғланиш учун, java дастурига маълумотлар базасини юклаб олиш керак. Ва ундан кейин боғланиш мумкин. Бу боғланишни амалга ошириш йўлларида бири қуйидагича(2.2 – расм):

```
import java.sql.*;
public class Test {
    public static void main(String[] a)
```

```

throws Exception {
Class.forName("org.h2.Driver");
Connection conn = DriverManager.
    getConnection("jdbc:h2:~/test", "sa", "");
// add application code here
conn.close();
}
}

```

2.2 – расм. Java мухитида H2 билан боғланиш.

Бу код ёрдамида биринчи драйверни юклаб олади ва боғланишни амалга оширади. “*org.h2.Driver*” – драйвер номи. Маълумотлар базаси URL си доим *jdbc:H2:* билан бошланади. *Sa* – фойдаланувчи номи. (sa – System Administrator (тизим администратори)).

2.3. SQL тили

Маълумотлар базаси дунёси тобора ягона бўлиб бормоқда. Бу жараён хар хил компьютер мухитларида фаолият кўрсатувчи ахборот тизимларини яратишда қўлланувчи ягона стандарт тил яратишни талаб қилди. Стандарт тил бир командалар тўпламини билган фойдаланувчиларга уларни шахсий компьютер тармоқ ишчи стантсияси ёки катта ЭХМ да ишлашларидан катъий назар маълумотни яратиш, излаш ва узатишга имкон беради.

SQL (Structured Query Language, одатда "sikvel" дейилади) маъноси *Таркибланган сўровлар тили*. Бу релятсион маълумотлар базаларида ишлашга имкон берадиган тилдир. Бу тил ифодаларининг хусусияти шундан иборатки улар маълумотларни қайта ишлаш протседураларига эмас натижаларига йўналтирилганДир. SQL ўзи маълумотлар қаерда жойлашгани, қандай индекслар ва ҳатто амалларнинг энг эффектив кетма кетлигини қўллаш кераклигини аниқлайди; бу деталларни маълумотлар базасига сўровларда

кўрсатиш керак эмас.

SQL тилининг ўзи IBM компаниясида МББТ DB2 яратиш жараёнида ишлаб чиқилган ва кенг кўламда RISS протсессорли машитналарда UNIC тизимлар асосида, ҳамда мейнфреймларда, суперкомпьютерлар асосида қурилган катта ҳисоблаш тизимларида қўлланилган.

Шу билан бирга мустақил бўлмасдан PL/SQL, ва Transact-SQL каби ички дастурлаш тилларига инкапсулятсия қилинади. 1986 йилда, ANSI (American National Standart Institute) SQL тилининг расмий стандартини ишлаб чиқди, 1992 йил бу стандарт кенгайтирилди. Бутун тил 30 га яқин операторларга эга бўлиб, баъзи версияларида сал кўпроқ, баъзиларида сал камроқ. Ҳар қандай МБ ҳар хил объектларга эга, Яъни жадваллар, протседуралар, функциялар, тасаввурлар, кетма кетликлар ва хоказо.

"Клиент-Сервер" технологиясига кўра, фойдаланувчи ЭХМ (Клиент) лар сўровлари махсус маълумотлар серверларида (Сервер) қайта ишланади, фойдаланувчи ЭХМ ларга фақат сўровни қайта ишлаш натижалари қайтарилади.

Табиийки Сервер билан мулоқот қилиш учун ягона тил керак ва бундай тил сифатида SQL танланди. Шунинг учун ҳамма замонавий релятсион МББТ версиялари (DB2, Oracle, Ingres, Информих, Сйбасе, Прогресс, Рдб) ва хаттоки норелятсион МББТ версиялари (масалан, Адабас) "Клиент_Сервер" технологияси ва SQL тилидан фойдаланадилар.

SQL тилида Маълумотларни жадвал кўринишда тасвирлашга йўналтирилган амALLар концепцияси кўп бўлмаган (30 дан кам) ифодалардан иборат компакт тил яратишга имкон берди.

Икки хил SQL мавжуд: **Интерактив** ва **Жойлаштирилган**. Кўп холларда иккала форма бир хил ишлайди, лекин икки хил фойдаланилади:

Интерактив SQL маълумотлар базаси ўзида фаолият кўрсатади ва буюртмачи фойдаланиши учун чиқиш хосил қилиш учун ишлатилади. SQL бу формасида, сиз команда киритсангиз, у даров бажарилади, ва сиз дархол

натижани(агар у мавжуд бўлса) кўришингиз мумкин.

Жойлаштирилган SQL бошқа тилда яратилган дастурга жойлаштирилган SQL командалардан иборат.

SQL Интерактив, ва жойлаштирилган формаларида кўп сонли гурухлар ёки суббўлимлар мавжуд. Улар ANSI томонидан эътиборга олинган ва концептуал даражада фойдали, лекин кўпчилик SQL дастурлар уларни алохида қайта ишламайди, шунинг учун улар аслида SQL командаларининг функционал категорияларидир.

DDL (*Маълумотларни Таърифлаш Тили*) - ANSI да Схемани таърифлаш тили, объектларни(жадваллар, индекслар, тасаввурлар ва хоказо) яратувчи командалардан иборат.

DML (*Маълумотларни Ўзгартириш Тили*) - бу ихтиёрий дақиқада жадвалларда қандай қийматлар сақланишини аниқловчи командалар мажмуасидир.

DCD (*Маълумотларни Бошқариш Тили*) фойдаланувчига маълум объектлар устида маълум таъсир ўтказишга рухсат бериш ёки бермасликни аниқловчи воситалардан иборат.

SQL Стандарти **ANSI** томонидан аниқланган ва хозирда **ISO** томонидан қабул қилинган. Лекин коммерциал маълумотлар базалари дастурлари ANSI ни огохлантирмасдан SQL ни кенгайтирадилар, яъни фойдали ҳисоблаган ҳар хил хоссалар кўшадилар.

SQL тилида маълумотлар турлари. SQL тилида қуйидаги асосий маълумотлар турлари ишлатилиб, уларнинг форматлари ҳар хил МББТ лар учун фарқ қилиши мумкин:

INTEGER - бутун сон (одатда 10 тагача қийматли рақам ва ишора).

SMALLINT - "қисқа бутун" (одатда 5 тагача қийматли рақам ва ишора).

- DECIMAL(p,q) - ўнли сон, p рақам ва ишорадан иборат ($0 < p < 16$). Ўнли нуқтадан сўнг рақамлар сони қорқали берилади ($q < p$, агар $q = 0$ бўлса, ташлаб юборилиши мумкин).
- FLOAT - ҳақиқий сон 15 та қийматли рақам ва бутун даражадан иборат. Даража МББТ типи билан аниқланади (масалан, 75 ёки 307).
- CHAR(n) - узунлиги ўзгармас, n га тенг бўлган символли қатор ($0 < n < 256$).
- VARCHAR(n) - узунлиги ўзгарувчи, n символдан ошмаган символли қатор ($n > 0$ ва ҳар хил МББТ ларда ҳар хил лекин 4096 дан кам эмас).
- DATE - махсус команда орқали аниқланувчи форматдаги сана; сана майдонлари бизнинг эрамиздан олдин бир неча мингйилликлардан бошланувчи ва бизнинг эрамиз бешинчи-ўнинчи мингйиллиги билан чекланган ҳақиқий саналарни ўз ичига олиши мумкин.
- TIME - махсус команда орқали аниқланувчи форматдаги вақт (кўзда тутилган бўйича hh.mm.ss).
- DATETIME - сана ва вақт комбинатсияси.
- MONEY - махсус команда орқали аниқланувчи форматдаги пул.

Жадвалларни яратиш. Жадваллар CREATE TABLE командаси билан яратилади. Бу команда қаторларсиз бўш жадвал яратади. CREATE TABLE командаси жадвал номини ва жадвал ўзини маълум тартибда кўрсатилган устунлар номлари кетма – кетлиги таърифи кўринишида аниқлайди. У

маълумотлар типлари ва устунлар ўлчовини аниқлайди. Ҳар бир жадвал жуда бўлмаганда битта устунга эга бўлиши керак.

CREATE TABLE командаси синтаксиси:

CREATE TABLE <TABLE-name >

(<column name> <data type>[(<size>)],

<column name> <data type>[(<size>)], ...);

Аргумент қиймати катталиги маълумот турига боғлиқдир. Агар сиз махсус кўрсатмасангиз, тизим автоматик қийматни ўрнатади.

Бундан буён қуйида келтирилган 3 та жадвалдан иборат маълумотлар базасини кўрамиз.

Жадвал Сотувчилар (Salespeople):

Cnum	Cname	City	Comm
11	Peel	London	0.12
12	Serres	San Jose	0.13
14	Motika	London	0.11

CNum – ҳар бир сотувчи уникал номери,

CName – сотувчи номи,

City – сотувчи адреси (шаҳар),

Comm – сотувчиларнинг ўнли шаклдаги коммиссион фойдаси.

Жадвал Буюртмачилар (Customers):

Cnum	Cname	City	Rating	CNum
21	Хоффман	London	100	11
22	Гиованни	Роме	200	13
23	Лиу	СанЖосе	200	12

Cnum – ҳар бир буюртмачи уникал номери,

CName – буюртмачи номи,

City – буюртмачи адреси (шаҳар),

Rating – буюртмачининг бошқалардан устунлик даражасини кўрсатувчи

код

(рейтинг),

CNum – шу буюртмачига тайинланган сотувчи номери.

Жадвал Буюртма (Orders):

Onum	AMT	Odate	CNum	CNum
38	4723.00	1990/10/05	26	11
310	1309.95	1990/10/06	24	12

ONum – ҳар бир сотиб олиш уникал номери,

AMT – сотиб олиш суммаси қиймати,

ODate – сотиб олиш санаси,

CNum – сотиб олувчи буюртмачи номери,

CNum – сотувчининг номери.

Мисол учун сотувчилар жадвалини яратишни кўриб чиқамиз:

Жадвалларни ўчириш. Жадвални ўчириш имконига эга бўлиш учун, жадвал эгаси (Яъни яратувчиси) бўлишингиз керак. Фақат бўш жадвални ўчириш мумкин. Қаторларга эга бўлган, тўлдирилган жадвални ўчириш мумкин эмас, Яъни жадвал ўчиришдан олдин тозаланган бўлиши керак. Жадвални ўчириш командаси қуйидаги кўринишга эга:

```
DROP TABLE < TABLE name >;
```

Масалан: DROP TABLE SalePeople;

Жадвални яратилгандан сўнг ўзгартириш. Жадвални ўзгартириш учун ALTER TABLE командасидан фойдаланилади. Бу команда жадвалга

Янги устунлар кўшиш, устунларни ўчириш, устунлар катталигини ўзгартириш, ҳамда чекланишларни кўшиш ва олиб ташлаш имкониятларига эга. Бу команда ANSI стандарти қисми эмас, шунинг учун ҳар хил тизимларда ҳар хил имкониятларга эга.

Жадвалга устун кўшиш учун команданинг типик синтаксиси:

```
ALTER TABLE <TABLE name> ADD <column name>  
      <data type> <size>;
```

Масалан:

```
ALTER TABLE Salepeople ADD Phone CHAR(7);
```

Сатрларни ўчириш. Сатрларни жадвалдан DELETE командаси билан ўчириш мумкин. У алоҳида қийматларни эмас фақат сатрларни ўчиради. DELETE қуйидаги форматга эга:

```
DELETE FROM <TABLE name | view name>  
      [WHERE search-condition];
```

Масалан, Сотувчилар жадвалидаги ҳамма сатрларни ўчириш учун, қуйидаги шартни киритиш мумкин:

```
DELETE FROM Salepeople;
```

Маълум сатрларни ўчириш учун предикатдан фойдаланилади. Масалан, жадвалдан Ахелрод сотувчини ўчириш учун:

```
DELETE FROM Salepeople  
      WHERE CNum = 13;
```

Майдон қийматларини ўзгартириш. Бу ўзгартириш UPDATE командаси ёрдамида бажарилади. Бу командада UPDATE ифодасидан сўнг жадвал номи ва SET ифодасидан сўнг маълум устун учун ўзгартириш кўрсатилади. UPDATE икки форматга эга. Улардан биринчиси:

```
UPDATE <TABLE name | view name>  
      SET column = expression [, column = expression] ...  
      [WHERE search-condition]
```

бу эрда expression - бу устун | ифода | константа | ўзгарувчи.

Иккинчи вариант:

```
UPDATE <TABLE name>
```

```
    SET column = expression, ...
```

```
    [ FROM TABLE-лист ]
```

```
    [ WHERE search-condition ]
```

Масалан, ҳамма буюртмачилар бахосини 200 га ўзгартириш учун қуйидагини киритишингиз мумкин:

```
UPDATE Customer
```

```
    SET Rating = 200;
```

Маълум сатрларни ўзгартириш учун DELETE даги каби предикатдан фойдаланиш керак. Масалан Peel (CNum=11) сотувчининг ҳамма буюртмачилари учун бир хил ўзгартиришни қуйидагича киритиш мумкин:

```
UPDATE Customer
```

```
    SET Rating = 200
```

```
    WHERE CNum = 11;
```

SET вергул билан ажратилган ихтиёрий сондаги устунларга қиймат тайинлаши мумкин. Ихтиёрий жадвал сатрлари учун қиймат тайинланиши мумкин, лекин бир вақтнинг ўзида фақат битта сатрга қиймат тайинланади.

SELECT сўров оператори. SELECT оператори МБ жадвалларидан натижавий тўплам олиш учун мўлжалланган ифодадир. Биз SELECT оператори ёрдамида сўров берамиз, у бўлса маълумотлар натижавий тўпланини қайтаради. Бу маълумотлар жадвал шаклида қайтарилади. Бу жадвал Кейинги SELECT оператори томонидан қайта ишланиши мумкин ва хоказо.

Оператор SQL92 стандартига кўра қуйидаги кўринишга эга:

```
SELECT -- ALL ----- схема , устун ----
```

```
    -- DISTINCT -- ---- * -----
```

FROM -- схема , Жадвал .. -----

WHERE -- излаш шарти -----

GROUP BY -- схема , устун -----

HAVING -- излаш шарти -----

ORDER BY – тартиблаш спетсификатори -----

Биринчи қоида, SELECT ифодаси ўз ичига албатта FROM ифодасини олиши керак. Қолган ифодалар керак бўлса ишлатилади.

SELECT ифодасидан сўнг сўровда қайтарилувчи устунлар рўйхати ёзилади.

FROM ифодасидан сўнг сўровни бажариш учун жадваллар номи ёзилади.

WHERE ифодасидан сўнг агар маълум сатрларни қайтариш лозим бўлса, излаш шарти ёзилади.

GROUP BY ифодаси гурухларга ажратилган натижавий сўров яратишга имкон беради.

HAVING ифодасидан гурухларни қайтариш шарти ёзилади ва GROUP BY билан бирга ишлатилади.

ORDER BY ифодаси маълумотлар натижавий тўпламини тартиблаш йўналишини аниқлайди.

OFFICES жадвалидаги ҳамма ёзувларни қайтарувчи содда сўров кўрамиз.

SELECT * FROM OFFICES

SELECT ёрдамида маълумотларни танлаш. SELECT оператори албатта "қайтарилувчи устунлар рўйхати " ни ўз ичига олиши керак, Яъни:

Ёзувларни тартиблаш, ORDER BY жумласи. Олдин кўрилган сўровларда натижалар ихтиёрий тартибда олинган эди. Агар ўқувчилар рўйхатини алфавит тартибда ёки товарлар нархини камайиш тартибда

чиқариш зарур бўлсачи? Бунинг учун SELECT оператори таркибида ORDER BY ифодаси кўзда тутилган. Унинг синтаксиси:

```
----- ORDER BY – устун номи ----- .  
-- устун тартиб рақами --- ----- ASS -----  
----- DESS -----  
----- , -----
```

Аввал қуйидаги мисолни кўрамиз: Ҳар бир оффис учун сотувлар ҳақиқий хажмларини REGIONлар номлари, ҳар бир REGIONда эса шаҳарлар номлари бўйича алфавит тартибида кўрсатиш.

```
SELECT CITY, REGION, SALES  
FROM OFFICES  
ORDER BY REGION, CITY
```

ORDER BY ифодасидан KEYин келган устун АСОСИЙ калитдир, ундан Кейинги устунлар иккинчи даражали калитлардир. Ёзувларни ўсиш ҳамда камайиш бўйича тартибланиш мумкин.

Масалан: Сотувлари ҳақиқий хажмлари камайиш тартибида оффислар рўйхатини чиқариш.

```
SELECT CITY, REGION, SALES  
FROM OFFICES  
ORDER BY SALES DESS
```

Сотувлар хажмларини DESS предикатини қўллаб камайиш тартибида чиқарамиз. Ўсиш тартибида чиқариш учун ASS предикати қўлланади. Бу предикат кўзда тутилган бўлиб, уни кўрсатиш шаррт эмас. Агар устун ҳисобланувчи бўлиб, номга эга бўлмаса унинг тартиб номерини кўрсатиш мумкин!

Масалан, қуйидагича: Сотувлар ҳақиқий ва режадаги хажмлари айирмаси камайиш тартибида оффислар рўйхатини чиқаринг.

```
SELECT CITY, REGION, (SALES - TARGET)
```

FROM OFFICES

ORDER BY 3 DESS

Шу каби ORDER BY ифодасида устунлар номлари ва номерлари, ҳамда DESS, ASS предикатларини қўллаб мураккаб тартиблаш шартларини ҳосил қилиш мумкин.

Масалан: REGIONлар номлари, ҳар бир REGIONда сотувлар ҳақиқий ва режадаги хажмлари айирмаси камайиш тартибида оффислар рўйхатини чиқаринг.

SELECT CITY, REGION, (SALES - TARGET)

FROM OFFICES

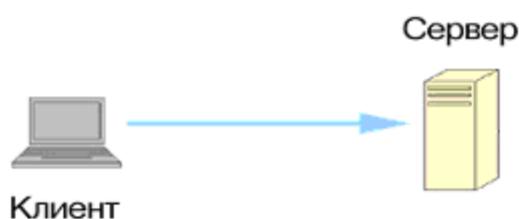
ORDER BY REGION ASS, 3 DESS

Фойдаланувчилар. SQL мухитида ҳар бир фойдаланувчи махсус идентификатсион ном, муружжат идентификаторига (ID) эга. Маълумотлар базасига юборилган команда маълум фойдаланувчи билан ёки бошқача айтганда махсус муружжат идентификатори билан боғланади. SQL маълумотлар базасида ID рухсат – бу фойдаланувчи номи ва SQL команда билан боғланган муружжат идентификаторига илова қилувчи махсус калит сўз USER дан фойдаланиши мумкин.

Регистратсия бу компьютер тизимида кириш ҳуқуқини олиш учун фойдаланувчи бажариши керак бўлган процедурадир. Бу процедура фойдаланувчи билан қайси муружжат ID си боғланишини аниқлайди. Одатда ҳар бир маълумотлар базасидан фойдаланувчи ўзининг ID сига эга бўлиши керак ва регистратсия жараёнида ҳақиқий фойдаланувчига айланади. Лекин кўп масалаларга эга фойдаланувчилар бир неча муружжат ID лари билан регистратсиядан ўтишлари, ёки бир неча фойдаланувчи битта муружжат ID сидан фойдаланишлари мумкин.

Клиент/сервер архитектура. Содда холда клиент/сервер архитектура амалий дастурдаги қайта ишлашни икки ёки ундан кўп

мантикий қисмларга ажратишга асосланган. Маълумотлар базаси қандайдир амалий дастур томонидан фойдаланиш учун яратилгандир. Соддалаштириб айтиш мумкинки маълумотлар базаси клиент/сервер архитектурасининг бир қисмини ташкил қилади. Маълумотлар базаси «сервер», ундан фойдаланувчи ҳар қандай амалий дастур «клиент». Одатда клиент ва сервер ҳар хил машиналарда жойлашган; кўп ҳолларда клиент амалий дастури маълумотлар базасига дўстона интерфейсдир. Қуйидаги график шаклда клиент/сервер содда тизими тасвири берилган.



2.3 - расм. Клиент/сервер архитектураси

Маълумотлар базаси билан ишлайдиган амалий дастурлар яратилганда авваламбор клиентни маълумотлар базаси билан боғлаш имкониятига эга бўлиш керак. Маълумотлар базалари яратувчи дастурчилардан конкрет тилга мўлжалланган, API ёрдамида боғланиш асосий механизмларни беркитадилар. Маълумотлар базаси билан ишловчи дастур яратганингизда сизни сўровларингизни тармоқ орқали маълумотлар базаси серверига узатиловчи ТСП/ИП пакетларига трансляция қилади.

Маълумотлар базасига мурожаат API ларининг ташқи кўриниши ҳар хил ва дастурлаш тилларига, кўп ҳолларда маълумотлар базасининг ўзига боғлиқ. MySQL учун API лар MSSQL билан ўхшаш қилиб яратилгани учун, биз кўрадиган API лар орасидаги фарқ минималдир.

Уч бўгинли архитектура. Шу пайтгача биз WWW ва бизнес амалий дастурлари билан ишлаш энг содда архитектураси клиент/сервер

архитектурасини муҳокама қилдик. Лекин бу архитектурани Амалий дастурлар ривожланиши билан такомиллаштириш анча мураккабдир. Бу архитектурада объектга-йўналтирилган дастурлаш имкониятларидан фойдаланиш ҳам қийин. Биринчи муаммо «нозик клиентлар» ҳақидаги бахсларда ўз аксини топди. Нозик клиентларга бўлган талаб, клиентга узатилаётган маълумотлар ўсиб боиш тендетсиясидан келиб чиқди. Бу муаммо PowerБуилдер ва ВисуалБасис ларда кўринди. Улар базадан маълумотларни GUI га олади ва бу маълумотлар устидаги ҳамма амALLарни GUI да бажаради.

Фойдаланувчи интерфейсини база ядросига боғлаб қўйиш фойдаланувчилар сони ва маълумотлар хажми ошиши билан ўзгартириш ва масштаблаш қийин бўлган дастурлар яратилишига олиб келади. Агар сизда фойдаланувчи интерфейси яратиш тажрибаси бўлса, фойдаланувчи хошишига қараб интерфейсни қайта ишлаб чиқиш муаммосига дуч келгансиз. Бундай қайта ишлашни камайтириш йўли GUI учун фақат битта вазифа – фойдаланувчи интерфейси вазифасини қолдириш керак. Фойдаланувчи бундай интерфейси чиндан ҳам нозик клиентдир.

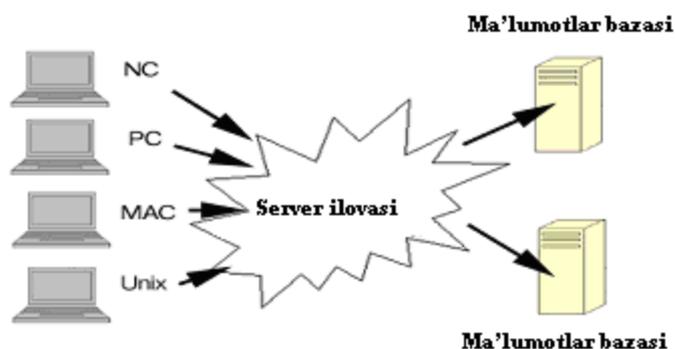
Нозик клиентлар – бугунда ягона йўналиш эмас. Бошқа йўналиш-коддан қайта фойдаланиш. Хар хил амалий дастурлар учун код бизнес логика деб аталган қайта ишлашга йўналтирилади. Агар бизнес логика фойдаланувчи интерфейсида жойлашган бўлса, коддан қайта фойдаланишни таъминлаш қийин бўлади. Бу муаммони хал қилиш йўли Амалий дастурни икки қисмга эмас уч қисмга ажратишдир. Бундай архитектура уч бўғинли дейилади.

Клиентдаги фойдаланувчи интерфейс ҳақида гапирганимизда, мантиқий фарқни назарда тутамиз. Нозик клиент бир тури «Ўта нозик клиент», бўлиб кўпчилик Web-саҳифа деб қабул қилади. Web-саҳифа динамик тарзда Web-серверда яратилиши мумкин. Бу ҳолда клиент ишининг кўп қисми серверда HTML-саҳифаларни динамик генератсия қилиш шаклида бажарилади.

2.2 - расмда кўрсатилган икки бўғинли архитектурани 2.3 - расмда кўрсатилган уч бўғинли архитектура билан солиштиринг. Биз фойдаланувчи интерфейс ива маълумотлар базаси орасида кўшимча қатлам жойлаштирдик. Бу янги қатлам амалий дастурлар сервери ўзида бирор соҳа учун умумий бўлган Амалий дастур иш мантиғи – бизнес мантиқни олади. Клиент ўрта ярус объектларини кўриш воситаси, маълумотлар базаси бўлса шу объектлар омборига айланади.

Сизни энг катта ютуғингиз, - фойдаланувчи интерфейси билан маълумотлар базасини ажратилишидир. Энди сизга маълумотлар базаси ҳақидаги маълумотни GUI га киритишга тўғри келмайди. Балким, маълумотлар базаси билан қандай ишлаш ҳақидаги ҳамма маълумотлар, ўрта ярусда жойлашиши мумкин.

Амалий дастурлар серверининг икки асосий вазифаси – маълумотлар базасига уланишларни изолятсия қилиш ва бизнес мантиқ учун марказлашган омборни таъминлаш. Фойдаланувчи интерфейси фақат маълумотларни киритиш ва акслантириш билан шуғулланади, маълумотлар базаси ядроси бўлса фақат маълумотлар базаси муаммолари билан шуғулланади. Маълумотларни қайта ишлашни марказлаштириш Амалий дастурлар серверининг битта дастурини ҳар хил фойдаланувчи интерфейслари ишлатиши мумкин ва ҳар гал янги амалий дастур яратилганда маълумотларни қайта ишлаш қоидаларини ёзиш керак бўлмай қолади.



2.4 - расм. Уч бўғинли архитектура

Объектга-йўналтирилган мурожаат ва ODBC. Агар сиз MySQL учун яратилган дастурни бошқа МББТ га кўчирмоқчи бўлсангиз ўз кодингизни шу ядро API сидан фойдаланадиган қилиб қайта ёзишингиз керак.

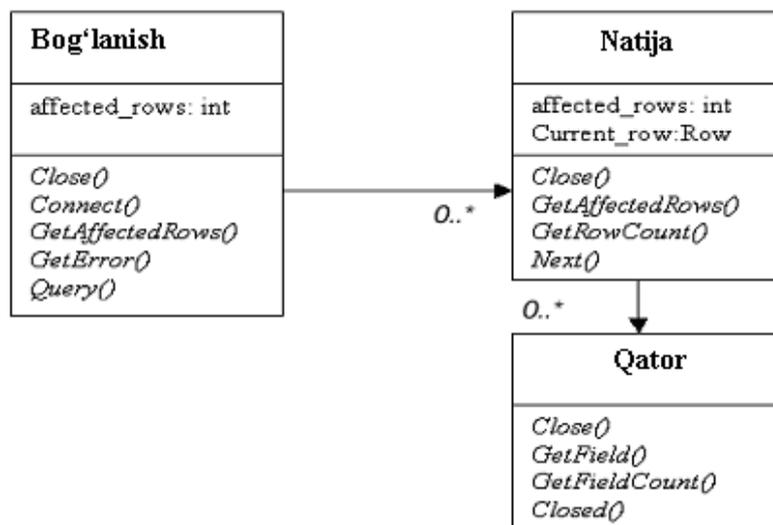
Лекин дастурчилар бошқа маълумотлар базасига кўчириш муаммосидан асосан халос бўлганлар. Уларда ягона API, Open DataBase Connectivity API (ODBC), ҳамма SQL-маълумотлар базаларига унификатсия қилинган интерфейс мавжуд.

ODBC ҳамма маълумотлар базаларига ягона интерфейс бўлгани учун, MySQL ва бошқа МББТ лар билан ишловчи дастурлар яратиш учун уни ўрганиб чиқиш этарли. Агар сиз керакли тарзда ODBC дан фойдалансангиз, сиз яратган дастурлар ихтиёрий МББТ билан ишлай олади.

ODBC ҳақида тушунча. Ҳамма API лар каби ODBC биргаликда маълум функциялар тўпламини таъминловчи синфлар ва интерфейслар тўпламидир. ODBC холида бу функциялар маълумотлар базасига мурожаатни таъминлайди. ODBC API ни ташкил қилувчи синфлар ва интерфейслар ихтиёрий турдаги маълумотлар мурожаат қилишдаги умумий тушунчалар абстрактсиясидир.

Масалан, Connection маълумотлар базаси билан боғланишни тасвирловчи интерфейсдир. Шунга ўхшаб ResultSet SQL SELECT командаси қайтарувчи натижавий тўпламни тасвирлайди. Табиийки маълумотлар базасига мурожаат конкрет деталлари унинг яратувчисига боғлиқ. ODBC бу деталлар билан ишламайди.

ODBC синфларини маълумотлар базасини дастурлашга объектга-йўналтирилган усуллар нуқтаи назаридан кўриб чиқамиз.



2.5 - расм. Маълумотлар базасига объектга-йўналтирилган мурожаат библиотекаси.

МББТ билан ишлашни учта асосий тушунча тасвирлайди: уланиш, натижавий тўплам ва натижавий тўплам сатрлари. 2.4 - расм бу объектларни UML-диаграммада кўрсатади.

UML - бу Янги Унификатсияланган моделлаштириш тили бўлиб, Гради Буч, Айвар Якобсон ва Джеймс Рамбо (Grady Booch, Ivar Jacobson, James Rumbaugh) томонидан объектга-йўналтирилган лойиҳалаш ва тахлилни хужжатлаш янги стандарти сифатида таклиф қилинган.

Маълумотлар базасига уланиш. Ихтиёрий мухитда маълумотлар базасига мурожаат уланишдан бошланади. Бизнинг объектга-йўналтирилган библиотекамизни яратиш Connection объектини яратишдан бошланади. Объект Connection сервер билан боғланишни ўрнатиш, зарур маълумотлар базасини танлаш, сўровларни узатиш ва натижа олишни билиши керак.

Connection синфи усуллари ҳамма МББТ лар учун бир хилдир. Лекин синф ичида, компилятсия қилинаётган библиотека учун хос бўлган ёпиқ аъзолар беркитилгандир. Боғланиш ўрнатишда маълумотлар билан боғланишни таъминлайдиган синф аъзолари фаркли бўлиб қолади.

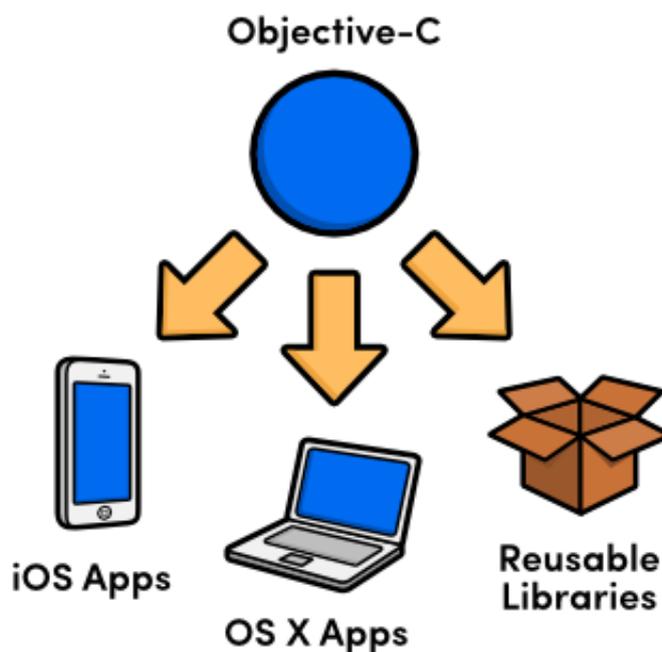
Маълумотлар базаси билан уланишни ўрнатиш. Бу API ёрдамида яратилган ҳамма амалий дастурларга маълумотлар базасига уланиш учун Connection синфи нусхасини унинг конструкторларидан бири ёрдамида яратиш керак бўлади. У каби узилиш учун Амалий дастур Connection нусхасини ўчириши керак. У тўғридан тўғри close() ва CONNECT() усуллариغا мурожаат қилиб Connection нусхасини қайтадан ишлатиши мумкин.

Маълумотлар базасидан узилиш. Connection яна бир мантиқий функцияси маълумотлар базаси билан алоқани узиш ва дастурдан беркитилган ресурсларни озод қилишдир. Бу функцияни Close () усули амалга оширади.

Маълумотлар базасига мурожаатларни бажариш. Боғланишни очиш ва ёпиш одатда маълумотлар базасига командалар юборилади. Connection синфи аргумент сифатида SQL команда олувчи Query() усули ёрдамида бажаради. Агар команда сўров бўлса 2-3. расмда кўрсатилган объект моделидан Result синфи нусхасини қайтаради. Агар команда маълумотларни янгилаётган бўлса, усул NULL қайтаради ва affected_rows қийматини ўзгартирилган сатрлар сонига тенг қилади.

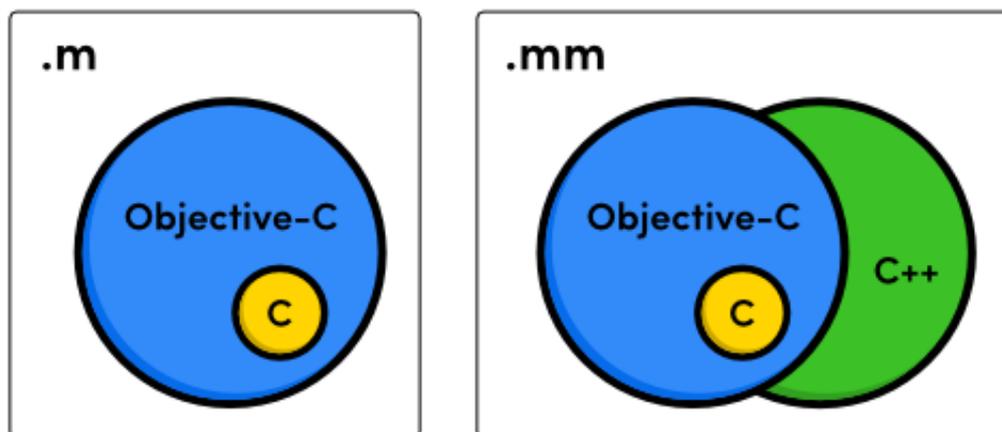
2.4. Objective C дастурлаш тили.

Objective-C Apple фирмасининг iOS ва OS X операцион тизимлари учун дастур яратадиган дастурлаш тили ҳисобланади. Objective C 100% объектга йўналтирилган дастурлаш тилидир.



2.6 – расм. Objective-C структураси.

Objective C тили 1980 йилларнинг бошларида Stepstone фирмасида ишловчи Brad Cox ва Tom Love лар томонидан яратилган. У NeXT компанияси томонидан NeXTSTEP операцион тизимига тегишли бўлган OS X ва iOS тизимлари учун асосий тил сифатида танланган. Objective C файллари **.m** кенгайтмалари орқали, хедер файллари эса **.h** кенгайтмалари орқали ифодаланади. Бу тилда C++ тили файллари ҳам ишлатиш мумкин, фақат улар **.mm** кенгайтмалари орқали ифодаланади.(2.6 - расм) **.h** файлларида асосан **public** ўзгарувчи ва функциялар эълон қилинади. **.m** файлларида **.h** файлларида эълон қилинган функцияларнинг нима иш бажариши ёзилади. **.m** файлларидаги ўзгарувчи ва методлар **private** бўлади. Албатта барча тилларда биринчи навбатда“import world” сўзини чоп этиш амалга ошириб кўрилади. Биз ҳам шуни амалга ошириб кўрамиз.



2.7 –расм. Objective-C кенгайтмалари.

.h файл кўриниши

```
@interface XYZPerson : NSObject
- (void)sayHello;
@end
```

.m файл кўриниши

```
#import "XYZPerson.h"
@implementation XYZPerson
- (void)sayHello {
NSLog(@"Hello, World!");
}
@end
```

Бу ерда:

NSObject – super класс

NSLog – console ойнасига натижа чиқариб кўриш

Инитсиализатсия қилиш. Objective с класслари яратилгандан кейин, уларни бошқа классда объект сифатида ишлатиш учун уни биринчи инитсиализатсия қилиш керак бўлади. Бунинг учун биринчи allocate (жойлатириш) бажарилади ва инитсиализатсия қилинади. Бу икки кадам бажарилмагунча объект ҳақиқий ҳисобланмайди. Инитсиализатсия қилиши куйидагича:

```
MyObject *o = [[MyObject alloc] init];
```

Бошқа тилларда мавжуд бўлган конструкторлар Objective c тилида бошқачароқ яратилади. Яъни инициализация қилиниш давомида уни ҳам эълон қилиб кетилади. Мисол учун:

```
MyObject *o = [[MyObject alloc] initWithString:myString];
```

Протоколлар. Протокол тушунчаси C++ тилида abstract класс, Жава ва C# тилларида эса interface деб аталади. Бунинг вазифаси multiple inheritance(Кўп ворислик) дир. Унинг кўриниши қуйидагича:

```
@protocol NSLocking
```

```
- (void)lock;
```

```
- (void)unlock;
```

```
@end
```

Бошқа дастурлаш тилларида бўлгани каби, Objective c тилида ҳам бир хил номли бир нечта функцияларни яратиш мумкин. Яъни буларнинг қабул қиладиган қийматлари ҳар хил бўлади:

```
- (void)setMyValue:(id)foo;
```

```
- (void)setMyValue:(id<NSCopying>)foo;
```

```
- (void)setMyValue:(NSNumber *)foo;
```

```
- (void)setMyValue:(NSNumber<NSCopying> *)foo;
```

C++ тилига ўхшаб Objective c тили ҳам C тилига объектга йўналтирилган хусусиятларни қўшилишидан ҳосил бўлган. Аммо бу икки тил ўзининг хусусиятларига кўра умуман фарқ қилади. Objective C тили бундан ташқари ўзининг анъанавий узун номлаш усули билан ҳам жуда машҳур. Мисол сифатида қуйидаги C++ ва Objective C тилларидаги функцияга мурожаат қилишни кўришимиз мумкин.

```
//C++
```

```
john->drive(“Corvette”, “Mary’s House”);
```

```
//Objective c
```

```
[john driveCar: @"Corvette" toDestination: @"Mary’s House”];
```

Бундан кўришиб турибдики, Objective c тили инсон тушуниши учун

жуда қулайдир.

Framework. Кўплаб дастурлаш тилларига ўхшаб Objective c тилида ҳам ўзининг стандарт кутубхоналари мавжуддир ва улар Framework лар деб аталади. Бу стандарт кутубхоналар **Cocoa**(OS X учун) ва **Cocoa touch** (iOS учун) мавжуд.

<u>Framework</u>	Таъриф
<u>Foundation</u>	Ўзида қаторлар, массивлар, бутун сонлар, мантикий ифодалар, луғатлар, тўпламлар ва бошқа тоифаларни ўзида мужжам этади.
<u>UIKit</u>	iOS қуриламалари билан боғловчи бир нечта контроллерни ўзида мужжамлаштиради.
<u>AppKit</u>	UIKit билан ўхшаш, фақат OS X қуриламалари учун
<u>CoreData</u>	Маълумотлар базаси билан боғловчи бир нечта классларни ўз ичига олади
<u>MediaPlayer</u>	Мусика, видео ларни ишга тушириш учун фойдаланиладиган классларни ўз ичига олади.
<u>AVFoundation</u>	Қуйироқ қисмдаги классларни ўз ичига олади. Яъни бу класслар аудио/видео ларни ишга тушириш ва ёзиш учун мўлжалланган.
<u>QuartzCore</u>	Дастурдаги расмларни ва аниматсияларни бошқариш учун керак бўлган класслардан иборат

Xcode. Objective c тилини ишлатиш учун Xcode компиляторидан фойдаланилади. У ўзида кўплаб имкониятларни мужжамлаштирган. Код ёзиш имконияти, interface қисми, қурилманинг симуляторлари, муккамал тест қилиш ва debug қилиш имкониятлари ва бошқалар.

main() функцияси. Бошқа тилларга ўхшаб Objective c тилида ҳам биринчи бўлиб main() функцияси ишга тушади. Xcode дастури автоматик

равишда main() функциясини яратади. У қуйидагича бўлади:

```
#import <Foundation/Foundation.h>
int main(int args, const char * argv[]){
    @autoreleasepool{
        NSLog(@"Hello, World!");
    }
    return 0;
}
```

3. ДАСТУРИЙ ТАЪМИНОТ ИШЛАБ ЧИҚИШДАГИ МОДУЛЛАР ВА УНДАН ФОЙДАЛАНИШ

3.1. Дастурий таъминот модуллари ва уларни яратиш

Электрон Луғат тайёрлаш давомида дуч кулган муаммолардан бири бу сўзни қандай усул орқали қидириш бўлади албатта. Фойдаланувчи бирор сўзни қидираётганда сўзни хато киритиши ёки киритиш давомида бир нечта харфларни тушуриб қолдириши мумкин. Буларни олдини олиш учун қуйидаги алгоритмларни кўриб чиқдим.

Левенштейн масофаси (Levenshtein distance). Левенштейн масофаси деганда, икки қаторнинг ўзаро фарқлари миқдори тушунилади. Бошқача сўз билан айтганда биринчи сўзни минимум ўзгартириш(қўшиш, ўчириш, ўрин алмаштириш) орқали иккинчисини ҳосил қилиш демакдир. 1965 йил Владимир Левенштейн ҳисоблаб топди ва уни номи билан атала бошланди. Буни усулни бундан ташқари “ўзгаришлар масофаси”(“edit distance”) деб ҳам юритилади.

Математик жиҳатдан Левенштейн масофаси қуйидагича топилади.

Бизга a ва b қаторлар берилган бўлсин, $lev_{a,b}(|a|, |b|)$ қуйидагига тенг:

$$lev_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} lev_{a,b}(i-1, j) + 1 \\ lev_{a,b}(i, j-1) + 1 \\ lev_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Агар $a_i = b_j$ тенглик бажарилса, $1_{(a_i \neq b_j)}$ 0 га тенг бўлади, акс холда 1 га тенг бўлади.

Шуни эсда тутиш керакки, минимумлар топишда биринчи элемент a қатордан битта белги ўчириш, иккинчи элемент битта белги қўшиш, учинчиси эса мос келиш ёки келмаслик билдиради.

Мисол учун, “kitten” ва “sitting” сўзлари учун Левнштейн масофаси 3 га тенг, чунки булар ўртаси энг камида қуйидаги 3 та ўзгартириш бажарилади:

1. `kitten` → `sitten` (“k” ни ўрнига “s” ни алмаштириш)
2. `sitten` → `sittin` (“e” ни ўрнига “i” ни алмаштириш)
3. `sittin` → `sitting` (охирига “g” ни қўшиш).

Одатда бу усул жуда узун каторлар учун унчалик қўл келмайди. Унинг рекурсия орқали ифодаси қуйидагича:

// `len_s` va `len_t` лар `s` ва `t` каторларнинг мос равишда узунлиги

```
int LevenshteinDistance(string s, int len_s, string t, int len_t)
{
    /* Agar ulardan biri bo'sh bo'lgan holat uchun */
    if (len_s == 0) return len_t;
    if (len_t == 0) return len_s;
    /* test ularning oxirgi belgilari mos tushishini tekshirish */
    if (s[len_s-1] == t[len_t-1]) cost = 0;
    else cost = 1;

    return minimum(LevenshteinDistance(s, len_s - 1, t, len_t ) + 1,
                   LevenshteinDistance(s, len_s , t, len_t - 1) + 1,
                   LevenshteinDistance(s, len_s - 1, t, len_t - 1) + cost);
}
```

Бу ерда агар қатор жуда узун берилса, амаллар сони жуда кўп бўлиши мумкин. Буни қисқартириш учун қуйидагича иш бажарамиз. Бизда рекурсия давомида битта қисм функция бир неча бор такрорланади. Буни олдини олиш учун биз такрорланган функцияни эслаб қолишимиз ва кейинги сафар учраганда буни текшириб қўйишимиз керак. Икки ўлчовли `d[][]` функция буни бажаришда бизга ёрдам беради. Унинг кўриниши қуйидагича:

```
int LevenshteinDistance(char s[1..m], char t[1..n])
```

```

{
  int d[0..m, 0..n] ni e'lon qilamiz
  d massivni hamma elementlarini tozalaymiz //ya'ni nolga initsializatsiya
qilamiz
  for i from 1 to m
  {
    d[i, 0] := i
  }
  for j from 1 to n
  {
    d[0, j] := j
  }
  for j from 1 to n
  {
    for i from 1 to m
    {
      if s[i] = t[j] then
        d[i, j] := d[i-1, j-1]
      else
        d[i, j] := minimum
          (
            d[i-1, j] + 1, // o'chirish
            d[i, j-1] + 1, // qo'shish
            d[i-1, j-1] + 1 // almashtirish
          )
    }
  }
  return d[m, n]
}

```

Юқоридаги мисолни матритсадаги кўриниши:

	k	i	t	t	e	n	
0	1	2	3	4	5	6	
s	1	1	2	3	4	5	6
i	2	2	1	2	3	4	5
t	3	3	2	1	2	3	4
t	4	4	3	2	1	2	3
i	5	5	4	3	2	2	3
n	6	6	5	4	3	3	2
g	7	7	6	5	4	4	3

Ҳамминг масофаси (hamming distance). Ҳамминг масофаси иккита узунликлари тенг қаторларнинг ўзаро фарқли белгилари миқдори билан ўлчанади. Бошқача сўз билан айтганда бир сўзни ҳосил қилиш учун иккинчи сўзни минимал белгиларини алмаштиришлар сони ҳисобланади.

Иккита a ва b иккилик қаторлар учун ҳамминг масофаси a XOR b амалидан ҳосил бўлган қийматнинг бирлар сонига тенг бўлади. Бу иккилик қаторлар дан ҳосил бўлган ҳамминг масофаси ҳамминг кубини номи билан ҳам машҳур.

1950 йилларда Ричард Ҳамминг Ҳамминг кодлар билан ишлаш жараёнида бу қийматни ўйлаб топгандан кейин, унинг номи билан яъни ҳамминг масофаси номи билан атала бошланган.

Дастурий кўриниши қуйидагича:

```
unsigned hamdist(unsigned x, unsigned y)
{
    unsigned dist = 0, val = x ^ y; // XOR amali

    while(val)
    {
```

```

++dist;
val &= val - 1;
}

return dist;
}

```

Soundex алгоритми. **Soundex** фонетик алгоритм ҳисобланиб, у сўзларни ўзаро товушига яъни талаффузига қараб гуруҳларга ажратади. Бу алгоритмнинг асосий мақсади ёзилиши ҳар хил, лекин талаффуз қилиниши бир-бирига яқин бўлган сўзларни топишдир. Бу алгоритм 1918 йилда **Роберт С. Руссел ва Маргарет Кинг Оддел** томонидан яратилган.

Фақат биринчи ҳарфи унли бўлган сўзларнигина шифрлаши мумкин, қолган ҳолатларда фақат ундош ҳарфларни шифрлайди. **Soundex** алгоритми фонетик алгоритмлар ичида энг машҳурларидан ҳисоблиб, оммабоп маълумотлар базаларидан бўлган **DB2, PostgreSQL, MySQL, MS SQL Server, Oracle**ларнинг асосий стандартларида **Soundex** алгоритми жойлаштирилган ва одатда синоним сўзлар учун ишлатилади. Ҳозирги кунларда ҳам энг замонавий алгоритмлардан бири ҳисобланади.

Soundex коди битта ҳарф ва учта рақамдан иборат бўлади. Ҳарф – берилган сўзнинг биринчи ҳарфи, рақамлар эса ундош ҳарфларнинг шифрланишидан ҳосил бўлган рақамлар. Бир хил эшитиладиган ҳарфлар бир хил кодга эга бўлади. Масалан, лаб ундошлари: В, F, Р ва V лар бир хил 1 орқали кодланади.

Шифрлашнинг тўлиқ кўриниши қуйидагича бўлади:

1. Сўзнинг биринчи ҳарфи сақлаб қолинади ва қолган барча унлиларни ташлаб юборилади.

2. Ундошларни қуйидаги тартибда рақамларга ажратилади:

- В, F, P, V → 1
- С, G, J, K, Q, X, Z → 2

- D, T → 3
- L → 4
- M, N → 5
- R → 6

3. Агар икки ёки ундан ортиқ бир хил ҳарфлар ёнма-ён келса, уларнинг фақат биттаси сақлаб қолинади. Қолганлари эса ташлаб юборилади.

4. Бу кетма-кетлик битта ҳарф ва 3 та рақам ҳосил бўлгунга қадар давом этади. Агар бизда ҳарфлар сони жуда кўп бўлса, ортиқча ҳарфларни ташлаб юборишга тўғри келади. Агар етарли ҳарф ва рақамларга эга бўлмасал, қолганларини нол билан тўлдириш кифоя.

Масалан,

“Robert” ва “Rubert” сўзлари учун бир хил “R163” кўринишида шифрланади.

“Salom” ва “Salom” учун “S450” бўлади.

Унинг дастурий кўриниши қуйидагича бўлади:

```
public class Soundex {
    public static final char[] MAP = {
        '0','1','2','3','0','1','2','0','0','2','2','4','5',
        '5','0','1','2','6','2','3','0','1','0','2','0','2'
    };
    public static String soundex(String s) {
        String t = s.toUpperCase();
        StringBuffer res = new StringBuffer();
        char c, prev = '?';
        for (int i=0; i<t.length() && res.length() < 4 &&
            (c = t.charAt(i)) != ','; i++) {
            if (c == '\\') continue;
            if (c>='A' && c<='Z' && c != prev) {
                prev = c;
                if (i==0)
                    res.append(c);
```

```

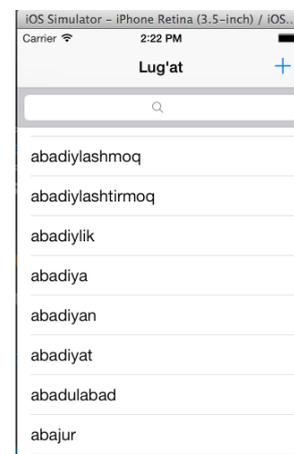
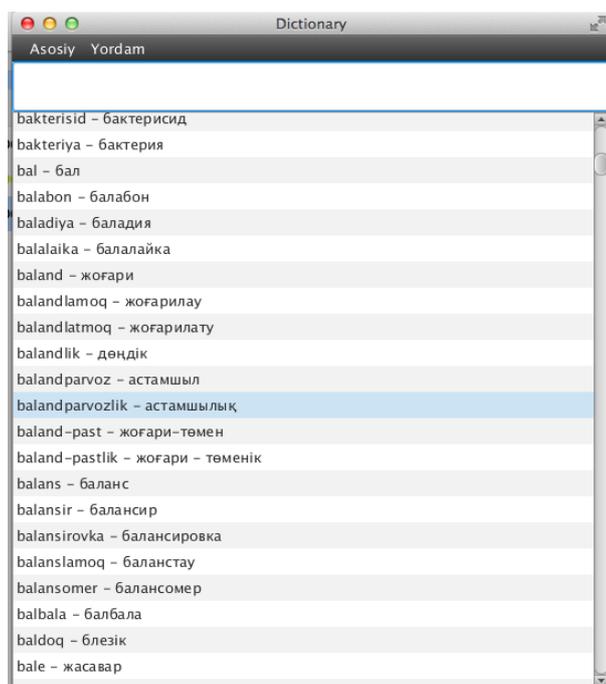
        else {
            char m = MAP[c-'A'];
            if (m != '0')
                res.append(m);
        }
    }
}
if (res.length() == 0)
    return null;
for (int i=res.length(); i<4; i++)
    res.append('0');
return res.toString();
}
public static void main(String[] args) {
    System.out.println(soundex("salom"));
    System.out.println(soundex("salem"));
}}

```

Хулоса қилиб шуни айтиш мумкинки, кўриб чиқилган усуллардан энг маъқул келадигани *soundex* алгоритми бўлди. Чунки бу усул сўзларни фонетик таҳлил қилган ҳолда қидиради.

3.2 Фойдаланувчига йўриқнома

Электрон луғатни компьютер ва мобил версиялари ишлаб чиқилди. Дастурий таъминотнинг сўз қидириш, янги сўз қидириш, сўзни таҳрирлаш ва бошқа имкониятлари мавжуд. Дастур имкониятлари қуйида келтирилган.

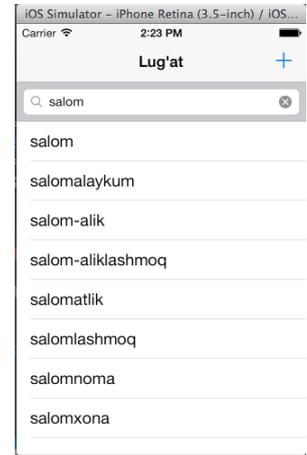
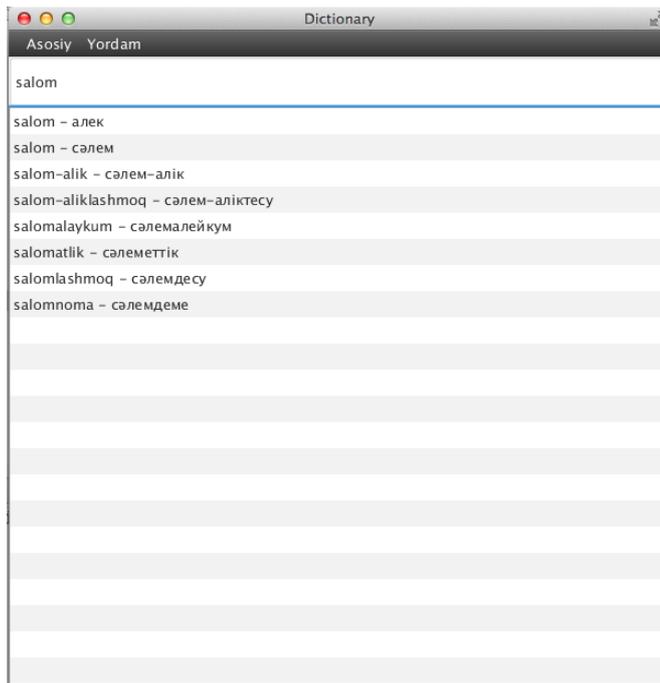


3.1 – расм. Дастур ойнасиннг асосий кўриниши

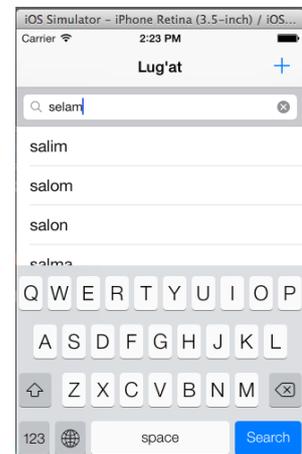
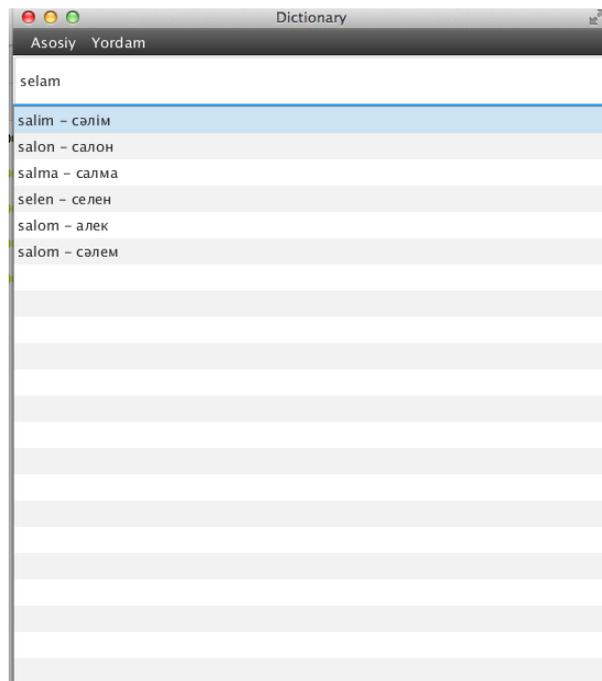
Бирор сўзни қидириш. Бирор бир сўзни қидириш қуйидагича амалга оширилади: Агар сиз бирор ўзбекча сўзни таржимасини қидирмоқчи бўлсангиз, қидирув майдонига лотин ҳарфлари орқали ўша сўзни киритиш керак. Агар қозоқча сўзни таржимасини қидирмоқчи бўлсангиз, у ҳолда крилл ҳарфлари орқали шу сўзни киритишингиз керак. (Қидириш қулай бўлиши учун ўзбек тили лотин ҳарфлари орқали, қозоқ тили крилл ҳарфлари орқали ифодаланган).

Қидирув икки усулда амалга оширилади:

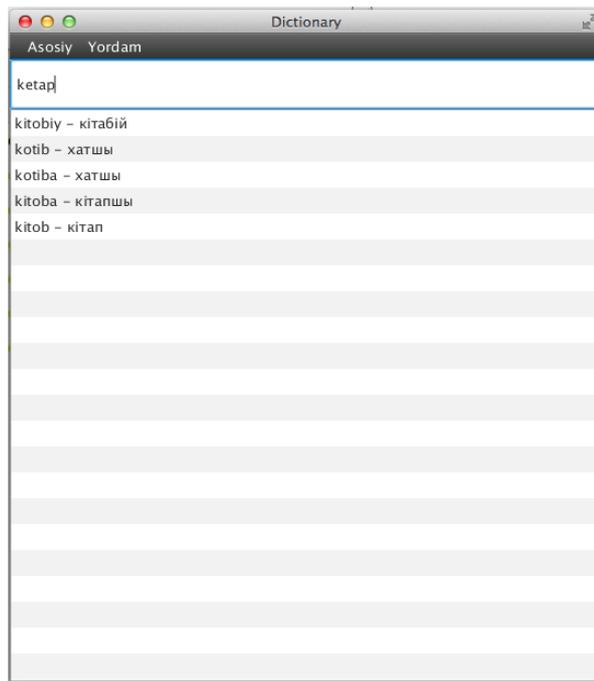
1. Киритилган сўзни префикс сифатида қараб қидириш (3.2 - расм)
2. Киритилган сўзга фонетик яқин сўзларни қидириш (3.3 ва 3.4 - расмлар)



3.2 – расм. 1-усулга мисол

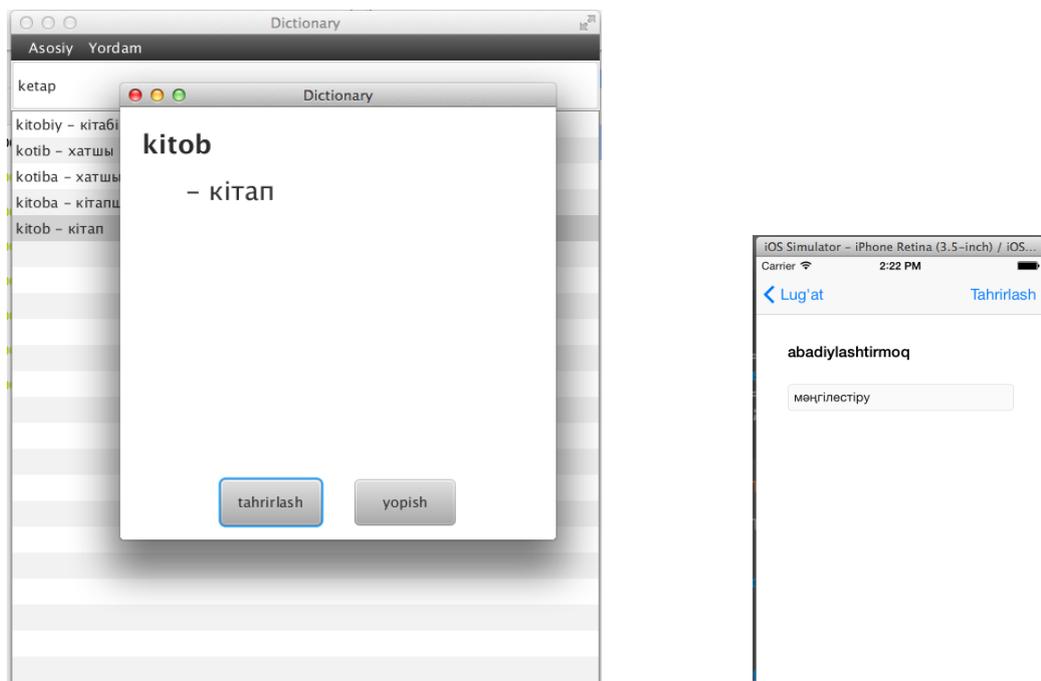


3.3 – расм. 2-усулга мисол



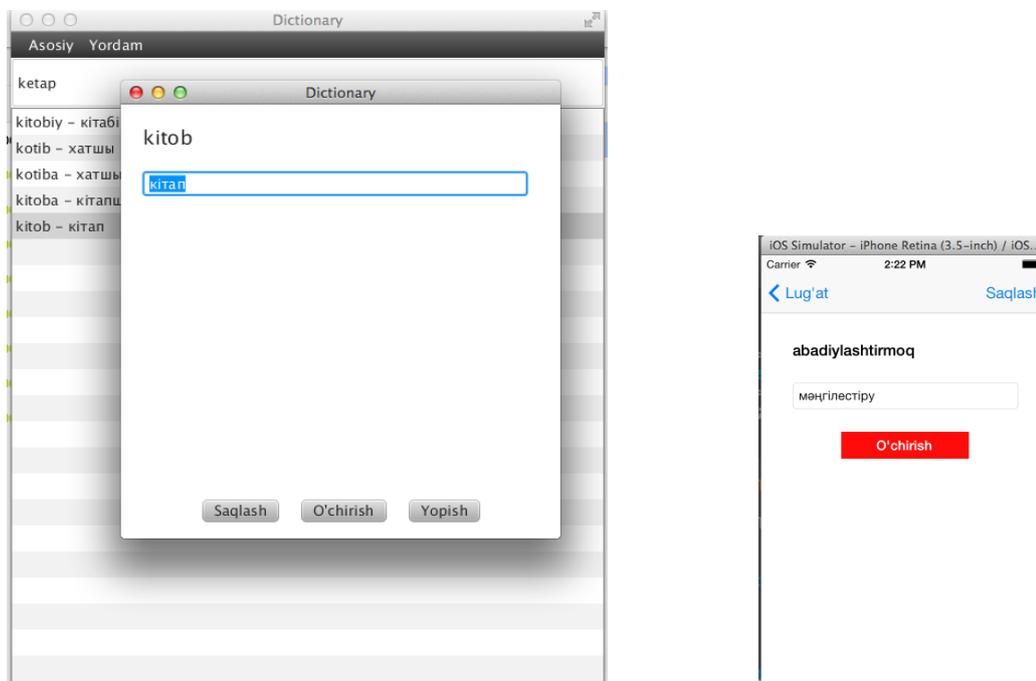
3.4 – расм. 2-усулга мисол.

Қўшимча имкониятлари. Агар топилган сўзни ўчирмоқчи ёки ўзгартирмоқчи бўлсангиз ўша сўзни устига бозсангиз қуйидаги (3.5 - расм) кўринишда ойна очилади.



3.5 – расм. Таҳрирлашдан олдинги ойна.

Сўзни ўзгартириш ёки ўчириб ташлаш учун <tahrirlash> тугмасини босинг. Бунда 3.6 – расм кўринишида ойна очилади.

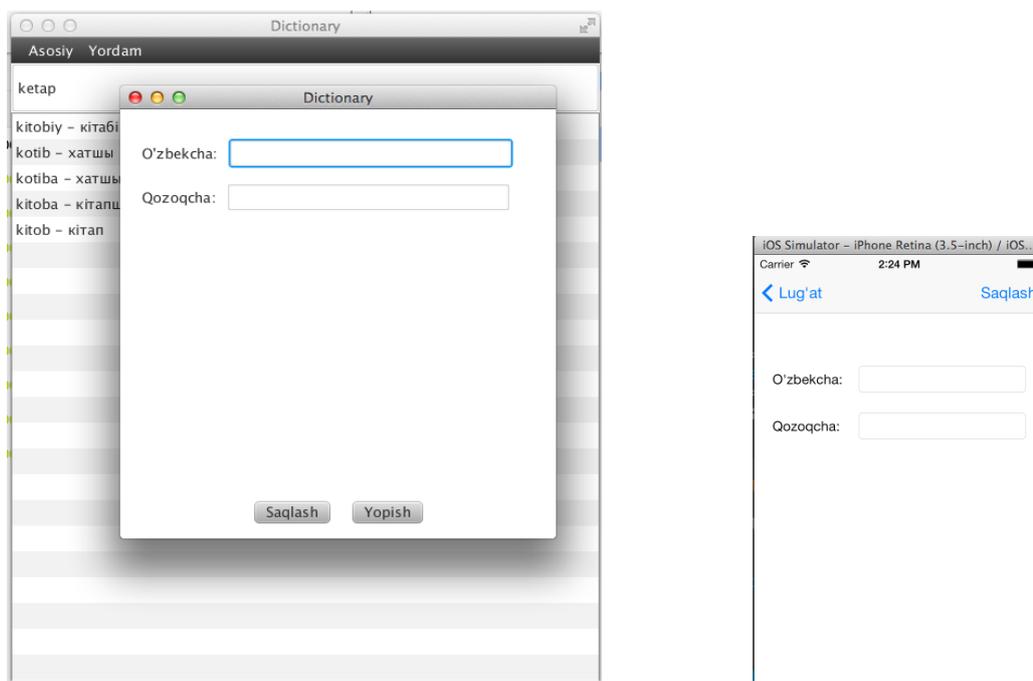


3.6 – расм. Таҳрирлаш ойнаси

Ўзгартирилгандан кейин сақлаш учун <Saqlash> тугмасини босиш керак. Агар ўчириб ташламоқчи бўлсангиз <O'chirish> тугмасини босиш кифоя. Акс ҳолда, агар ҳеч қандай ўзгартириш амалга оширмасангиз ёки ўзгартирганингизни сақлаш ниятингиз бўлмаса <Yopish> тугмасини босинг.

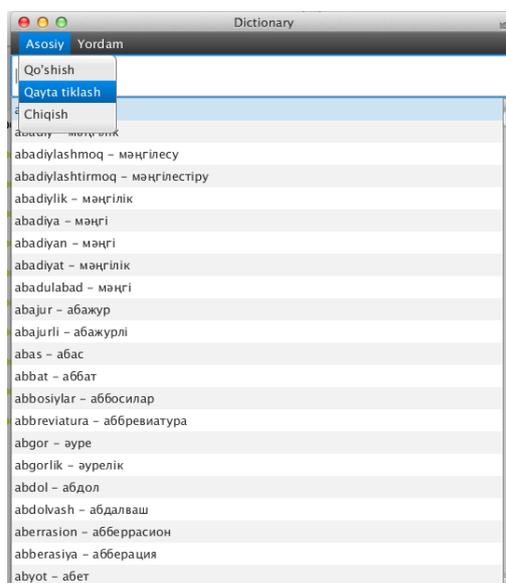
Asosiy меню элементлари.

Qo'shish. *Qo'shish* орқали биз луғатга янги сўз киритишимиз мумкин. Буни амалга ошириш учун 3.7 – расм кўринишидек майдончаларни мос равишда тўлдириш ва сақлаш тугмасини босиш кифоя.



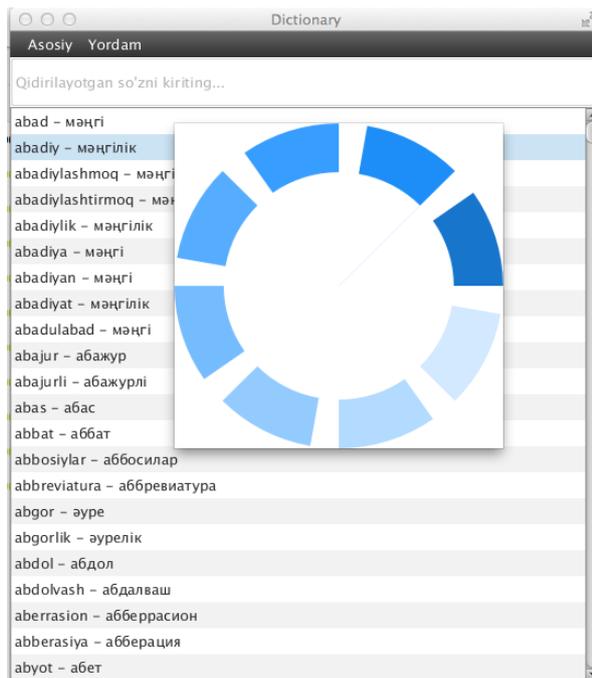
3.7 – расм. Янги сўз қўшиш.

Qayta tiklash. Дастурда ишлаш мобайнида биз созларга қандайдир ўзгартиришлар киритдик, ва яна бундан ташқари янги сўзлар қўшдик. Агар бу ўзгартишлар давомида қандайдир хатоликларга йўл қўйилган бўлса, ва базани бошланғич ҳолатга келтирмоқчи бўлсак. *Qayta tiklash* орқали буни амалга ошириш мумкин. Буни амалга ошириш қуйидагича: <Asosiy> -> <Qayta tiklash>(3.8 - расм).



3.8 – расм. Қайта тиклаш менюсини танлаш.

Сўзларни юклаш давомида куйидаги чиқади:



3.9 – расм. Қайта тиклаш жараёни

4. ҲАЁТ ФАОЛИЯТИ ХАВФСИЗЛИГИ

Фан-техника тараққиёти инсоннинг меҳнат фаолиятида талай қулайликларни яратиш билан бирга инсон ҳаёти учун зарарли бўлган баъзи бир омилларни келиб чиқишига ҳам олиб келди. Саноатда электр қувватидан кенг фойдаланиш йўлга қўйилди. Ҳозирги пайтда ҳар қандай электр қурилма электр токи билан ишлайди. Шу сабабли электр токи таъсирида рўй бериши мумкин бўлган бахтсиз ходисалар ва улардан сақланиш муҳим масалалар қаторига киради. Электр токининг энг хавфли томони шундаки, бу хавфни олдинроқ сезиш имкони йўқ. Шунинг учун ҳам электр токи хавфига қарши ташкилий ва техник чора тадбирларни белгилаш, тўсиқ воситалари билан таъминлаш, шахсий ва жамоа тизимларини ўрнатиш ниҳоятда муҳим.

4.1. Корхоналарда меҳнат муҳофазасига оид ишларни ташкил қилиш

Корхоналар маъмурияти ва муҳандис-техник ходимларнинг асосий вазифалари меҳнат ҳақидаги қонунлар мажмуи ҳамда “Хавфсизлик йўллари ва ишлаб чиқариш санитарияси қоидалари” билан белгиланади. Ишлаб чиқаришда шикастланиш ва касб касалликларини камайтириш ҳамда уларнинг олдини олишга оид меҳнат муҳофазаси бўйича ишларни амалга ошириш, тадбирларига умумий раҳбарлик ҳамда бу ишга жавобгарлик корхона раҳбари унинг ўринбосари-бош муҳандис зиммасига юклатилади.

Корхона раҳбари:

-ишлаб чиқаришда шикастланиш ва касб касалликларининг олдини олувчи ташкилий техник тадбирларни режалаштиришга;

-ана шу тадбирлар учун ўз вақтида маблағ ажратишга ва уларни ўтказишга доир руйхатларни тасдиқлашга ҳамда меҳнат шароитини мустаҳкамлаш ва соғломлаштириш учун ажратилган маблағнинг тўғри сарфланишини назорат қилиб боришга;

-меҳнат муҳофазасига доир жамоа шартномалари ва битимларининг

бажарилишини тaминлашга;

-мехнат ва дам олиш тартиби, аёллар ҳамда ўсмирлар меҳнaтини муҳофaзa қилиш ҳақидаги қонунларга амал қилишга;

-қасаба уюшмаси техник назоратчилари ва жамоатчи назоратчилар ҳамда маҳаллий қасаба уюшмаси қўмитаси комиссиясининг меҳнат муҳофазасига доир буйруқларини бажаришга;

-ишлар ва қасбларнинг айрим турлари учун хавфсизлик йўллари бўйича йўриқномаларни тасдиқлашга;

-ишчи-хизматчиларни ўз вақтида амалдаги меъёрларга мувофиқ қоржом, махсус пойафзал, яққа тартибдаги ҳимоя воситалари ва махсус озиқ-овқатлар билан тaминлашга мажбур.

Бош муҳандис:

-ҳамма цехлар ва бўлинмалар бошлиқларининг хавфсизлик йўллари ҳамда ишлаб чиқариш санитариясига доир қонун чиқарувчи меъёрлар ва қоидаларни бажаришларини мунтазам равишда назорат қилиб бориш;

-амалдаги хавфсизлик йўллари ва ишлаб чиқариш санитарияси қоидалари ҳамда мавжуд ишлаб чиқариш шароитига мувофиқ, қасблар ва ишлар турлари бўйича хавфсиз ишлаш йўллари ҳамда усулларига доир йўриқномаларни ишлаб чиқаришга қўлланишига раҳбарлик қилиш;

-хавфсиз ишлаш усуллари, йўл-йўриқларини ўрганиш юзасидан ўқув ишлари олиб борилишини назорат қилиш;

-ишчиларнинг дастлабки ва даврий тиббий кўриклардан ўтказилишини назорат қилиш;

-хавфсиз ишлаш усуллариининг оммавий тадбиқ қилиниши, хавфсизлик хоналарида маърузалар, суҳбатлар ўтказилиши, хавфсизлик йўлларига оид плакатлар ва огоҳлантирувчи ёзувлар тайёрланишини назорат қилиш;

-қасаба уюшмаси ташкилоти билан биргаликда меҳнат муҳофазаси, хавфсизлик йўллари ва ишлаб чиқариш санитариясига оид ишларнинг ҳолатини текшириш ҳамда меҳнат муҳофазаси юзасидан қабул қилинган қарорларнинг ишчилар томонидан бажарилишини назорат қилиш;

-замонавий тузилишдаги тўсик техникаси, сермехнат жараёнларни автоматлаштириш, шамоллатиш ва санитария-маиший ускуналарни жорий этиш;

-мехнат муҳофазаси, ишлаб чиқариш маданияти ва техник эстетика бўйича тажриба алмашиш ишига раҳбарлик қилиш;

-ишчиларга ўз вақтида сифатли коржома, махсус пойафзал ва якка тартибдаги ҳимоя воситалари берилишини назорат қилиш;

-хавфсизлик йўллари, ишлаб чиқариш санитариясига доир амалдаги қоидаларга, касаба уюшмаси техник назоратчиси ва жамоатчи назоратчиларнинг меҳнат муҳофазасига доир буйруқларга амал қилинишини назорат этиш;

-белгиланган муддатларда ишлаб чиқаришда рўй берган шикастланиш тўғрисидаги ҳисоботларни, меҳнат шароитини соғламлаштиришга ажратилган маблағларнинг сарфланишини назорат қилади.

Бош механик, энергетик:

-бинолар иншоотлар, энерготизмлар, турли ускуналарни профилактик кўздан кечириш ва режа асосида тузатишларнинг тўғри ташкил этилиши ҳамда ўз вақтида ўтказилишига, шунингдек, тузатиш ишларининг хавфсиз бажарилишига;

-кранлар ва бошқа турдаги юк кўтариш механизмлари ҳамда дастгоҳлари, механик ускуналардан босим остида ишлайдиган бак, сув иситиш қозонлари, аппаратлар, идишлар ҳамда ускуналарни ўз вақтида техник текширувдан ўтказилишига;

-номенклатурадаги тадбирларга доир битимга мувофиқ меҳнат муҳофазасига оид ташкилий-техник тадбирларнинг ўз вақтида амалга оширилишига жавобгарлик;

-электр жиҳозлари, куч ва ёритиш электр тармоқлари, электр тақсимлаш ускуналари, яшиндан ҳимоялагичнинг соз ҳолатда бўлишини мунтазам назорат қилишга;

-шамоллатиш қурилмалари ва иситиш тизимларининг тегишли ҳолатда

бўлишини назорат қилишга масъул.

Цех бошлиқлари, усталар:

-ишчиларнинг меҳнат муҳофазаси, хавфсизлик йўллари ва ишлаб чиқариш санитариясига доир қоида ҳамда меъёрларга амал қилишларини тامينлашга, хавфли ва зарарли меҳнат шароити билан боғлиқ ишларни бажаришда эҳтиёткорлик чораларининг бажарилишини назорат қилишга;

-меҳнат муҳофазаси, хавфсизлик йўллари ва ишлаб чиқариш санитариясига доир амалдаги қоидалар ҳамда меъёрларга мувофиқ, хавфсиз ишлаш йўллари ва усуллари юзасидан йўриқномалар ишлаб чиқаришда қатнашишга;

-барча ишчиларга хавфсиз ишлаш йўллари ва усулларини ўргатишга, шунингдек, ўз тасарруфидаги бўлинма ишчиларига хавфсизлик бўйича йўриқномалар беришга мажбур.

Меҳнат муҳофазаси, хавфсизлик йўллари ва ишлаб чиқариш санитариясига доир ишларни ташкил қилишга жавобгар бўлган меҳнат муҳофазаси муҳандиси зиммасига қуйидаги вазифалар юклатилган:

-бошланғич йўл-йўриқларни бериш;

-амалдаги қонунларнинг, вазирликлар ва идораларнинг қарорлари ҳамда фармойишларининг, шунингдек, хавфсизлик йўлларига доир қоида ва меъёрларнинг цехлар, бўлимлар раҳбарлари томонидан бажарилишини назорат қилиш;

-йўриқномалар ишлаб чиқишда қатнашиш ҳамда уларнинг тўғри қўлланилишини текшириш;

-буйруқ ва фармойишлар лойиҳаларини тайёрлаш;

-меҳнат шароитини яхшилашга доир тадбирлар ишлаб чиқиш, ташкилий-техник тадбирлар режалари лойиҳаларини ишлаб чиқиш ва уларнинг бажарилишини назорат қилиш;

-меҳнат муҳофазаси ва хавфсизлик йўлларига оид мукамал тўсиқлар ва сақловчи ускуналарни ишлаб чиқишда илмий-текшириш институтлари ва илғор корхоналарнинг шу соҳадаги таклифларини ишлаб чиқаришга жорий

этишда қатнашиш;

-корхонани қишки ва ёзги шароитда ишлашга тайёрлаш тадбирларини ишлаб чиқишда қатнашиш ҳамда уларнинг амалга оширилишини назорат қилиш;

-жамоа шартномасида кўзда тутилган меҳнат шароитларини соғломлаштириш ва енгиллаштириш тадбирларини бажарилишини текшириш;

-бинолар, ускуналар, аппаратлар, ускуналарни қуриш, қайта қуриш, капитал тузатиш лойиҳаларини кўриб чиқувчи ва уларни фойдаланишга қабул қилиб олувчи комиссияларда қатнашиш;

-ишчиларга йўл-йўриқ бериш ҳамда муҳандис-техник ходимлар ва ишчиларнинг хавфсизлик йўллари курсида ўқитилишини ташкил этиш;

-дастлабки ва даврий тиббий текширувларнинг ўз вақтида ўтказилишини назорат қилиш;

-иш хоналаридаги кўринадиган жойларга меҳнат муҳофазаси, хавфсизлик йўллари ва ишлаб чиқариш санитариясига доир амалдаги ҳамма қарорлар, қоида ва меъёрлардан бир нусхадан олиб қўйиш;

-хавфсизлик йўллари хоналарини жиҳозлаш, хавфсизлик йўлларига доир стендлар ташкил этиш, плакатлар ва огоҳлантирувчи ёзувларни осиб қўйиш;

-ишлаб чиқариш билан боғлиқ кўнгилсиз ҳодисаларнинг рўй бериш сабабларини текширишда қатнашиш ҳамда уларни бартараф этиш, олдини олиш тадбирларини ишлаб чиқиш;

-ишлаб чиқариш билан боғлиқ кўнгилсиз ҳодисаларни ҳисобга олиб ва қайд қилиб бориш, ишлаб чиқаришда шикастланишларни таҳлил қилиш;

-хавфсизлик йўллари ишларини яхши йўлга қўйган ходимларни тақдирлаш ҳамда хавфсизлик йўллари талаблари ва қоидаларини бузганларни қонунда белгиланган тартибда жавобгарликка тортиш тўғрисида корхона раҳбариятига таклифлар бериш.

Хавфсизлик йўллари бўйича муҳандис ушбу ҳуқуқларга эга:

-хавфсизлик йўллари талаблари ва қоидаларининг бузилишларини бартараф этиш ҳақида бўлинмалар, бўлимлар раҳбарларига кўрсатмалар бериш (бундай кўрсатмалар фақат корхона раҳбари ёки бош муҳандис (техник раҳбар) томонидан бекор қилиниши мумкин;)

-ишловчиларнинг ҳаёти ва соғлиги учун яққол хавф пайдо бўлганда бўлимлар, дастгоҳлар ва ускуналарда ишлашни тақиқлаб қўйиш ёки тўхтатиш ҳамда бу ҳақда дарҳол корхона раҳбариятига маълум қилиш;

-хавфсизликни таъминлай олмайдиган, талабга жавоб бермайдиган ускуналар, асбоблар, мосламаларни фойдаланишдан чиқариб ташлаш чораларини кўриш;

-цехлар, бўлинмалар раҳбарларидан ишлаб чиқариш билан боғлиқ бўлган кўнгилсиз ҳодисаларни мунтазам равишда ҳисобга олиб боришни ва ўз вақтида текширишни талаб қилиш;

-ишлаб чиқариш бўлимининг раҳбари билан биргаликда, талаблар ва қоидаларни бузганларни вақтинча ишдан четлатиш.

Хавфсизлик йўллари бўйича муҳандис бевосита корxonанинг бошлиғи ва бош муҳандисга бўйсунди. У ўз ишини маҳаллий касаба уюшмаси кўмитаси, меҳнат муҳофазаси бўйича комиссия, шунингдек, Меҳнат ва аҳолини ижтимоий муҳофаза қилиш вазирлиги саноат техник назоратчилари билан ҳамкорликда амалга оширади.

4.2. Техноген характердаги фавқулодда ҳолатларнинг содир бўлиши.

Техноген ҳалокатлар

Катта ҳудудларда портлаш, ёнғин, радиоактив, химиявий ва биологик зарарланишларни ҳамда инсонлар ҳаётига хавф солиб, гуруҳли ўлимларга олиб келувчи, ишлаб чиқариш жараёнининг кескин ишдан чиқиши билан кечадиган ҳодисалар, яъни машина ва механизмларни қўққисдан, кутилмаганда фойдаланиш даврида ишдан чиқиб авария ҳамда ҳалокатларга олиб келиши техноген ҳалокатлар деб аталади.

Техноген ҳалокатларга саноат объектларидаги, қурилиш, темир йўл, ҳаво ва автомобил транспорти, сув транспортидаги қувурлар, газ-нефть қувурлари ва бошқа шу каби объектлардаги авариялар мисол бўлади. Бундай авариялар натижасида ёнғинлар ва портлашлар келиб чиқиши, аҳоли яшаш ва саноат биноларининг бузилиши, радиацион, химиявий ва биологик заҳарланишлар вужудга келиши, ҳар хил авариялар оқибатида нефть маҳсулотлари ва заҳарли моддаларни оқиши билан ер, сув ва ҳавонинг ифлосланиши, аҳоли ҳаётига ва атроф-муҳитга катта хавф туғилиши рўй беради.

Техноген ҳалокатлар ташқи табиий омиллар таъсирида, жумладан, табиий офатлар оқибатида, бино ва иншоотларни лойиҳавий ва ишлаб чиқариш нуқсонлари, камчиликлари ва ишлаб чиқариш технологиясини бузилиши натижасида рўй бериши мумкин.

Антропоген ҳалокатлар - инсониятнинг хўжалик фаолияти туфайли юзага келувчи антропоген омиллар таъсирида биосферанинг сифат жиҳатидан ўзгариши ва натижада инсонлар ҳаётига, ўсимлик ва ҳайвонот дунёсига ҳамда атроф-муҳитга таҳдид солувчи ва катта хавф туғдирувчи ҳодисалардир.

Бундай экологик ҳаракатердаги антропоген ҳалокатларга тупроқни интенсив равишда деградацияланиши ва оғир металлар (кадмий, қўрғошин, симоб, хром ва бошқалар) ҳамда бошқа зарарли моддалар билан ифлосланиши, атмосферани зарарли химиявий моддалар, шовқин, электр магнит майдони ва ионли нурланишлар билан ифлосланиши, кислотали ёмғирлар, озон қатламини емирилиши, йирик саноат шаҳарларида ҳарорат инверсиясининг юзага келиши, сув ресурсларини ифлосланиши ва шу каби инсоннинг турмуш тарзи сифатига таъсир этувчи, уларнинг ҳаётига таҳдид солувчи ҳодисалар киради.

Ижтимоий-сиёсий ва ҳарбий-сиёсий можаролар, икки давлатнинг ўзаро қарама-қаршилиги натижасида урушларнинг келиб чиқиши, урушда оммавий қирғин қуролларидан фойдаланиш хавфининг туғилиши ва шунга боғлиқ ҳолда бошқа турдаги муаммолар, масалан, ҳарбий можаролар вақтидаги қочоқлар муаммоси, юқумли касалликларни келиб чиқиши ва уларни катта

худудларда тарқалиш хавфларини ортиши ҳамда миллий кризислар, минтақавий можароларни юзага келиш ҳолатларидир.

Фавқулодда ҳолатлар хавфи тарқалиш тезлигига кўра қуйидаги турларга бўлинади: кутилмаганда (ер кимирашлар, портлашлар, транспорт авариялар ва бошқалар), шиддатли (ёнғинлар, гидродинамик авариялар ва бошқалар), ўртача (сув босишлар, вулқонлар отилиши, радиоактив моддалар чиқиши билан кечадиган авариялар ва шу кабилар), аста-секин тарқалувчи хавфлар (курғоқчиликэпидемия, саноаттозалаш иншоотларининг авариялари, тупроқларни ифлосланиши ва сувларни зарарли химиявий моддалар билан ифлосланиши ва бошқалар).

Фавқулодда ҳолатлар тарқалиш масштабининг кўрсаткичларига унинг тарқалиш ҳудуди ўлчамидан ташқари, хавфли омилларни ташкилий, ижтимоий, иқтисодий ва шу каби муҳим боғланишларга бевосита таъсир этиши ҳам киради. Бундан ташқари таърифлаш белгиларига фавқулодда ҳолатлар оқибатларининг даражаси, яъни, унинг асоратлари ҳам муҳим кўрсаткичлардан ҳисобланади. Чунки фавқулодда ҳолатлар кичик худудларда, кичик масштабда содир бўлсада, унинг оқибати жуда аянчли ва трагедияли бўлиши мумкин. Шу сабабли, фавқулодда ҳолатлар категориясини аниқлашда фавқулодда ҳолатлар юз берган майдон (худуд) ҳолатини ва фавқулодда ҳолатларнинг оқибатларини баҳолаш талаб этилади. Фавқулодда ҳолатлар оқибатларини ўрганиш ва баҳолаш, уни бартараф этишга қанча куч ва ресурс ажратиш лозимлигини аниқлашга асос бўлади.

ХУЛОСА

Битирув малакавий ишини бажариш давомида қуйидаги натижаларга эришилди:

- Ҳозирги вақтдаги мавжуд луғатларни таҳлил қилинди;
- Ўзбекча-русча ва қозоқча-русча луғатлар асосида ўзбекча-қозоқча луғатни ишлаб чиқилди;
- Ўзбек-қозоқ луғатни маълумотлар базасини шакллантирилди
- Н2 маълумотлар базаси бошқариш тизимини ўрганилди;
- JavaFX кутубхонаси ва Н2 маълумотлар базаси ёрдамида ўзбек-қозоқ луғатини яратилди;
- Электрон луғатдан фойдаланиш қўлланмасини ишлаб чиқилди;

Фойдаланилган адабиётлар

1. «Юксак маънавият – енгилмас куч». Ислом Каримов. Тошкент «Маънавият» 2008.
2. Ўзбекистон Республикаси Президентининг «Компьютерлаштиришни янада ривожлантириш ва ахборот-коммуникация технологияларини жорий этиш тўғрисида» 2002 йил 30 майдаги ПФ-3080-сон Фармони.
3. Ўзбекистон Республикаси Вазирлар Маҳкамасининг «Компьютерлаштиришни янада ривожлантириш ва ахборот-коммуникация технологияларини жорий этиш чора-тадбирлари тўғрисида» 2002 йил 30 июндаги 200-сон Қарори
4. Ўзбекистон Республикаси «Ахборот эркинлиги принциплари ва кафолатлари тўғрисида» 2002 йил 12 декабрдаги 439-П-сон Қонуни.
5. Ўзбекистон Республикаси «Ахборотлаштириш тўғрисида» 2003 йил 11 декабрдаги 560-П-сон Қонуни.
6. Жумаев Т. Туркий тилларнинг қиёсий-тарихий таҳлили. Ўқув қўлланма 156 б.
7. К.Бектаев. Большой казахско-русский русско-казахский словарь. 2010 йил. Алматы.
8. Назиров Ш.А., Кадиров У. Э. Давлат тилидаги элементар саволларнинг лексик-морфологик анализаторини яратиш. «Современные проблемы алгоритмизации и программирования». 5-7 сентябрь 2001 г. Ташкент. - – С.413-414.
9. Фленов М. Transact-SQL. –СПб.: БХВ – Петербург, -576с.
10. Назиров Ш.А. Туркий тиллар учун электрон таржиманинг дастурий таъминоти яратиш муаммолари // ТАТУ хабарлари №4/2011 Тошкент. 57-60 бет.
11. Назиров Ш.А., Рахманов Қ.С., Хожиев С.Н., Тўхтамуродова Ш.Ш. Табиий тилларда луғатларни куриш усуллари // “Аниқ фанларни ўқитишнинг долзарб муаммолари” мавзусидаги республика илмий-амалий

- анжумани материаллари. 2013 йил 22-23 ноябрь, Гулистон. 114-117б.
12. Назиров Ш.А., Рахманов Қ.С., Тўхтамуродова Ш.Ш. Туркий тиллар учун электрон луғатларни куриш // “Аниқ фанларни ўқитишнинг долзарб муаммолари” мавзусидаги республика илмий-амалий анжумани материаллари. 2013 йил 22-23 ноябрь, Гулистон. 121-123б.
 13. Рахманов Қ.С., Убайдуллаев А.М., Тўхтамуродова Ш.Ш. Луғат ва унинг турлари // “Аниқ фанларни ўқитишнинг долзарб муаммолари” мавзусидаги республика илмий-амалий анжумани материаллари. 2013 йил 22-23 ноябрь, Гулистон. 127-129б.
 14. Экология и безопасность жизнедеятельности: Учебное пособие для студентов ВУЗов/ ред. Л. А. Муравий, 2002.
 15. Белов С.В. Безопасность жизнедеятельности М.: Высшая школа. 2003.
 16. Ёрматов Ғ.Ё., Исамухамедов Ё.У. Мехнатни муҳофаза қилиш. Дарслик. Ўзбекистон нашриёти. Тошкент 2002.
 17. Programming iOS7, Fourth Edition. Matt Neuburg. USA 2014. Published by O'Reilly Media.
 18. iOS7 Programming Cookbook/ Vanda Nahavandipoor. USA 2014. Published by O'Reilly Media.
 19. www.ziyonet.uz .
 20. <http://www.google.com>
 21. <http://www.stackoverflow.com>
 22. www.oracle.com
 23. www.wikipedia.org
 24. www.cprogramming.com
 25. developer.apple.com

ИЛОБА

```
package sample;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.ListView;
import javafx.scene.control.TextField;
import javafx.scene.input.MouseEvent;
import javafx.stage.Modality;
import javafx.stage.Stage;
import uz.tuit.uzqaz.uz.tuit.uzqaz.Api;
import uz.tuit.uzqaz.uz.tuit.uzqaz.entity.EmptyTranslation;
import uz.tuit.uzqaz.uz.tuit.uzqaz.entity.Translation;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;

public class SampleController {
    public static ListView<Translation> resultListView;
    public static TextField searchField;
    public static List<Translation> results;

    public static void run() {

        Api.loadDB();

        searchField.setOnKeyReleased(new
```

```

EventHandler<javafx.scene.input.KeyEvent>() {
    @Override
    public void handle(javafx.scene.input.KeyEvent keyEvent) {
//        System.out.println(searchField.getText());
//        System.out.println(keyEvent.getText());

        printResult(searchField.getText());
    }
});

resultListView.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent mouseEvent) {

System.out.println(resultListView.getSelectionModel().getSelectedItem());

System.out.println(resultListView.getSelectionModel().getSelectedIndex());

        Translation trans =
resultListView.getSelectionModel().getSelectedItem();
        if (trans == null || trans.isEmpty()) {
            return;
        }

        try {
            FXMLLoader fxmLoader = new
FXMLLoader(getClass().getResource("InfoForm.fxml"));
            Parent root = (Parent) fxmLoader.load();
            InfoFormController controller = fxmLoader.getController();

```

```
controller.setWord(resultListView.getSelectionModel().getSelectedItem());
```

```
    Stage primaryStage = new Stage();
```

```
    primaryStage.setTitle("Dictionary");
```

```
    primaryStage.setScene(new Scene(root, 400, 400));
```

```
    primaryStage.initModality(Modality.WINDOW_MODAL);
```

```
    primaryStage.initOwner(searchField.getScene().getWindow());
```

```
    primaryStage.showAndWait();
```

```
    printResult(searchField.getText());
```

```
    } catch (IOException e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
    }
```

```
});
```

```
}
```

```
private static final List<Translation> empty = Arrays.asList((Translation)new  
EmptyTranslation());
```

```
private static void printResult(String string) {
```

```
{
```

```
    List<Translation> translations = Api.search(string);
```

```
    if (translations.isEmpty() || translations.size() == 0) {
```

```

        resultListView.getItems().setAll(empty);
    } else {

        resultListView.getItems().setAll(translations);

    }

}

}

public void exitAction(ActionEvent actionEvent) {
    System.exit(0);
}

public void resetAction(ActionEvent actionEvent) {
    System.out.println("populating...");
    Api.populateDB();
    System.out.println("populated");
}

public void addAction(ActionEvent actionEvent) {
    try {
        FXMLLoader fxmLoader = new
FXMLLoader(getClass().getResource("AddForm.fxml"));
        Parent root = (Parent) fxmLoader.load();
        AddFormController controller = fxmLoader.getController();

        Stage primaryStage = new Stage();
        primaryStage.setTitle("Dictionary");
        primaryStage.setScene(new Scene(root, 400, 400));
    }
}

```

```

        primaryStage.initModality(Modality.WINDOW_MODAL);
        primaryStage.initOwner(searchField.getScene().getWindow());
        primaryStage.showAndWait();
        printResult(searchField.getText());

    } catch (IOException e) {
        e.printStackTrace();
    }
}

}

package uz.tuit.uzqaz.util;
import uz.tuit.uzqaz.uz.tuit.uzqaz.entity.Translation;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class DB {

    private static Connection getConnection() {
        try {
            Class.forName("org.h2.Driver");
            return DriverManager.getConnection("jdbc:h2:~/uzqaz_dic", "sa", "");
        } catch (Exception ex) {
            throw new RuntimeException(ex);
        }
    }
}

```

```
}
```

```
private static List<String> load(String query) {  
    try {  
        Connection conn = null;  
        try {  
            conn = getConnection();  
            Statement st = conn.createStatement();  
            ResultSet rs = st.executeQuery(query);  
            List<String> result = new ArrayList<String>();  
            while (rs.next()) {  
                result.add(rs.getString(1));  
            }  
            return result;  
        } finally {  
            if (conn != null) {  
                conn.close();  
            }  
        }  
    } catch (Exception ex) {  
        throw new RuntimeException(ex);  
    }  
}
```

```
private static void update(String query) {  
    try {  
        Connection conn = null;  
        try {  
            conn = getConnection();  
            Statement st = conn.createStatement();
```

```

        st.executeUpdate(query);
    } finally {
        if (conn != null) {
            conn.close();
        }
    }
} catch(Exception ex) {
    throw new RuntimeException(ex);
}
}

```

```

public static String getSoundex(String a, String b){
    return "1234";
}

```

```

public static boolean isDBReady() {
    List<String> r = load("SELECT 1 AS count FROM
information_schema.tables WHERE table_name = 'TRANSLATIONS'");
    return !r.isEmpty();
}

```

```

public static void createDB() {
    String createQuery =
        "CREATE TABLE translations(" +
            "uzbek_word VARCHAR(100) NOT NULL," +
            "qazaq_word VARCHAR(100) NOT NULL," +
            "order_no INT," +
            "PRIMARY KEY (uzbek_word, qazaq_word))";
    update("DROP TABLE IF EXISTS translations");
    update(createQuery);
}

```

```
}
```

```
private static String q(String s) {
```

```
    return s.replace("'", "");
```

```
}
```

```
public static void delete(String uzbek, String qazaq) {
```

```
    String query = "DELETE FROM translations WHERE uzbek_word=" +  
q(uzbek) + " AND qazaq_word=" + q(qazaq) + "";
```

```
    update(query);
```

```
}
```

```
public static void saveTranslation(String uzbek, String qazaq, int orderNo) {
```

```
    String query = "SELECT 1 AS cc FROM translations WHERE  
uzbek_word=" + q(uzbek) + " AND qazaq_word=" + q(qazaq) + "";
```

```
    List<String> r = load(query);
```

```
    if (!r.isEmpty()) {
```

```
        return;
```

```
}
```

```
    query = "INSERT INTO translations VALUES(" + q(uzbek) + "," +  
q(qazaq) + "," + orderNo + ")";
```

```
    update(query);
```

```
}
```

```
public static List<Translation> loadAll() {
```

```
    try {
```

```
        String query = "SELECT uzbek_word, qazaq_word FROM translations";
```

```
        Connection conn = null;
```

```
        try {
```

```
            conn = getConnection();
```

```

Statement st = conn.createStatement();
ResultSet rs = st.executeQuery(query);
List<Translation> result = new LinkedList<Translation>();
while (rs.next()) {
    String uzbek = rs.getString(1);
    String qazaq = rs.getString(2);
    Translation translation = new Translation(uzbek, qazaq);
    translation.isUzb = true;
    result.add(translation);
}
return result;
} finally {
    if (conn != null) {
        conn.close();
    }
}
} catch(Exception ex) {
    throw new RuntimeException(ex);
}
}
}

```

```

package uz.tuit.uzqaz.util;
import java.io.InputStream;
import java.util.Scanner;
public class Resource {

    public static void populateDB() {
        Scanner sc = null;

```

```

try {
    sc = new Scanner(Resource.class.getResourceAsStream("source.txt"));
    int k = 0;
    while (sc.hasNextLine()) {
        String line = sc.nextLine();
        //      System.out.println(line);
        String[] words = line.split("=");
        String uzbek = words[0].trim().toLowerCase();
        if (uzbek.isEmpty() || words.length <= 1) {
            continue;
        }
        String[] qazaqs = words[1].split(",");
        for(int i = 0; i < qazaqs.length; i ++) {
            String qazaq = qazaqs[i].trim().toLowerCase();
            if(!qazaq.isEmpty()) {
                DB.saveTranslation(uzbek, qazaq, i);
            }
        }
        k++;
        //      if (k == 1000) break;
    }
    System.out.println("FINISHED");

} catch (Exception ex) {
    throw new RuntimeException(ex);
} finally {
    if (sc != null) {
        sc.close();
    }
}
}

```

```
}
```

```
}package uz.tuit.uzqaz.uz.tuit.uzqaz.entity;
```

```
public class Soundex {
```

```
    /* Implements the mapping
```

```
    * from: AEHIOWYBFPVCGJKQ SXZDTLMNR
```

```
    * to: 0000000011112222222334556
```

```
    */
```

```
    public static final char[] MAP = {
```

```
        //A B C D E F G H I J K L M
```

```
        '0','1','2','3','0','1','2','0','0','2','2','4','5',
```

```
        //N O P W R S T U V W X Y Z
```

```
        '5','0','1','2','6','2','3','0','1','0','2','0','2'
```

```
    };
```

```
    public static String soundex(String s) {
```

```
        String t = s.toUpperCase();
```

```
        StringBuffer res = new StringBuffer();
```

```
        char c, prev = '?';
```

```
        for (int i=0; i<t.length() && res.length() < 4 &&
```

```
            (c = t.charAt(i)) != ','; i++) {
```

```
            if (c == '\\') continue;
```

```
            if (c>='A' && c<='Z' && c != prev) {
```

```

        prev = c;
        if (i==0)
            res.append(c);
        else {
            char m = MAP[c-'A'];
            if (m != '0')
                res.append(m);
        }
    }
}

if (res.length() == 0)
    return null;
for (int i=res.length(); i<4; i++)
    res.append('0');
return res.toString();
}

}

package uz.tuit.uzqaz.uz.tuit.uzqaz.entity;

import uz.tuit.uzqaz.util.DB;

import java.util.*;

public class Data {

    private long[][] a = new long[101][101];
    private List<Translation> words = new LinkedList<Translation>();

    public void save(Translation t) {

```

```

words.remove(t);
words.add(t);
DB.saveTranslation(t.uzbek, t.qazaq, 0);
}

```

```

public List<Translation> search(String text) {
    if (text.equals("")) {
        return words;
    }
    List<Translation> result = new ArrayList<Translation>();
    int k = 0, kk = 0;
    for (Translation t : words){
        if (t.uzbek.startsWith(text)) {
            t.isUzb = true;
            result.add(t);
            k++;
        } else if (t.qazaq.startsWith(text)) {
            t.isUzb = false;
            result.add(t);
            k++;
        }
        if (k == 80) break;
    }

    if (k != 0) {
        Collections.sort(result, new Comparator<Translation>() {
            @Override
            public int compare(Translation o1, Translation o2) {
                return o1.isUzb ? o1.uzbek.compareTo(o2.uzbek) :
o1.qazaq.compareTo(o2.qazaq);

```

```

        }
    });
}

return result;
}
public List<Translation> searchSoundex(String text){
    List<Translation> result = new ArrayList<Translation>();
    String textSoundex = Soundex.soundex(text);
    if (textSoundex == null) return null;
    int k = 0;

    for (Translation t : words){

        if (textSoundex.equals(t.uzbekSoundex)){
            result.add(t);
            if (k++ == 80) break;
        }

    }

    return result;
}
public void load() {
    words = DB.loadAll();
}

public int wordCount() {
    return words.size();
}

```

```
public void delete(Translation translation){  
    words.remove(translation);  
}  
  
}
```