ЎЗБЕКИСТОН РЕСПУБЛИКАСИ АЛОҚА, АХБОРОТЛАШТИРИШ ВА ТЕЛЕКОММУНИКАТЦИЯ ТЕХНОЛОГИЯЛАРИ ДАВЛАТ ҚЎМИТАСИ
ТОШКЕТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ

«Ҳимояга рухсат»

АТДТ кафедраси мудири

Акбаралиев Б.Б.
(каф.муд. имзоси, Ф.И.Ш.)

«_____»_____2014 й.

БИТИРУВ МАЛАКАВИЙ ИШИ

Мавзу: “SUDOKU” ўйини дастурини android 4.3 платформаси учун ишлаб чиқиш

Битирувчи	_____ (имзо)	Ишкобилов Д.Б. (Ф.И.Ш.)
Раҳбар	_____ (имзо)	Рўзибоев О.Б. (Ф.И.Ш.)
Тақризчи	_____ (имзо)	Жабборов Х.Ф. (Ф.И.Ш.)
ХФХ дан маслахатчи	_____ (имзо)	Амурова Н.Ю. (Ф.И.Ш.)

ТОШКЕНТ - 2014

МАЗМУННОМА

Мазкур битирув малакавий иши “Sudoku” ўйини android платформасида мобил телефонлар учун ишлаб чиқишга бағишланган. Ушбу ишнинг эътиборли жиҳати шундаки, Sudoku ўйинини фойдаланувчилар ўз мобил телефонлари орқали фойдаланишлари мумкин. Sudoku ўйинини яратишда асосан java дастурлаш тили ва andorid платформаси компоненталаридан фойдаланилган. Дастур Java (Dalvik Virtual Machine) виртуал машинасида бажарилади.

АННОТАЦИЯ

Данная квалификационная выпускная работа посвящена в разработке в платформе Android игры Sudoku для мобильных телефонов. Особенность данной работы в том что пользователи смогут воспользоваться данной игрой используя свои мобильные телефоны. В разработке игры Sudoku использован программный язык Java и компоненты платформы Android. Программа выполнена виртуальной машине Java (Dalvik Virtual Machine).

SUMMARY

This qualification outlet is devoted to develop a platform of Android Sudoku game for mobile phones. Feature of this work is that users will be able to take this game by using their mobile phones. In the development of the game Sudoku used Java programming language and platform components Android. Program made in virtual machine Java (DalvikVirtual Machine).

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ АЛОҚА, АХБОРОТЛАШТИРИШ ВА ТЕЛЕКОММУНИКАТЦИЯ ТЕХНОЛОГИЯЛАРИ ДАВЛАТ ҚЎМИТАСИ
ТОШКЕТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ
Дастурий инжиниринг факультети АТДТ кафедраси йўналиши
5521900- Информатика ва ахборот технологияси

ТАСДИҚЛАЙМАН
Кафедра мудири
Акбаралиев Б.Б._____
2014 йил "__"_____

Битирув малакавий ишига
Т О П Ш И Р И Қ
Ишкобилов Даврон Баходирович
(фамилияси, исми, оатсининг исми)
1. Иш мавзуси "Хусусий стоматология клиникалари учун ахборот тизимини яратиш"

2. 2014 йил "25" Февралдаги 154-16 -сонли буйруқ билан тасдиқлайман
3. Ишни ҳимояга топшириш муддати : 02 Июн, 2014 йил
4. Ишга оид дастлабки маълумотлар: тадқиқот тизимининг тавсифи, тизимни яратишга техник топшириқ, объектнинг тизимли ва ишчи лойихаси ва ҳ.к.лар
5. Ҳисоблаш-тушунтириш ёзувнинг мазмуни (ишлаб чиқиладиган масалалар рўйхати): танланган мавзунинг долзарблигини асослаш; тадқиқоднинг мақсади ва уни амалга оширишда ҳал қилиниши лозим бўлган масалаларни аниқлаштириш; тадқиқод предмети ва объектини аниқлаш ва тавсифлаш; тадқиқод натижаларини таҳлил қилиш ва қайта ишлашни олиб бориш; хулосалар келтириб чиқариш, олинган натижаларни баҳолаш ва таълим соҳасига тадбиқ этиш учун таклифлар бериш.
6. График материаллар рўйхати:1) Android платформасида Sudoku ўйинини яратиш босқичлари, 2) Android платформасида Sudoku ўйинини структураси 3) Sudoku ўйиниинг интерфейслари , 4) хулоса
7. Топшириқ берилган сана _____

Раҳбар _____
(имзо)
Топшириқ олдим _____
(имзо)

8. Ишнинг айрим бўлимлари бўйича маслаҳатчилар

Қисм	Маслаҳатчи ўқитувчининг Ф.И.Ш.	Имзо,сана	
		Топшириқ берилди	Топшириқ олинди
Кириш	Рўзибоев О.Б.	27.02.2014	
Назарий қисм	Рўзибоев О.Б.	26.03.2014	
Асосий қисм	Рўзибоев О.Б.	24.04.2014	
Амалий қисм	Рўзибоев О.Б.	12.05.2014	
Якуний қисм	Рўзибоев О.Б.	26.05.2014	
Ҳаёт фаолияти хавфсизлиги	Амурова Н.	02.06.2014	

9. Ишни бажариш графиги

Т/р	Иш қисмларининг номи	Бажариш мудати	Раҳбар (маслаҳатчи белгиси)
1	Мобил қурилмалар учун android платформасида иловалар ишлаб чиқиш муҳити	27.02.2014	
2	Android платформасида “SUDOKU” ўйинини яратиш технологияси	26.03.2014	
3	Ҳаёт фаолияти хавфсизлиги	24.04.2014	
4	Хулоса ва тавсиялар	26.05.2014	
5	Ҳаёт фаолияти хавфсизлиги	02.06.2014	

Битирувчи_____ 2014 йил «___» _____
(имзо)

Раҳбар_____ 2014 йил «___» _____
(имзо)

МУНДАРИЖА

КИРИШ..... 6

1. МОБИЛ ҚУРИЛМАЛАР УЧУН ANDROID ПЛАТФОРМАСИДА ИЛОВАЛАР ИШЛАБ ЧИҚИШ МУҲИТИ

1.1. Эркин ва очиқ кодли дастурий таъминотлар қиёсий таҳлили..... 9

1.2. Android операцион тизимига кириш…… 14

1.3. Andorid операцион тизимининг хусусиятлари…….. 21

1.4. Sudoku ўйини қоидалари ва математик ечимни топиш усуллари……

1.5. Масала қўйилиши..... 26

2. ANDROID ПЛАТФОРМАСИДА “SUDOKU” ЎЙИНИНИ ЯРАТИШ ТЕХНОЛОГИЯСИ

2.1 Android платформаси архитектураси ва компоненталари……..... 27

2.2. Android платформасида иловаларни ишлаб чиқиш учун java асослари..... 32

2.3. Android платформасида "SUDOKU ўйинини ишлаб чиқиш босқичлари..... 36

2.4. “SUDOKU” ўйини фойдаланувчи йўриқномасини ишлаб чиқиш..... 48

3. ҲАЁТ ФАОЛИЯТИ ХАВФСИЗЛИГИ

3.1. Компьютер хоналарида мехнат шароитларини тўғри ташкил этиш жойини ташкил этилиши……..... 51

3.2. Ёнғин хавфсизлиги..... 54

Якуний хулоса..... 60

Фойдаланилган адабиётлар..... 61

Илова..... 63

КИРИШ

Ўтган йили иқтисодиётнинг барча тармоқларига ва кундалик ҳаётимизга ахборот-коммуникация технологияларини кенг жорий этиш масаласига ўта муҳим эътибор қаратилди. Хусусан, «Шаҳарлараро коммутация марказларини янги авлод тармоқ технологиялари асосида модернизация қилиш ва кенгайтириш» (NGN), «FTTx технологияси бўйича кенг қамровли форматда фойдаланиш имконини берадиган оптик тармоқларни ривожлантириш», «Ўзбекистон Республикаси минтақаларида EVDO технологияларини жорий этиш орқали CDMA-450 мобиль тармоғини ривожлантириш» каби ва бошқа йирик инвестиция лойиҳаларини амалга ошириш эътиборимиз марказида бўлди. Ушбу лойиҳаларнинг самарали ниҳоясига етказилиши дунёнинг замонавий алоқа ва ахборот воситалари юқори даражада ривожланган мамлакатлари қаторига қўшилиш, ишбилармонлик соҳасидаги алоқаларни кенгайтириш бўйича қўшимча имкониятлар яратади[1].

Ахборот коммуникатция технологиялари ёрдамида дастурий маҳсулотлар ишлаб чиқариш ва жорий этиш муҳим аҳамият касб этади. Ҳозирги кунда иқтисодиётни ривожлантириш суръатлари ишлаб чиқарилаётган ҳар қандай дастурий махсулотнинг сифати ва самарадорлигига эришиш борасида ўз талабларини қўймоқда. Ишлаб чиқариш жараёнлари ва ташкилотларда ҳужжат айланишини автоматлаштириш ҳамда тизимлаштириш, сарф - харажатларни камайтириш, белгиланган вазифаларни бажаришни тезлаштириш имконини яратиб, дастурий махсулот ишлаб чиқарувчи компанияларга кенг доирадаги истеъмолчилар учун мўлжалланган универсал маҳсулотларни яратиш имконини бермоқда.

Мамлакатимизда дастурий маҳсулот ишлаб чиқарувчи компаниялар томонидан энг янги маҳсулотларни ишлаб чиқиш ва интегратсиялаштиришга доир кенг кўламли лойиҳалар амалга оширилди. Ёш дастурчиларни тайёрлаш

ва қўллаб-қувватлаш маркази томонидан мамлакатимиздаги барча ахборот-кутубхона марказида ахборот кутубхоналарининг **KaDATA** дастурий таъминоти техник иловаси ташкил этилди. Тошкент шаҳар урология маркази поликлиника ва статсионарида бизнес-жараёнларни автоматлаштириш учун **MedDATA** дастурий таъминотини жорий қилиш кўзда тутилмоқда. Шу билан бирга, ихтисослаштирилган "**Sugurta DATA**" дастурий таъминоти ишлаб чиқилиб, "Alskom" суғурта компаниясида синаб кўрилмоқда. Давлат органларида электрон ҳужжат айланиш жараёнида фойдаланилаётган дастурий ечимларнинг 70 фоиздан ортиғини мамлакатимизда ишлаб чиқилган маҳсулотлар ташкил этмоқда.

Президентимиз томонидан қабул қилинган 2014 йил, 3 апрелдаги Иқтисодиётнинг реал секторига ахборот-коммуникация технологияларини жорий қилиш чора-тадбирлари тўғрисида"ги №ПҚ-2158-сонли қабул қилинди.

Қарорга мувофиқ, қуйидагилар иқтисодиётнинг реал секторига АКТни жорий қилишнинг муҳим вазифа ҳамда йўналишлари этиб белгиланди:

• хўжалик бирлашмалари ва ишлаб чиқариш корхоналарининг молиявий-хўжалик фаолиятида ҳисобот топшириш, электрон ҳужжат алмашинуви жараёнларига миллий дастурий таъминот маҳсулотларидан фойдаланган ҳолда ахборот-коммуникация технологияларини кенг жорий қилиш;

• замонавий АКТни жорий қилиш орқали маҳсулот ва хизматлар сифатини ошириш, ишлаб чиқаришда уларнинг таннархининг арзонлашишига эришиш, моддий-техник ресурслардан оқилона фойдаланиш, маҳсулот истеъмолчилари билан мунтазам алоқани йўлга қўйишга эришиш;

• Интернет орқали корхоналар томонидан ишлаб чиқарилаётган маҳсулотлар бозорини ривожлантириш, экспорт қилинишини таъминлаш мақсадида чора-тадбирлар ишлаб чиқиш;

• Ишлаб чиқариш корхоналарининг илмий-ишлаб чиқариш, олий таълим муассасалари, илмий-тадқиқот ва лойиҳалаштириш ташкилотлари билан

ўзаро алоқаларини йўлга қўйиш орқали ишлаб чиқаришда янги инновацияларни жорий қилиш;

• акциядорлик компанияларида, ассоциацияларда, йирик ташкилотларда АКТни жорий қилиш бўйича мутахассис ва кадрлар малакасини оширишга доир тизимни йўлга қўйиш.

Қарор билан 2014-2015 йилларда иқтисодиётнинг реал секторида ахборот технологияларини кенг жорий қилишга доир чора-тадбирлар, лойиҳалар рўйхати тасдиқланган. Мазкур қарорнинг амалга оширилиши АКТ маҳсулотларига бўлган талаб ва эҳтиёжнинг ошиши, лойиҳаларга юртимиз дастурий таъминот ишлаб чиқувчиларини жалб қилган ҳолда, миллий АКТни ривожлантиришга эришиш имконини яратади.

Юқорида баён этилган мулоҳазалардан келиб чиққан ҳода Битирув малакавий иш “Sudoku” ўйини дастурини android 4.3 платформаси учун ишлаб чиқиш” мавзуси танланди. Ушбу битирув малакавий ишда Android платформасидан фойдаланилди.

Битирув малакавий иш таркиби кириш, 4 бўлим, хулоса, фойдаланилган адабиётлар рўйхати ва иловалардан иборат. БМИнинг биринчи бўлимида Мобил қурилмалар учун android платформасида иловалар ишлаб чиқиш муҳити ҳақида батафсил маълумот берилган. Иккинчи бўлимда Android платформасида “sudoku” ўйинини яратиш технологиясига бағишланган.

БМИнинг сўнги учинчи бўлимида ҳаёт фаолияти хавфсизлигигига бағишланган бўлиб, унда иш жойларини ташкил этиш қилиш, меҳнат қобилияти ва унинг динамикаси тўғрисида фикр юритилган. БМИнинг асосий амалий натижалари ва хулосалар баён этилиб, дастур кодлари иловада келтирилган.

1. МОБИЛ ҚУРИЛМАЛАР УЧУН ANDROID ПЛАТФОРМАСИДА ИЛОВАЛАР ИШЛАБ ЧИҚИШ МУҲИТИ

1.1. Эркин ва очиқ кодли дастурий таъминотлар қиёсий таҳлили

Очиқ кодли дастурий таъминот – бу дастурий таъминотни ишлаб чиқаришнинг шундай усулики, унда дастурларнинг яратилаётган дастлабки коди очиқ, яъни барча кўриб чиқиши ва ўзгартириш киритиши учун очиқ бўлади.

Эркин дастурий таъминот – бу уни тарқатиш, ривожлантириш ва ўрганиш эркинлигидир.

Очиқ кодли дастурий таъминот ва эркин дастурий таъминот ғояларининг яқинлигини ҳисобга олиб, бу шарҳ доирасида улар “Эркин ва очиқ кодли дастурий таъминот” (ЭОКДТ) умумий тушунчаси остида кўрилади.

Ўзбекистон тараққиёти учун ЭОКДТларнинг аҳамияти қуйидаги омиллардан иборат:

- Миллий саноатни ва илмий потенциални ривожлантириш
- Импорт ўрнини босиш ва валюта маблағларини тежаш
- Хавфсизликни ва ишончлиликни таъминлаш
- Очиқ стандартлар ва ишлаб чиқарувчига қарам бўлмаслик
- Ақлий мулк химояси ва компьютер қароқчилиги билан кураш

- Маҳаллийлаштириш
- Халқаро алоқаларни ривожлантириш
- Рақобатни ривожланиши
- Миллий дастурий таъминот саноатини ривожланиши
- Кичик ва ўрта бизнес учун имкониятлар
- Турмуш даражасини ошириш
- Умумий таълим даражасини ошириш

Ўзбекистонда ЭОКДТни татбиқ қилиш, потенциални ривожланишига олиб келиши мумкин, шу жумладан, дастурий таъминотни ишлаб чиқишда ҳам ва истиқболли илмий тадқиқотларни ўтказишда ҳам. Ўзбекистон шундай ахборот тизимларини яратадики, қайсики, ўзи учун зарур, ўз навбатида, бу фанни, таълимни ва иқтисодиётни янада ривожланишига кўмак беради. Дастлабки кодларни борлиги ЭОКДТни тезкорлик билан истиқболли илмий ишланмаларни ва саноатни эҳтиёжларини қониқтириш учун мослаштиришга ёрдам беради. ЭОКДТдан фойдаланишнинг яна бир муҳим натижаларидан бири, пул оқимларини миллий бозор томонга қараб қайта тақсимланишидир. Шундай қилиб, ЭОКДТ қаерда ундан фойдаланилаётган бўлса, ахборот технологияларини ички бозорини ривожлантиришга ёрдам беради.

Кўп мамлакатлар учун АКТни ривожланишига асосий тўсиқ, ёпиқ тижорат дастурий таъминот учун юқори лицензион тўловлардир. Яқин келажакда бу масала Ўзбекистон учун ҳам, мамлакатни муаллифлик ҳуқуқини ҳимоя қилишнинг ***Берн конвенциясига*** қўшилиши муносабати билан, долзарб бўлиб қолади.

Бундай ёпиқ тижорат дастурий таъминотнинг асосий қисми Ўзбекистондан ташқарида ишлаб чиқарилганлиги сабабли, у Ўзбекистонга фақат импорт қилинади, истиқболда уни харид қилиш сезиларли қимматли валюта захираларини улушини ютиб юбориши мумкин. Шу билан бирга, бу пуллар бошқа муҳимроқ мақсадларга йўналтирилиши мумкин эди. ЭОКДТдан фойдаланиб Ўзбекистон, ўхшаш маҳсулотларни импорт қилишни

қисқартириб, мустақил равишда зарур бўлган дастурий таъминотни ўзи мослаштириши ва ишлаб чиқариши мумкин.

ЭОКДТнинг хавфсизликни таъминлашдаги афзаллиги дастлабки кодларнинг борлигидир, бу эса, ҳар кимсага – мустақил экспертлар ҳам шу жумладан – хавфсизликни таҳлил қилиш босқичларида ўз ҳиссасини қўшишга имкон беради. Тижорат дастурий таъминотининг бу йўналишдаги афзаллиги эса, лицензион тушумлар ҳисобидан хафвсизликни таъминлаш соҳасидаги тадқиқотларни молиялаш имконидадир.

2000 йилни ўзидаёқ, Франциянинг солиқ ва божхона муассасалари Red Hat Linux ОТга ўтганлар, буни биринчи галда хавфсизлик мулоҳазаларидан келиб чиқиб қилганлар.

Очиқ стандартлар фойдаланувчига, у ҳох ҳукумат бўлсин ёки ҳох хонадондаги фойдаланувчи бўлсин, дастурий пакетларни, платформаларни ва ишлаб чиқарувчиларни танлашдаги керак бўлган эпчилликни ва эркинликни беради. Ишлаб чиқарувчиларнинг хусусий стандартлари фойдаланувчини фақат бир ишлаб чиқарувчининг маҳсулотидан фойдаланишга мажбур қилади ва амалда истеъмолчини ишлаб чиқарувчи мурувватига қолдиради, у эса очиқ форматлар ва бошқа ишлаб чиқарувчиларнинг форматлари билан мослаштиришга мадад беришдан бош тортиши ёки бу ишни фойдаланувчига қўшимча ҳақ учун бажаришини таклиф қилади.

Ўзбекистонда 2005 йил мобайнида бунинг учун бир неча сезиларли ҳаракатлар қилинди: муаллифлик ҳуқуқини ҳимоя қилиш тўғрисидаги Берн конвенцияси имзоланди, Ўзбекистоннинг энг машҳур компаниялари дастурий таъминот ишлаб чиқарувчиларни муаллифлик ҳуқуқига риоя қилиш декларациясини имзоладилар. Бироқ, буларни ҳаммасини ҳаракатга келтириш талаб қилинади, хусусан мавжуд қароқчи дастурий таъминотни легаллаштириш, бу эса, экспертлар фикрига кўра, юзлаб миллионлардан то бир неча миллиард АҚШ доллари миқдорида лицензион тўловлар

қилинишига олиб келиши мумкин. Бундай тўловаларга қарши бошқа йўл тутиш – ЭОКДТни татбиқ қилиш мумкин.

ЭОКДТ ишлаб чиқиш ва ундан фойдаланиш модели бутун дунёдаги кўпчилик одамларни бирлаштиради. Шунинг учун, Ўзбекистонда ундан фойдаланиш маслакдошлар орасида тўғридан-тўғри халқаро алоқаларни ўрнатиш имконини беради, бу ўз навбатида албатта, халқаро тижорат лойиҳаларини ташкил қилишга олиб келади, бу эса Ўзбекистонни экспорт потенциалини энг самарали йўл билан амалга оширишга имкон беради.

ЭОКДТнинг энг кучли томонларидан биридир. Фойдаланувчилар ЭОКДТни ўз эҳтиёжларига кўра, ўзларининг ҳудудларининг маданий ўзига хос хусусиятларини ҳисобга олиб катта маблағ сарфламай созлаб олишлари мумкин. Бошланиш учун бир неча малакали техник мутахассислар ва таржимонлар бўлиши кифоя, улар ЭОКДТни асосий таркибий қисмларини минимал интерфейс даражасида маҳаллийлаштириши мумкин, чунки тўла маҳаллийлаштириш – содда бўлмаган масала ва кўп вақт талаб қилинади. Бундан кейин эса, ҳар ерда ҳозир фойдаланувчилар ёрдам берадилар. Бунинг учун уларнинг ҳар бири хабар, ёрдам сахифалари ва шу кабиларнинг таржимасини ўз қисмини таклиф қиладилар холос.

Сервер компьютери дастурлари

Электрон почта

- www.postfix.org
- http://www.exim.org/
- http://www.dovecot.org
- http://www.xmailserver.org/

Тармоқ хизматлари

- http://www.fs-security.com
- http://www.openswan.org/
- http://www.shorewall.net
- www.snort.org
- http://dspam.nuclearelephant.com/

- http://www.simonzone.com/software/guarddog/

Веб-серверлар

- http://www.apache.org
- http://tomcat.apache.org
- http://www.zope.org
- http://community.roxen.com

Маълумотлар базасини бошқариш тизимлари

- http://www.mysql.com
- http://www.postgresql.org
- http://www.firebirdsql.org
- www.openoffice.org
- http://www.csqldb.com
- http://www.ingres.com

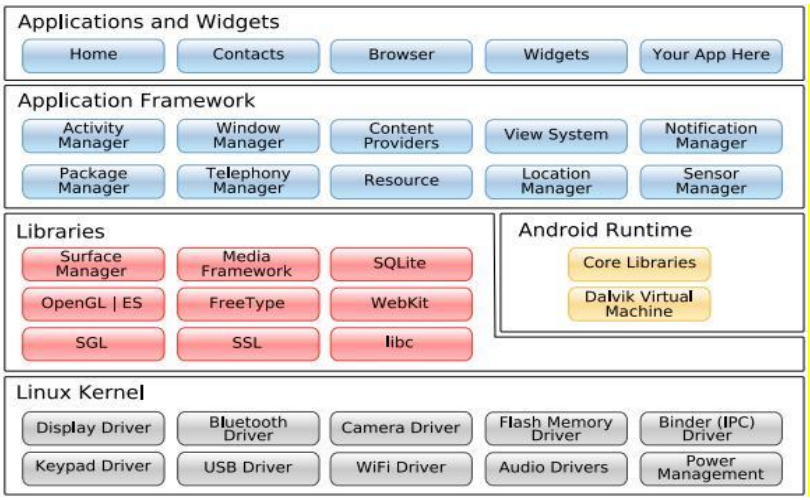
Бошқа сервер дастурлари

- http://plone.org
- http://www.postnuke.com
- http://www.list.org
- http://www.ejabberd.im
- http://www.jabber.org

1.2. Android операцион тизимига кириш

Andorid Google компанияси томонидан яратилган мобил операцион тизимдир. У смартфонлар ва мобил қурилмалар учун очиқ платформадир. Android тизим қуйидагилардан иборат:

- Andorid илова: Мобил фойдаланувчилар томонидан ишлатиладиган дастур. Android иловалар Java дастурлаш тилида яратилади.
- Framework илова: Framework Android иловани ва Dalvik виртуал машинасини, хамда библиотекаларни улайди.
- Linux ядро: ядро Linux мобил қурилмалар аппарат таъминотига хизмат кўрсатади.



1.2.1.-Расм. Android архитектураси

Java да ёзилган Andorid иловалар Dalvik виртуал машинаси учун байт – код сифатида компиляция қилинади. Framework илова ва Dalvik виртуал машинаси Andorid иловалар бажариш учун асосий тузилмани ташкил этади. Andorid иловалар Dalvik виртуал машинаси ёрдамида турли спецификацияли қурилмаларга эга смартфонларда бажарилади. Биринчи расмда Android платформаси тузилмаси кўрсатилган. Android илова ишлаб чиқиш учун қуйидаги дастурлар ўрнатилган бўлиши керак.

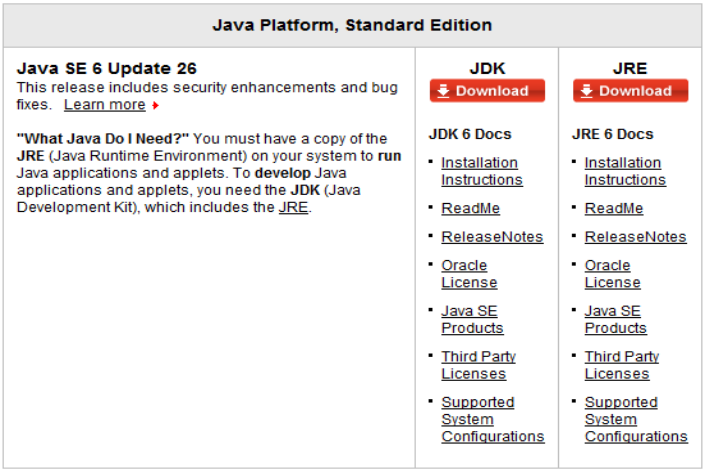
- Java Development Kit (JDK)
- Android Software Development Kit (SDK)
- Eclipse

Android илова Java тилида яратилиши учун аввал JDK ўрнатилиши, сўнгра Android SDK ўрнатилиши лозим. Eclipse Java учун IDE вакилидир. Биз Android иловалар яратиш учун Eclipse дан фойдаланамиз. Eclipse ўрнатгандан сўнг, Android SDK бошқариш учун ёрдамчи инструментлар ўрнатамиз.

1. JDK ўрнатиш

Сиз JDK ни Oracle қуйидаги сахифасидан юклаб олишингиз мумкин:

http://www.oracle.com/technetwork/java/javase/downloads/index.html



1.2.2. Oracle.com. сайти саҳифаси.

JRE (Java Runtime Envinronment) фақат Java дастурларни бажариш учун ишлатилади. Шундай қилиб биз JDK юклашимиз зарур. Юклашдан сўнг JDK ўрнатиш учун дастурни ишга туширинг.

2. Android SDK ўрнатиш

Сиз Android SDK ни қуйидаги илова орқали юклашингиз мумкин:

http://developer.android.com/sdk/index.html

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r12-windows.zip	36486190 bytes	8d6c104a34cd2577c5506c55d981aebf
	installer_r12-windows.exe (Recommended)	36531492 bytes	367f0ed4ecd70aefc290d1f7dcb578ab
Mac OS X (intel)	android-sdk_r12-mac_x86.zip	30231118 bytes	341544e4572b4b1afab123ab817086e7
Linux (i386)	android-sdk_r12-linux_x86.tgz	30034243 bytes	f8485275a8dee3d1929936ed538ee99a

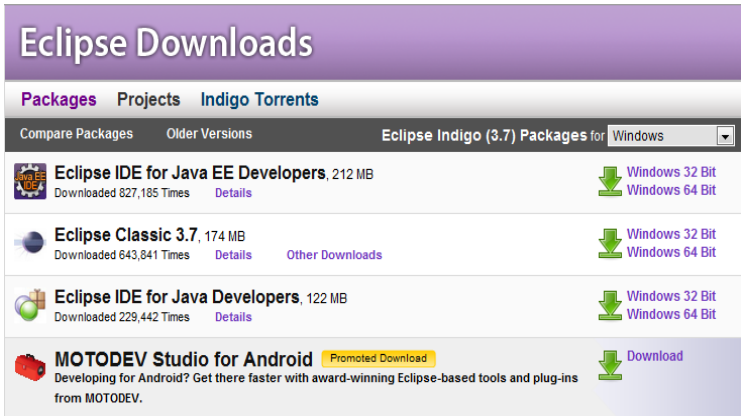
1.2.3.-расм.developer. android.com сайти саҳифаси

Операцион тизим турига қараб пакетни юклашингиз мумкин. Юклашдан сўнг Android SDK ни ўрнатишингиз мумкин.

3. Ecplise ўрнатиш

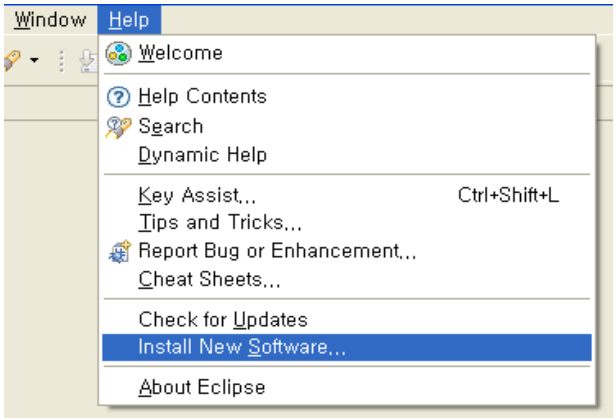
Сиз Eclipse ни қуйидаги илова орқали юклашингиз мумкин:

http://www.eclipse.org/downloads/



1.2.4.-расм Eclipse.org веб сайти саҳифаси.

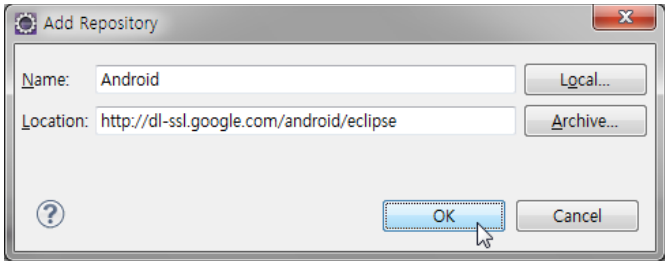
Eclipse турли версияларидан "Eclipse IDE for Java Developers" танланг. Eclipse ўрнатиш дастурига эга эмас. Сиз юкланган файлни очишингиз билан Eclipse дан фойдаланишингиз мумкин.



1.2.5-расм. Eclipse редатори ойнаси.

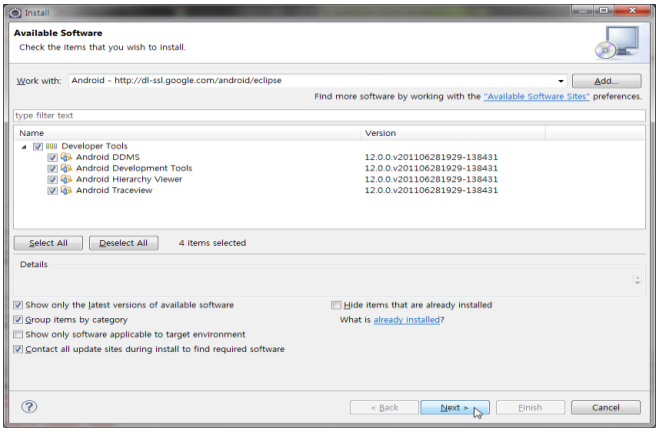
Eclipse ўрнатилгандан сўнг, Eclipse да Android иловаларни ишлаб чиқиш учун зарур бўлган Android Development Tools ўрнатишингиз мумкин. Олдин Eclipse ишга туширинг ва Help менюсидан "Install New Software" пунктини танланг. Сўнгра "Add" тугмасини босинг ва пайдо бўлган дарчада қуйидагиларни киритинг:

- Исм: ADT
- Жойлашув: http://dl-ssl.google.com/android/eclipse



1.2.6-расм Eclipse ойнаси.

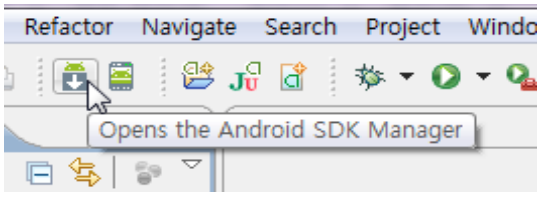
Ҳозирча дарчаларни яратиш учун қуйидаги инструментларни кўришингиз мумкин. Ҳамма инструментларни танланг. Бу Android иловалар учун Eclipse ўрнатиш охирги қадами.



1.2.7-расм. Eclipse ойнаси.

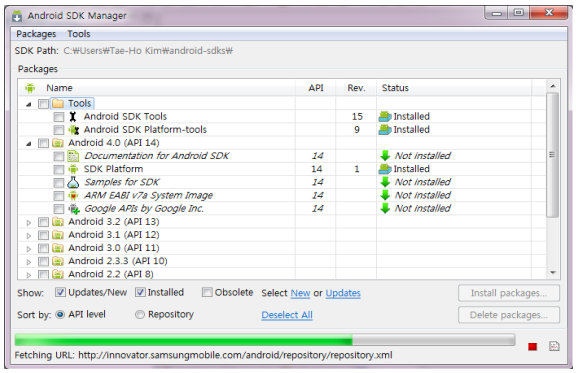
ADT ўрнатгандан сўнг Eclipse ни қайта ишга туширинг ва Android SDK мухим элементларини ўрнатинг.

4. Android SDK ўрнатиш



1.2.8.-расм. Eclipse ойнаси.

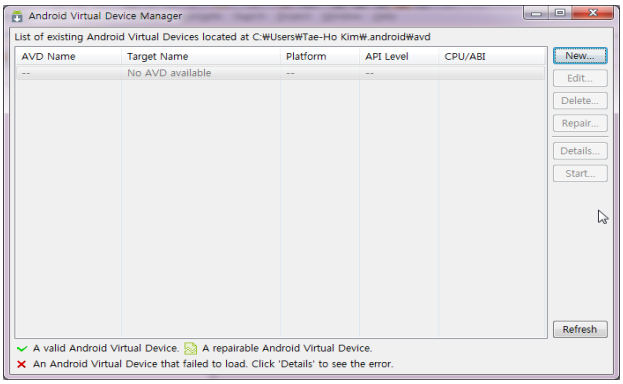
Юқоридаги расмдаги тугмани босинг ва Android SDK менеджерини қайта ишга туширинг.



1.2.9-расм. Eclipse ойнаси.

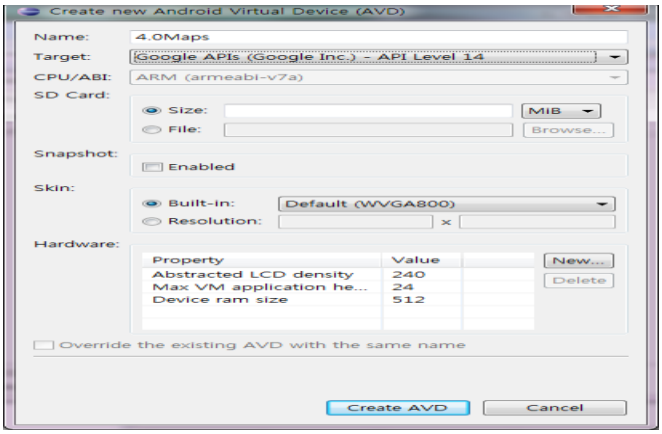
5. AVD ни созлаш

Оддий x86 ПК Android иловани бажара олмайди. Иловаларни ишга тушириш учун Android Virtual Device (AVD) виртуал муҳити зарур. Биз ҳамма ишлаб чиқариш воситаларини ўрнатгандан сўнг AVD ни созладик. AVD менеджерни Eclipse дан ишга тушириш мумкин.



1.2.10-расм. Eclipse ойнаси.

Ҳозирча AVD менеджерида AVD йўқ. AVD қўшиш учун, "New .." тугмаси босилади. AVD яратиш дарчаси қуйидаги кўринишга эга:



1.2.11-расм. Eclipse ойнаси.

"Name" майдонига AVD номини киритинг ва илова яратиш учун зарур платформани "Target" майдонида танланг. "SD Card" майдонида сақловчи хажми ва "Skin" майдонида экран катталигини танланг. Шундан сўнг "Create AVD" тугмасини босинг. Сиз "Start" тугмасини босиб яратилган AVD ни бошлашингиз мумкин.

1.2.12 - Расм. Android тизимини **AVD** экрани.

Сиз ишлаётган Android тизимни **AVD** экранида кўришингиз мумкин (2 расм). Android тизим **AVD** да ва мобил қурилмаларда бир хил.

Eclipse дан фойдаланиб Android илова яратиш. Бу бўлимда Eclipse дан фойдаланиб Android илова яратиш ёритилади. Android илова яратиш жараёни қуйидагилардан иборат:

1. Eclipse да Android лойиха яратиш,
2. Экран дизайни (**XML** ресурси орқали)
3. Java кодни яратиш
4. Лойихани бажариш (**AVD** орқали)
5. Илова натижасини текшириш. Агар у тўғри ишламаса, илова хатоларини тўғрилаш

Eclipse дан фойдаланиб Android илова яратиш жараёнида содда қадамлардан бошлаб, керак бўлса мураккаб қадамларга ўтилади.

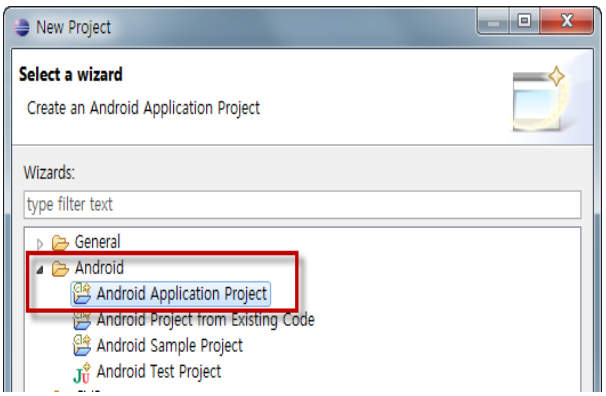
Илова яратамиз:

- Eclipse ни ишга туширинг.

Eclipse ни ўрнатилган каталогдан ишга туширинг.

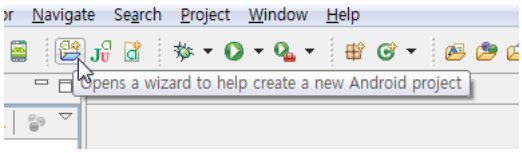
- " Select a workspace" дарчасидан лойиха папкасини текширинг ва ОК тугмасини босинг.
- Andorid лойиха яратинг.

Менюдан олдин File → New → Project сўнгра ‘Android Application Project’ танланг.



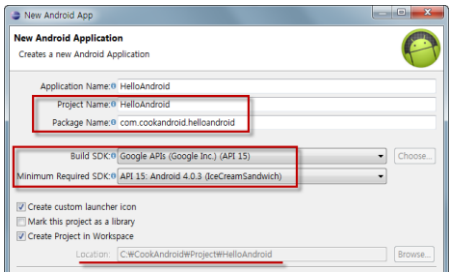
1.2.13-расм. Eclipse ойнаси.

Ёки ADT томонидан генерация қилинган тугма ёрдамида лойиха яратишингиз мумкин.



1.2.14-расм. Eclipse ойнаси.

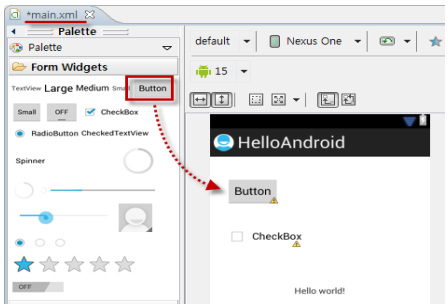
- ‘New Android Application’ дарчаси пайдо бўлганда лойиха номини "Project Name" майдонига киритинг. Бундан ташқари пакет номини киритиш ва ‘BuildSDK’ учун Android тизимини танлаш лозим. Биз Android 4.0.3 танладик.



1.2.15-расм. Eclipse ойнаси.

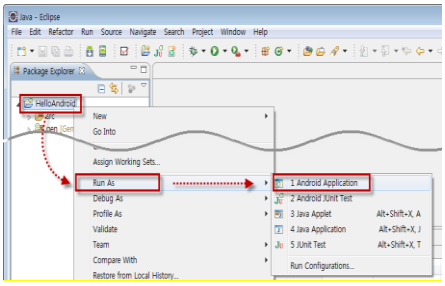
- Сиз "Next" тугмасини босиб кейинги қадамга ўтишингиз мумкин.
- Бу Android илова учун лойиха қуриш қадамлари. Энди илова дизайнини тахрирлаш лозим. Агар сиз Android иловалар учун асосий макетни кўрмаётган бўлсангиз Eclipse чап панелида жойлашган ‘res/layout’‘PackageExplorer’ да макет файлини танланг.

- Виджетлар жойлашуви XML макет файлида мос код фрагментини генерация қилади. Энди пастдаги расмда кўрсатилганидек тугмани панелдан экранга кўчиринг.



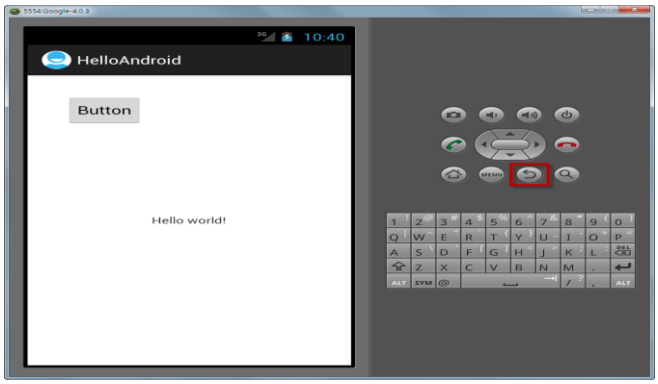
1.2.16-расм. XML макет файли.

- Биз Android иловалар учун Java код ёзишимиз мумкин. ‘Package Explorer’ да олдинги қадамларда киритилган пакет номига босинг. Бу холда дастур учун Java код кўришингиз мумкин.
- Лойихани ишга тушириш. Пакетлар лойихани танланг ва контекстли менюни чақириш учун сичқонча ўнг клавишини босинг. Сиз Run As → Android application танлашингиз мумкин.



1.2.17-расм. XML макет файли.

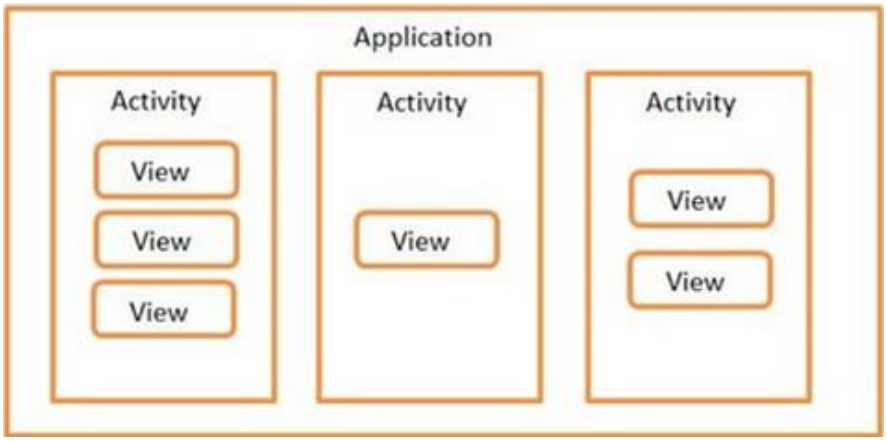
AVD лойихани бажаришни бошлайди. Маълум вақтдан сўнг лойихани AVD да кўриш мумкин.



1.2.18-расм. AVD да кўриш ойнаси.

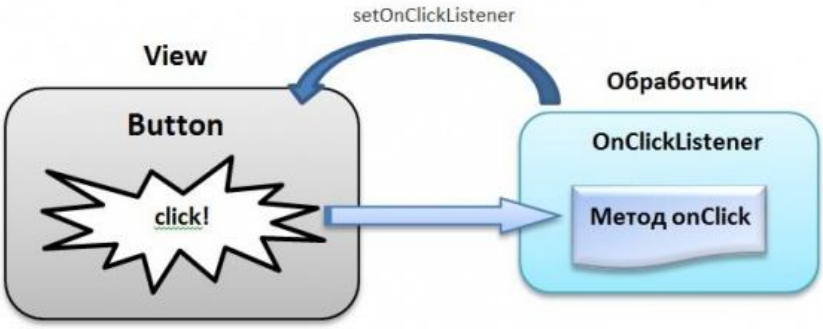
1.3 Andorid операцион тизимининг хусусиятлари

Бир Activity ойнанинг барча хусусиятлари билан ишлай олади. Мисол учун C# да форм1.cc, Андроидда эса Activity деб номланади. Activity тушунчасини батафсил тушуниб олишимиз учун Почта клиент дастурини мисол қиламиз. Биринчи Activity имизда(ойнамизда) – хатлар рўйхати, иккинчи Activity имизда хатларни ўқиш яъни кўриш, учунчи Activity имизда почтамизнинг созлашлари. Activity имиз таркибида турли хил компонентлардан ташкил топади ва буни биз View номи билан кўрсатиб кетамиз.



1.3.1.-Расм. View компонеталар.

Яни бу ерда дастуримиз **Application** деб номланган ва унинг таркибида 3 та **activity** ташкил топганлиги юқоридаги расмда ифода этган.



1.3.2.-Расм**.** Ҳодисалар билан ишлаш.

APPLICATION NAME: BUTTON_EVENT
PROJECT NAME: BUTTON_EVENT
PACKAGE NAME: UZ.PROJECTS.BUTTON_EVENT
TARGET SDK: API 17 ANDROID 4.2(JELLY BEAN)
CREATE ACTIVITY: MAINACTIVITY
ACTIVITY_MAIN.XML:

Бизда текист учун **TextView** ва **OK**, **Cancel Button**лари мавжуд. **App**имизнинг ишлаши ишлаши қуйдагича: **OK** тугмаси босилганда **TextView**га **OK** тугмасини босдингиз ва **Cancel** тугмаси босилганда сиз Cancel тугмасини босдингиз деб чиқарамиз. **MainActivity.java** классимизни очамиз ва onCreate методимизга ўтиб view элементлари билан ишлаш учун уларнинг обектини ҳосил қилиб view элементларини идси билан боғлаймиз.

```
public class MainActivity extends Activity {
TextView tvOut;
Button btnOk;
Button btnCancel;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
```

```
// найдем View-элементы
tvOut = (TextView) findViewById(R.id.tvOut);
btnOk = (Button) findViewById(R.id.btnOk);
btnCancel = (Button) findViewById(R.id.btnCancel);
}
}
```

Энди эса тест қиб кўрамиз.

1.3.3-расм. AVD да кўри ш ойнаси.

1.4. Sudoku ўйини қоидалари ва математик ечимини топиш усуллари

Судоку (яп. 数独 *су:доку* деб талаффуз этилади ва “бир ҳил сон” маъносини англатади) - машҳур сонли бошқотирма ҳисобланади. Японча дан таржима қилганда “су:” -“рақам”, “доку” эса – “бўлак турувчи” деб таржима этилади. Баъзан **судоку**ни “сеҳирли квадрат” деб ҳам аташади бироқ бу унчалик ҳам тўғри эмас. Чунки судоку 9-тартибли лотинча квадратдир. Судоку дунёнинг кўпчилик журнал ва газеталарида босиб чиқарилади ва уни ечиш энг машҳур дам олиш усулларидандир.

Ўйин қоидалари. Ўйин майдони 9х9 ҳажмли квадратда ташкил топган бўлиб, у ҳар томонидан 3 та катакка бўлинган ва умумий катаклар сони 81тани ҳосил қилади. Бу катакларда ўйин бошида бир неча сонлар туради (1 дан 9 гача) ва уларни йўл кўрсатувчилар деб аташади. Ўйинчидан бўш

катакларни 1 дан 9 гача бўлган сонлар билан шундай тўлдириш талаб этиладики, натижада ҳар бир катакда, ҳар бир устунда ва ҳар бир кичик 3х3 квадратда ҳар бир сон фақат бир марта учрасин.

Судоку мураккаблиги унинг олдиндан тўлдирилган катакларига боғлиқ эмас, бу уни ечиш учун қўлланиладиган усулга боғлиқ холос. Баъзи бошқатирмаларни бир неча минут ичида ечиш мумкин, баъзиларини ечишга эса соатлаб вақт сарфланиши мумкин.

Ечимни топиш усуллари. Энг яхши ечимни топиш усули – бу номзод-сонларни катак чап бурчак тепасига ёзиш, кейин бу катакдан ўйин қоидасига кўра мумкин бўлмаган сонлар ўчирилади. Шундан кейин бу катакни эгаллаши мумкин бўлган сонларни кўриш мумкин бўлади. Судокуни шошмасдан ўйнаш тавсия этилади, чунки кишига завқ берувчи ўйин ҳисобланади. Тўғри тузилган бошқотирма фақат битта ечимга эга бўлади.

Аввало кўпроқ тўлдирилган қаторларга, устунларга ва 3х3 блокларга қараш керак. Чунки қаерда вариантлар кам бўлса, ўша жойда ечиш осонроқ бўлади. Катакни тўлдирганда устун, қатор ва 3х3 блокга эътибор қаратиш керак. Барча қолган 8 сон иккиланмайдими текшириш керак.

Судоку ўйинида 3х3 блокда бир неча очиқ катак қолганда ва бу сонга фақат битта катак мос келса, у ҳолда айнан шу сонни катакка ёзиш керак. Катакларни тўлдиришдан олдин ёзилаётган соннинг шу устун, шу сатр ёки 3х3 блокнинг бошқа катагида учрамаслигига ишонч ҳосил қилиш керак.

Бир устун, сатр ёки 3х3 блокда 3 та исталган номзод-сонлар {1,2; 1,2; 1,3} бўлганда учинчи катак сони 3 бўлиши шарт. Чунки, агар бу сон 1 бўлганда биринчи икки катакдан бирида 2 сони бўлган бўлар эди, бошқасида эса ҳеч нима бўлмаган бўлар эди – бундай эса бўлиши мумкин эмас. Чунки барча катаклар тўлдирилиши керак.

Бошқатирмани ечишнинг тезлигини оширишнинг икката стратегияси мавжуд. Буни қуйидаги мисолда кўриб чиқамиз.

Мисол:

2					5	8	9	
	6			9		7	1	
9			7					4
				6			5	8
		6				4		
5	9			3				
3					8			7
	4	9		2			8	
	8	5	1					9

1.4.1-расм. “SUDOKU” квадратик ойнаси.

9 сони 6 та 3х3 блокларда 6 марта учраяпти. Шу сабабли 9 сонини иккиланмасдан марказий пастки 3х3 блокнинг чап юқори бурчагига қўйиш мумкин. Шунингдек, тепадаги иккинчи ўнг 3х3 блокнинг бири нчи қатори биринчи катагига ҳам қўйиш мумкин. 9 сонини марказий 3х3 блокнинг фақатгина иккинчи қатор учинчи катагига қўйиш мумкин.

Мисол:

	4			8				7
		3	7		2			9
			6	3	5			
	7					2		
9	1						3	8
		5					7	
			1	2	7			
8			4		3	1		
2				5			6	

1.4.2-расм. “SUDOKU” квадратик ойнаси.

Юқори қатор 3х3 блоклар ўртаси ва пастки қатор 3х3 блоклар ўртаси деярли тўлдирилган. Юқори блок ўртасида ечмлмаган учта сон бор -1, 4 ва 9. Бундай ҳолатни таҳлил қилиб 4 сонини блок марказига, 1 сонини ўнг юқори

бурчакка, 9 сонини эса чап юқори бурчакка ёзиш мумкин. Бунга ўхшаш пастки марказий 3х3 блок билан ҳам шундай қилиш мумкин. Унда 6, 8 ва 9 сонлари йўқ. Катаклар кетма-кет тўлдирилади: 6 сонини марказга, 9 сонини пастки ўнг бурчакка, 8 сонини эса пастки чап бурчакка ёзамиз.

1.5. Масаланинг қўйилиши.

“Sudoku” ўйини дастурини android 4.3 платформаси учун ишлаб чиқиш масаласи қўйилди. Бу дастурни ҳозирги вақтда энг тез ривожланаётган android платформасида яратишдан иборат.

Ушбу масалани ечиш учун қуйидаги вазифаларни амалга ошириш лозим:

- “Sudoku” ўйинини яратишда android платформасининг компоненталарини ўрганиб чиқиш;
- Android платформасида илова яратиш жараёнларини ўрганиш
- “Sudoku” ўйинини алгоритми ва дастурини ишлаб чиқиш;
- “Sudoku” ўйини интерфейсини ишлаб чиқиш;
- “Sudoku” ўйинини мобил қурилмаларда ишлатиш учун фойдаланувчи йўриқномасини ишлаб чиқиш.

2. ANDROIDДА ПЛАТФОРМАСИДА “SUDOKU” ЎЙИНИ МОБИЛ ИЛОВАСИНИ ЯРАТИШ ТЕХНОЛОГИЯСИ

2.1. Android платформаси архитектураси ва компоненталари

Android турли компонентларга эга. Баъзи мантиқий компонентлар платформалар билан боғлиқ. Баъзилари физик файлларда ажратилади. Қуйида Android компонентлари хақида маълумот келтирилган.

Ресурслар - Android иловадаги бирламчи кодда(ёки бажарилувчи) ташқари файллар. Улар тасвирлар, товуш ва жойлашувлар каби файлларни ажратилган форматларда сақлаш учун ишлатилади. Дастурчилар одатда ресурсларни турига қараб турли маълумотномаларда сақлайдилар. Қуйида ресурслар учун маълумотномалар намуналари келтирилган.

• res/anim: XML да анимация

• res/drawable: Тасвирлар

• res/layout: Экран жойлашуви

• res/values: сатрлар, тўпламлар ва бошқа статик қийматлар

• res/xml: XML да бошқа маълумотлар

• res/raw: Аниқланмаган ўзга маълумотлар

Биз бу файлларни Android лойихасида Eclipse ёрдамида кўрамиз.

Ресурсларга мурожаат (Referring resources). Биз ресурслардан Android лойихага файллар қўшилгандан сўнг фойдаланишимиз мумкин. Ресурсларга мурожаат қилишнинг икки усули мавжуд: ресурс файли орқали ва берилган файлда.

Биз ресурс файлида бошқа ресурсларга мурожаат қилиш учун "@" белгидан фойдаланамиз:

- @ ресурс тури / имя ресурс номи

Қуйида бошқа ресурсларга мурожаат мисоллари келтирилган.

- @string/hello
- @layout/main
- @drawable/icon

- @android:id/text1

Биз кодда ресурсга мурожаат қилганда ADT томонидан автоматик яратилган объект R дан фойдаланишимиз мумкин (gen/director да жойлашган):

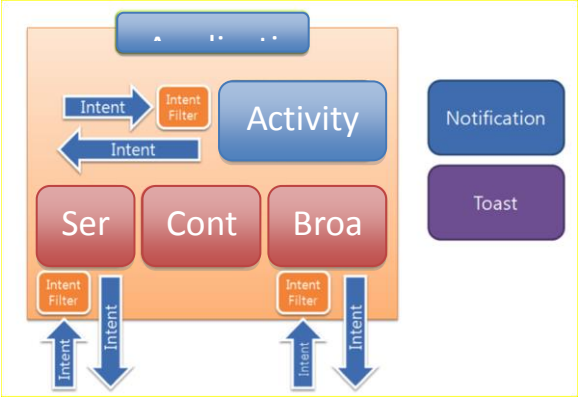
➢ R. (ресурс тури).(ресурс номи)

Қуйида ресурсларга мурожаат мисоллари келтирилган.

- R.string.hello
- R.layout.main
- R.drawable.icon
- R.raw.sound
- android.R.id.text1

XML файллари в layout/directory да являются энг кенг қўлланадиган ресурслардир.

Фаолият (Activity). Android илова бир неча компонентлардан иборат. Қуйидаги расмда компонентлар орасидаги муносабат кўрсатилган.



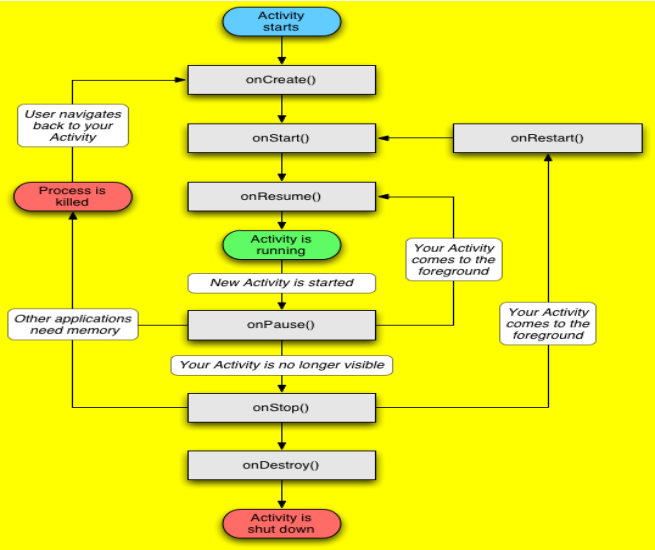
2.1.1-расм. Компонентлар орасидаги муносабат кўрсатилган.

Фаолият илова экранига мос келади. Шундай қилиб, фаолият Android иловада фойдаланувчи билан ўзаро мулоқот қилиши мумкин. Ягона фаолият дарча яратади. Иловаларда кўп қайтарилувчи фаолиятлар бўлиши мумкин. Android ядроси бу фаолият. Фаолият уч статусга эга:

- Актив: Фаолият Android қурилма экранида ишлайди. У фойдаланувчилар билан ўзаро мулоқот қилиши мумкин.
- Тўхтатиб турилган: Фаолият Android қурилма экранида кўринади, лекин фойдаланувчилар билан ўзаро мулоқот қилолмайди.
- Тўхтатилган: Фаолият экранда кўринмайди. (Бошқа Фаолият ёки фойдаланувчи экрани уни тўсиши мумкин)

Агар фаолият тўхтатиб турилган бўлса, Android қурилмада хотира етарли бўлмаса бутунлай тўхтатиши мумкин.

Фаолият берилган коддан ва экран қуриш учун жойлашувдан иборат. Жойлашув – экранни лойихалаш учун ресурс. Дастурчилар Eclipse ёрдамида жойлашувни лойихалаштириши мумкин: таърифлаш ва виджетларни тайёрлаш. Фаолият бажарилиш бўйича хаёт циклига эга. Қуйидаги 2.1.1-расмда хаёт цикли кўрсатилган.



2.1.2.-Расм. Ҳаёт цикли.

Расмда кўрсатилганидек фаолият яратилганда, ички функциялар бажарилиш холатига қараб чақирилади. Масалан, Android тизимли чақириғи **onCreate(),** фаолият яратилганда чақирилади. Усул **onCreate()** кўп чақирилади, чунки инициализация фазасида бажарилиши лозим бўлган код яратамиз.

Усуллар хақида батафсил маълумот:

- onCreate():

- фаолият яратилаётганда чақирилади

- инициализацияни қайта ишлайди

- UI сақловчи боғламни олади ва фаолият қандай тугаганини кўрсатади

- **onStart()**

- фаолият экранда кўринганда чақирилади

- фойдаланувчи билан ўзаро таъсир кўрсатолмайди

- **onResume()**

- фаолият актив бўлганда чақирилади

- фойдаланувчи билан ўзаро таъсир қилиши мумкин

- **onPause()**

- фаолият фокус йўқотганда чақирилади

- фаолият экранда пайдо бўлади, лекин фойдаланучи билан ўзаро таъсир қилолмайди

- **onStop()**

- фаолият экранда пайдо бўлмаганда чақирилади

- агар Android да хотира етарли бўлмаса фаолият тўхтатилиши мумкин

- **onRestart()**

- фаолият қайта ишга тушганда чақирилади

- **onStar()** ва **onResume()**

- **onDestroy()**

- фаолият бузилганда чақирилади.

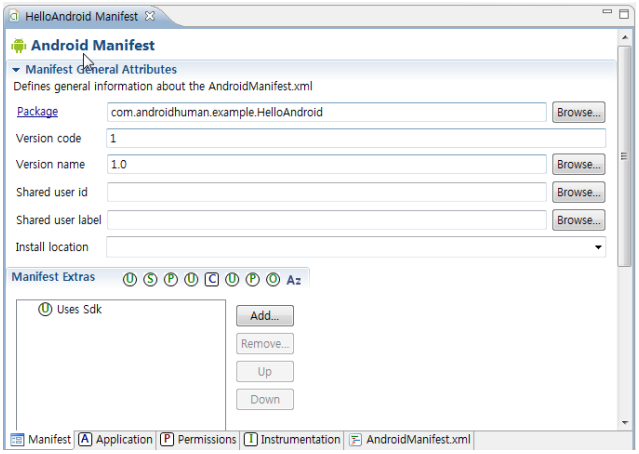
Декларация (Manifest). Декларация Android илова тузилмалари хақида ахборотга эга. Бу ахборот ўз ичига илова номи, компонентлар, рухсатлар ва хоказоларни олади.

Декларация қуйидаги маълумотларга эга:

- Илова пакети номи.
- Иловада компонентлар.
- Илова операцияси учун рухсат.

- Илова бажариш учун SDK Минимал версияси (API даражаси).
- Иловада ишлатилган библиотекалар.

Декларацияни кўриш учун навигацион бар чап панелида ‘Android Manifest.xml’ босиш мумкин. Кейинги расмда Eclipse декларация учун беш табуляцияни кўрсатади.



2.1.3-расм. Eclipse декларация ойнаси.

Декларация учун қуйидаги тавсифлар мавжуд:

- Декларация: Илова учун умумий ахборот.
- Илова: Илова атрибутлари, компоненталар ва библиотекалар тўғрисида ахборот
- Рухсат: Рухсат иловада талаб қилинади
- AnroidManifest.xml: XML да Декларация. Биз XML ни тўғридан тўғри ўзгартиришимиз мумкин.

Истак (Intent). Истак Android илова компонентлари орасида алоқа қилиш учун ишлатилади: Истак икки турга бўлинади: ошкор ва ошкор эмас. Ошкор истак хабар юбориш учун мақсадли компонент ошкор кўрсатилганда ишлатилади. Ошкор бўлмаган истак ўзининг мақсадини конкрет аниқламайди. Бу мақсад махсус атрибутидир. Шундай қилиб Android тизимларга фаолият атрибутларини регистрация қилиш учун ‘Intent filter’,

ўрнатамиз. Истак мақсад компонентаси номини ва белгиланган вақтда бажарилувчи фаолиятни ўз ичига олиши мумкин.

Хизмат кўрсатиш (Service). Хизматлар – фаолиятлардан фарқли фон режимида ишлаб, экранда кўринмайдиган компонентлардир. Хизматлар ноактив режимда бажарилади. Масалан мусиқа чалиш. Фойдаланувчилар интернет-серфинг каби баъзи масалаларни ечаётганда, биз мусиқа эшиттириш имкониятига эга бўлишимиз керак. Шундай қилиб хизматлар кўрсатишдан мустақил равишда масалаларни ечиш учун ишлатилади.

2.2. Android платформасида иловаларни ишлаб чиқиш учун java асослари

Java дастурлаш тили **Sun Microsystems** компанияси томонидан 1991 йилда ишлаб чиқилган. Java кўп сохаларда 1995 ва 1997 йилларда JDK 1.0 ва JDK 1.1 ишлаб чиқилгандан сўнг қўллана бошланди. Java объектга - йўналтирилган тил бўлиб, содда синтаксисга эга. Аммо Java компиляторлари Java виртуал машинасида бажарилувчи байт – генерация қилади. Оддий компиляторлар эса конкрет платформа учун иккилик код генерация қилади. Байт-код ихтиёрий платформада бажарилиши мумкин, агар платформа Java виртуал машинасига эга бўлса.

Java дастурларни яратиш. Java кодлар одатда .java кенгайтмасига эга ва Java компилятор берилган кодни .class кенгайтмали байт-кодга айлантиради. Биз байт-кодни java номли виртуал машина орқали бажаришимиз мумкин. Содда мисол тариқасида экранга "Hello world!". чиқарувчи кодни кўрамиз. Қуйидаги кодни териб hello.java сифатида сақланг:

```
public class hello {
    public static void main(String args[]) {
            System.out.println (“Hello world!”);
    }
}
```

Файлни "javac" яъни , Java компилятор ёрдамида компиляция қилиш.

$ javac hello.java

Java-компилятор “hello.class.” файлни генерация қилади. Энди файлни "java" орқали бажаришга беринг ва қуйидаги сатрни кўришингиз мумкин.

$ java hello

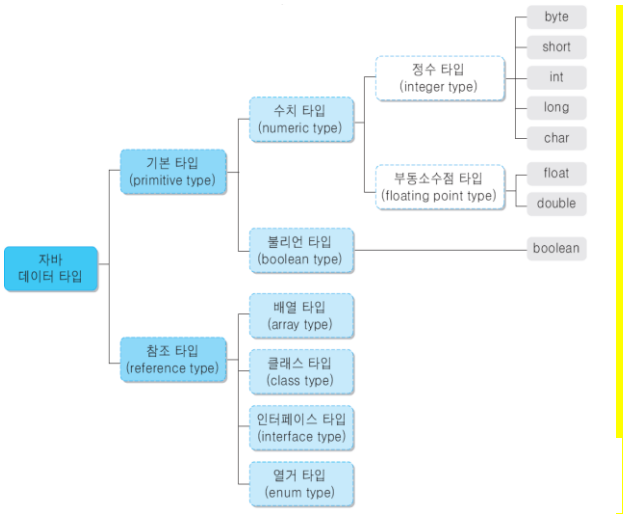
Hello world!

Java да ўзгарувчилар. Ўзгарувчи - координата сақлаш учун жой. Сиз ўзгарувчини турини кўрсатган холда таърифлашингиз лозим. Қуйидаги мисолда икки ўзгарувчи: ‘sum’ ва ‘msg’ таърифи келтирилган.

int sum = 3 + 4;

String msg = “Hi!”;

Ўзгарувчи ‘sum’ ‘int’ яъни бутун тип сифатида таърифланган, ва ‘msg’ ўзгарувчи ‘string’, яъни белгилар сатри сифатида таърифланган. Ва икки ўзгарувчи мос равишда И две переменные инициализируются как 7 ва "Hi!" қийматлари билан инициализация қилинган. Қуйида Java қўлловчи маълумотлар турлари графиги келтирилган.



2.2.1-расм. “Java” тилида қўлловчи маълумотлар турлари графиги.

Java да назорат операторлари. 1) Оператор ‘if’ асосий бошқарув операторларидан бири. Агар шарт бажарилса ‘then’ блоки бажарилади. Акс холда ‘else’ блоки бажарилади. Шаблон ‘if (condition) { (then block) } else { (else block) }.’ . Қуйида мисол келтирилган.

```
if (a > 3) {
    b = 4;
    c += 1;
} else {
    b  = 5;
```

```
}
```

Юқоридаги мисолда агар 'a' ўзгарувчи 3 дан катта бўлса, ‘b’ ўзгарувчига 4 қиймат берилади ва 'c' ўзгарувчи биттага оширилади. Акс холда 'b' ўзгарувчига 5 қиймат берилади.

2) Оператор‘while’ цикл операторлари вакилидир. Токи шарт бажарилса, цикл танасидаги операторлар бажарилади.

```
while (a < 10) {
    a = a + 1;
}
```

То 'a' ўзгарувчи 10 дан кичик бўлса, ўзгарувчи қиймати 1 га ошади.

3) Оператор ‘for' хам цикл операторидир. Java тилида икки хил for оператори қўлланади. Улардан бири Один из них является С- тилидаги каби бўлиб инициализация, шарт ва қиймат оширишдан иборат. Иккинчиси массивлар учун қўлланади. У хар бир итерация учун массивдан навбатдаги элемент ажратади.

```
int a[6] = [3,4,2,6,7,1];
for (int i = 0; i< 6; i += 1) {
    System.out.println(“> “ + a[i]);
}
for (int i : a) {
    System.out.println(“> “ + i);
}
```

Юқоридаги мисолда for оператори бир хил ишни икки усулда бажаради. Биринчи усулда for оператори 'i' ўзгарувчини 0 дан до 5 гача оширади ва хар бир итерацияда a[i] ни чиқаради. Иккинчи усулда 'a' массивдан элементлар танланади ва "i" ўзгарувчига қиймат сифатида берилади.

Java да объектлар. Java является объектга- йўналтирилган дастурлаш тилидир. Яъни биз объектлар асосида дастур яратамиз. Объект турли типдаги ўзгарувчилар ва уларни қайта ишловчи функциялар («усул») билан боғланган. Аниқ таърифланган объект маълумотлар туридир. Объектдан

фойдаланиш учун уни хотирада қуриш лозим. Биз хотирада қурилган объектларни "instantces” сифатида чақирамиз. Шундай қилиб дастурчи объект яратиш учун синф таърифини келтиради, нусхасини яратади ва ўзгарувчида сақлайди.

Синфлар таърифланганда ‘class’ калит сўзидан фойдаланилади

```
class Point {
    int x, y;

    Point(int x, int y) {
        this.x = x;
        this.y = y;
    }

    Void moveTo(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
```

'Point' номли синф икки ўзгарувчига эга: "x" ва "y". Синф билан бир хил номга эга ўзгарувчи бу ‘constructor’, ва нусха яратилганда чақирилади. Бу синф ‘moveTo’ номли усулга эга. Биз усулни нусха сақланувчи ўзгарувчи орқали чақирамиз.

Биз нусха яратганда ‘new’ калит сўзидан фойдаланамиз.

```
Point p = new Point(11,12);
p.moveTo(20,20);
```

Калит ‘new’ сўзи конструктор билан ишлатилади ва нусха яратади. Яратилган нусха ‘p’ ўзгарувчида сақланади. Ўзгарувчи ‘p’ типи ‘Point’, чунки ‘Point’ типи нусхасини сақлаши лозим. Биз 'p' ўзгарувчи билан ‘moveTo’ га мурожаат қиляпмиз. Шунга эътибор бериш лозимки ўзгарувчилар ёки синфлар усуллари ёки пакетларни кўрсатиш учун ‘. (period)’ ишлатамиз.

2.3. SUDOKU ўйинини Android платформасида ишлаб чиқиш босқичлари

Android бу ўзига хос операцион тизим. Дастурни ишлаб чиқувчи яхши натижаларга эришиш учун унинг барча нгозик ўзига хос жихатларини билиши шарт. Баъзибир қийинчиликлар мавжуд бўлим, дастурлашда уларга ахамият бериш керак. Уларни қисқача келтириб ўтамиз:

1) Дастурларни ўрнатишда шу дастур кенглигидан 2 маротаба (ёки 4 маротаба) катта жой талаб қилади.

2) Ўрнатилган флешка билан ишлаш тезлиги 10 маротаба тушиб кетади, агар ундаги кенглик камайса.

3) Ҳар бир жараён 16 Мб (баъзида 24 Мб) ҳажмда опретав хотирани банд қилиши мумкин.

Android асосида унумли дастурларни ишлаб чиқиш тартиби. Android асосида унумли дастурларни ишлаб чиқишга бир қанча тавсиялар бор. Уларни яна Effective Java — Joshua Bloch ва .Net Compact Framework — И.Салмре платформаларида мобил қурилмалари учун дастурлаш тўғрисидаги китоб ёрдамида кенгайтириш мумкин. **Android** асосида унумли дастурларни ишлаб чиқишнинг асосий тартибларини келтириб ўтамиз.

Стратегик:

1) Манбаларни тежаш керак.

2) Ҳолатларга учун дарҳол реакция кўрсатиш керак ва фойдаланувчи билан тескари алоқа ўрнатган ҳолда.

3) Дастурнинг унумдорлиги бош мақсад. Дастурлаш жараёнида унинг унумдорлигини дойимо оптимал йўлини бажариш керак уни кейинга қодлирмасдан.

4) Бажарилиш вақтини ўлчаш керак, протоколлаштириш ва дастурнинг ишлашини тахлиллаш, кодларнинг кичик қисмларини, объектнинг хаётилик вақтини, хоирадан жой ажратилишини. Ўлчанмайдиган нарсаларни оптималлаштириб бўлмайди.

Тактикага асосан:

1) Ортиқча объектларни тузишдан қочинг.

2) Имконият даражасида усулларни статик кўринишида бўлсин.

3) Кенгликларга тўғри боғланишлар қилинг даллол усули билн фойдаланмасдан.

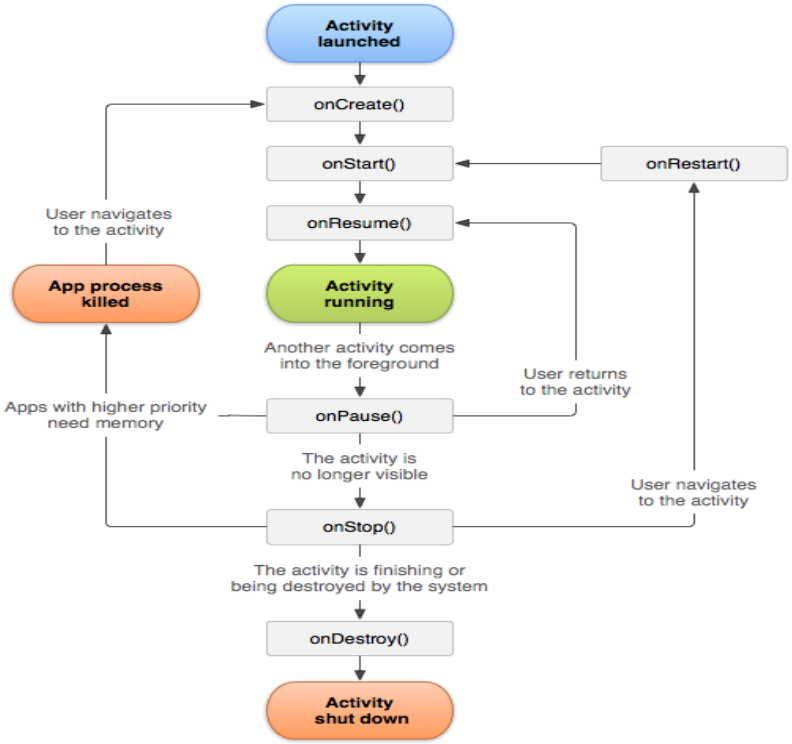
4) Константалар учун **static final** ларни ишлатинг.

5) Тўлиқ турдаги оддий ўзгарувчилар учун **enum** ни кўп ишлатманг.

Android нинг муҳим жиҳатлари. Android – Linuxга асосланган. Ядро билан дастур орасида API қатлами ва кутубхона билан тўлдирилган код қатлами бор. Дастур Java (Dalvik Virtual Machine) виртуал машинасида бажарилади.

Android да бир нечта дастурларни ишга тушириш мумкин. Лекин уларнинг фақат биттаси, асосийси экранни эгаллайди. Айни пайтдаги дастур ёрдамида аввалгисини ёки янгисини ишга тушириш мумкин. Бу худди браузернинг тарихи билан ишлашга ўхшайди.

Фойдаланувчининг ҳар бир экрани **Activity коди** синфи билан тақдим этилган. Турли **Activity** лар жараёнларда сақланади. **Activity** ҳатто жараёнлардан ҳам кўпроқ яшаши мумкин. **Activity** тўҳтатилиш ва барча керакли ахборотлар билан сақланган ҳолда қайта ишга туширилиши мумкин.



2.3.1-расм. Ҳаёт цикли.

Android махсус **Intent** ишлаш жараёнига асосланган механизмни. Амални бажариш керак бўлганда (қўнғироқ қилиш, хат жўнатиш, дарчани кўрсатиш)да **Intent** чақирилинади. Шунингдек **Android** да **Linux** демонлари каби сервислар бўлиб, улар фон ортида керакли амалларни бажаришади (масалан, қўшиқни тинглаш). Дастурлар аро маълумот алмашишда **Content providers** ишлатилинади.

Судоку ўйининг лойиҳасини кўриб чиқамиз. Ишлаши учун **Android SDK** ва **Eclipse** керак. Қандай ўрнатиш ва бошлаш тўғрисида <u>қуйида</u> ёзилган. Eclipse лойиҳасни юклаш учун қуйидаги кетма-кетликни бажариш керак:

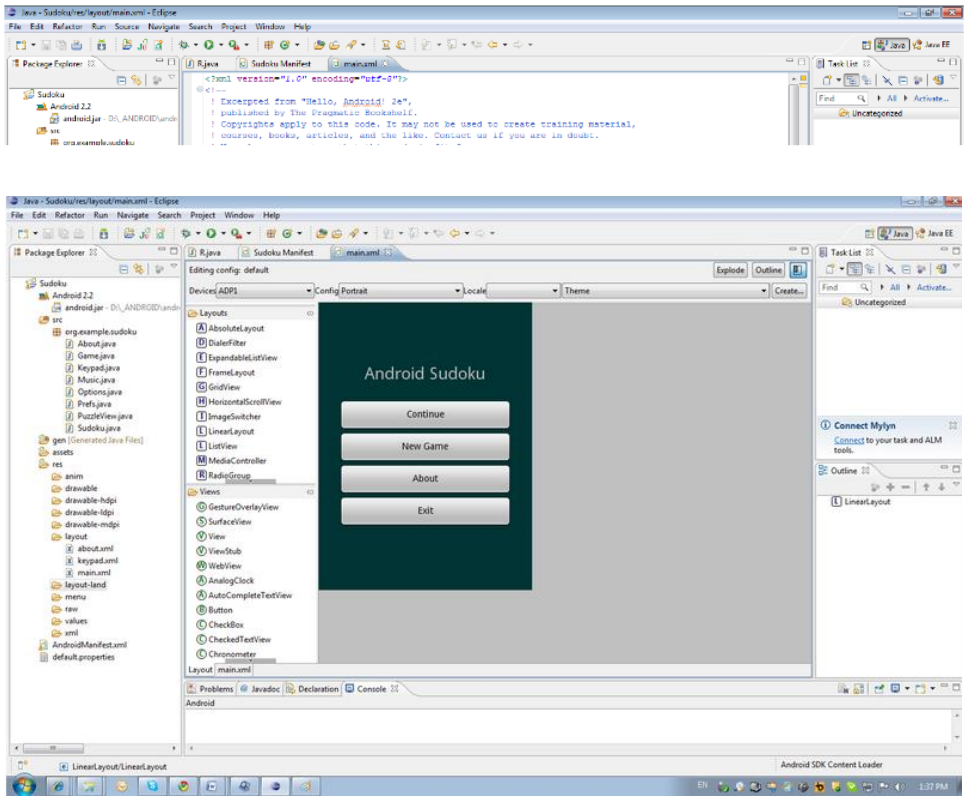
1) Eclipse иш кенглигига лойиҳани архивдан чиқариш.

2) Менюдан **File->New->Android Project** амалини бажариш керак.

3) **New Android Project** бўлимидан **Create project from existing source** ни танлаш.

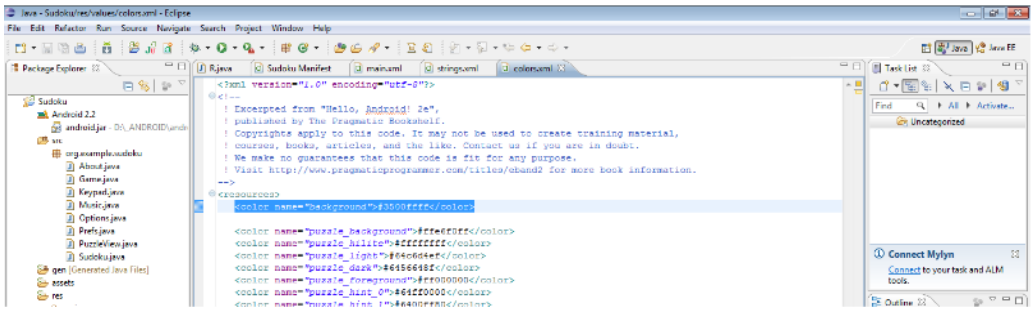
4) **Location** майдонида лойиҳа жойлашган папкага манзилни кўрсатиш. **Next** ни босиш. **Меню дастурлар.**

Ўйналиш менюси **res/layout/main.xml** файлида ёзилган. Барча элементлар битта устун кўринишида котейнерда **LinearLayout** сақланади.



2.3.2-расм. Eclips да “SUDOKU” ўйинини дастури.

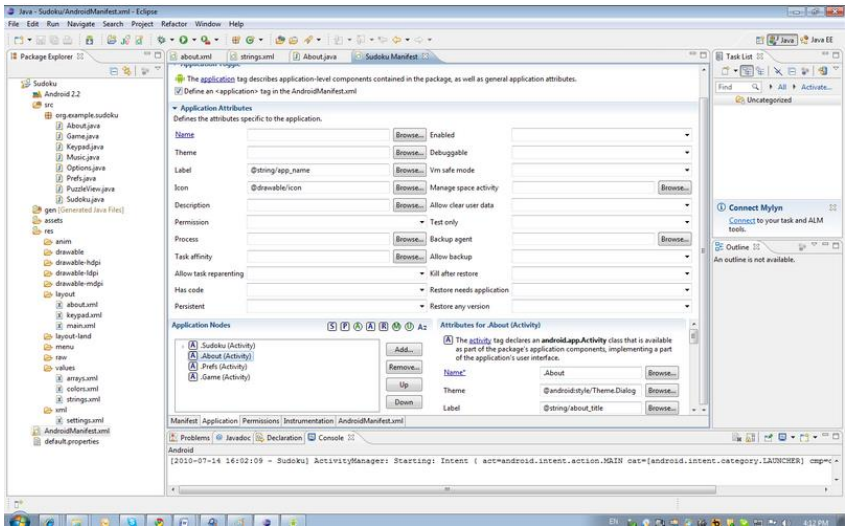
Манбалар. Аҳамият беринг барча матн ёзувлар **(android:text)** манбасидан олинади. Масалан, **android:text="@string/main_title"** ёзуви матнни **res/values/string.xml** да **main_title** ноим билан излашни кўрсатади. Фоннинг ранги **(android:background="@color/background")** манбасидадир, лекин color.xml файлида **(#3500ffff)** кўрсатилган.



2.3.3-расм. Eclips да “SUDOKU” ўйинини дастури.

Кодлардан фойдаланишга рухсат олиш учун бошқарув элементлари id га эга бўлишлари шарт. Тугмалар шахсий id сига эгадир (**android:id="@+id/continue_button"**) ушбу тугмага босилганида турли амаллар бажарилиши учун. Қўшув белгиси тугма учун **/gen/org.example.sudoku/R.java (public static final int continue_button=0x7f0b000b**;) файлининг идентификаторини тузишни сўрайди. Ушбу файл автомат тарзда генерацияланади ва уни ўзгартириш тавсия этилмайди. Файл R синфини ўзида мужассамлаштиради, у орқали интерфейнинг исталган элементига ёки бошқа манбаларга мурожаат этиш мумкин.

Ойналарни тузиш. Дастур тўғрисида ахборот берувчи ойна тузишни кўриб чиқамиз. Ушбу ойна тўғрисида маълумот **/res/layout/about.xml** файлида жойлашган. Синф коди эса **/src/org.example.sudoku/About.java** файлида келтирилган. **AndroidManifest.xml** файли **Activity** билан узвий боғланган. Ушбу файлни редактор ёки XML кўринишида танишиш мумкин. Редакторнинг турли бўлинмаларида файлнинг турли қисмларини танлаш мумкин. Application қисмида **Activity** нинг маълумотлари жойлашган. Аҳамият беринг Theme нинг маълумотлари **@android:style/Theme.Dialog** мазмун моҳиятига эга. Шунинг учун ойна кўпроқ қулай диалог кўринишга эга.



2.3.4-расм. Eclips да “SUDOKU” ўйинини дастури.

About тугмасин босиш орқали Sudoku синфига мансуб ахборот дастур ойнасини юклаш амалга оширилади. **Sudoku** синфи шундай тузилганки, у ўзи Click (public class Sudoku **extends Activity implements OnClickListener**) ҳолатни бошқаради. Public void **onClick(View**) усулида қайси тугма орқали ҳодиса юкланганлиги текширилади, ҳамда керакли код ишга туширилади. **About** ойнасини чақириш учун керакли **Intent** бажарилади.

```
case R.id.about_button:
Intent i = new Intent(this, About.class);
startActivity(i);
break;
```

Ҳодисаларни қайта ишловчилар албатта аниқ бошқарув элементларга ҳам ўрнатилиши мумкин. Масалан: **Keypad** синфида алоҳида тугмаларга ишлов берувчи **setListeners()** синф усули ўрнатилади.

2.3.5-расм. AVD ойнаси.

Оддий диалог. Фойдаланувчига мушкуллик даражасини танлаш имконини бериш керак. Бу кичик диалог бўлиб унда бир неча вариантлардан бирини танлаш шарт. **AlertDialog** синфидан фойдаланамиз. Фойдаланувчи **New Game** тугмасини босади. Қайта ишлашда бу **Sudoku – onClick** синф

усулидир. Сўнг **openNewGameDialog** усули чақирилади. У эса ўйиннинг мушкуллик даражасини таклиф этади ва танланган мушкуллик даражадаги ўйинни ишга туширади. Ушбу диалог **AlertDialog** синфи ёрдамида қурилади.

```
private void openNewGameDialog()
{
   new AlertDialog.Builder(this)
      .setTitle(R.string.new_game_title)
      .setItems(R.array.difficulty,
       new DialogInterface.OnClickListener()
            {
         public void onClick(DialogInterface dialoginterface, int i)
                    {
           startGame(i);
         }
      })
      .show(); }
```

Аҳамият беринг диалогнинг ичидагилари(тугмалар мажмуаси) **R.array.difficulty** қатор массивларидан ташкил топган. Шу ернинг ўзида тугмаини босилганда амалга ошириладиган ҳодиса белгиланади, у ўз навбатида ўйиннинг мушкуллик даражасини танланганидан сўнг **startGame** усулини чафирган ҳолда ўйинни ишга туширади.

График. Game синфи ўйинниг мантиғига жавобгар. Вазифа юкланади ва ютиқнинг талаби текширилади. Game синфи XMLда баён қилинмаган, у код ёрдамида тузилади. **onCreate** усулида тузилади.

View:

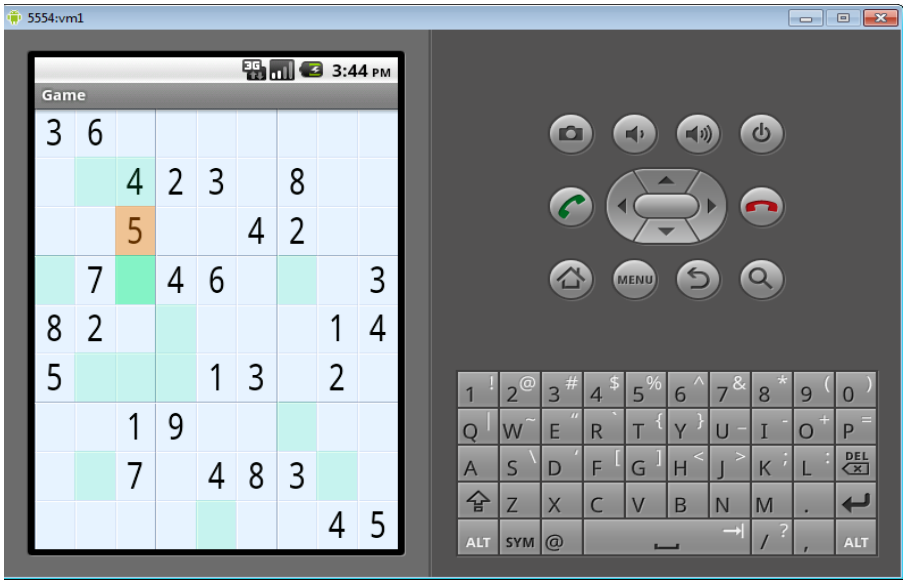
```
puzzleView = new PuzzleView(this);
setContentView(puzzleView);
puzzleView.requestFocus();
```

PuzzleView – бу **View**га таълуқли синфдир. Унда ўйин майдони чизилади, ҳамда экранга тегилганда қайта ишлаш олатлари амалга ошади (**onTouchEvent** усули) ва (**onKeyDown** усули).

Android да чизиш жараёнини билан танишамиз. Чизиш учун **onDraw** усулини юклаш керак. Усул **Canva**s объекти орқали чизиш имкониятига эга бўлади. Ранглар бериш учун Paint синфи объети тузилади. Ранглар **ARGB** форматида берилади. Рангларни манбааларда сақлаш яхшироқдир (файл **colors.xml**). Масалан матнни чизишда у уни бўяш, текстлаш, шрифт белгилари тўғрисида ахборотни сақлайди.

Canvas чизиш ва бўяш тўғрисида барча усуллар маълумотномаларини мужассамлаштирган (**drawRect**, **drawLine**, **drawPath**, **drawText** ва бошқалар).

Графиканинг ишини яхшилаш учун **onDraw** метод ичида ортиқча объектларни тузишдан сақланинг.



2.3.6-расм. AVD ойнаси.

Мусиқа. Мусиқани ишга туширишда MediaPlayer синфидан фойдаланилади. Ўйин учун мусиқа манбалларга жойлаштирилган бўлади. Керакли файллар нусхасини /res/raw папкага жойлаштирилса бўлди(**WAV, AAC, MP3, WMA, AMR, OGG, MIDI** форматида).

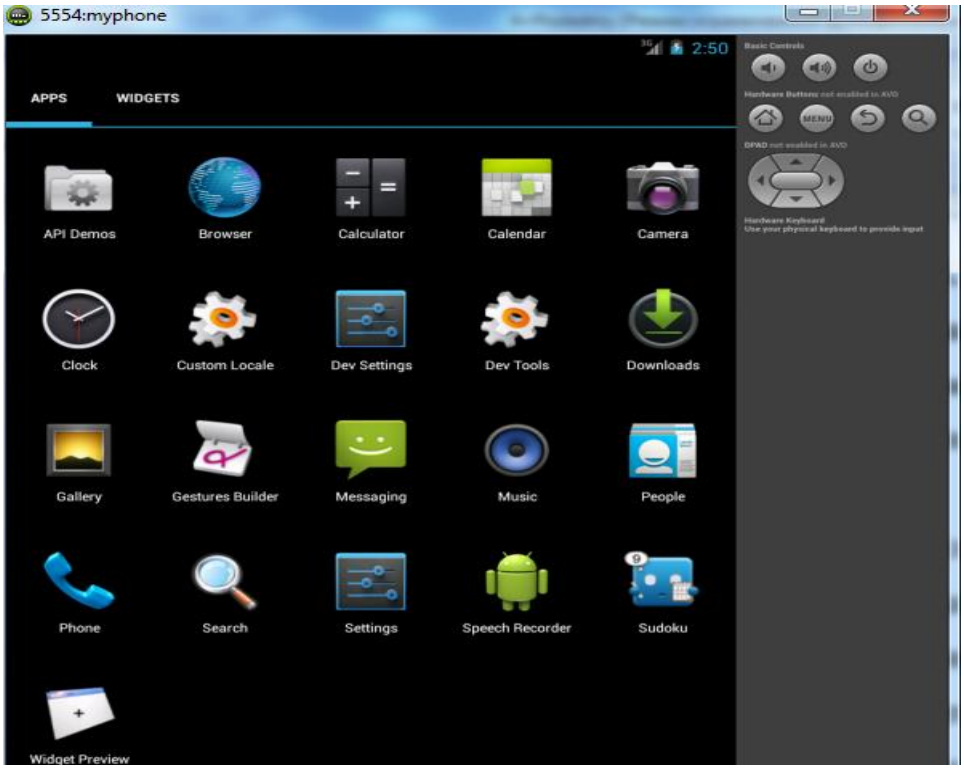
Аввалам бор MediaPlayer синф экземплярини тузамиз: бу ерда context оддий синф бўлиб мусиқани ишга тушишини бошқаради, **resource** –

мусиқанинг манбааси идентикатори. **start, stop** ва **release** мусиқани бошқаришда ишлатилинади.

Ўйинда мусиқа бош менюда ишга туширилади (Sudoku синф орқли туширали) ўйин жараёни вақтида.

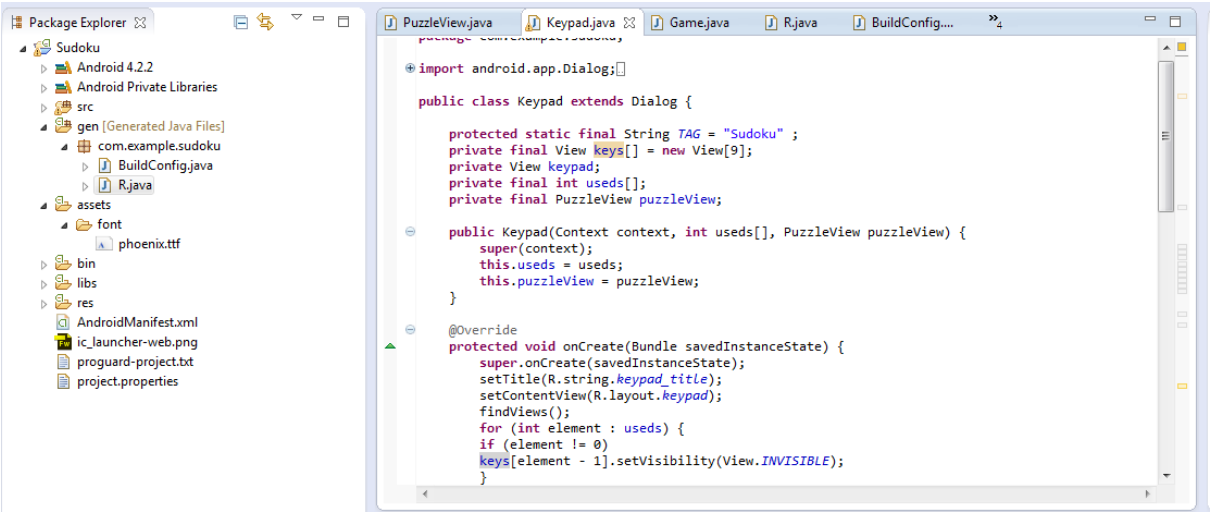
Андроид тизимида ишловчи “Sudoku” ўйини дастурини android платформаси яратишни бошлаймиз. Бизнинг дастур JAVA дастурлаш тили ёрдамида ва eclipse редакторида ёзилган. Eclipse редакторини ишга туширамиз.(2.4.1-расм)

2.3.7-Расм. Eclipse дастурини ишга тушириш.

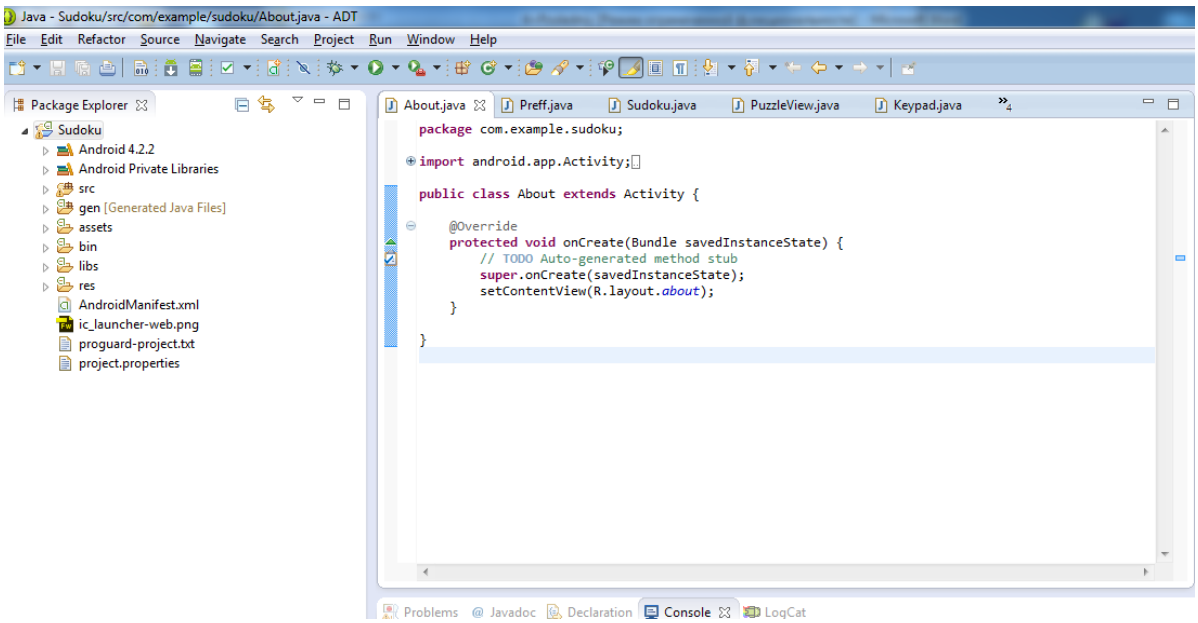


2.3.8-Расм. ADT дастурининг асосий ишчи майдони.

Android тизимига дастур ойналари кўриниши layout.xml файлларида жойлашган бўлса, бу ойналар функцияларини Activity java classларида жойлашган бўлади. Шунинг учун ҳам ойна кўриши .xml файлларда акс эттирилиб шу файл Activity classга чақирилиб қўйилади.



2.3.9-Расм. Javada код ёзиш майдони.

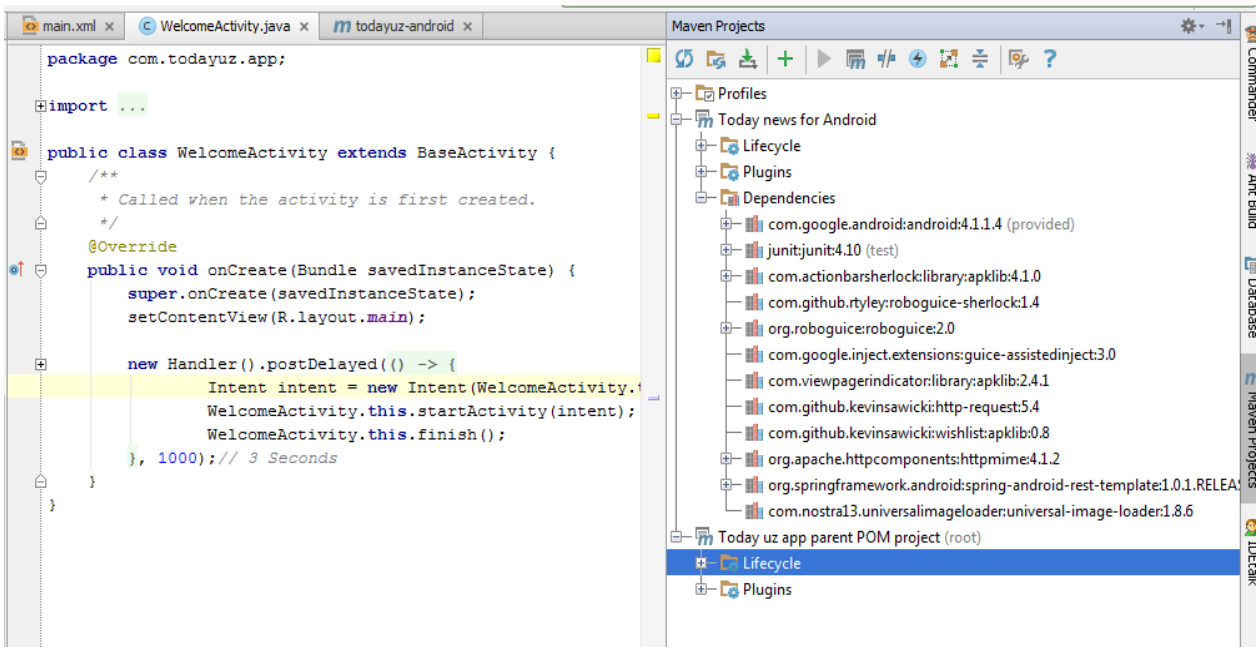


2.3.10-расм. Javada код ёзиш майдони.

Дастуримизда фойдаланиладиган java ва android кутубхоналарини maven ёрдамчи дастури орқали қўшишимиз мумкин. Maven– бу java дастурлаш тилидаги барча кутубхоналар жойлашган дастур ҳисобланади. Maveннинг ҳам pom.xml файли мавжуд бўлиб шу файлга керакли кутубхоналар қуйидаги кўринишда ёзилади.

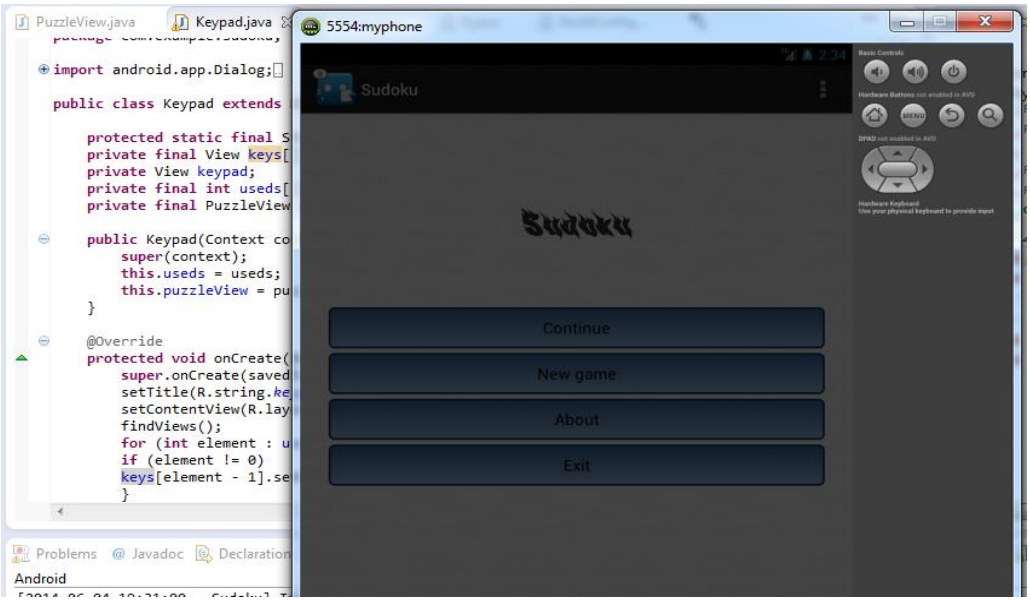
```
<dependencies>
    <dependency>
        <groupId>com.google.android</groupId>
        <artifactId>android</artifactId>
        <scope>provided</scope>
        <version>${android.version}</version>
    </dependency>
</dependencies>
```

Maven pom.xml га ёзилган кутубхоналар орқали проектга қўшилади.



2.3.11-Расм. Javada код ёзиш майдони.

Кутубхоналарни қўшиб олдик ва кейинги босқичдаги ишларга ўтамиз ва дастур интерфейси ва асосий кодларни киритамиз (асосий кодлар иловада келтирилган). Дастурни бехато ишлаш учун уни бир неча марта текшириб хатолари тузатилади, охирида дастур тайёр бўлади. Дастуримиз бир неча ойналари 3.1.5-расмда кўрсатилган



2.3.12-Расм. Дастурни ишга тушириш ойнаси.

Дастуримизда чап менюдан фойдаланилган. Меню ойнасини кўриниши menu_layout.xmlга чизиб чиқилган. Меню кўриниши 3.1.6-расмда келтирилган. Бу ойнани яратишда android стандарт компоненталаридан фойдаланилган.

- **LinearLayout;**
- **RelativeLayout;**
- **TextView;**
- **ImageView;**
- **Button;**
- **EditText;**
- **RadioButton;**
- **CheckBox ва ҳ.к.**

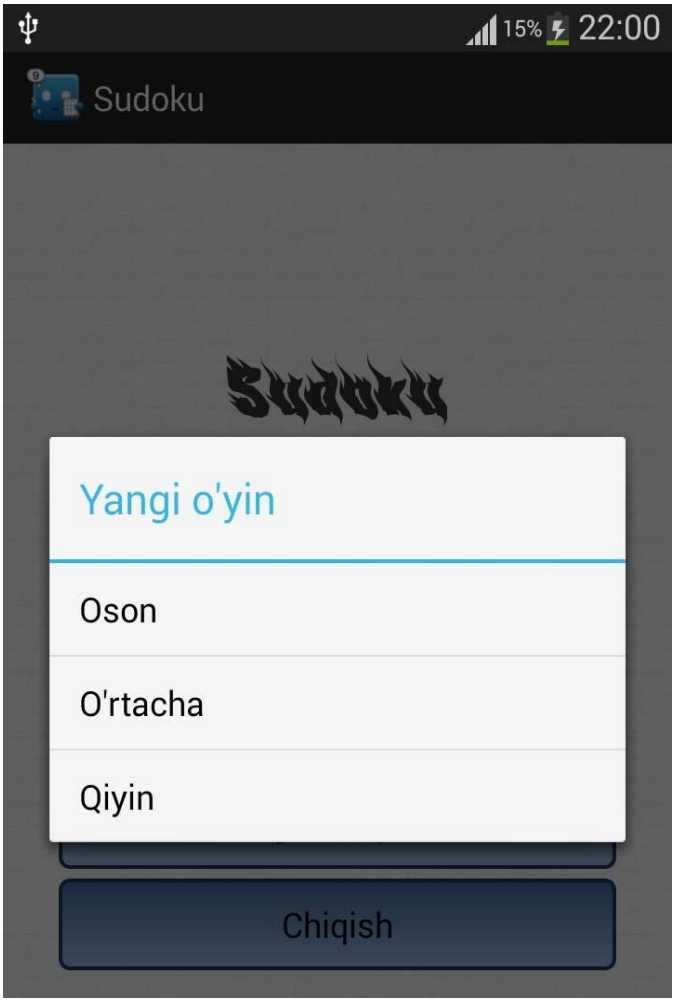
2.4.“SUDOKU” ўйини фойдаланувчи йўриқномасини ишлаб чиқиш

Фойдаланувчи “SUDOKU” ўйнашдан олдин ўйин қоидалари билан батафсил танишган бўлиши керак. Мазкур ўйин бошқотирмали ўйин ҳисобаланди. Биламизки android дастурлари кириш қисмида дастур logo типи ва номи кўрсатилган ойна бир неча секундга очилади ва кейин ойнага ўтиб кетилади:

2.4.1-Расм. “SUDOKU” ўйини асосий ишчи ойнаси.

3 - секунддан кейин дастуримизни бош ойнаси очилади. Бу ойнада Sudoku ўйинини ишлатиш мумкин бўлади. Юқоридаги 2.4.2 – расмда “SUDOKU” ўйининг бош менюси келтирилган бўлиб, улар қуйидаги қисмлардан иборат:

- Ўинни давом эттириш – бу қисмда фойдаланувчи ўйиндан фойдаланаётган вақтда ўйиндан чиқиб кетган ҳолатда, ўйинни кеган жойидан давом эттириш имкониятига эга бўламиз;
- Янги ўйин – Фойдаланувчи ўйинни бошлаши мумкин бўлади;
- Ўйин ҳақида – “SUDOKU” ўйини қачон ишлаб чиқарилган ва android платформаси версияси кўрсатилади;
- Чиқиш – Ўйиндан чиқиш мумкин.

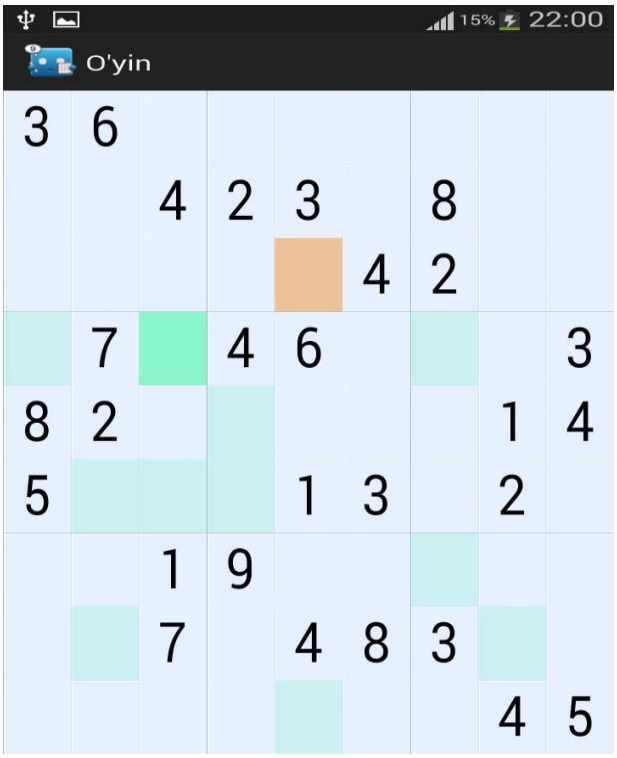


2.4.2-Расм. “SUDOKU” ўйини даражалари.

2.4.2-Расмда “SUDOKU” ўйини ўйнаш даражалари қуйидагича берилган:

- Осон – Ўйинни умуман ўйнамаган фойдаланувчи шу даражадан ўтиши тавсия этилади;

- Ўртача – Бу даража фойдаланувчилар ўрта даржадан кўпроқ фойдаланишади;
- Қийин – Ўйинни яхши ўйнайдиган фойдаланувчилар бу даражадан фойдаланишлари мумкин.



2.4.3-Расм. “SUDOKU” ўйини ўйнаш ойнаси.

Ўйин майдони 9х9 ҳажмли квадратда ташкил топган бўлиб, у ҳар томонидан 3 та катакка бўлинган ва умумий катаклар сони 81тани ҳосил қилади. Бу катакларда ўйин бошида бир неча сонлар туради. Фойдаланувчи бўш катакларни 1 дан 9 гача бўлган сонлар билан шундай тўлдириш талаб керакки, натижада ҳар бир катакда, ҳар бир устунда ва ҳар бир кичик 3х3 квадратда ҳар бир сон фақат бир марта учрасин. Ўйинни шу тариқа давом эттириш тавсия этилади.

3. ҲАЁТ ФАОЛИЯТИ ХАВФСИЗЛИГИ

3.1. Компьютер хоналарида меҳнат шароитларини тўғри ташкил этиш жойини ташкил этилиши

Компьютер хоналарида иш жойи тўғри ташкил этилиши иш унумдорлигини ошириш, чарчашни олдини олиш,иш жойидаги жихозларни ва ускуналарни тўғри жойлаштиришниомиллар, рангларни тўғри танлай билишдир. Ускуналар шундай жойлашиши керакки ишчилар ортиқча харакатсиз, ўзини зўриқтирмасдан осонгина фойдаланиш.

Иш жойида меҳнат шароитини яхшилаш ишларига бир қанча омилларни ҳисобга олган ҳолда ташкил қилинади.Буларга ташкилий, техникавий,санитария –гигиена,табиий-иқлим омиллари киради.

Ташкилий омиллар –ишнинг ташкил этиш шакли интизом,меҳнат жараёни устидан қилинадиган назоратнинг ҳолати, меҳнат муҳофазаси, ишчи ходимларнинг касбий тайёргарлик даражаси, техник омиллар жараёнларини механизациялаш ва автоматлаштириш даражаси, бошқарувда электрон–ҳисоблаш техникаларида компьютерларни қўллаш, ҳимоялаш воситаларининг созлиги ва етарлиги киради.

Санитария-гигиена омиллари -иш жойининг санитария ҳолатига жавоб бериш-бермаслиги, эрганомик омил машина ва ускуналарни инсон билан ўзаро ҳаракатда бўлганда машина элементларининг мос келиши. Бунда техникани тезлик параметрларига тегишли, ишчи органларидан келаётган маълумотлар хажми, иш жойининг ташкил этилиш даражаси, бошқариш органларининг қулай жойлашганлиги оператор ўриндиғининг конструкцияси киради.

Психофизиологик омиллар-меҳнатнинг оғирлиги ва қизғинлиги, жамоадаги психологик вазият, ишчиларнинг бир-бири билан ўзаро муносабати, жисмоний зўриқиш, асабий-психик зўриқишлаш меҳнат шароитининг инсон организмига таъсири ўрганган ҳолда қуйидагиларни амалга ошириш керак:

-ишда бажарилаётган жараёнлар тез ва тез талаблар чегарасида ва ҳаракатлантирувчи майдоннинг энг қулай доирасида амалга ошириш:

-ишлаб чиқариш биноларида ҳаво муҳитини текшириш;

-ишлаб чиқаришдаги метрологик омилларини аниқлаш;

-маҳаллий ва умумий титрашни аниқлаш;

-ишлаб чиқаришдаги шовқин даражасини аниқлаш;

-иш жойининиг ёритилганлигини аниқлаш;

-нурланганликни текшириш;

-ҳаво алмашувини текшириш.

Компьютер хоналарида ишлаганда меҳнат шароитларини яратишдаги энг кўп кўйиладиган хатоларга қуйидагилар киради:

1. Иш жойларни етарлича катта эмаслиги.
2. Хона ҳарорати ва намлиги талаб килинадиган меъёрларга мос келмаслиги.
3. Хона ва иш жойларни етарлича ёритилмаганлиги.
4. Монитордан чиқаётган паст частотали элктромагнитни майдонида нурланишнинг кўп даражада ажралиши.
5. Иш вақтини ва дам олиш вақтларини тўғри тақсимланганлиги.

Компьютер операторлари, дастурчилар ва бошқа ҳисоблаш техникаси ишчилари шовқин, электр токи, статик электр каби физик, хафли ва зарарли физик омиллар таъсирида бўлади.

Кўпгина ҳисоблаш техникаси билан ишловчи ҳодимлар ақлий зўриқиш, кўриш ва эшитиш анализаторларини зўриқиши, эмоционал зўриқиш каби психофизик омиллар таъсирида бўладилар.

Толиқиш пайдо бўлиши иш пайтида марказий асаб тизимида пайдо бўладиган ўзгаришларга боғлиқ. Масалан кучли шовқин ранг ажратишни кийинлаштиради, куриш ўткаирлиги, ёруғликка мослашишни пасайтиради, ахборот қабул қилишни қийинлаштиради ва 5-12 фоизга иш унумини пасайтиради.

90 ДБ шовқинни узоқ вақт давомидаги таъсири иш унумдорлигини 30-60 фоизга пасайтиради.

Ҳисоблаш техникаси билан ишловчи ишчилар тиббий текширувдан ўтказилганда иш унумдорлиги пасайишдан ташқари шовқин эшитиш қобилиятини хам пасайтиради. Комбинациялашган зарарли омиллар таъсирида кўп вакт қолиш касбий касалланишга олиб келиши мумкин.

Электр қурилмалари яъни компьютернинг барча қурилмалари инсон учун хафли ҳисобланади. Чунки компьютерда ишлаётганда инсон ток кучланиши таъсирида бўлган қисмларга тегиб кетиши мумкин.

Электр қурилмаларнинг специфик хавфи: бу электр ўтказувчилар, изоляцияси шкастланиши натижасида ток таъсирига тушиб қолган компьютер корпуси Электр токининг таъсири фақат ток инсон танасидан ўтганда сезилади. Электр шикастланишидан ҳимояланишда электр қурилмалари тўғри жойлаштирилиши, электр ўтказувчи сим ва кабелларни тўғри уланиши мухим ўрин эгаллайди.

Иш жойларида статик электрнинг разрядли токи кўпроқ компьютернинг бирор бир элементига тегиб кетиш натижасида хосил булади. Бундай разрядлар инсонга хавф туғдирмайди, ёқимсиз таъсирдан ташқари компьютерни ишдан чиқишига олиб келади. Симлар изоляцияси шикастланганда ток таъсирини камайтириш учун иш хоналарининг поллари бир қаватли поливинил хлоридли антистатик линолиум билан қопланиши лозим. Ҳимоялашни бошқа усули ионлаштирилган газ билан зарядларни нейтраллаш.

Компьютер хоналарининг катталиги у ерда ишловчи ходимлар ва компютерлар сонига мос келиши зарур. Иш жойларини ташкиллаштиришга, яна харорат, ёруғлик, ҳаво тозалиги, шовиқиндан ҳимояланганлик параметрлари ҳисобга олинади.

Санитар меъёрларига кўра бир ишчи учун иш жойининг хажми 15 $м^3$, иш майдони эса 4,5 $м^2$ дан кам булмаслиги керак. Хонанинг баландлиги полдан шифтгача 3-3,5 м булиши керак.

Компьютер хоналарида одатда ён томонлама табиий ёритилганлик қўлланилади. Табиий ёритилганликда шимол ёки шимолий-шарққа қаратилган ёруғлик дарчаларидан фойдаланиш керак, бунда табиий ёритилганлик коэфиценти 1,2-1,5 % кам бўлмаслиги шарт.

Компютерларни подвалларда жойлаштириш рухсат этилмайди.

Компьютер хоналари ва иш жойларида табиий ёритилганлик қўлланиши зарур.Бошқа холларда эса сунъий ёритилганликни қўллаш мумкин.

Иш хужжатлари жойлашган стол усти ёритилганлиги 300-500 лк булиши керак. Ёруглик манбаини шундай жойлаштириш керакки, бунда ёруғлик кўзни қамаштирмаслиги керак, кўриш майдонидаги ёруғлик манбаининг ёрқинлиги 200 кд /м^2 ошмаслиги керак.

Компьютер жойини шундай жойлаштириш керакки, бунда табиий ёруғлик иложи борича ёндан тушиши лозим.

Компьютер столининг баландлиги иложи борича 680-800 мм булиши керак. Иш столи оёклар учун баландлиги 600 мм, эни 500 мм дан кам бўлмаган, чуқурлиги тизза даражасида 450 мм дан кам бўлмаслиги, узатилган оёқ даражасида 650 мм дан кам бўлмаслиги керак.

Экран монитори кўздан энг узоғи билан 600-700 мм бўлиши керак, лекин харф ва шрифтларнинг ўлчамига қараб 500 мм дан яқин бўлмаслиги керак.

Хоналарни рангли жиҳозланиши ишни санитар-гигиена шароитларини яхшилашга қаратилади, иш унумини ошишига ҳизмат қилади.Хоналарни бўялиши инсон асаб тизимлари, кайфиятига ва охир-оқибат иш унумига таъсир этади.

Компьютер хоналарини рангини техник жиҳозлар ранги билан бир хил рангда бўяш мақсадга мувофик. Хоналар ва жиҳозлар ранглари юмшоқ бўлиши ва ялтироқ бўлмаслиги лозим.

3.2.Ёнғин хавфсизлиги

Электр қурилмалари, компьютерлардан фойдаланишда турли хилдаги ёнишлар хавфи доим мавжуддир. Замонавий компьютерларда электрон схемаларнинг элементларини жойлашиш зичлиги жуда юқоридир, улаш

симлари, коммунакацион кабеллар бир-бирига жуда яқин жойлашган. Улардан ток оққанда, ката миқдорда иссиқлик ажралади, баъзи бўлимларда ҳарорат 80-100С гача кўтарилиши мумкин. Бу уларнинг изоляция қобиғининг эришига, ўтказгич қисмларининг очилиб қолишига, оқибатда қисқа туташув бўлиб, учқун чиқиши ва ёниб кетишига олиб келиши мумкин.

Ортиқча иссиқликни йўқотиш учун ҳавони конденционерлаш ва вентиляция тизими хизмат қилади. Лекин бу тизмлар машиназаллари ва бошқа хоналар учун қўшимча ёнғин хавфини юзага келтиради, чунки бир тарафдан, ёнғин содир бўлганда, уларни хоналарга тезда тарқалишига ёрдам беради.

Электр қурилмаларига ток алоҳида ёнғин хавфи бўлган кабел симлари орқали узатилади, ёнувчи изоляция материалининг мавжудлиги, электр учқуни ва яқинлашиш қийинлиги кабел линияларидан ёнғин ёнғин чиқиши ва ривожланиши эҳтимоли янада катталигидан далолат беради. Шунинг учун кабел симлари ёнмайдиган материаллардан тайёрланган, олиб-қўйилувчи технологик пол остидан ўтиши керак. Ҳисоблаш марказларидаги хоналарда ёнғин жўмраклари йўлкаларга зина майдончаларига, кириш жойларига ўрнатилади. Олов ўчиргичлари 40-50 см^2 га битттадан ўрнатилади. Ёнғин сигнализаторлари ва автоматик ёнғин ўчириш қурилмалари ўрнатилади.

Ёнгинлар саноат корхоналари , халк хужалигинингхамма тармоклари, кишлок хужалиги ва турар жойларда юз бериши мумкин булган ,етказадиган зарари жихатидан табиий офатларга тенглашиши мумкин булган ходиса хисобланади . Ёнгинлар катта моддий зарар келтириши билан бирга , огир бахтсиз ходисалар , захарланиш ,куйиш натижасида кишилар хаётини олиб кетган холлар куплаб учрайди .

Шунинг учун хам ёнгинга қарши кураш барча фуқароларнинг умумий бурчи ҳисобланади ва бу ишлар давлат микёсида амалга оширилади.

Умуман ёнғин чиқмаслигини таьминлаш , ёнғин чиққан тақдирда ҳам унинг ривожланиб, тарқалиб кетишининг олдини олиш, моддий бойликларни, инсон саломатлиги ва унинг хаётини саклаб колишга

каратилган чора тадбирлар бўлиб, бу масалалар меҳнатни муҳофаза қилишнинг таркибий қисми ҳисобланади .

Бизнинг вазифамиз ёнгин хакида асосий тушунчалар бериш билан бирга, унга қарши самарали кураш олиб бориш, ёнғинни учиришда кулланиладиган бирламчи воситалар , хар хил тадбирлар билан ўқувчиларни таништиришга қаратилган .

Ёнғиннинг сабаблари: иситиш печларини куриш ёки ишлатиш қоидаларини бузиш, ишлаб чиқариш ёки уйда оловни эҳтиётсизлик билан ишлатиш.,керосин билан ишлаётганда ёритиш ёки қиздириш асбобларидан нотўғри фойдаланиш ёки нотўғри ўрнатиш.яшин ёки статик электр разрядларини ишлатиш. Машиналар ва ишлаб чиқариш жихозларинг носозлиги хамда уларни ишлатиш қоидаларига роя қилмаслик сабаб бўлади.

Ён-инни олдини олиш учун тадбирлар: ташкилий ,техникавий тадбирлар қўллаш керак бўлади.

3.2.1.Саноат корхоналарини ёнғинга ва портлашга хавфи бўйича тоифаланиши

Ҳар бир саноат корхонаси унинг ишлаб чиқариш технологиясиишлатадиган ҳом-ашёси,чиқарадиган маҳсулоти ва жойлашган биноларнинг констукциясига кўра ёнғин чиқишига,портлашига ва ёнғин чиққан тақдирда унинг тарқалишига,ёнғиннинг асоратига асосланган ҳолида ёнғинга ва портлашга хавфли даражаси белгиланилади.Қурилиш норма ва қоидаларига асосан корхоналар,складлар ва портлаш хавфи бўйича бешта тоифага бўлинади.

А тоифа- ёнғинга ва портлашга хавфли корхона. Буларга сув .кислород ва бир-бири билан бирикиш натижасида портлаш ва ёниши мумкин бўлган моддаларни ишлатадиган корхоналари: алангаланиш куйи чегараси хонадаги ҳаво хажмига нисбатн 10% миқдорни ташкил қилиши мумкин бўлган ёнувчи газларни ишлатадиган корхоналар: хона хажмига нисбатан 5% миқдорни ташкил қилиши мумкин бўлган ва буғланинг алангаланиш харорати 28 с гача бўлган суюқликлар билан иш олиб борадиган корхоналар. Улар

олтингугуртли углерод.эфир. ацетон ва бошқа шунга ўхшаш моддалр оладиган корхоналар.

Б тоифа- ёнғинга ва портлашга хавфли корхона.Бу тоифага куйи алангаланиш чегараси ҳаво хажмига нисбатан 10 %дан ортиқ бўлган ёнувчи газлар билан иш олиб борадилар.шунингдек чақнаш харорати28 С дан 61 С гача бўлган суюқликлар хамда ишлаб чиқариш жараёнида чақнаш хароратигача суюқликлар Билан ишлайдиган ва чанглар хона хажмининг 5% дан кўпроқ миқдорида тўпланалдиган.портловчи аралашма ҳосил қилиши мумкин бўлган саноат корхоналри киради. Манна шундай саноат корхоналари сирасига аммиак хайдовчи компрессор станциялари.деталларни керасин билан ювиб тозлаш корхоналри киради.

В тоифа - ёнғинга хавфли тоифа. Бу тоифага буғларнинг чақнаш харорати 61 с дан юқори бўлган суюқликлар қуйи алангаланиш чегарси 65г/м3 дан ортиқ бўлган ёнувчи чанглар ва толалар. Шунингдек кислород билан бириккан ҳолда ёнувчи моддалар ва қаттиқ ёнувчи моддалар билан ишловчи корхоналар киради. Кўмир кукуни ҳосил қилувчи .ёғочсозлик корхоналри киради.

Г тоифа - ёнғинга хавфли тоифа.Бу тоифага ёнмайдиган жисм материалларга.қиздириб.чўғлатиб ва эритиб ишлов берадиган ва ишлов бериш давомида нурли иссиқлик.учқун ва аланга чиқариш мумкин бўлган.қаттиқ суюқ ва газсимон моддалр ёқилғи сифатида ишлатиладиган корхоналар.Қозонхоналар эритиш ва қуйиш цехлари матен цехлари киради.

Д тоифаси- ёнғинга хавфсиз тоифа Бунга ёнмайдиган жисмлар ва материалларга совуқ ишлов берадиган корхоналар киради. Қурилиш машина созлик корхоналари. Корхоналарни лойхалашда ва қуришда ёнғинга қарши кураш тадбирлари.

Корхоналарни қуриш ва лойхалашда унинг бажарадиган иш моҳиятидан келиб чиқадиган талабларидан.унга техник мустахкамлик санитария-гигена ва иқтисодий талаблардант ташқари унга ёнғин хавфи ва ёнғинга қариши

тура олиш талаблари хам қўйилади. Давлат стандартларига асосан ҳамма қурилиш констукциялари ёниши бўйича уч группага бщлинади.

Ёнмайдиган контрукциялар- бунга катта харорат ёки аланга таъсирида ёниб, кулга ёки кўмирга айланмайдиган қурилиш констукциялари киради. Маслан металл конструкциялари, ва минерал материаллари.

Қийин ёнадиган конструкциялар-бунга катта харорат ёки кучли аланга доимий таъсир этганда тутаб ёнадиган,аланга таъсири йўқалиши билан ўчадиган саноат конструкциялари киради. Буларга мисол қилиб ўтга қарши ишлов берилган ёғоч констукциялар ва чиқиндилардан тайёрланган ярим органик ва ярим менирал моддалардан тайёрланган констукциялар.

Ёнадиган конструкциялар- буларга аланга ёки катта харорат ёндирувчи восита бўлиб кеёин алага олиб кетилгандан кейин ҳам ёнишда давом этадиган контструкциялар киради. Буларга ёғоч материаллари .қурилишда ишлатиладиган турли туман пласмассалар киради

Конструкцияларнинг ўтга чидамлилиги соатларда белгиланилади. Мана шу чегар соатларнинг катталигига қараб курилиш констукцияларининг ўтга чидамлилик даражаси белгиланилади ва улар рим рақами билан белгиланилади.

1 даражадаги ўтга чидамлиликка эга бўлган бинолар асосий деворлари Зина поя майдонлари ва колонналарининг ўтга чидамлилик чегараси 2.5 соатдан кам бўлмаслиги.ташқи девор ва оролиқ деворлар 0.5 соатдан кам бўлмаслик керак. Ll-даражадаги биноалр эса юқоридаги кўрсатгичлари 2.1 ва 0.25 соатни ташкил қилади. Ўтга чидамлиги 111 бўлган биноларнинг ҳамма қисмлари ёнмайдиган бўлади. .биноларнинг пойдеворлари қийин ёнадиган ва томлари кўтарувчи конструкцияга эга бўлган ёнадиган бўлиши керак.

4 даражали биноларда катта бинони қисмларга ажратадиган эшик-деразасиз махсус ёнғинга қарши идеворлари ёнмайдиган бўлади. V даражадаги бинолар учун эса ўтга чидамлиликнинг минимал миқдори белгиланмайди.

Қурилиш конструкцияларини ўтга чидамлилик даражасини ортириш имкониятлари мавжуд Маслан, металларни ўтга чидамлилиги ниҳоятда паст бўлиб 15-20 мин. Ичида эгилиб –букилиб кетади,агар ўтга чидамли бўёқлар билан бўялса чидамлилиги бир мунча ортади.алебастер ёки цемент аралашмалари Билан сувалса унинг ўтга чидамлилиги 1 соатга ортади. .агар гипс плиталари билан қопласак .плиталар қалинлиги 6 см дан кам бўлмаса, уларнинг ўтга чидамлилиги 3 соатга етади.

Ёғоч конструкцияларни ўтга чидамлилигини оширишга уни суваш ва калинлиги 20 мм бўлиши керак шунда ўтга чидамлилиги 20-25 минутга етади. Яна ёғочни ўтга чидамлилигини оширишда антиприн деб аталувчи ёғоч устига сепиилади, бу препарат ёғ ва шимилиб ёнишини қийинлаштиради.

ХУЛОСА

Ушбу битирув малакавий ишида android тизими учун Sudoku ўйини дастури яратилди. Ушбу ишнинг эътиборли жиҳати шундаки, Sudoku ўйинида фойдаланувчилар ўз мобил телефонлари орқали фойдаланишлари мумкин. Sudoku ўйинини яратишда асосан java дастурлаш тили ва andorid платформаси компоненталаридан фойдаланилди. Дастур Java (Dalvik Virtual Machine) виртуал машинасида бажарилди. Мазкур ишда қуйидаги ишлар амалга оширилди:

- “SUDOKU” ўйинини яратишда android платформасининг компоненталарини ўрганиб чиқилди;
- Android платформасида илова яратиш жараёнлари ўрганилди;
- “SUDOKU” ўйинини алгоритми ва дастури ишлаб чиқилди;
- “SUDOKU” ўйини интерфейсини ишлаб чиқилди;
- “SUDOKU” ўйинини мобил қурилмаларда ишлатиш учун фойдаланувчи йўриқномасини ишлаб чиқилди.

Фойдаланилган адабиётлар

1. Ўзбекистон Республикаси Президенти Ислом Каримовнинг мамлакатимизни 2013 йилда ижтимоий-иқтисодий ривожлантириш якунлари ва 2014 йилга мўлжалланган иқтисодий дастурнинг энг муҳим устувор йўналишларига бағишланган Вазирлар Маҳкамасининг мажлисидаги маърузаси.
2. И.А. Каримовнинг 2012-йил 21-мартдаги “Замонавий ахборот-коммуникация технологияларини янада жорий этиш ва ривожлантириш чора-тадбирлари тўғрисида”ги Қарори.
3. И.А. Каримовнинг 2012-йил 16-октябрдаги “Ахборот-коммуникация технологиялари соҳасини тубдан такомиллаштириш ва ривожлантириш тўғрисида”ги Фармони.
4. И.А. Каримовнинг 2013 йил 20-сентябрдаги “Мамлакатимизда дастурий таъминот воситалари ишлаб чиқарувчиларни рағбатлантиришни янада кучайтириш чора-тадбирлари тўғрисида” ги қарори.
5. И.А. Каримовнинг 2014 йил 3 апрелдаги “Иқтисодиётнинг реал секторига ахборот-коммуникация технологияларини жорий қилиш чора-тадбирлари тўғрисида”ги қарори.
6. Брайн Х., Билл Ф. Программирование под Android. Питер.2014г.-592с.
7. П. Дейтел, Х. Дейтел, Э. Дейтел, М. Android для программистов. Создаем приложения, Питер. 2013г.
8. Б.Джошуа, Java. Эффективное программирование, издательство: Лори. 2013г.-310 с.
9. Рето Майер, Android 4. Программирование приложений. издательство: Эксмо. 2013г.- 816с.
10. Голощапов А. Л. Google Android. Программирование для мобильных устройств (Профессиональное программирование). БХВ-Петербург.2011г.-440с.

11.Internetda dasturlash. Muammoli ma'ruzalar matni. O.Jakbarov, T.Jo'raev. Namangan – 2004.

12. SQL va ma'lumotlar bazalarining keyingi dasturlash. SH.Nazirov, R.Qobulov. toshkent - 2006.

13. Ma’lumotlar bazalari. B.A. Begalov, E.E.Inog‘omjonov, Sh. Nasreddinova. Toshkent – 2006.

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. Yormatov G’.Y., Isamuhamedov Y.U. Mehnatni muhofaza qilish. Darslik. O’zbekiston nashryoti. Toshkent 2002.

ИЛОВА

```
package com.example.sudoku;


import android.app.Activity;
import android.os.Bundle;


public class About extends Activity {


    @Override
    protected void onCreate(Bundle savedInstanceState) {
            // TODO Auto-generated method stub
            super.onCreate(savedInstanceState);
            setContentView(R.layout.about);
    }


}
package com.example.sudoku;


import android.content.Context;
import android.os.Bundle;
import android.preference.PreferenceActivity;
import android.preference.PreferenceManager;
```

```
public class Preff extends PreferenceActivity {

    public final static String OPT_MUSIC = "music";
    public final static boolean MUSIC_DEF = true;
    public final static String OPT_HINT = "hint";
    public final static boolean HINT_DEF = true;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.settings);
    }

    public static boolean getMusic(Context context){
        return
PreferenceManager.getDefaultSharedPreferences(context).getBoolean(OPT
_MUSIC, MUSIC_DEF);

    }

    public static boolean getHint(Context context){
```

```
            return
PreferenceManager.getDefaultSharedPreferences(context).getBoolean(OPT
_HINT, HINT_DEF);


        }


    }
    package com.example.sudoku;


    import android.os.Bundle;
    import android.view.Menu;
    import android.view.MenuInflater;
    import android.view.View;
    import android.view.View.OnClickListener;
    import android.app.Activity;
    import android.content.Intent;
    import android.view.MenuItem;
    import android.widget.TextView;
    import android.app.AlertDialog;
    import android.content.DialogInterface;
    import android.graphics.Typeface;
    import android.util.Log;
```

```
public class Sudoku extends Activity implements OnClickListener {

    private final static String TAG = "Sudoku";
    private TextView logo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sudoku);

        // Обработчик для кнопки продолжить
        View continueButton = findViewById(R.id.continueButton);
        continueButton.setOnClickListener(this);
        // Обработчик для кнопки Новая игра
        View newGameButton = findViewById(R.id.newGameButton);
        newGameButton.setOnClickListener(this);
        // Обработчик для кнопки Об Игре
        View aboutButton = findViewById(R.id.aboutButton);
        aboutButton.setOnClickListener(this);
        // Обработчик для кнопки Выход
        View exitButton = findViewById(R.id.exitButton);
```

```
        exitButton.setOnClickListener(this);


        logo = (TextView) findViewById(R.id.logo);
        Typeface typeface = Typeface.createFromAsset(getAssets(),
"font/phoenix.ttf");
        logo.setTypeface(typeface);
    }


    @Override
    protected void onResume() {
        super.onResume();
        Music.play(this, R.raw.main);
    }


    @Override
    protected void onPause() {
        super.onPause();
        Music.stop(this);
    }


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
        // TODO Auto-generated method stub
    super.onCreateOptionsMenu(menu);
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.sudoku, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
        // TODO Auto-generated method stub
        //super.onOptionsItemSelected(item);

        switch (item.getItemId()) {
        case R.id.settings:
                Intent settinsMenu = new Intent(this, Preff.class);
                startActivity(settinsMenu);
                return true;



        }
        return false;
}
```

```
public void onClick(View v) {
        switch (v.getId()) {
        case R.id.continueButton:
                startGame(Game.DIFFICULTY_CONTINUE);
                break;
        case R.id.aboutButton:
                Intent about = new Intent(this, About.class);
                startActivity(about);
                break;
        case R.id.newGameButton:
                openGameDialog();
                break;
        case R.id.exitButton:
                finish();
                break;
        default:
                break;
        }
}

private void openGameDialog()
```

```
{
        new AlertDialog.Builder(this)
                .setTitle(R.string.new_game)
                .setItems(R.array.difficult,                    new
DialogInterface.OnClickListener() {


                        @Override
                        public void onClick(DialogInterface dialog, int
which) {
                                // TODO Auto-generated method stub
                                startGame(which);
                        }
                }).show();
}


private void startGame(int i)
{
        Log.d(TAG,"clicked on " + i);
        Intent intent = new Intent(this, Game.class);
        intent.putExtra(Game.KEY_DIFFICULTY, i);
        startActivity(intent);
}
```

```
}
package com.example.sudoku;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.Paint.FontMetrics;
import android.graphics.Paint.Style;
import android.graphics.Rect;
import android.os.Bundle;
import android.os.Parcelable;
import android.util.Log;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.view.animation.AnimationUtils;

public class PuzzleView extends View {

    private static final String TAG = "Sudoku";
    private Game game;
    private float width; // ширина одного тайла
```

```
private float height; // высота одного тайла
private int selX; // координата x выделенной области
private int selY; // координата y выделенной области
private final Rect selRect = new Rect();

// запоминание курсора
private static final String SELX = "selX" ;
private static final String SELY = "selY" ;
private static final String VIEW_STATE = "viewState" ;
private static final int ID = 42;


public PuzzleView(Context context) {
        super(context);
        this.game = (Game) context;
        setFocusable(true);
        setFocusableInTouchMode(true);
        setId(ID);
}

@Override
protected void onSizeChanged(int w, int h, int oldw, int oldh) {
```

```
        width = w / 9f;
        height = h / 9f;
        getRect(selX, selY, selRect);
        Log.d(TAG, "onSizeChanged: width " + width + ", height "
        + height);
        super.onSizeChanged(w, h, oldw, oldh);
    }

    private void getRect(int x, int y, Rect rect) {
        rect.set((int) (x * width), (int) (y * height), (int) (x
        * width + width), (int) (y * height + height));
    }

    @Override
    protected void onDraw(Canvas canvas) {
        // Рисование фона...
        Paint background = new Paint();
        background.setColor(getResources().getColor(
        R.color.puzzle_background));
        canvas.drawRect(0, 0, getWidth(), getHeight(), background);
        // Рисование доски...
        // Определение цветов для линий решетки
```

```
Paint dark = new Paint();
dark.setColor(getResources().getColor(R.color.puzzle_dark));
Paint hilite = new Paint();
hilite.setColor(getResources().getColor(R.color.puzzle_hilite));
Paint light = new Paint();
light.setColor(getResources().getColor(R.color.puzzle_light));
// Рисование вспомогательных линий решетки
for (int i = 0; i < 9; i++) {
        canvas.drawLine(0, i * height, getWidth(), i *
height,
        light);
        canvas.drawLine(0, i * height + 1, getWidth(), i *
height
        + 1, hilite);
        canvas.drawLine(i * width, 0, i * width,
getHeight(),
        light);
        canvas.drawLine(i * width + 1, 0, i * width + 1,
                getHeight(), hilite);
        }
        // Рисование основных линий решетки
for (int i = 0; i < 9; i++) {
        if (i % 3 != 0)
```

```
                    continue;
                    canvas.drawLine(0, i * height, getWidth(), i *
height,
                    dark);
                    canvas.drawLine(0, i * height + 1, getWidth(), i *
height
                    + 1, hilite);
                    canvas.drawLine(i * width, 0, i * width,
getHeight(), dark);
                    canvas.drawLine(i * width + 1, 0, i * width + 1,
                    getHeight(), hilite);
                    }

            // Рисование чисел...
            // Определение цвета и стиля для чисел
            Paint foreground = new Paint(Paint.ANTI_ALIAS_FLAG);
            foreground.setColor(getResources().getColor(
            R.color.puzzle_foreground));
            foreground.setStyle(Style.FILL);
            foreground.setTextSize(height * 0.75f);
            foreground.setTextScaleX(width / height);
            foreground.setTextAlign(Paint.Align.CENTER);
            // Рисование числа в центре тайла
```

```
FontMetrics fm = foreground.getFontMetrics();
// Центровка по оси X: использование выравнивания (и координаты центральной точки)
float x = width / 2;
// Центровка по оси Y: сначала измеряем повышение/понижение
float y = height / 2 - (fm.ascent + fm.descent) / 2;
for (int i = 0; i < 9; i++) {
for (int j = 0; j < 9; j++) {
canvas.drawText(this.game.getTileString(i, j), i
* width + x, j * height + y, foreground);
}
}
// Рисование выделения...
Log.d(TAG, "selRect=" + selRect);
Paint selected = new Paint();
selected.setColor(getResources().getColor(
R.color.puzzle_selected));
canvas.drawRect(selRect, selected);

// Нарисовать подсказки...
// Выбрать цвет подсказки, основываясь на количестве оставшихся ходов
```

```
            if(Preff.getHint(getContext())){
                Paint hint = new Paint();
                int c[] = {
getResources().getColor(R.color.puzzle_hint_0),
                getResources().getColor(R.color.puzzle_hint_1),
                getResources().getColor(R.color.puzzle_hint_2), };
                Rect r = new Rect();
                for (int i = 0; i < 9; i++) {
                for (int j = 0; j < 9; j++) {
                int movesleft = 9 - game.getUsedTiles(i, j).length;
                if (movesleft < c.length) {
                getRect(i, j, r);
                hint.setColor(c[movesleft]);
                canvas.drawRect(r, hint);
                }        }        }    }
        }

        @Override
        protected Parcelable onSaveInstanceState() {
            Parcelable p = super.onSaveInstanceState();
            Log.d(TAG, "onSaveInstanceState" );
            Bundle bundle = new Bundle();
```

```
        bundle.putInt(SELX, selX);
        bundle.putInt(SELY, selY);
        bundle.putParcelable(VIEW_STATE, p);
        return bundle;
    }

    @Override
    protected void onRestoreInstanceState(Parcelable state) {
        Log.d(TAG, "onRestoreInstanceState" );
        Bundle bundle = (Bundle) state;
        select(bundle.getInt(SELX), bundle.getInt(SELY));

super.onRestoreInstanceState(bundle.getParcelable(VIEW_STATE));
    }
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        Log.d(TAG, "onKeyDown: keycode=" + keyCode + ", event="
        + event);
        switch (keyCode) {
            case KeyEvent.KEYCODE_DPAD_UP:
            select(selX, selY - 1);
            break;
```

```
case KeyEvent.KEYCODE_DPAD_DOWN:
select(selX, selY + 1);
break;
case KeyEvent.KEYCODE_DPAD_LEFT:
select(selX - 1, selY);
break;
case KeyEvent.KEYCODE_DPAD_RIGHT:
select(selX + 1, selY);
break;
case KeyEvent.KEYCODE_0:
case  KeyEvent.KEYCODE_SPACE:  setSelectedTile(0);
break;
case KeyEvent.KEYCODE_1: setSelectedTile(1); break;
case KeyEvent.KEYCODE_2: setSelectedTile(2); break;
case KeyEvent.KEYCODE_3: setSelectedTile(3); break;
case KeyEvent.KEYCODE_4: setSelectedTile(4); break;
case KeyEvent.KEYCODE_5: setSelectedTile(5); break;
case KeyEvent.KEYCODE_6: setSelectedTile(6); break;
case KeyEvent.KEYCODE_7: setSelectedTile(7); break;
case KeyEvent.KEYCODE_8: setSelectedTile(8); break;
case KeyEvent.KEYCODE_9: setSelectedTile(9); break;
case KeyEvent.KEYCODE_ENTER:
```

```
                case KeyEvent.KEYCODE_DPAD_CENTER:
                game.showKeypadOrError(selX, selY);
                break;
                default:
                return super.onKeyDown(keyCode, event);
            }
            return true;
        }
            private void select(int x, int y) {
            invalidate(selRect);
            selX = Math.min(Math.max(x, 0), 8);
            selY = Math.min(Math.max(y, 0), 8);
            getRect(selX, selY, selRect);
            invalidate(selRect);
            }
            @Override
        public boolean onTouchEvent(MotionEvent event) {
            if (event.getAction() != MotionEvent.ACTION_DOWN)
            return super.onTouchEvent(event);
            select((int) (event.getX() / width),
            (int) (event.getY() / height));
            game.showKeypadOrError(selX, selY);
```

```
        Log.d(TAG, "onTouchEvent: x " + selX + ", y " + selY);
        return true;
}
        public void setSelectedTile(int tile) {
        if (game.setTileIfValid(selX, selY, tile)) {
        invalidate();// можно изменить подсказки
        } else {
        // Число не подходит для этого тайла
        Log.d(TAG, "setSelectedTile: invalid: " + tile);
        }
                        Log.d(TAG, "setSelectedTile: invalid: " + tile);
        startAnimation(AnimationUtils.loadAnimation(game,
        R.anim.shake));
}}
```