

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Коммерциялық емес акционерлік қоғамы
« ҒҰМАРБЕК ДӘУКЕЕВ атындағы АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ
БАЙЛАНЫС УНИВЕРСИТЕТІ»
«ІТ – инжиниринг» кафедрасы

«Қорғауға жіберілді»
Кафедра меңгерушісі

_____ (аты – жөні, ғылыми дәрежесі, атағы)

_____ « ____ » _____ 20 ____ ж.
(қолы)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: Мегалолис бағдаршамдарын реттеудің зияткерлік ақпараттық жүйесін құру

Мамандығы: 5В070400 - «Есептеу техникасы және бағдарламамен қамтамасыз ету»

Орындаған: Тұрбеков Қабылет

Тобы: ВТк-16-1

Жетекші: т.ғ.к., доцент, Тусупова Б.Б.

Кеңесшілер:

Экономикалық бөлім: к.э.н., доцент _____ Габелашвили
К.Р.

« ____ » _____ 2020ж.

Өміртіршілігі қауіпсіздігі: к.б.н., доцент _____ Мусаева
Ж.К.

« ____ » _____ 2020ж.

Есептеу техникасын қолдану: аға оқытушы _____ Айтқулов
Ж.С.

« ____ » _____ 2020ж.

Норма бақылаушы: аға оқытушы _____ Абсатарова
Б.Р.

« ____ » _____ 2020ж.

Пікір жазушы: т.ғ.к., доцент _____ Тусупова Б.Б.

« ____ » _____ 2020ж.

Алматы 2020
ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Коммерциялық емес акционерлік қоғамы
**«ҒҰМАРБЕК ДӘУКЕЕВ атындағы АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ
БАЙЛАНЫС УНИВЕРСИТЕТІ»**
Басқару жүйелері және ақпараттық технологиялар институті
Есептеу техникасы және бағдарламалық қамтамасыз ету мамандығы
«IT – инжиниринг» кафедрасы

Дипломдық жобаны орындауға берілген
ТАПСЫРМА

Студент: Тұрбеков Қабылет Алтайұлы

Жоба тақырыбы: Мегалолис бағдаршамдарын реттеудің зияткерлік ақпараттық жүйесін құру

«__» _____ 20__ ж. №_____ университет бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «_____» _____ 2020 ж.

Жобаға бастапқы деректер (талап етілетін жоба нәтежелерінің параметрлері және нысанның бастапқы деректері): Бұл дипломдық жоба Мегалолис бағдаршамдарын реттеудің зияткерлік ақпараттық жүйесін құру үшін негізделген.

Диплом жобасындағы әзірленуі тиіс сұрақтар тізімі немесе диплом жобасының қысқаша мазмұны:

- а) Пәндік саланы талдау;
- б) Программалық қамтаманы жобалау;
- в) Программалық қамтаманы құру;
- г) Экономикалық бөлім;
- д) Өміртіршілік қауіпсіздігі;
- е) А қосымшасы. Программа мәтіні;

Негізгі ұсынылатын әдебиеттер:

1. Python is a programming language // Сайттың электрондық нұсқасы <https://www.python.org/>
2. Cameron Davidson-Pilon. Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference // Addison-Wesley Data & Analytics, 2015.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep learning
4. Christopher Bishop. Pattern Recognition and Machine Learning

Жоба бойынша бөлімшелерге қатысты белгіленетін кеңесшілер

Бөлімшелер	Кеңесшілер	Мерзімі	Қолы
Экономикалық бөлім	Габелашвили К.Р.	15.04.2020 – 30.04.2020	
Өміртіршілігі қауіпсіздігі	Мусаева Ж.К.	15.04.2020 – 30.04.2020	
Бағдарламалық қамтама	Айтқулов Ж.С.	13.05.2020 – 18.05.2020	
Норма бақылау	Абсатарова Б.Р.	13.05.2020 – 18.05.2020	

Диплом жобасын дайындау
КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Теориялық бөлім	17.02.2020 - 16.03.2020	
2	Бағдарламалық қосымшаны жобалау	17.03.2020 - 05.04.2020	
3	Қосымша әзірлеу бөлімі	06.04.2020 - 09.05.2020	

Тапсырманың берілген уақыты «__» _____ 20__ ж.

Кафедра меңгерушісі _____ Досжанова А.А.

Жоба жетекшісі _____ Тусупова Б.Б.

Орындалатын тапсырманы қабылдаған студент _____ Тұрбеков Қ.А.

АҢДАТПА

Қазіргі таңда заманауи технологиялардың дамуына байланысты күнделікті өмірде бұл технологиялар өте қажетті заттардың бірі болып табылады. Себебі бүгінде жасалынып жатқан сервистер адамзат өмірінде орын алатын тіршіліктерді жеңілдетуде.

Дипломдық жұмыстың мақсаты күнделікті өмірге өте қажетті және адамдардың уақытын үнемдейтін ақылды бағдарламаларды реализация жасайтын программалық қамтама болып табылады.

Ақылды бағдарламалардың көмегі әсіресе үлкен қалаларда яғни мегаполистерді көп септігін тигізетін болады. Көліктердің көптігіне байланысты үлкен қалаларда кептелістер жиі орын алады. Соның салдарынан төтенше жағдай орын алғанда, жедел-жәрдем, өрт сөндіру және т.б көліктер уақытында төтенше жағдай орын алған жерге жетуге үлкен көмегін тигізеді. Программалық қамтама қаладағы кептелістерді азайта отырып және қаладағы жол ережесін бұзушыларды бақылайтын болады.

Бұл программалық жабдықтама коммерциялық немесе коммерциялық емес идеяда қолдануға болатын бағдарлама және алдыңғы уақытта да жетілдіріледі.

АННОТАЦИЯ

Сегодня, благодаря развитию современных технологий, эти технологии являются одной из самых важных вещей в повседневной жизни. Потому что услуги, которые разрабатываются сегодня, облегчают жизнь людям.

Целью дипломной работы является внедрение интеллектуальных светофоров, которые необходимы для повседневной жизни и экономят время людей.

Помощь умных светофоров будет особенно полезна в крупных городах, например мегаполисах. Пробки распространены в крупных городах из-за большого количества транспортных средств. В результате, в случае возникновения чрезвычайной ситуации, машины скорой помощи, пожарные машины и т. Д. Могут оказать большую помощь в доставке на место чрезвычайной ситуации. Программное обеспечение уменьшит пробки на дорогах в городе и контролирует нарушителей правил дорожного движения в городе.

Это программное обеспечение представляет собой программу, которая может использоваться в коммерческих или некоммерческих целях и будет улучшена в будущем.

ANNOTATION

Today, thanks to the development of modern technologies, these technologies are one of the most important things in everyday life. Because the services that are being developed today make life easier for people.

The purpose of the thesis is the introduction of intelligent traffic lights, which are necessary for everyday life and save people time.

The help of smart traffic lights will be especially useful in large cities, such as megacities. Traffic jams are common in large cities due to the large number of vehicles. As a result, in the event of an emergency, ambulances, fire engines, etc. can be of great help in delivering the emergency situation. The software will reduce traffic congestion in the city and controls violators of traffic rules in the city.

This software is a program that can be used for commercial or non-commercial purposes and will be improved in the future.

МАЗМҰНЫ

Кіріспе	8
1 Жалпы бөлім	9
1.1 Жүйе туралы түсінік	9
1.2 Мегалополис бағдаршамдарын реттеудің ақпараттық жүйесінің мақсаты, міндеті	11
1.3 Мегалополис бағдаршамдарын реттеудің ақпараттық жүйесін құру үшін қолданылатын технологиялар	14
1.4 Python программалау тілінің кітапханалары мен фреймворктары	16
2 Жобалау бөлімі	22
2.1 UML бірыңғайландырылған модельдеу тілі	22
2.2 Rational Rose құралы	25
2.3 Тізбек диаграммасы	26
2.4 Прецеденттер диаграммасы	31
2.5 Класстар диаграммасы	34
3 Қолданбалы бөлім	38
3.1 Мегалополис бағдаршамдарын реттеудің ақпараттық жүйесінің программасы	38
4 Экономикалық бөлім	46
4.1 Еңбек ресурстары	46
4.2 Техникалық жабдықтар	48
4.3 Техникалық-бағдарламалық камтамасыз ету	48
4.4 Салық шығындарын есептеу	49
4.5 Амортизацияны есептеу	50
4.6 Электр энергиясына шығындарды есептеу	50
4.7 Үстеме есептеу	51
4.8 Даму шығындарының сметасы	52

4.9 Жобаны іске асыру бағасы	53
5 Өміртіршілік қауіпсіздігі	56
5.1 Жұмыс орнының эргономикасы және еңбек жағдайларын талдау.....	56
5.2 Жасанды жарықтандыруды нүктелік әдіспен есептеу.....	64
Қорытынды	70
Әдебиеттер тізімі	71
А қосымшасы	72
Б қосымшасы	77

КІРІСПЕ

Бүгінгі трафик проблемасының басты себептерінің бірі - бұл қозғалысты басқаруда қолданылатын әдістер. Тікелей трафик сценарийіне басты назар аударылмайды, бұл трафикті басқарудың тиімсіз жүйелеріне әкеледі. Бұл трафик таймерлері алдын ала белгіленген уақытты көрсетеді. Егер бағдарламашы таймерлері қозғалысты реттейтін уақытты көрсетсе, қажет емес жасыл сигналдарға кететін уақыт үнемделеді. Әр жолақ үшін таймер - бұл трафикті басқарудың қарапайым тәсілі. Егер сол таймерлер нақты уақытты болжаса, онда автоматты түрде жүйе тиімдірек болады. Бұл жоба трафикті басқару жүйесін нақты уақыттағы трафиктің жағдайына байланысты шешім қабылдау тұрғысынан ақылды етуге бағытталған. Бұл шешімдер қызыл, жасыл немесе сары жарықтандырудың әдеттегі жүйесі көліктің қозғалысын барлық бағытта қалай қарайтынына қарамай, көлік құралдарының қозғалысын басқарудың қарапайым кестелерін қолданудың орнына, жолдағы көлік құралдарының қазіргі көлеміне негізделгендей болуы керек. Бұл жобада басқару жүйесімен қозғалатын барлық көліктер оңай анықталатын және әртүрлі көлік құралдарынан алынған деректерді сұрыптауға және көлік құралын бақылауға болатындай етіп, жекеленген құралдардың көмегімен оңай ажыратуға болады. Қазіргі трафикті басқару жүйесін жетілдірумен және оның трафикті басқарудағы тиімділігімен қатар, бұл жұмыс төтенше жағдай кезінде логистиканы жақсартуға және төтенше жағдай орын алған жағдайда қажетсіз кептелістерді азайтуға айтарлықтай ықпал етеді. Сонымен қатар, бұл жүйе жол апаттары болуының ықтималдылығын азайтады және жол ережесін бұзушылар мен қылмыскерлерді іздеу үшін де қолданылуы мүмкін. Сондықтан да, бұл жүйенің тиімділігі өте көп.

1 Жалпы бөлім

1.1 Жүйе туралы түсінік

Интеллектуалды көлік жүйелері мен қызметтерін ақпараттық-коммуникациялық технологияларды көлік инфрақұрылымымен, көлік құралдарымен және пайдаланушылармен біріктіру ретінде анықтауға болады. Урбанизация мен көлік кептелісінің артуы біздің көлік жүйелерімізді барынша тиімді пайдалану үшін жедел қажеттілік туғызады. Нақты уақыттағы трафикті басқару - бұл жол желісін оңтайлы пайдалануға қол жеткізуге бағытталған заманауи қалалық трафикті басқару жүйесінің ажырамас бөлігі. Күрделі трафик желісі үшін нақты уақыт режиміндегі трафикті тиімді басқару қиын болуы мүмкін. Сигналдық жүйенің жұмысы трафиктің сигналдық жүйесін интеграцияланған мультимодальды тасымалдау жүйесінің шағын құрамдас бөлігі ретінде қарастыратын соңғы тенденциямен қиындатады. Көшеде және көлік жолдарында қозғалыстың тиімді жүруі үшін жол сигналдары мен басқа басқару құрылғыларын оңтайландыру зияткерлік көлік жүйесінің қозғалыс менеджментінің дамыған жүйесінің маңызды бөлігі болып табылады. Трафикті басқарудың кеңейтілген жүйесі үшін трафиктің жай-күйін бағалау және оның бірқалыпты жүріп жатқанын айту қиын болуы мүмкін. Соңғы бірнеше жылда көп агенттік жүйелер әр түрлі, гетерогенді және таратылған ақпарат көздерінің өсіп келе жатқан қол жетімділігін тиімді пайдалану үшін шешуші технологияға айналды. Зерттеушілер бірнеше жылдар бойы көптеген әдістерді қолданды және проблемалық домендеріне көп агенттік жүйелерді енгізу үшін әртүрлі құралдарды қолданды. Зерттеушілер осы автономды көп агенттік жүйелерді жақсырақ түсіне отырып, олардың жұмысын жақсарту үшін көптеген мүмкіндіктер енеді және жетілдірілген жүйелер одан да күрделі қолданбалы домендер үшін қолданыла алады. Интеллектуалды бағдарламалық жасақтама агенті - дербес компьютерлік бағдарлама, ол белгілі бір компьютерге қатысты тапсырмаларды орындауда және соңғы пайдаланушымен өзара әрекеттеседі. Кез-келген агентте әрқашан белгілі бір ақыл-ой деңгейі болады. Интеллект деңгейі алдын-ала анықталған рөлдер мен оқу орындарына байланысты өзгеруі мүмкін. Мульти-агенттік жүйе - бұл үлкен жүйені бір-бірімен байланысатын және үйлестіретін және оны оңай кеңейте алатын бірнеше кіші жүйелерге ыдырайтын агенттердің жиынтығы. Агенттерге негізделген модельдеу дегеніміз - бірнеше субъектілер өздерінің жергілікті ортадағы жағдайларға тұрақты түрде жауап беретін, күрделі жүйенің күрделі әрекетін еліктейтін модельдер. Қалалық көлік жүйесі - бұл көптеген нысандар қатысатын өте күрделі жүйе және олардың арасындағы қарым-қатынас күрделі. Сондықтан, оқушы агент үшін ең маңызды мәселелер - бұл басқа агенттердің мінез-құлқы мен интеллект деңгейін бағалау. Интеллектуалды қозғалыс жүйесі адамдарға маңызды ақпараттармен бөлісе отырып, көліктік желілерден көбірек қауіпсіздікті және қоршаған ортаға аз әсер етуге

мүмкіндік береді. Интеллектуалды қозғалыс жүйесі бүкіл көлік жүйесіне тиімді және тиімді жұмыс істеуге көмектеседі. Интеллектуалды қозғалыс жүйесі пайдаланушыларды, көлік жүйелерін және көлік құралдарын қазіргі заманғы ақпараттық-коммуникациялық технологиялар арқылы біріктіреді. Интеллектуалды қозғалыс жүйесі саяхатшылардың қауіпсіздігін, тиімділігі мен жайлылығын айтарлықтай жақсартып алады. Интеллектуалды қозғалыс жүйесі жүк жөнелтушілер мен тасымалдаушыларға жүкті белгіленген жерге сенімді және тиімді жеткізуге көмектеседі. Интеллектуалды қозғалыс жүйесі көлік жүйелерін басқаратын адамдарға, яғни клиенттерге сапалы қызмет көрсетуге көмектеседі. Және де интеллектуалды қозғалыс жүйесі бізде бар жол және теміржол жүйелерінен ең жақсы құндылықты алуға көмектеседі. Ақылды қозғалыс жүйесі трафиктің ағынын біртіндеп, кідірістерді, отын шығынын, ауа мен шудың ластануын азайтады. Интеллектуалды қозғалыс жүйесі қоғамдық көліктерді ыңғайлы және қол жетімді етуге көмектеседі. Интеллектуалды қозғалыс жүйесі жиі сағана артында жұмыс істейді. Апаттық көліктер апат болған жерге тезірек жете алады, интеллектуалды қозғалыс жүйесі апатты анықтап, төтенше жағдай қызметтерін хабардар етіп, ең жақын әрекет ету бөлімін ала алады. Адамдар өздерінің қымбат уақытының көп бөлігін жұмыс, мектеп, сауда және әлеуметтік іс-шараларға жұмсайтын болады, сондай-ақ бағдаршаммен айналысуға жұмсайды. Жедел жәрдем мен жедел жәрдем көлігі (полиция, өрт сөндіру қызметі және т.б.) кез-келген шақыру орнына кептеліссіз жете алатын болады. Ірі жолдарда қозғалыс барлық бағыттарда әрқашан жұмыстан бұрын, жұмыс уақытында немесе одан кейінгі уақытта ауыр болады. Ең жаманы, әуе жолдарын (орташа адам үшін қымбат болуы мүмкін), су жолдарын (паромдар мен қайықтардың жағдайына байланысты үшінші әлем елдерінде қауіпті болуы мүмкін) немесе темір жолдарды (мысалы, әуе жолдарын) пайдалануда балама көлік жүйесін пайдалануда қиындықтар бар. Біз сияқты үшінші әлемдегі қарапайым және архаикалық дизайн. Осы басқа балама қол жетімді және пайдалану мүмкін болған жағдайда да, жол бойында көлікпен жүру әрқашан қажет болады. Автомобиль көлігіне сұраныс артқан сайын, көліктер жолды пайдалану үшін бәсекеге түсетін орындарға жетеді. Бұл көлік кептелісіне әкеледі. Бағдаршамның көптеген жүйелері уақыт аралық механизмде жұмыс істейді, ол белгілі бір уақыт аралығында түстерін өзгертеді. Бұл белгілі бір жағдайларда тиімсіз болуы мүмкін. Таймерді басқару тетігі ашық ілмекті басқару жүйесінің бір түрі болғандықтан, басқарылатын айнымалыдан (қозғалыс көлемі) контроллерге кері байланыс жоқ (жарықтандыруды қолдана отырып, трафикті басқару жүйесі). Мысалы, жасыл түстің ұзақтығы әрқашан жолдың кептелу-болмауына бірдей болады. Бұл проблемаларды шешу үшін басқарудың жаңа жүйесін құрған дұрыс; ақылды және ақылды басқару жүйесі. Ақылды бағдаршам жүйесі көлік құралдарының болуын немесе болмауын сезеді және соған сәйкес әрекет етеді. Интеллектуалды бағдарлау жүйелерінің идеясы жүргізушілер бағдаршамның өзгеруін күтіп, қажетсіз уақытты жұмсамайды. Интеллектуалды қозғалыс жүйесі трафикті әртүрлі жолдармен

анықтайды. Барлық технологиялық енгізілімдермен бірге көлік кептелісі толығымен шешілмеген және бұл бізді күнделікті жұмысқа (және одан) жолда қарсы алатын ұзын-сонар кезектер мен кептелістерден көрінеді. Жылдар бойына трафикті басқарудың бірнеше жүйесі ойлап табылғанына қарамастан, бұл басқару элементтерін ақылды ету қажеттілігіне баса назар аударуға болмайды. Бұл жоба зияткерлік көлік жүйесі (ITS) және қозғалыс экономикасы бойынша бірқатар ғылыми-зерттеу жұмыстарымен тығыз байланысты. Ол енгізілетін логикалық блоктардан басқа, қазіргі кездегі трафикті басқару инфрақұрылымы мен жарықтандыру жүйесін пайдаланады. Бүкіл идея - қазіргі трафикті басқару жүйесін кері байланыс желісімен теңестіру, әрі осылайша оны ақылды және ақылды ету.

1.2 Мегалополис бағдарламдарын реттеудің ақпараттық жүйесінің мақсаты, міндеті

Қазіргі замандағы ақпараттық немесе цифрлық технологиялардың дамуына орай көптеген программалық қызметтер даму үстінде. Бұл уақытқа дейін адамдар көбіне механикалық жүйелерді пайдаланып келді. Бірақ заманның дамуына байланысты уақыт өте келе механикалық жүйелерді пайдалану адамдарға қолайсыздық тудыра бастады. Нақтырақ айтатын болсақ, қаладағы бағдарламдар. Қаладағы бағдарламдар ең басында құрастырылған жүйе бойынша жұмыс жасап келе жатыр. Дегенмен, ол кездегі заманмен қазіргі заманды салыстыруға келмейді. Ол кезде қала ішіндегі қозғалыс бүгінгі күнгі қозғалысқа қарағанда әлдеқайда аз болатын. Сол үшін де ол бағдарламдар сол кездегі қозғалысты басқара алған. Бірақ, бүгінгі таңда көліктердің көбейуіне байланысты бұл бағдарламдар өз жұмысын дұрыс жасамайды және соның салдарынын қалада үлкен кептелістер орын алады. Бүгінгі трафик проблемасының басты себептерінің бірі - бұл қозғалысты басқаруда қолданылатын әдістер. Тікелей трафик сценарийіне баса назар аударылмайды, себебі бұл трафикті басқарудың тиімсіз жүйелеріне әкеледі. Бұл трафик таймерлері алдын ала белгіленген уақытты көрсетеді. Егер бағдарлам таймерлері қозғалысты реттеудің дұрыс уақытын көрсетсе, қажет емес жасыл сигналдарға кететін уақыт үнемделеді. Әр жолақ үшін таймер - бұл трафикті басқарудың қарапайым тәсілі. Егер сол таймерлер нақты уақытты болжаса, онда автоматты түрде жүйе тиімдірек болады. Бұл жоба трафикті басқару жүйесін нақты уақыттағы трафиктің жағдайына байланысты шешім қабылдау тұрғысынан ақылды етуге бағытталған. Бұл шешімдер қызыл, жасыл немесе сары түске айналатын кәдімгі жарықтандыру жүйесі қозғалыс жолындағы көлік құралдарының қазіргі көлеміне негізделетіндей болуы керек. Жобаның негізгі міндеттері:

- Жол қиылыстарында физикалық өзгерістердің аздығымен, қозғалыс кезіндегі жағдайларға оңай бейімделетін, арзан электронды компоненттерді қолдана отырып, нақты уақыт режимінде трафикті

- басқарудың қарапайым, сенімді автоматтандырылған жүйесін жобалау мүмкіндігін зерттеу;
- Жол қиылысында барлық бағыттар бойынша көлік құралдары мен жаяу жүргіншілер қозғалысының жылдам өтуін қамтамасыз ету мүмкіндігін зерттеу. Электрондық компоненттердің жұмыс қабілеттілігі мен дәлдігін зерттеу;
- Жол сақшысының жұмысын азайту;
- Жобаның пайдалылығы мен дәрежесін зерттеу;
- Болашақта жетілдіруге болатын практикалық шешім ұсыну;

Бұл жобада басқару жүйесімен қозғалатын барлық көліктер оңай анықталатын және әртүрлі көлік құралдарынан алынған деректерді сұрыптауға және көлік құралын бақылауға болатындай етіп, жекеленген құралдардың көмегімен оңай ажыратуға болады деп болжанады. Бір жолақты қозғалыс модельдеуде қолданылады. Қазіргі трафикті басқару жүйесін жетілдірумен және оның трафикті басқарудағы тиімділігімен қатар, бұл жұмыс төтенше жағдай кезінде логистиканы жақсартуға және іскери операциялардың қажетсіз кідірістерін азайтуға айтарлықтай ықпал етеді деп күтілуде. Басқа күтулерге тоқталатын болсақ. Олар:

- ықтимал соқтығысу немесе апат жағдайларын азайтуға көмектеседі, сол арқылы автомобиль жолдарын пайдаланатын жүргізуші мен жаяу жүргіншінің сенімін жақсартады;
- қозғалыс жүйесінің бағыттарына байланысты түйіспелердегі уақытты жоғалту қорқынышын болдырмайды, өйткені жүйе тәулік бойы үздіксіз жұмыс істей алады, осылайша трафикті басқару қызметі үнемі қол жетімді болады;
- Төтенше жағдайларды жоюға және арнайы қызметтегі көліктерді тез тазартуға көмектеседі; Бұл адамның күш-жігерін және күш-қуатын азайтады;
- Сонымен қатар оны жол ережесін бұзушылар мен қылмыскерлерді іздестіру үшін қолдануға болады;

Жол қозғалысының инженерлік зерттеулерінің арқасында кезекті бағдаршамның моделін алуға мүмкіндік берді, онда көлік құралдары бағдаршаммен басқарылатын қиылысқа келіп, кезекке тұрады. Бірнеше зерттеу жұмыстары көше ені бойынша әр жолақта кезектің ұзындығын және тәуліктің белгілі бір уақытында күтілетін көлік құралдарының санын бағалауға бейімделген әртүрлі әдістерді жасады. Алайда кезек моделіндегі бағдаршамның тиімділігіне көлік құралдарының ағынына кедергі келтіретін көлік құралдарының бұзылуы немесе жол-көлік оқиғалары сияқты күтпеген оқиғалардың туындауы әсер етті. Кезек моделіне негізделген осы әдістердің ішінде кезектерді анықтау алгоритмі болды. Алгоритм қозғалыс және көлік құралдарын анықтау операцияларынан тұрды, олардың екеуі де жарық жағдайындағы вариациялардың әсерін азайту үшін көріністің жиегін алуға

негізделген. Нақты уақыт режимінде және қозғалыс сигналдарын бақылауға арналған, білімге негізделген жүйені Файнлер және басқалар (1997) сипаттаған. Бұл жүйе екі үрдістегі оқу жүйесі болды. Бірінші процесс бір жол қиылысында және көше тораптарындағы тұрақты қозғалысты басқаруды оңтайландырды, ал екінші саты трафиктің кенеттен өзгеруіне жауап беру кезінде болжалды, яғни кенеттен реактивті басқарумен айналысты. Орталықтандырылмаған басқару моделі 1999 жылы сипатталған. Бұл модель әр түрлі бағыттарға бару құны тұжырымдамасына негізделген көп бағытты бағдарлау мен нақты уақыттағы бағдаршамның үйлесімі болды. Электрондық трафигтік сигнал болашақ интеллектуалды тасымалдау ортасында дәстүрлі бағдаршам жүйесін жетілдіреді деп сенеді, өйткені оның машиналарға оңай көрінуінің артықшылығы барын Хуанг пен Миллер (2004) сипаттаған. Олардың жұмысы трафиктің негізгі электронды хаттамалық негізін және оның екі туындысын, жол қиылыстарына арналған сенімді протоколды және аялдама сигналдарына арналған бір протоколды ұсынды. Бұл хаттамалар алушы көліктерге шағылысудан туындаған мүмкін шатасуларға қарамастан, сигналдың бағытын анық анықтауға мүмкіндік берді. Сондай-ақ, авторлар қосымшалардың біреуін үлгі қосымшасын құру үшін қалай пайдалану керектігін көрсетті: қызыл жарықпен дабыл беру жүйесі, сонымен қатар орналасу жүйесінің белгісіздігінен туындаған ықтимал сәйкессіздік қауіптері және оларды басқару құралдарын талқылау туралы мәселелерді көтерді. Қазір көптеген елдер қалалардағы көлік жүйесіне әсер ететін және күрделі дилемма тудыратын көлік кептелісі проблемаларынан зардап шегуде. Жол қозғалысының офицерлері мен флагманын автоматты қозғалыс жүйелеріне ауыстырғанына қарамастан, әсіресе кептеліс түйіндері болған кезде, кептелісті оңтайландыру әлі де маңызды мәселе болып отыр. Автомобильдер санының тез артуы және жолды пайдаланушылар санының үнемі өсуі жеткілікті ресурстары бар алға тартылған инфрақұрылыммен қатар жүрмейді. Жаңа жолдар салу, айналмалы және айналмалы жолдар салу, айналма жолдар құру және жолдарды қалпына келтіру арқылы ішінара шешімдер ұсынылды. Алайда, жол проблемасы әр түрлі параметрлерді тартуға байланысты өте күрделі. Біріншіден, трафик ағымы көбінесе таңертең және күндізгі уақытта трафиктің ең көп уақыты болатын күннің уақытына байланысты болады; демалыс күндері ең төменгі жүктеме көрсетілген аптаның күндері, дүйсенбі және жұма күндері, әдетте, қалалардан шет аймақтарға және кері бағытқа бағытталған тығыз қозғалыс байқалады, сәйкесінше демалыс және жаз сияқты жылдың уақыты. Екіншіден, бағдаршамның қазіргі жүйесі қатаң кодталған кідірістермен жүзеге асырылады, онда шамдардың ауысу уақыты ұяшықтары тұрақты түрде орнатылады және нақты уақыт қозғалысының ағымына тәуелді болмайды. Үшінші мәселе - қиылыстағы бір жарықтың жай-күйі, оған іргелес қиылыстардағы қозғалыс ағымына әсер етеді. Сондай-ақ, кәдімгі қозғалыс жүйесі көлік кептелісін нашарлататын апаттар, жол жұмыстары және бұзылған автомобильдер жағдайын қарастырмайды. Сонымен қатар, жедел жәрдем, құтқару машиналары, өрт сөндіру бригадасы, полиция және VIP

адамдар сияқты үлкен приоритетті көліктердің қиылысы арқылы адамдардың қозғалыста болуы проблеманың маңызды мәселесі болып табылады. Сонымен, жолақтармен өтетін жаяу жүргіншілер жол қозғалысы жүйесін де өзгертеді. Кәдімгі қозғалыс жүйесін көліктің ауыр кептелісін шешу, көлік проблемаларын жеңілдету, қозғалыс көлемі мен күту уақытын азайту, жалпы жүру уақытын азайту, автомобильдердің қауіпсіздігі мен тиімділігін арттыру, денсаулық сақтау, экономикалық және экологиялық салалардағы артықшылықтарды кеңейту үшін жаңарту қажет. Жиналу мәселесінің ең дұрыс шешімі - жиналған тараптардың ең алдымен қиылысты кесіп өтуіне мүмкіндік беру.

1.3 Мегалополис бағдарламдарын реттеудің ақпараттық жүйесін құру үшін қолданылатын технологиялар

Жоғарыда ұсынылған шешімді Python-дағы OpenCv кітапханасының көмегімен немесе суреттегі автомобильдердің санын есептеу, сонымен қатар әр түрлі көлік түрлерін ажырату үшін объектілерді анықтау арқылы машиналық оқыту арқылы жүзеге асырылуы мүмкін.

Python - интерпретацияланған, жоғары деңгейлі, жалпы мақсаттағы бағдарламалау тілі. Гидо ван Россум жасаған және 1991 жылы алғаш шығарылған Python-ның дизайн философиясы кодтың оқылымдылығын, оның кендігінің айтарлықтай қолданылуын баса көрсетеді. Оның тілдік құрылымы мен объектіге бағытталған тәсілі программисттерге кіші және ірі жобаларда нақты, логикалық код жазуға көмектеседі. Python динамикалық түрде теріледі және қоқыс жинайды. Ол бірнеше бағдарламалау парадигмаларын қолдайды, соның ішінде процедуралық, объектілік-бағытталған және функционалды бағдарламалау. Python кеңейтілген стандартты кітапхананың арқасында көбінесе «аккумулятормен жабдықталған» тіл ретінде сипатталады. Python 1980 жылдардың аяғында ABC тілінің ізбасары ретінде пайда болды. 2000 жылы шығарылған Python 2.0 тізімдер мен анықтамалық циклдерді жинауға қабілетті қоқыс жинау жүйесі сияқты функцияларды ұсынды. 2008 жылы шығарылған Python 3.0, тілге түбегейлі сәйкес келмейтін, және Python 2-нің көптеген коды Python 3-те өзгертілмеген. Python 2 тілі, яғни Python 2.7.x ресми түрде 2020 жылдың 1 қаңтарында тоқтатылды (алғаш рет 2015 жылға жоспарланған), содан кейін ол үшін қауіпсіздік патчтары мен басқа да жетілдірулер берілмейді. Python 2 өмірінің соңына дейін тек Python 3.5x және одан кейінгі нұсқаларға қолдау көрсетіледі.

Python интерпретаторлары көптеген операциялық жүйелер үшін қол жетімді. Программисттердің жаһандық қоғамдастығы CPython-ды қолданады, оған ашық негізде сілтеме жасалады. Python Software Foundation коммерциялық емес ұйымы Python және CPython дамыту үшін ресурстарды басқарады және басқарады.

```
19 guess_me = 7 ✓
20 if guess_me < 7:
21     print('too low') ✓
22 elif guess_me > 7:
23     print('too high')
    File "<ipython-input-3-ff33f686ae87>", line 1
        elif guess_me > 7:
            ^
SyntaxError: invalid syntax
24 else:
25     print('just right')
```

1.1-сурет – Python программалау тілінің синтаксисі

Python тілінің дамуы 1980 жылдардың соңында басталды, Голландиялық CWI институтының мүшесі Гуидо ван Россум. Үлестірілген Amoeba OS кеңейтілген сценарий тілін қажет етті, Гуидо бос уақытында Python-ды жаза бастады, ABC тілі үшін негіздердің бір бөлігін алды (Гуидо бағдарламалауды үйретуге бағытталған осы тілдің дамуына қатысты). 1991 жылдың ақпанында Гуидо бастапқы мәтінді alt.sources жаңалықтар тобында жариялады. Басынан бастап Python объектіге бағытталған тіл ретінде жасалынған.

Мейірімді, жауапкершілікті пайдаланушылар қоғамдастығы Гуидоның дизайнерлік түйсігі, Python-дың сәттілік факторларының бірі болып саналады. Тілдің дамуы PER құжаттарын жасаудың, талқылаудың, таңдаудың және жүзеге асырудың нақты реттелген процесі бойынша жүзеге асырылады (Python Enhancement Proposal) - Python дамыту бойынша ұсыныстар.

2008 жылғы 3 желтоқсанда, кең тестілеуден кейін Python 3000-ның алғашқы нұсқасы шығарылды (немесе Python 3.0, Py3k аббревиатурасы да қолданылады). Python 3000 көптеген архитектуралық кемшіліктерді Python-дың ескі нұсқаларымен үйлесімділіктің максималды (бірақ толық емес) сақталуымен шешеді. Бүгінгі таңда екі дамуға қолдау көрсетіледі (Python 3.x және 2.x), бірақ Python 2.7 қолдау 2020 жылы аяқталады.

Python динамикалық теруді қолдайды, яғни айнымалы түрі тек жұмыс уақытында анықталады. Сондықтан, «шаманы айнымалыға берудің» орнына «мәнді белгілі бір атпен байланыстыру» туралы сөйлескен жөн. Python-да кіріктірілген типтер бар: логикалық, жолдық, Юникодтық жол, дәлдіксіз бүтін сан, өзгермелі нүкте нөмірі, күрделі сан және басқалар. Python-дағы жинақтардан: тізім, тупле (өзгермейтін тізім), сөздік, жиынтық және басқалары жинақталған. Барлық мәндер объектілер, соның ішінде функциялар, әдістер, модульдер, кластар. Сіз жаңа типті классты жазу арқылы немесе кеңейту модуліндегі жаңа типті анықтай аласыз (мысалы, C түрінде жазылған). Класс жүйесі мұрагерлікті (жалғыз және көп) және метапрограммалауды қолдайды. Кірістірілген көптеген түрлерден және кеңейту түрлерінен мұрагерлік мүмкін.

Барлық нысандар анықтамалық және атомдық болып бөлінеді. Атомдыққа `int`, `long` (3-нұсқада `int`-тің кез-келген саны кіреді, өйткені 3-нұсқада өлшем шегі жоқ), күрделі және басқалары бар. Атомдық объектілерді тағайындау кезінде олардың мәні көшіріледі, ал сілтеме үшін объектіге сілтеме ғана көшіріледі, сондықтан тағайындаудан кейінгі екі айнымалы мән де бірдей мәнді қолданады. Анықтамалық нысандар өзгермелі және өзгермейтін. Мысалы, ішектер мен құлыптар өзгермейтін болып табылады, ал тізімдер, сөздіктер және көптеген басқа объектілер өзгеріске ұшырайды. Python-дағы үзінді - бұл өзгермейтін тізім. Көптеген жағдайларда торлар тізімдерге қарағанда тезірек жұмыс істейді, сондықтан егер сіз реттілікті өзгертуді жоспарламасаңыз, онда оларды қолданған дұрыс.

Енді жоғарыда айтып өткен OpenCV кітапханасына қысқаша түсінік беріп кетейін.

1.4 Python программалау тілінің кітапханалары мен фреймворктары

1.4.1 OpenCV

OpenCV-Python – бұл компьютердің көру мәселелерін шешуге арналған Python байланыстырушы кітапханасы. OpenCV-Python NumPy-ді қолданады, ол MATLAB стиліндегі синтаксисі бар сандық операциялар үшін өте оңтайландырылған кітапхана болып табылады. Барлық OpenCV массив құрылымдары NumPy массивіне және одан түрлендіріледі. OpenCV-ды Intel-де 1999 жылы Гари Бадский бастаған, ал алғашқы шығарылымы 2000 жылы шыққан. Вадим Писаревский Гари Бадскийге Intel-дің ресейлік OpenCV бағдарламалық жасақтамасын басқару үшін қосылды. 2005 жылы OpenCV 2005 жылы DARPA Grand Challenge жеңімпазы болған Stanley-де қолданылды. Кейін оның белсенді дамуы жоба жетекшісі Гари Бадский мен Вадим Писаревскийдің қолдауымен Willow Garage қолдауымен жалғасты. Қазір OpenCV компьютерлік көру және машиналық оқытуға қатысты көптеген алгоритмдерді қолдайды және күннен-күнге кеңейіп келеді. OpenCV C ++, Python, Java және т.б. бағдарламалаудың көптеген түрлерін қолдайды және Windows, Linux, OS X, Android және iOS сияқты әртүрлі платформаларда қол жетімді. CUDA және OpenCL негізінде жоғары жылдамдықты GPU операциялары үшін интерфейстер де белсенді дамуда. Бұл жобада қолданатын технологиялар тек осы кітапханамен ғана тоқталып қалмайды. Олар: TensorFlow, NumPy, SciPy, Matplotlib, pillow, H5py, Keras және ImageAI. Енді осылардың әрқайсысына қысқаша тоқталып өтетін болсам:

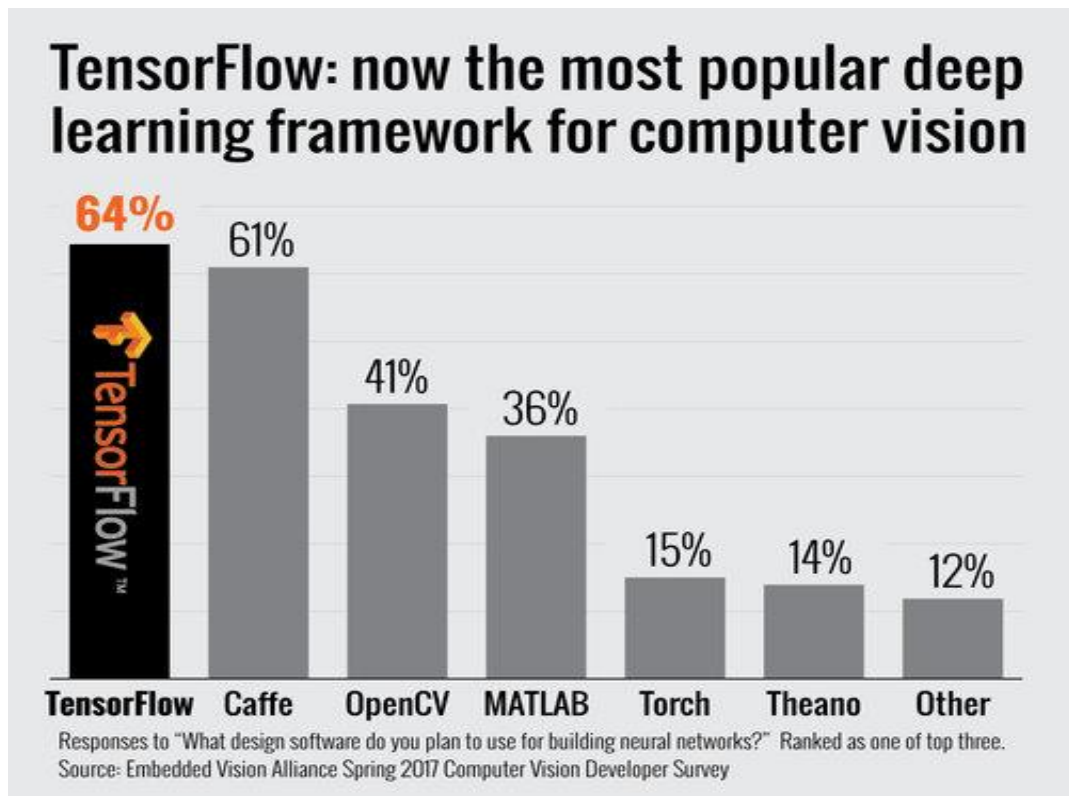
1.4.2 TensorFlow

TensorFlow - бұл машинада оқуға арналған ашық бастапқы платформа. Онда құралдардың, кітапханалардың және қоғамдық ресурстардың жан-

жақты, икемді экожүйесі бар, бұл зерттеушілерге ML-дегі ең жаңа жағдайларды жасауға және әзірлеушілерге ML-мен жұмыс істейтін қосымшаларды оңай құруға және қолдануға мүмкіндік береді. TensorFlow - ақпараттар ағынына және әртүрлі міндеттерді орындауға арналған бағдарламалауға арналған ашық және ашық бағдарламалық қамтамасыздандыру кітапханасы. Бұл математикалық символикалық кітапхана, сонымен қатар нейрондық желілер сияқты машиналық оқу үшін қолданылады. Ол Google-да зерттеу үшін де, өндіріс үшін де қолданылады. TensorFlow - бұл Google Brain екінші буын жүйесі. 1.0.0 нұсқасы 2017 жылдың 11 ақпанында шығарылды. Анықтамалық нұсқаулық бір құрылғыда жұмыс істеген кезде, TensorFlow бірнеше процессорлар мен GPU-де жұмыс істей алады (графикалық өңдеу қондырғыларында жалпы мақсатты есептеу үшін CUDA және SYCL кеңейтімдері бар). TensorFlow 64 биттік Linux, macOS, Windows және Android және iOS қоса мобильді есептеу платформаларында қол жетімді. Оның икемді архитектурасы әр түрлі платформаларда (CPU, GPU, TPU) және жұмыс үстелінен бастап серверлер кластерлеріне мобильді және жиекші құрылғыларда есептеулерді оңай орналастыруға мүмкіндік береді. Және де Tensorflow таңдағанымның тағы да бір себептері:

- Оңай модель құру - ML модельдерін жылдам модельдеуді тездетуге және оңай шешуге мүмкіндік беретін Керас сияқты интуитивті жоғары деңгейлі интуитивті API көмегімен оңай құрастырыңыз және үйретіңіз.
- ML өндірісі кез-келген жерде - Бұлтты, жергілікті, шолғышта немесе құрылғыда қай тілде сөйлесеңіз де, модельдерді оңай үйретіңіз және орналастырыңыз.
- Зерттеуге арналған қуатты тәжірибе - Концепциядан кодқа, жаңа үлгілерге және тезірек жариялауға жаңа идеяларды қабылдауға арналған қарапайым және икемді архитектура.

Және оның таңымалдылығы және сұраныстын көп болуы да менің Tensorflow таңдауыма себеп болды.



1.2-сурет – TensorFlow кітапханасының танымалдылығы

1.4.3 NumPy

NumPy - Python-мен бірге ғылыми есептеулердің іргелі пакеті. Оның құрамында мынадай заттар бар:

- N-өлшемді массивтің қуатты нысаны
- күрделі (хабар тарату) функциялары
- C / C ++ және Fortran кодтарын біріктіруге арналған құралдар
- пайдалы сызықтық алгебра, Фурье түрлендіруі және кездейсоқ сандар мүмкіндіктері

Ғылыми қолданудан басқа, NumPy жалпы деректердің тиімді көпөлшемді контейнері ретінде де қолданыла алады. Еркін мәліметтер типтерін анықтауға болады. Бұл NumPy-ге әр түрлі мәліметтер базасымен үздіксіз және жылдам біріктіруге мүмкіндік береді. NumPy бірнеше шектеулермен қайта пайдалануға мүмкіндік беретін BSD лицензиясы бойынша лицензияланған.

1.4.4 SciPy

SciPy («Sigh Pie» деп аталады) - бұл Python негізделген математика, ғылым және инженерияға арналған бастапқы коды ашық бағдарламалық жасақтама. Атап айтқанда, бұл негізгі пакеттер:

- NumPy N өлшемді массивтің базалық пакеті
- IPython жақсартылған интерактивті консоль

- SciPy кітапханасы ғылыми есептеу үшін негізгі кітапхана
- Matplotlib кешенді 2-өлшемді сюжеттер
- SymPy символдық математика
- Pandas деректер құрылымы және талдау

1.4.5 Matplotlib

Matplotlib - бұл Python 2D сюжеттік кітапханасы, ол әр түрлі баспа форматтарында және платформаларда интерактивті орталарда жарияланымның сапалық көрсеткіштерін шығарады. Матлотлибті Python сценарийлерінде, Python және IPython снарядтарында, Jupyter ноутбукінде, веб-қосымшаның серверлерінде және төрт графикалық пайдаланушы интерфейсінің құралдар жинағында пайдалануға болады. Matplotlib оңай және қиын нәрселерді жасауға тырысады. Кодтардың бірнеше жолымен сіз сюжеттер, гистограммалар, күш спектрлері, штрих-диаграммалар, қателіктер, шашыраңқалар және т.б. жасай аласыз. Мысалдар үшін сюжеттердің үлгілері мен нобай галереясын қараңыз. Pyplot модулін қарапайым жоспарлау үшін MATLAB интерфейсі қамтамасыз етіледі, әсіресе IPython-мен біріктірілгенде. Энергия тұтынушысы үшін сіз сызық стильдерін, қаріп сипаттарын, осьтердің қасиеттерін және т.б., объектіге бағытталған интерфейс арқылы немесе MATLAB пайдаланушыларына таныс функциялар жиынтығы арқылы толықтай басқара аласыз.

1.4.6 Pillow

Pillow - бұл Алекс Кларк пен салымшылардың достық PIL шанышқысы. PIL - бұл Фредрик Лундтың және үлес қосушылардың Python Imaging кітапханасы. Бұл кітапхана суреттермен жұмыс істеу үшін өзінің септібін тигізеді.

1.4.7 H5Py

H5py пакеті - HDF5 екілік деректер форматына арналған Pythonic интерфейсi. Бұл сізге көптеген сандық деректерді сақтауға және бұл деректерді NumPy-мен оңай басқаруға мүмкіндік береді. Мысалы, сіз дискіде сақталған көп терабайттық мәліметтер жиынтығын нағыз NumPy жиымдары сияқты бөлуге болады. Мындаған деректер жиынтығын бір файлда сақтауға болады, бірақ қалағаныңызша санатқа бөліп, белгілей аласыз. H5py сөздік және NumPy массивінің синтаксисі сияқты түзу NumPy және Python метафораларын қолданады. Мысалы, файлдағы деректер жиынтығын қайталай аласыз немесе деректер жиынтығының .shape немесе .dtype атрибуттарын тексере аласыз. Жұмысты бастау үшін HDF5 туралы ерекше ештеңе білудің қажеті жоқ. Пайдалануға болатын жоғары деңгейлі интерфейсден басқа h5py HDF5 C API-ге объектіге бағытталған Cython

орауына негізделген. HDF5-тен C-дан бастап, h5py-ден жасай алатын барлық дерлік нәрсе. Ең жақсысы, сіз жасайтын файлдар кеңінен қолданылатын стандартты екілік форматта, сіз оны басқа адамдармен, соның ішінде IDL және MATLAB сияқты бағдарламаларды қолданушылармен де алмасуға болады.

1.4.8 Keras

Keras - бұл Python-да жазылған ашық нейрондық-желілік кітапхана. Ол TensorFlow, Microsoft Cognitive Toolkit, R, Theano немесе PlaidML-тің жоғарғы жағында жұмыс істей алады. Терең нейрондық желілермен жылдам тәжірибе жасауға мүмкіндік беретін, ол қолданушыға ыңғайлы, модульді және кеңейетін болуға бағытталған. Ол ONEIROS (Open Neuro-Electronic Intelligent Robot Operating System) жобасының зерттеу жұмысының аясында жасалды, және оның негізгі авторы әрі жөндеушісі - Google инженері Франсуа Шоллет. Шоллет сонымен қатар XSeqs терең нейрондық желі моделінің авторы. Егер терең оқу кітапханасы қажет болса, Керасты қолданған дұрыс болады. Себебі:

- Прототиптерді оңай және тез жасауға мүмкіндік береді (қолданушының ыңғайлылығы, модульділігі және кеңеюі арқылы).
- Реттелетін желілерді де, қайталанатын желілерді де, олардың екеуін де қолдайды.
- CPU және GPU-де біркелкі жұмыс істейді.

1.4.9 ImageAI

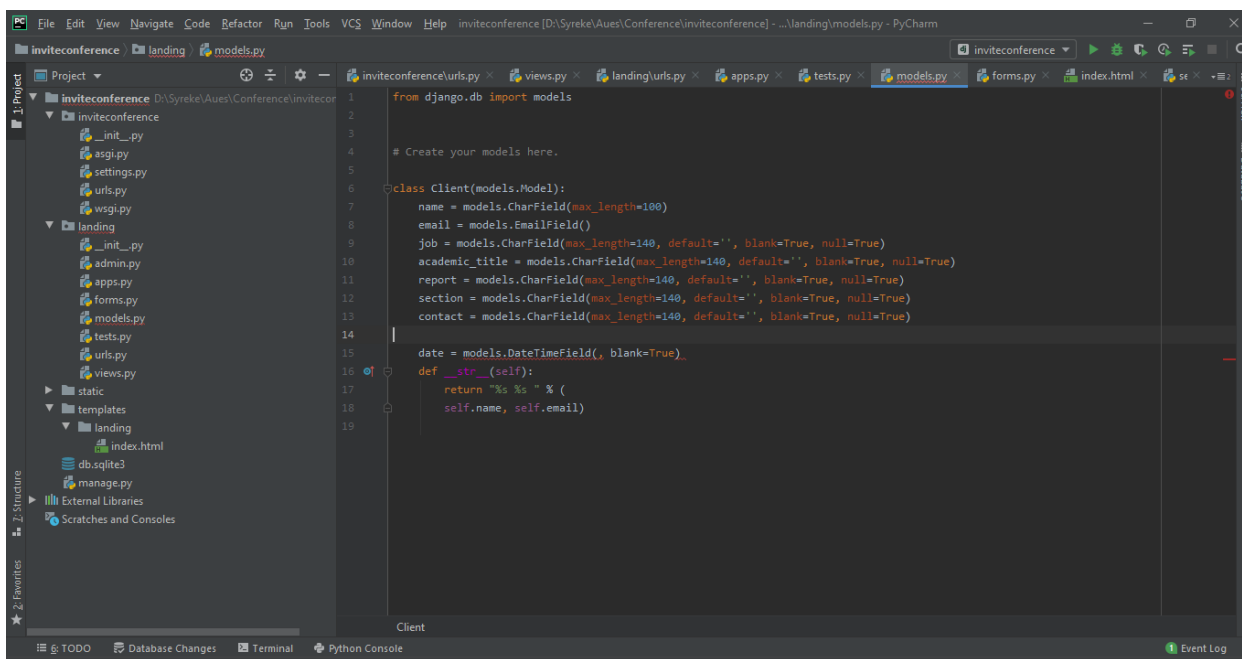
ImageAI - бұл қарапайым және бірнеше жолдардан тұратын Deep Learning және Computer Vision мүмкіндіктері бар қосымшалар мен жүйелерді құруға, әзірлеушілерге, оқытушылар мен студенттерге мүмкіндік беру үшін салынған питон кітапханасы. Бұл құжаттама ImageAI-де қол жетімді барлық сыныптар мен функциялар туралы егжей-тегжейлі ақпарат беру үшін, сонымен қатар бірнеше кодтық мысалдармен қамтамасыз етілген.

1.4.10 PyCharm

PyCharm - ReSharper әзірлеген бірдей адамдар болып табылатын JetBrains әзірлеген редактор және түзету құралы. ReSharper - Windows әзірлеушілерінің кодты қайта кодтау және NET кодын жазу кезінде олардың өмірін жеңілдету үшін пайдаланатын керемет құрал. ReSharper қағидаларының көпшілігі PyCharm кәсіби нұсқасына қосылды. PyCharm - бұл Python бағдарламалау тілі үшін біріктірілген даму ортасы. Кодтық талдау құралдарын, графикалық түзеткішті, блокты тексеруге арналған құралмен қамтамасыз етеді және Django-да веб-әзірлеуді қолдайды. PyCharm IntelliJ IDEA негізінде JetBrains жасаған және бұл программа Windows, MacOS, Linux үйлесімді платформалық даму ортасы. PyCharm Community Edition (тегін

нұсқа) Apache лицензиясы бойынша лицензияланған, ал PyCharm Professional Edition (ақылы нұсқа) - жеке меншік бағдарламалық жасақтама. PyCharm-ның мүмкіншіліктерін айтатын болсақ олар:

- Статикалық кодты талдау, синтаксисті бөлектеу және қателер.
- Жобалық және бастапқы кодтық навигация: жобаның файлдық құрылымын көрсету, файлдар, сыныптар, әдістер мен әдістердің арасында жылдам ауысу.
- Рефакторинг: атауды өзгерту, әдісті шығару, айнымалымен таныстыру, тұрақты енгізу, әдісті көтеру және төмендету т.б.
- Django жақтауын қолдану арқылы веб-дамыту құралдары
- Python-ге арналған түзеткіш
- Кіріктірілген қондырғыларды сынау құралдары
- Google App Engine көмегімен дамыту
- Нұсқаны басқаруды қолдау: өзгертулер тізіміне және біріктірулерге қолдау көрсететін Mercurial, Git, Subversion, Perforce және CVS үшін жалпы пайдаланушы интерфейсі



1.3-сурет – PyCharm құру ортасының интерфейсі

2 Жобалау бөлімі

2.1 UML бірыңғайландырылған модельдеу тілі

Объектті-бағытталған анализі мен жобалаудың көптеген қолданыстағы әдістері (ОБАЖ) модельдеу тілімен бірге, модельдеу процесін бейнелеуді жатқызады. Моделдеу тілі бағдарламалық қамтамасыз етудің «сызбасын» құрастырудың стандартты құралы болып табылады.

1984 жылдан 1994 жылға дейін ОБАЖ-дың түрлі әдістерінің саны оннан елуден астамға дейін өсті. Модельдеудің унифицирленген тілі UML (Unified Modeling Language) – бұл осы әдістер болашағының ұрпағы. UML құрау фактілі түрде 1994 ж. аяғында, Гради Буч және Джеймс Рамбо әдістерді біріктіру жұмысын бастаған кезде Booch және OMT (Object Modeling Technique) Rational Software компаниясының эгидасы астында. 1995 ж., осыған OOSE (Object-Oriented Software Engineering) әдісінің құраушысы Ивар (Айвар) Якобсон қосылды. Осылайша, UML аталған әдістердің тікелей байланысы мен унификациясы болып табылады және оларды жаңа мүмкіндіктерімен толықтырады.

1997 жылдың қаңтарында UML объектті-бағытталған әдістер мен технология облысында модельдеудің стандартты тілін құру бойынша конкурсқа ұсынылған және стандарт ретінде қабылданған болатын. UML сңғы аңғымдағы нұсқасы – 1.3 (1998 ж. күзде қабылданған болатын)

UML тілі барлық БҚ компания-өңдеушілеріммен құрал ретінде қабылданған болатын (Microsoft, IBM, Hewlett-Packard, Oracle, Sybase и др.) Одан басқа, барлық әлемдік өндірушілер CASE-құралдарды, Rational Software (Rational Rose) басқа, UML-ды өзінің өнімдерінде (Paradigm Plus 3.6, System Architec, Microsoft Visual Modeler for Visual Basic, Delphi, JBuilder, PowerBuilder және басқалары) қолдайды.

Унифицирленген Модельдеу Тілі (UML) – бағдарламалық жүйенің артефактілерін құжаттау және құрылымдастыру, арнайыландыру, визуализация тілі. UML түсіну концептуалды моделін меңгеру қажет, ол өзіне үш құрама бөліктерді қосады: тілдің негізгі құрылыстық блоктары, олардың үйлесу ережелері және тілдің жалпы механизмі үшін кейбір жалпы ережелері. UML тілінің сөздігі өзіне үш құрылыс блоктарын қосады:

- болмыстар;
- қатынастар;
- диаграммалар.

Болмыстар – бұл модельдің негізгі элементі болып табылатын абстракций.

Қатынастар - қарапайым аралықтағы семантикалық байланыс.

Диаграмма - көріністің графикалық элементі.

Қатынастар түрлі болмыстарды байланыстырады, болмыс жиынтығының қызығушылығын көрсететін диаграммаларды топтастырады.

UML-да төрт түрлі болмыстар бар:

- құрылымды;

- тәртіпті;
- топтастырушы;
- аннотациялық.

Болмыстары тілдің негізгі объектті-бағытталған блоктары болып табылады. Олардың көмегімен керекті модельдерді құруға болады.

Құрылымдық болмыстар – бұл UML тіліндегі моделіндегі зат есіп атаулары. Тәртіп бойынша, олар өз алдына модельдің статистикалық бөлігін көрсетеді, олар жүйенің концептуалды немесе физикалық бөліктеріне сәйкес келеді. Құрылымды болмыстың жеті түрлілігі бар:

- Класс (Class) – жалпы атрибуттар, операциялар, семантикалар және объектілердің көрінісі. Графикалық көрініс: оның атрибуты, операциясы, атауы берілген тікбұрыш.
- Интерфейс (Interface) – бұл класспен немесе компонентпен ұсынылатын операциялар жиынтығы.

Графикалық көрінісі: дөңгелек, оның астынан атау жазылады.

Кооперация (Collaboration) – өзара әрекеттесуді нықтайды, ол өз алдына бірігіп жұмыс істей отырып кейбір кооперативті нәтижені шығаратын рольдер мен басқа элементтердің жиынтығын көрсетеді. Графикалық көрінісі: пунктирлі сызықпен шектелген эллипс.

Прецедент (Use case) – орындалатын әрекеттердің, белгілі тұлғаға мағыналы тізбегінің көрінісі (әртіс, Actor).

Графикалық көрінісі: үздіксіз сызықпен шектелген эллипс.

Үш келесі болмыстар – кластарға ұқсас, алайда олар ерекшеліктер қатарына ие болады және сондықтан UMLдың жеке болмыстарына шығарылған. Белсенді класс (Active class) – бұл объектілері немесе жіптері (threads) бір немесе бірнеше процестерге енгізілген класс, және сондықтан олар басқарушы әрекетті көрсете алады. Графикалық көрінісі: жуан сызықпен белгіленген қарапайым класс тәріздес.

Компонент (Component) – бұл жүйенің физикалық алмастырушы бөлігі, ол интерфестер жиынтығына сәйкес келеді және оның жүзеге асырылуын қамтамасыз етеді. Компонент, ереже бойынша, өз алдына логикалық элементтердің физикалық жиынтығын, класстар мен интерфейстер және кооперация тәріздес. Графикалық көрінісі: бөлімшелері бар тікбұрыш тәрізді.

Түйін (Node) – бағдарламалық кешенді жобалау кезінде бар болатын жүйенің шынайы элементі және өз алдына жадының кейбір көлеміне ие болатын есептеу қорын көрсетеді. Графикалық көрінісі: әдетте куб ретінде көрсетіледі. Компоненттер мен түйіндер жүйенің физикалық болмыстарына сәйкес келеді, сол уақытта бірінші бес – концептуальды немесе логикалық болмысқа сай келген. Осы болмыстардың осылайша түрлері бар болады: актерлер, сигналдар, утилиттер (класс түрі), процестер және жіптер (активті клас түрлері), қосымшалар, құжаттар, файлдар, кітапханалар, беттер және кестелер (компоненттер түрі). Тәртіпті болмыстар UML тілінің үлгілерінің динамикалық құрастырушысы болып табылады. Бұл тіл етістіктері: олар үлгінің тәртібін уақыттарда және кеңістікте бейнелеп түсіндіреді:

Өзара әсерлесу (Interaction) – бұл тәртіп, болмысы объект арасында нақты контекст аясында хабарламалармен алмасуда нақты мақсатқа жетуде құралады. Графикалық белгіленуі: сәйкес операцияның аты бейнеленетін тілше.

Автомат (State machine) – күйлердің тізбегін анықтайтын тәртіп алгоритмі, сол арқылы объект немесе зара әрекеттесу өзінің өмірлік цикл созындысында түрлі құбылыстарға жауап ретінде өтеді. Графикалық көрінісі: доғаланған бұрышпен тікбұрыш, атауы бар болатын, мүмкін ішкі күйі де. Топтастырылатын болмыстар UML үлгісінің ұйымдастырушы бөлігі болып табылады. Бұл үлгіні қоюға болатын блоктар.

Пакеттер (Packages) – топтаушы болмыстардың жалғыз типі - өз алдына элементтердің топтарға ұйымдастырудың әмбебап механизмын ұсынады. Пакетке құрылысты, тәртіптік және басқа топтаушы болмыстарды орналастыруға болады. Бағдарламалаудың жұмыс уақытында бар болатын компоненттен айырмашылығы пакеттер таза концептуалды сипат алады, яғни өңдеу уақытында ғана бар болады. Графикалық белгілеу: закладкамен папка түрінде, кейде атау және –құрамындағыларды құрайтын.

Аннотациялы болмыстар – UML үлгілерінің түсіндіргіш бөлігі. Бұл қосымша суреттеуге арналған түсініктер, түсіндірулер немесе үлгі - элементіне ескерту. Аннотациялық элементтердің негіздік үлгісі бар ол-ескерту.

Ескерту (Note) –Бұл жай ғана түсініктерді бейнелеуге арналған символ немесе элементке немесе элементтердің тобына қосылған шектеулер. Графикалық белгілеуі: мәтіндік немесе графикалық түсінігі бар бүгілмелі жақпен тікбұрыш түрінде. Жиірек ескертулер диаграммаларды формалды немесе формальсіз мәтін түрінде айтуға болатын шектеулермен немесе түсініктермен жабдықтау үшін қолданылады.

Қатынастар туралы айтатын болсақ, UML тілінде қатынастың төрт типі анықталған. Олар:

- тәуелділік;
- ассоциация;
- жалпылау;
- жүзеге асыру.

Тәуелділік (Dependency) – екі болмыс арасындағы семаникалық қатынас, осыдан олардың біреуінің өзгерісі тәуелсіздің, тәуелдіге әсер етуі мүмкін. Графикалық белгілеуі: түзу пунктир сызық түрінде, жиі белгісі бар болатын тілшемен.

Ассоциация (Association) – құрылысты қатынас, байланыстардың жиынтығын суреттеуші; байланыс - бұл объектілер арасындағы бірігу. Ассоциацияның түрлігі агрегирлеу болып табылады (Aggregation)- құрылысты дәл осылай жиі оның бүтін және жеке бөлімдер арасындағы қатынасты атайды. Графикалық белгілеуі: түзу сызық түрінде (кейде аяқталатын тілмен немесе таңба ұстаушы), қайсының жанында қосымша белгілеулері қатыса алады, мысалы еселілік және рольдердің аттары.

Жалпылау (Generalization) – бұл «мамандандыру|жалпылау» қатынасы, қайсыда мамандандырылған элемент объектісі (ұрпақ) жалпыланған элемент объектісінің орнына қойылуы мүмкін (ата-анасының немесе атасының). Осылайша ұрпақ (Child) өзінің ата анасы (Parent) тәртібін мұра етеді. Графикалық белгілеу: боялмаған тілмен пен сызық түрінде, ата-анасына белгілейтін.

Жүзеге асыру (Realization) – бұл топтастырғыштар арасындағы семантикалық қатынас, қайсыда бір топтастырғыш «келісімді» анықтайды, ал басқасы оның орындалуына кепіл болады. Орындау қатынастары екі жағдайда кездеседі: біріншіден, интерфейстер және оларды іске асырушы класпен немесе компоненттер арасындағы, ал - екіншіден, прецеденттер және олардың іске асырушы кооперация арасында. Графикалық белгілеуі: пунктир сызық түрінде боялмаған тілмен, қандай да бір жалпылау және тәуелділік қатынасының ортасы сияқты. Төрт суреттелген элементтер UML үлгісіне қосуға болатын қатынастардың негізгі типтері болып табылады. Олардың сонымен қатар вариациясы (Refinement) да, трассировка (Trace), қосу және кеңейту де (тәуелділіктерге арналған) бар болады.

2.2 Rational Rose құралы

Rational Rose – дегеніміз автоматтандыру үрдістерін талдау және ақпараттық жүйені жобалау үшін арналған, сонымен қатар әртүрлі тілдердегі кодтарды генерациялауға және жоба құжатнамаларды шығаруға арналған Rational Software Corporation фирмасының объектілі-бағытталған CASE-құралдары. Rational Rose UML тіліне негізделіп жобалау және объектілі бағытталған талдау әдістерін қолданады. Rational Rose осы болжамасы C , Visual C , Visual Basic, Java, PowerBuilder, CORBA Interface Definition Language (IDL) бағдарламалар үшін кодтар генерациясын және ANSI SQL, Oracle, MS SQL Server, IBM DB2, Sybase үшін мәліметтер қорының генерация бейнеленуін, сонымен қатар диаграмма түріндегі жобалау құжаттарын және егжей- тегжейлерін іске асырады. Rational Rose жаңа жобаларда бағдарламалық компоненттерінің қайта қолдануын қамтамасыз ететін бағдарламалар мен мәліметтер қорының реверстік инжинирингтің құралдарынан тұрады. Rational Rose-де жұмыс істеудің негізі жүйе архитектурасының статикалық және динамикалық аспектілерін анықтайтын UML егжей- тегжейі мен диаграммаларды құру болып табылады. Rational Rose құрамындағы келесі алты негізгі құрылымдық компоненттерді белгілеуге болады: репозиторий, қолданушының графикалық интерфейсі, проектті қарау құралдары (браузер), проектті бақылау құралдары, документтердің статистикалық және генераторлық құралдарын жинау. Оларға сонымен қатар кодтар генераторлары (әрбір тіл үшін жеке) және реверстік инжинирингті қамтамасыз ететін C үшін анализатор кіреді. Репозиторий проектің мәліметтер қоры болып табылады. Браузер иерархия кластары бойынша орын ауыстыру, диаграммалардың бір түрінен екінші түріне ауысуды проект бойынша

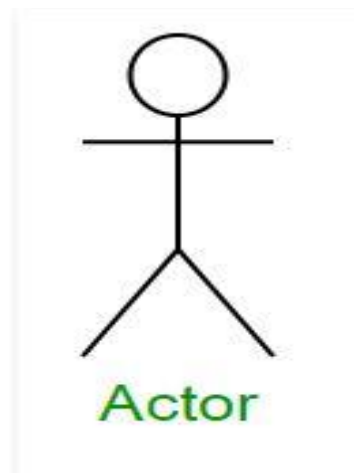
«навигацияны» қамтамасыз етеді. С тіліндегі автоматы түрдегі генерация кодтар бағдарламасының құралдары компоненттер мен диаграммалар кластарында болатын бағдарламаны қолданады да тақырыптар және класстардың файлдары -және объектілерін қалыптастырады. Осындай түрмен бағдарламаның «қанқасын» жасайды да ол С тілінде тура бағдарламалауда анықталады. С -тағы кодтар анализаторы жеке бағдарламалық модуль тәрізді негізделген. Оның тағайындалуы С - тегі қолданушы анықтайтын текстерінде болатын бағдарламаның негізінде Rational Rose-дегі проекттер модулін құру. Жұмыс барысында анализатор тексттердің дұрыстылығын және қателердің болдырмауын іске асырады. Оның жұмыс қорытындысында алынған модель бүтіндей немесе бөлшектей әртүрлі проекттерде қолданылуы мүмкін. Анализатордың кіріс және шығыс күі бойынша кең мүмкіншіліктері бар. Мысалы файлдар типін, компилятор қорларын , қандай анықтама моделге кіруі және қандай модель элементтері экранға шығуы екенің анықталуы мүмкін. Осындай мүмкіншіліктермен Rational Rose/C бағдарламалық компоненттердің қайта қолданылуын қамтамасыз етеді. Rational Rose арқылы мына диаграммаларды пайдалануға болады:

- күй диаграммасы;
- тізбек диаграммасы;
- қызмет диаграммасы;
- компоненттер диаграммасы.
- класстар диаграммасы;
- прецеденттер диаграммасы;
- таратылған диаграмма;
- компоненттер диаграммасы;
- кооперация диаграммасы;
- әрекеттесу диаграммасы.

2.3 Тізбек диаграммасы

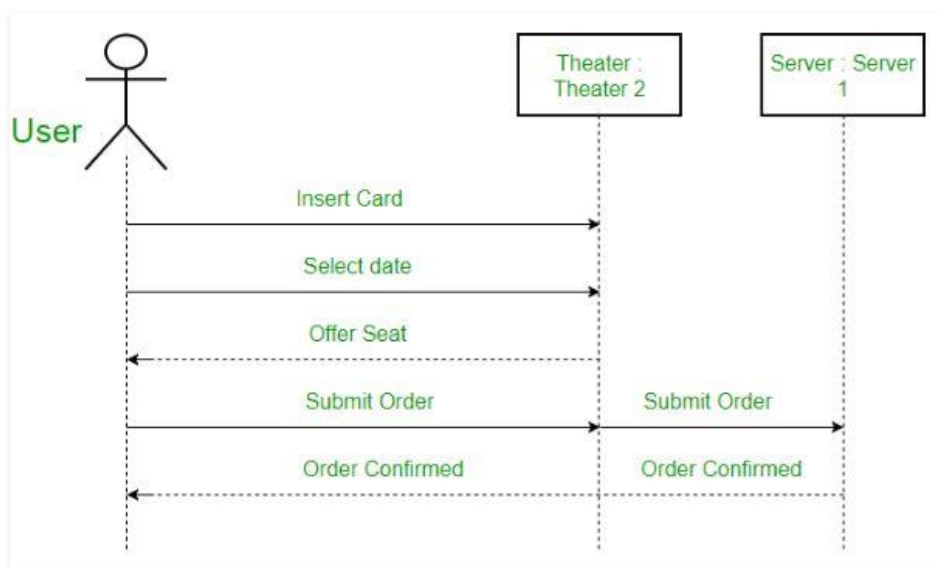
UML тізбек диаграммалары - бұл әрекеттерді қалай жүзеге асыратындығын көрсететін өзара әрекеттесу диаграммалары. Олар ынтымақтастық шеңберіндегі нысандардың өзара әрекетін түсіреді. Тізбек диаграммалары уақыттың фокустары болып табылады және олар хабарлардың қай уақытта және қай уақытта жіберілетінін көрсету үшін диаграмманың тік осін пайдаланып көрнекіліктің ретін көрсетеді. Тізбектік диаграммасы жай объектілердің өзара әрекеттесуін тізбектелген тәртіппен, яғни осы өзара әрекеттесулер болатын ретпен бейнелейді. Реттеу диаграммасына сілтеме жасау үшін оқиға оқиғалары мен сценарийлер терминдерін де қолдана аламыз. Бірізділік диаграммалары жүйелік функциядағы объектілерді қалай және қандай ретпен сипаттайды. Бұл диаграммаларды жаңа және қолданыстағы жүйелерге қойылатын талаптарды түсіну және түсіну үшін кәсіпкерлер мен бағдарламалық жасақтама жасаушылар кеңінен қолданады. Тізбек диаграммасы туралы жазбалар:

Актерлер - UML диаграммасындағы актер жүйе мен оның объектілерімен өзара әрекеттесетін рөл түрін білдіреді. Бұл жерде актер әрдайым UML диаграммасын қолдана отырып, модельдейтін жүйенің шеңберінен тыс болатындығын ескерген жөн.



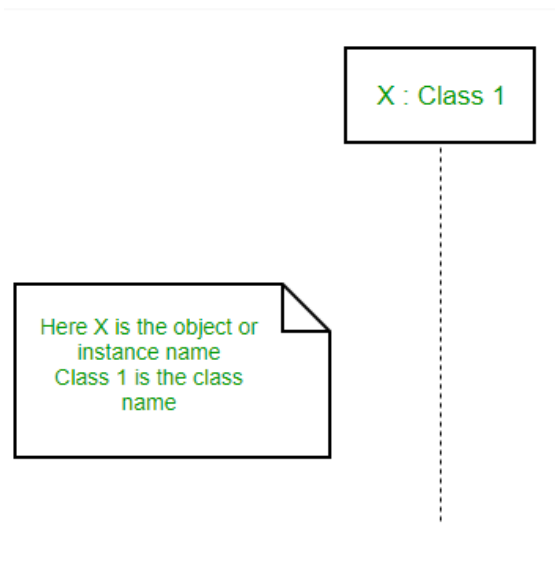
2.1-сурет – Актердың белгіленуі

Біз әр түрлі рөлдерді бейнелеу үшін актерлерді қолданамыз, соның ішінде адам қолданушылары және басқа сыртқы тақырыптар. Біз UML диаграммасында актерлерді жеке адамның жазбасын қолдана отырып ұсынамыз. Тізбек диаграммасында бірнеше актер бола аламыз. Мысалы, мұнда орынды брондау жүйесіндегі пайдаланушы жүйеден тыс жерде және жүйенің бөлігі болып табылмайтын жерде актер ретінде көрсетіледі.



2.2-сурет – орындарды брондау жүйесімен әрекеттесетін актер

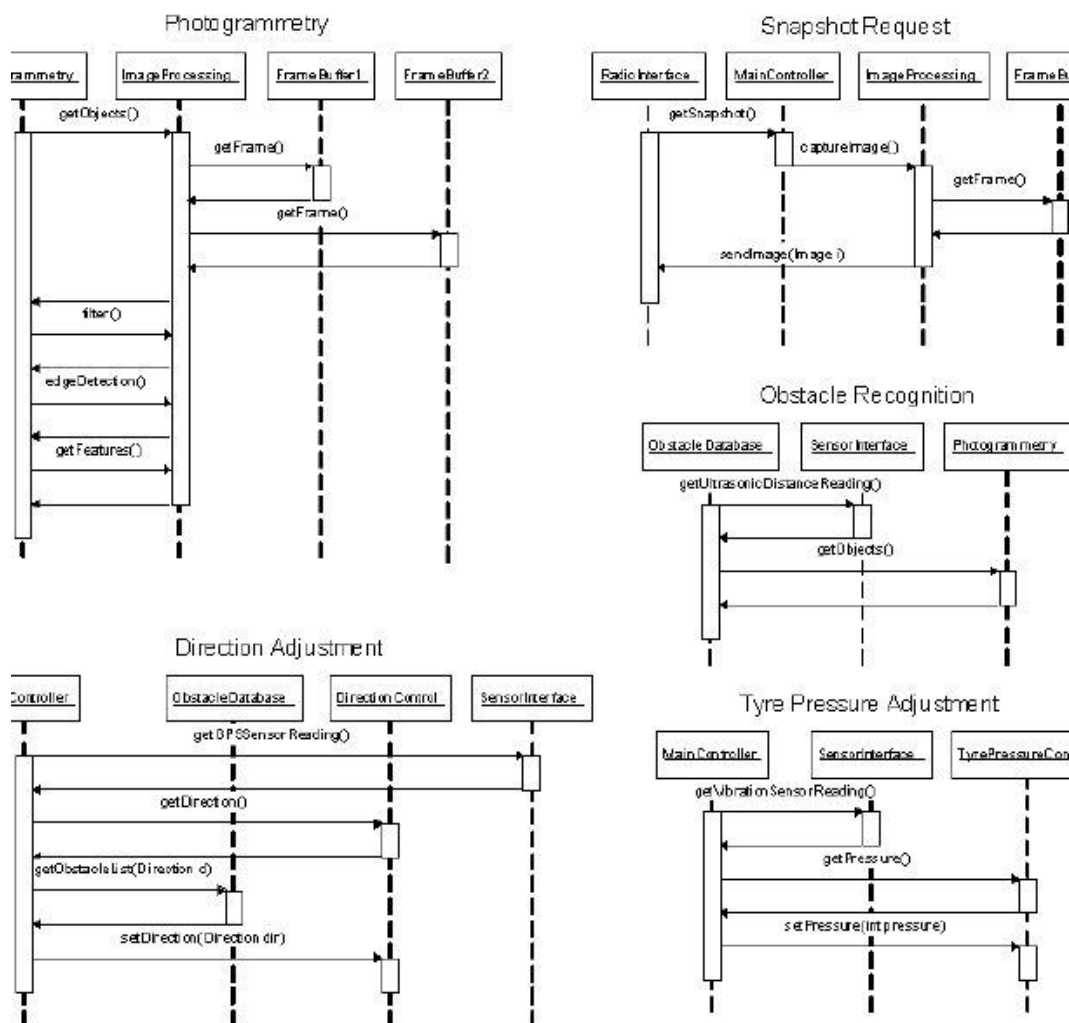
Өмірлік сызық - бұл тізбектік диаграммадағы жеке қатысушыны бейнелейтін аталатын элемент. Осылайша, кезек-кезек диаграммасындағы әр инстанция өмір сызығымен ұсынылған. Сызықтық элементтер жоғарғы жағында тізбектік диаграммада орналасқан. UML-дегі өмір сызығын атауға арналған стандарт келесі форматқа сәйкес келеді.



2.3-сурет – Өмірлік сызық сипаттамасы

Біз өмір сызығын басы деп аталатын тіктөртбұрышта оның аты мен түрімен көрсетеміз. Басы жоғарыда көрсетілгендей тік сызық сызықтың жоғарғы жағында орналасқан (баған деп аталады). Егер біз аты аталмаған дананы модельдегіміз келсе, біз сол үлгі бойынша жүреміз, тек қазір өмір жолының аты бос қалдырылады.

Өмір сызығы мен актер арасындағы айырмашылық - өмір сызығы әрқашан жүйенің ішіндегі затты бейнелейді, ал актерлер жүйеден тыс нысандарды бейнелеу үшін қолданылады. Төменде тізбектік сызбаның мысалы келтірілген:

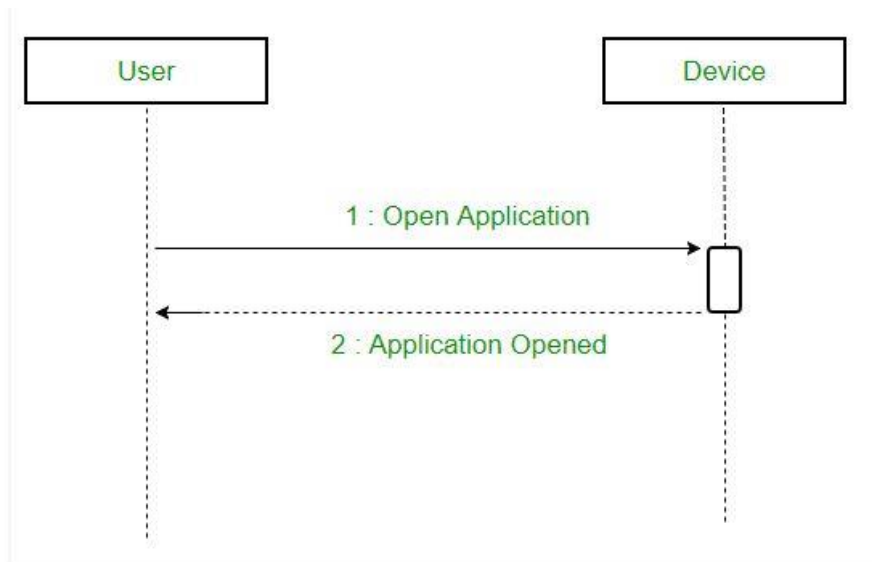


2.4-сурет – Тізбек диаграммасы

Хабарлар - нысандар арасындағы байланыс хабарламаларды қолдану арқылы бейнеленеді. Хабарламалар өмір жолында бірізді түрде шығады. Біз хабарларды көрсеткілерді қолдана отырып ұсынамыз. Сызықтық сызықтар мен хабарламалар бірізділік диаграммасының өзегін құрайды.

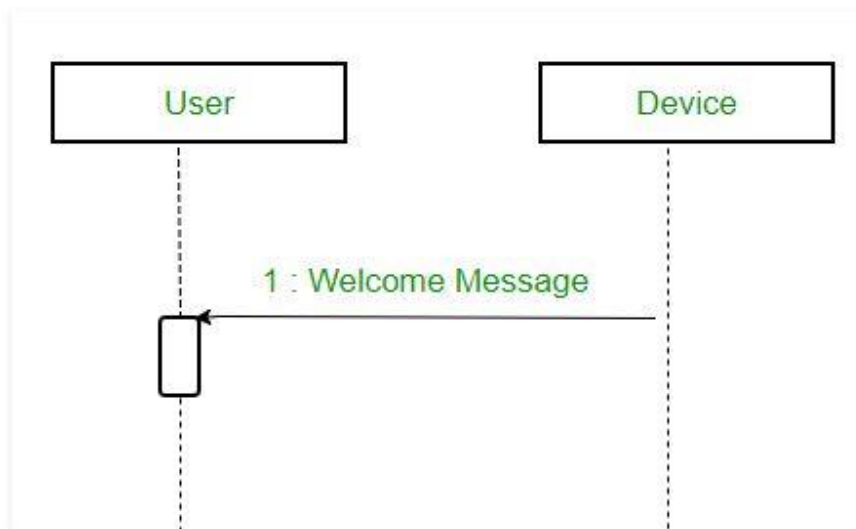
Хабарламаларды келесі санаттарға кеңінен жіктеуге болады:

Синхронды хабарламалар. Синхронды хабарлама өзара әрекеттесу алға жылжудың алдында жауап күтеді. Жіберуші алушы хабарламаны өңдеуді аяқтағанша күтеді. Қоңырау шалушы алушының алдыңғы хабарламаны, яғни жауап хабарламасын алғанын білген кезде ғана жалғасады. Объективті бағдарламалаудағы қоңыраулардың көп саны синхронды. Біз синхронды хабарламаны ұсыну үшін қатты көрсеткі басын қолданамыз.



2.5-сурет – синхронды хабарламалар тізбегінің диаграммасы

Асинхронды хабарлар – асинхронды хабар қабылдағыштан жауап күтпейді. Өзара әрекеттесу алдыңғы хабарламаны өңдейтін немесе қабылдамайтын қабылдағышқа қарамастан алға жылжиды. Асинхронды хабарламаны ұсыну үшін біз бағытталған жебенің басын қолданамыз.



2.6-сурет – асинхронды хабарламалар тізбегінің диаграммасы

Хабарлама жасау - біз дәйектілік диаграммасында жаңа нысанды енгізу үшін Create хабарламасын қолданамыз. Белгілі бір хабарлама қоңырауы объект жасауды қажет ететін жағдайлар болады. Ол нүктелік көрсеткімен көрсетілген және оның Хабарламаны құру символы екенін көрсету үшін оған белгіленген сөз жасаңыз. Мысалы, электрондық коммерция веб-сайтында жаңа тапсырысты құру Тапсырыс класының жаңа объектісін жасауды қажет етеді.

Хабарламаны жою - нысанды жою үшін Жою хабарламасын қолданамыз. Нысан жадқа бөлінген немесе жүйеде жойылған кезде біз хабарламаны жою таңбасын қолданамыз. Бұл жүйеде объектінің пайда болуын жояды. Бұл хпен аяқталатын көрсеткі арқылы ұсынылады.

Өздігінен хабарлама - объект өзіне хабарлама жіберу керек болған кезде белгілі бір сценарий туындауы мүмкін. Мұндай хабарламалар «Өздігінен хабарлар» деп аталады және U пішінді көрсеткімен ұсынылады.

Хабарламаға жауап беру - жауап хабарлары қабылдағыштан жіберушіге жіберілген хабарламаны көрсету үшін қолданылады. Біз нүкте сызығы бар ашық стрелка көмегімен қайтару / жауап хабарламасын ұсынамыз. Өзара әрекеттесу жауап алушы жіберген кезде ғана алға жылжиды.

Табылған хабар - табылған хабар белгісіз дереккөз хабарлама жіберетін сценарийді көрсету үшін қолданылады. Ол аяғынан бастап өмір сызығына бағытталған көрсеткі арқылы ұсынылған. Мысалы, жабдықтың істен шығуы сценарийін қарастыру. Бұл бірнеше себептерге байланысты болуы мүмкін және біз аппараттық құралдың ақауына не себеп болғанын білмейміз.

Жоғалған хабар - жоғалған хабар алушы жүйеге беймәлім сценарийді көрсету үшін қолданылады. Ол өмір сызығынан соңғы нүктеге бағытталған жебе арқылы ұсынылған. Мысалы, ескерту жасалған сценарийді қарастыру. Ескерту пайдаланушыға немесе басқа бағдарламалық жасақтама / объект үшін өмірлік сызық өзара әрекеттесетін болуы мүмкін. Баратын жер алдын-ала белгісіз болғандықтан, біз Жоғалған хабарлама таңбасын қолданамыз.

Сақшылар - жағдайларды модельдеу үшін біз UML-де сақшыларды қолданамыз. Олар хабарлама ағынын шектеу керек деген сылтаумен шектеу қажет болған кезде қолданылады. Сақшылар бағдарламалық жасақтама жасаушыларға жүйеге немесе белгілі бір процеске қатысты шектеулер туралы хабарлау үшін маңызды рөл атқарады.

Тізбек диаграммаларды қолдану:

- Күрделі функцияның, операцияның немесе процедураның артындағы логиканы модельдеу және бейнелеу үшін қолданылады.
- Олар сонымен қатар UML пайдалану жағдайларының диаграммаларын көрсету үшін қолданылады.
- Ағымдағы немесе болашақ жүйелердің толық функционалдығын түсіну үшін қолданылады.
- Хабарламалар мен тапсырмалар жүйеде объектілер немесе компоненттер арасында қалай қозғалатынын елестетіңіз.

2.4 Прецеденттер диаграммасы

Прецеденттер диаграммасы қарапайым болып пайдаланушының жүйемен өзара әрекеттесуінің көрінісі болып табылады, ол қолданушы мен қолданушы қатысатын әр түрлі пайдалану жағдайларын көрсетеді. Прецеденттер диаграммасы жүйені пайдаланушылардың әр түрлі типтерін және әр түрлі пайдалану жағдайларын анықтай алады және көбінесе басқа

диаграммалармен бірге жүреді. Прецеденттер шеңберлермен немесе эллипстермен ұсынылған. Прецеденттер диаграммасы - UML-дегі динамикалық немесе мінез-құлық диаграммасы. Активтерді және регистрлерді қолдана отырып, жүйенің функционалдығын модель мысалдарын қолданыңыз. Прецеденттер - бұл жүйе жасауы керек әрекеттер, қызметтер мен функциялар жиынтығы. Бұл тұрғыда «жүйе» дегеніміз - веб-сайт сияқты өңделетін немесе басқарылатын нәрсе. «Актерлер» - бұл жүйе ішінде белгілі рөлдер бойынша жұмыс жасайтын адамдар немесе ұйымдар. Прецеденттер диаграммалары жүйенің функционалдық қажеттіліктерін визуализациялау үшін маңызды болып табылады, бұл дизайнды және дамудың басымдылығын өзгертеді. Олар сонымен қатар жүйеге әсер етуі мүмкін кез-келген ішкі немесе сыртқы факторларды анықтауға көмектеседі және ескеру қажет. Олар жүйеден тыс жоғары сапалы талдауды қамтамасыз етеді. Пайдалану жағдайлары диаграммасы жүйенің осы функцияның қалай іске асырылатындығы туралы толғандырмай, актерлермен қалай әрекеттесетінін көрсетеді.

Прецеденттердің өзі барлық мүмкіндіктер туралы егжей-тегжейлі егжей-тегжейлі егжей-тегжейлі мәлімдей алады, алайда прецеденттер диаграммасы жүйенің жоғары деңгейінің көрінісін қамтамасыз етеді. Бұған дейін «Кейс-сызбаларды қолдану - бұл сіздің жүйеңіздің сызбасы» деп айтылған. Олар жүйенің іс жүзінде не істеуі керек екенін қарапайым және графикалық түрде ұсынады.

Олардың қарапайым сипатына байланысты кейс-схемаларды қолдану мүдделі тараптар үшін жақсы байланыс құралы бола алады. Сызбалар нақты әлемге еліктеуге тырысады және мүдделі тарапқа жүйенің қалай жасалатынын түсінуге мүмкіндік береді. Шиа мен Ли прецеденттер диаграммаларын қолдануға жарамды жағдайдың бар-жоғын немесе олардың қажет еместігін анықтау үшін зерттеулер жүргізді. Пайдалану жағдайларының диаграммалары жүйенің ниетін мүдделі тараптарға неғұрлым жеңілдетілген түрде жеткізетіндігі және олар «класс диаграммаларынан гөрі толықтай түсіндірілгені» болды.

Схемаларды қолдану мақсаты - жүйенің жоғары деңгейде көрінісін қамтамасыз ету және қызығушылық танытқан адамдарға талаптарды қарапайым адамдармен жеткізу. Қосымша диаграммалар мен құжаттаманы жүйенің толық функционалды және техникалық көрінісін қамтамасыз ету үшін пайдалануға болады.

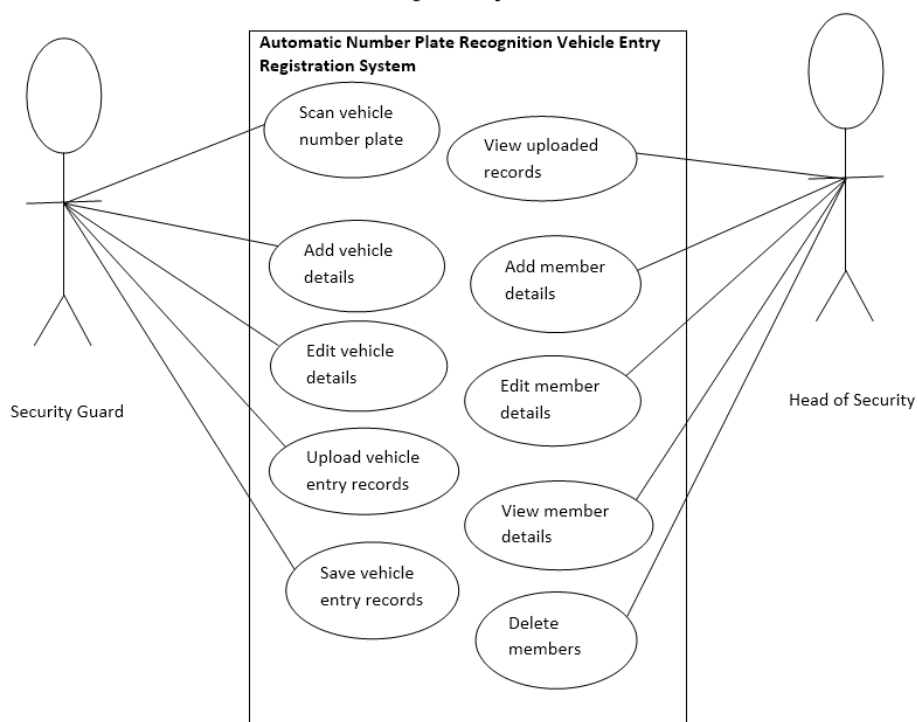
Прецеденттер диаграммасы көптеген егжей-тегжейлерге енбейді. Мысалы, оның қадамдар орындалу ретін модельдейтінін күтпеңіз. Оның орнына пайдалану жағдайларының дұрыс диаграммасы пайдалану жағдайлары, актерлер және жүйелер арасындағы өзара байланыстың жоғары деңгейлі шолуын бейнелейді. Сарапшылар мәтіндік сипаттаманы қолдану жағдайын толықтыру үшін кейс-сызбаларды қолдануды ұсынады.

UML - диаграммаларды құру үшін қолдануға болатын модельдеу құралы. Прецедент таңбаланған сопақ пішінді көрсетілген Stick фигуралары

процестегі актерлерді білдіреді, ал актерлердің жүйеге қатысуы актер мен пайдалану корпусының арасындағы сызықпен модельденеді. Жүйенің шекарасын бейнелеу үшін қолдану қорабына қорап салыңыз.

UML преценттер диаграммалары келесі операцияларды орындауға өте қолайлы:

- Жүйені қолданушылардың өзара әрекеттесу мақсаттарын көрсету;
- Жүйеде функционалдық талаптарды анықтау және ұйымдастыру;
- Жүйенің мазмұны мен талаптарын көрсету;
- Пайдалану жағдайындағы оқиғалардың негізгі ағынын модельдеу;



2.7-сурет – преценттер диаграммасы

Преценттер диаграммасының компоненттері:

- Актерлер: жүйемен өзара әрекеттесетін қолданушылар. Актер адам, ұйым немесе сіздің қосымшамен немесе жүйеңізбен өзара әрекеттесетін сыртқы жүйе бола алады. Олар деректерді шығаратын немесе тұтынатын сыртқы нысандар болуы керек.
- Жүйе: Актерлер мен жүйенің өзара әрекеттесуінің белгілі бір реттілігі. Жүйені сценарий деп те атауға болады.
- Мақсаты: көп жағдайда қолданудың соңғы нәтижесі. Табысты диаграмма мақсатқа жету үшін қолданылатын әрекеттер мен нұсқаларды сипаттауы керек.

2.5 Класстар диаграммасы

Бағдарламалық жасақтамада бірыңғай модельдеу тіліндегі (UML) классикалық диаграмма - бұл жүйенің кластарын, олардың атрибуттарын, операцияларын (немесе әдістерін) және объектілер арасындағы қатынасты көрсете отырып, жүйенің құрылымын сипаттайтын статикалық құрылым диаграммасы.

Класс диаграммасы объектіге бағытталған модельдеудің негізгі құрылыс блогы болып табылады. Ол қосымшаның құрылымын жалпы тұжырымдамалық модельдеу үшін және модельдерді бағдарламалық кодқа аударатын егжей-тегжейлі модельдеу үшін қолданылады. Класс диаграммаларын мәліметтерді модельдеу үшін де қолдануға болады. Класс диаграммасындағы сабақтар негізгі элементтерді, қосымшадағы өзара әрекеттестікті және бағдарламаланатын сыныптарды білдіреді.

Диаграммада сабақтар үш бөліктен тұратын қораптармен берілген:

- Жоғарғы бөлікте сынып атауы жазылған. Қалың және ортасында басылады, ал бірінші әріп бас әріппен жазылады.
- Ортаңғы бөлікте сыныптың атрибуттары бар. Олар сол жаққа тураланған және бірінші әріп кіші әріппен жазылған.
- Төменгі бөлікте сынып орындай алатын операциялар бар. Олар сонымен қатар сол жақта орналасқан және бірінші әріп кіші әріппен жазылған.

Жүйені жобалауда бірқатар класстар анықталып, сынып диаграммасында топтастырылған, бұл олардың арасындағы статикалық қатынастарды анықтауға көмектеседі. Егжей-тегжейлі модельдеумен қатар, тұжырымдамалық дизайн сыныптары көбінесе бірнеше ішкі сыныптарға бөлінеді. Жүйелердің әрекетін одан әрі сипаттау үшін осы класс диаграммаларын күй диаграмма немесе UML күй машинасымен толықтыруға болады.

UML атрибуттар мен әдістер сияқты сынып мүшелерін ұсыну механизмдерін және олар туралы конструкторлар сияқты қосымша ақпаратты ұсынады. Класс мүшесінің көрінуін анықтау үшін (яғни, кез-келген атрибут немесе әдіс), бұл жазбалар мүшенің атының алдына қойылуы керек:

- + Public
- Private
- # Protected
- ~ Package

Класс диаграммасының мақсаты:

- Жүйеде классификаторлардың статикалық құрылымын көрсетеді;
- Диаграмма UML тағайындаған басқа құрылымдық диаграммалар үшін негізгі жазбаны ұсынады;
- Әзірлеушілерге және команданың басқа мүшелеріне де пайдалы;

- Бизнес-аналитиктер жүйелерді бизнес тұрғысынан модельдеу үшін сынып диаграммаларын қолдана алады;

UML класс диаграммасы:

- Класстар жиынтығы және
- Класстар арасындағы қатынастар жиынтығы

Жүйеде ұқсас рөлдері бар объектілер тобының сипаттамасы, олар мыналардан тұрады:

Класс белгілері үш бөлімнен тұрады:

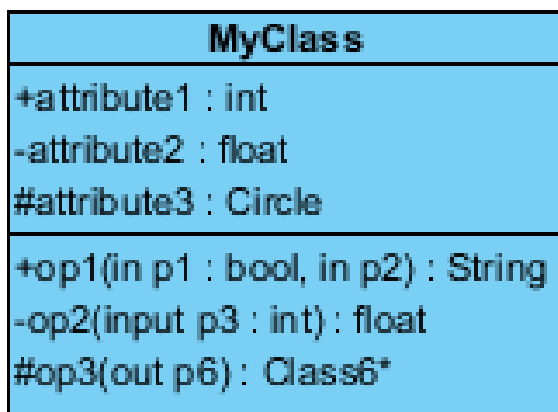
- Класс атауы
- Класс атауы бірінші бөлімде пайда болады.

Класс атрибуттары:

- Атрибуттар екінші бөлімде көрсетілген.
- Атрибут түрі қос нүктеден кейін көрсетіледі.
- Атрибуттар кодтағы мүшелік айнымалыларға (мәліметтер мүшелеріне) сәйкес келеді.

Класс операциялары (әдістері):

- Операциялар үшінші бөлімде көрсетілген. Бұл сынып ұсынатын қызметтер.
- Әдістің қайтару түрі әдіс қолтаңбасының соңында қос нүктеден кейін көрсетіледі.
- Әдіс параметрлерінің қайтару түрі параметр атауынан кейін қос нүктеден кейін көрсетіледі.
- Класс әдісіндегі операциялардың картасы.



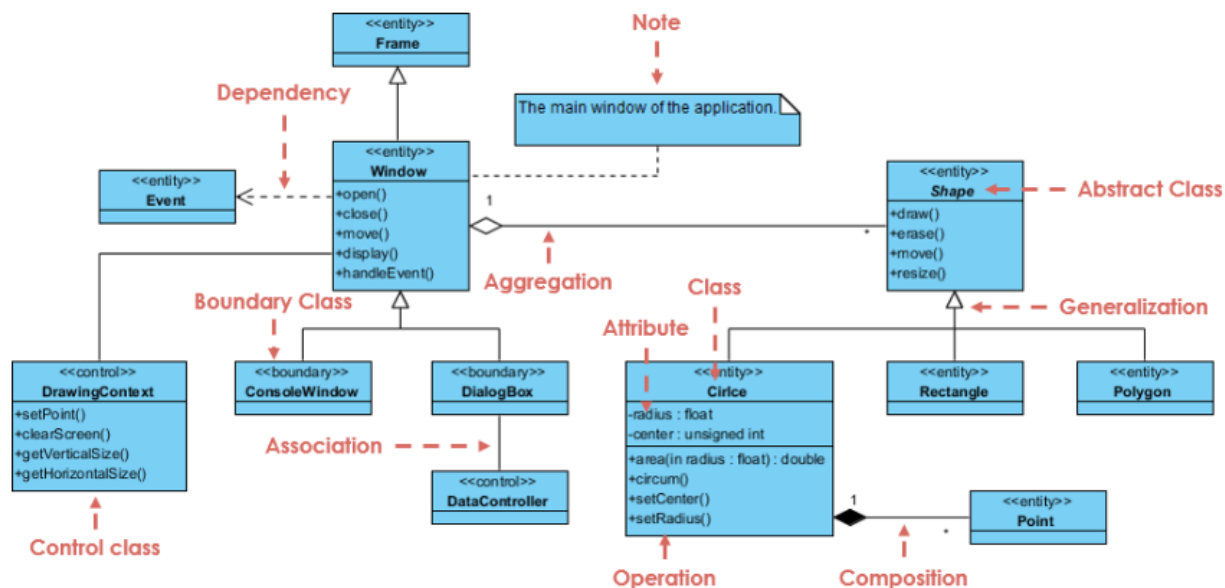
2.7-сурет – класс диаграммасының синтаксисі

Класстың графикалық көрінісі - MyClass жоғарыда көрсетілгендей:

- MyClass-те 3 атрибут және 3 операция бар;
- P3 op2 параметрі int типіне жатады;
- op2 өзгермелі мәнді қайтарады;
- op3 көрсеткіштің мәнін (a * арқылы белгіленген) Class6-ға қайтарады;

Класс басқа класстармен бір немесе бірнеше қатынастарға қатысуы мүмкін.

Класс диаграммасында класстарға немесе қатынастарға арналған жазбалар болуы мүмкін. Ескертпелер сұр түспен көрсетілген.



2.8-сурет – класс диаграммасының байланыстары

Жоғарыда көрсетілген диаграмманың мағынасын төмендегі тармақтар арқылы оқып түсіндіруге болады.

Пішін - бұл дерексіз сынып. Бұл курсивпен көрсетілген.

Пішін - бұл суперкласс. Шеңбер, төртбұрыш және көпбұрыш Shape-тен алынған. Басқаша айтқанда, шеңбер - бұл Shape. Бұл жалпылау / мұрагерлік қатынастар.

DialogBox пен DataController арасында байланыс бар.

Пішін - бұл терезенің бөлігі. Бұл агрегаттық қатынас. Пішін терезесіз болады.

Нүкте - шеңбердің бөлігі. Бұл композиция қатынасы. Шеңбер шеңберсіз мүмкін емес.

Терезе оқиғаға байланысты. Алайда, оқиға Терезеге тәуелді емес.

Шеңбердің атрибуттары - радиус және центр. Бұл субъект сыныбы.

Шеңбердің әдіс атаулары - аудан (), шеңбер (), setCenter () және setRadius().

Шеңбердегі радиус параметрі float типіндегі параметр болып табылады.

Сынып шеңберінің әдісі () double типінің мәнін береді.

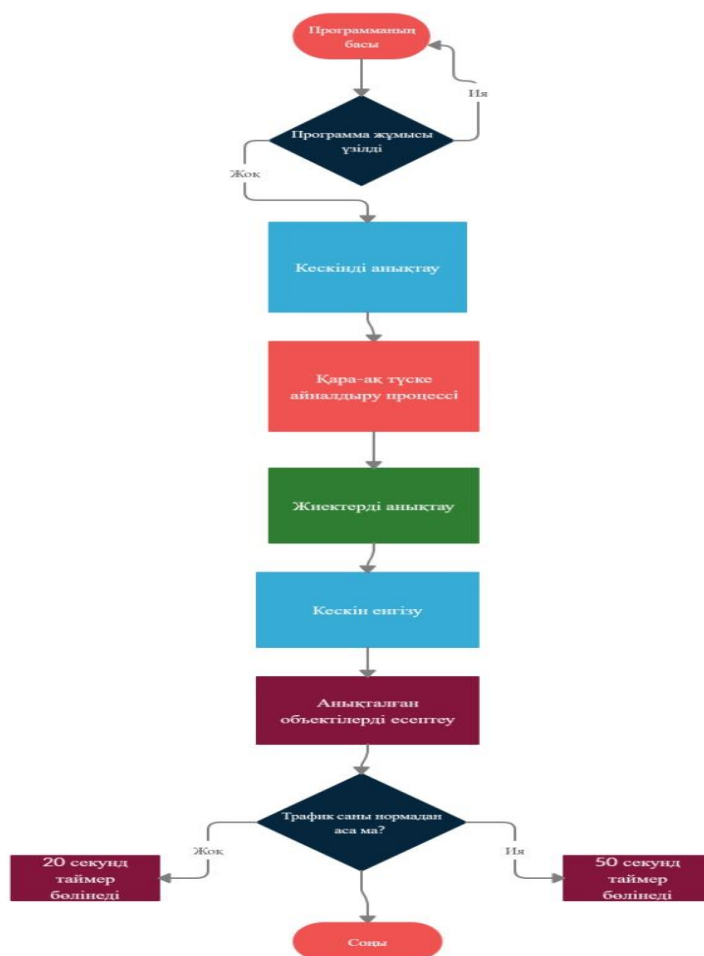
Төртбұрыштың атрибуттары мен әдіс атаулары жасырылған. Диаграммадағы кейбір басқа сыныптарда олардың атрибуттары мен әдіс атаулары жасырылған.

Біз класс диаграммаларын бағдарламалық жасақтаманы әзірлеудің өмірлік циклінің әртүрлі кезеңдерінде қолдана аламыз және әдетте алға жылжу кезінде сынып диаграммаларын үш түрлі перспективада (нақтылау деңгейлері) біртіндеп модельдеу арқылы қолдана аламыз:

Тұжырымдамалық перспектива: Диаграммалар нақты әлемдегі заттарды сипаттау ретінде түсіндіріледі. Осылайша, егер сіз тұжырымдамалық перспективаны қарастыратын болсаңыз, онда зерттелетін саладағы тұжырымдамаларды көрсететін диаграмма жасаңыз. Бұл ұғымдар табиғи түрде оларды жүзеге асыратын сыныптарға қатысты болады. Концептуалды перспектива тілден тәуелсіз болып саналады.

Ерекшеліктің перспективасы: Диаграммалар бағдарламалық жасақтаманың абстракцияларын немесе сипаттамалары мен интерфейсі бар компоненттерін сипаттайды, бірақ нақты іске асыруды талап етпейді. Осылайша, егер сіз техникалық сипаттаманың перспективасын қарастыратын болсаңыз, онда біз бағдарламалық жасақтаманың интерфейстерін қарастырамыз, іске асыруды емес.

Іске асыру перспективасы: Диаграммалар бағдарламалық қамтамасыз етуді белгілі бір технология мен тілде сипаттау ретінде түсіндіріледі. Осылайша, егер сіз іске асырудың перспективасын қарастыратын болсаңыз, біз бағдарламалық қамтамасыз етуді енгізуді қарастырамыз.



2.9-сурет – программа алгоритмінің блок схемасы

3 Қолданбалы бөлім

3.1 Мегалолис бағдаршамдарын реттеудің ақпараттық жүйесінің программасы

Қазіргі трафиктік проблемалардың бірі – жол қозғалыстарын басқарудың қолдану әдістері болып табылады. Егер бағдаршам таймерлері қозғалысты реттеудің дұрыс уақытын көрсетсе, қажет емес жасыл сигналдарға кететін уақыт үнемделеді. Әр жолақ үшін таймер - бұл трафикті басқарудың қарапайым тәсілі. Егер сол таймерлер нақты уақытты болжаса, онда автоматты түрде жүйе тиімдірек болады.



3.1-сурет – бағдаршамды дұрыс басқару үлгісі

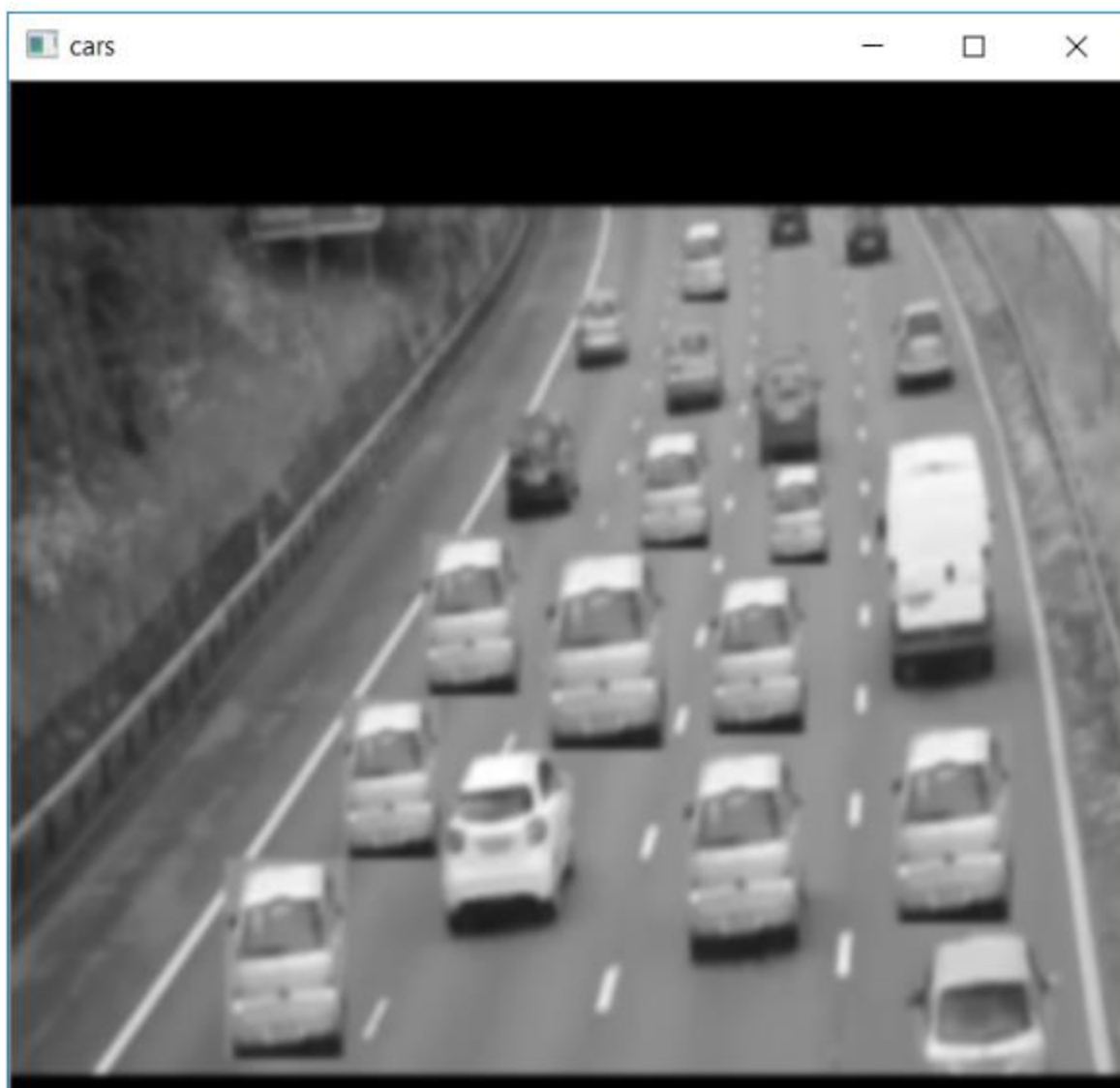
Жиналу мәселесінің ең дұрыс шешімі - жиналған тараптардың алдымен қиылысты кесіп өтуіне мүмкіндік беру. Жоғарыда ұсынылған шешімді Python-дағы OpenCv кітапханасында немесе Object Detection арқылы Machine Learning көмегімен суреттегі автокөліктердің санын санау, сондай-ақ әр түрлі көлік түрлерін ажырату үшін қолдануға болады.

OpenCv әдісі:

- Көше қиылысының екі жағына камера қою және қиылыстың екі жағында көлік құралдарының санын алу үшін суреттерді өңдеу.

Түсірілген сурет мынадай кезекте өңделеді:

1. Python-да OpenCv Library көмегімен оқылады
2. қажеттілікке сәйкес өзгертілді
3. кара-ақ түске ауыстырылады
4. Гаусс шуын жою
5. Тұз бен бұрыштың шуын жою
6. Жуылу
7. Анықтамалық суреттен айырмашылықты алады
8. Пайыздың өзгеруін табады
9. Автокөлік нөмірін алады



3.2-сурет – Қара-ақ түске айналдыру процесі



3.3-сурет – Жиіектерді анықтау кескіні

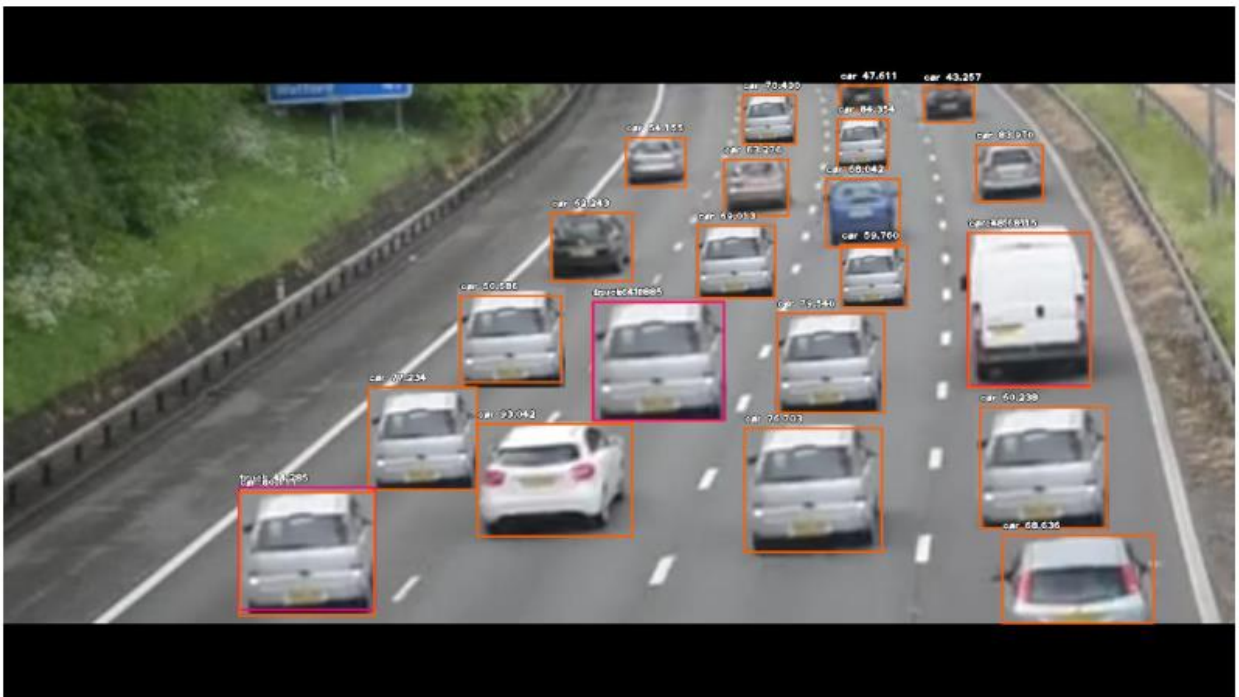
Машиналарды оқыту әдісі - қиылыстың әр жағында әр түрлі көліктердің санын алу үшін қиылыстың екі жағының суретін түсіріп, оны объектіні анықтау модуліне тапсырады.

Түсірілген сурет:

1. Жолын анықтау үшін алдын-ала өңделген.
2. Модель мұра болады, немесе жаңа - автомобильдер мен әртүрлі көлік құралдарын анықтау үшін жасалады.
3. Кескін жүктеліп, барлық анықталулар сақталады және жабылады.
4. Автомобильдер анықталған кезде есептеледі.



3.4-сурет – Кескін енгізу



3.5-сурет – Анықталған объектілер

Бұл сандар қалай басқарылады? Тізімде сақталғаннан кейін, автокөліктердің саны кептелісті тезірек жою үшін басқарылады.

Бір көліктің 3 жолақты жолға қосқан үлесін ескере отырып, қиылысты кесіп өту үшін 2 секунд уақыт керек, егер автомобильдер санының мәні 10-дан аспаса (Қалыпты мән), онда трафиктің сағат тілі бойынша қалыпты ағымы сақталады, қалған жиналған алгоритм пайда болады.

```
if List[MaxIndex]<=NormVal :  
    NormalTrafficS(List)  
else :  
    CongestedTrafficS(List)
```

3.6-сурет – Трафикті анықтау синтаксисі

Шарттар:

Цикл аяқталды, яғни сигнал жасыл болса, ол тағы да жасыл түске енеді, барлық қалған 3 сигнал мүмкіндікке ие болады. Максималды күту уақыты, яғни кез-келген тарапты күтудің максималды уақыты - 180 секунд немесе 3 минут. Автокөлік қорабы аз, мысалы, егер екі жақта автомобильдер өте аз болса, содан кейін бұл бірнеше машинаны максималды 180 секунд күтуге мәжбүр етудің орнына сәйкес жағы жасыл түске айналады. Бір жақты жағдайда тек бір жақты қозғалыс ағыны бар, және қиылыстың басқа жақтары бос болса, тиісті жаққа толық артықшылық беріледі. Бұл эвакуация сияқты жағдайларда өте пайдалы.

Трафикті жинау алгоритмдерінің түрлері туралы айтатын болсам алгоритмнің екі түрі бар. Жобада алгоритмдердің екі түрі әзірленген:

- Айнымалы алгоритм (Variance-Based Algorithm);
- Плиталарды бөлу алгоритмі (Slab Division Algorithm);

Ықтималдықтар теориясы мен статистика бойынша дисперсия - бұл сандар жиынының таралғандығын өлшеу әдісі. Дисперсия кездейсоқ шаманың күтілетін шамадан қаншалықты ерекшеленетінін сипаттайды. Айырмашылық жеке (байқалатын) және күтілетін шаманың арасындағы квадраттық айырмашылықтардың орташа мәні ретінде анықталады.

```
VarianceVal=Variance(List)  
if VarianceVal<100:  
    return 25  
elif VarianceVal<200:  
    return 40  
elif VarianceVal<400:  
    return 50  
else: #maximum 650  
    return 60
```

3.7-сурет – Айнымалы алгоритмін программада реализациялау

Төрт жағынан вагондар санының өзгеруі есептелгеннен кейін, бағдарлама сандардың қаншалықты таралғанын біледі және дисперсия мәніне қарай екі жағынан да жасыл жарықтың уақытын бөледі. Дисперсияға сәйкес таймерлер мәліметтерді аз тарату үшін қолданылады, сонымен қатар максималды мәнді жақындатуға тырысады, ал тізімнің екінші және үшінші максималды санынан аз болған жағдайда, бұл тізімді қиындатады және біртектілігін сақтайды. Жасыл сигналға уақытты бөлу үшін дисперсияны қолдану тиімділікті арттырады, өйткені алгоритм статикалық емес, динамикалы және жағдайға тәуелді болады.

Мысалы:

Тізім = [40,50,45,43] төмен дисперсияға әкеледі (13.25), осылайша екі жаққа бірдей және салыстырмалы түрде аз уақыт беріледі, жасыл сигнал әділ және тиімді болады.

Тізім = [1,2,4,60] жоғары дисперсияға әкеледі (624.6) және Автокөлік жағдайының аздығын ескергеннен кейін тиісті уақыттық модельдер бөлінеді.

Тізім = [5,20,30,45] нәтиже қалыпты дисперсияға әкеледі (212.5) және ең көп дегенде 50 секунд таймері бөлінеді.

Плиталар төменгі және жоғарғы шекаралары бар сандар жиынтығынан тұрады. Плиталар жасау және плиталар негізінде автокөлік санын бөлу және қажетті уақытты бөлу толып жатқан көлік ағынын бірқалыпты ету үшін жасалуы мүмкін.

```
...
SLABS
Congestion_0 - 00-10 -->20 secs
Congestion_1 - 10-40 -->40 secs
Congestion_2 - 40-60 -->60 secs
...
```

3.8-сурет – Плиталарды бөлу алгоритмін программада реализациялау

Жол қиылысының екі жағындағы автомобильдердің саны есептелгеннен кейін, плиталарды әр түрлі жақтарға уақытты белгілеу үшін пайдалануға болады, олар қиылысты кесіп өтуді және жоғарыда аталған жағдайларды сақтауды ескере отырып, автомобильдердің ең көп санын құрайды. трафик ағындары тиімді, тегіс және әділетті. Плиталарға таймерлер бөлінген, сол плитадағы автомобильдердің ең көп санын жарты автомобильдердің қиылысты өтуіне мүмкіндік бере отырып бөлу керек, бір автомобильдің үш жолақты жолдағы үлесін ескеріп 2 секундты құрайды. Мысалы:

Тізім = [10,20,30,40] бастапқыда 1 және 2-плитаның үш элементі бар және кейін мәндерге сәйкес өзгереді.

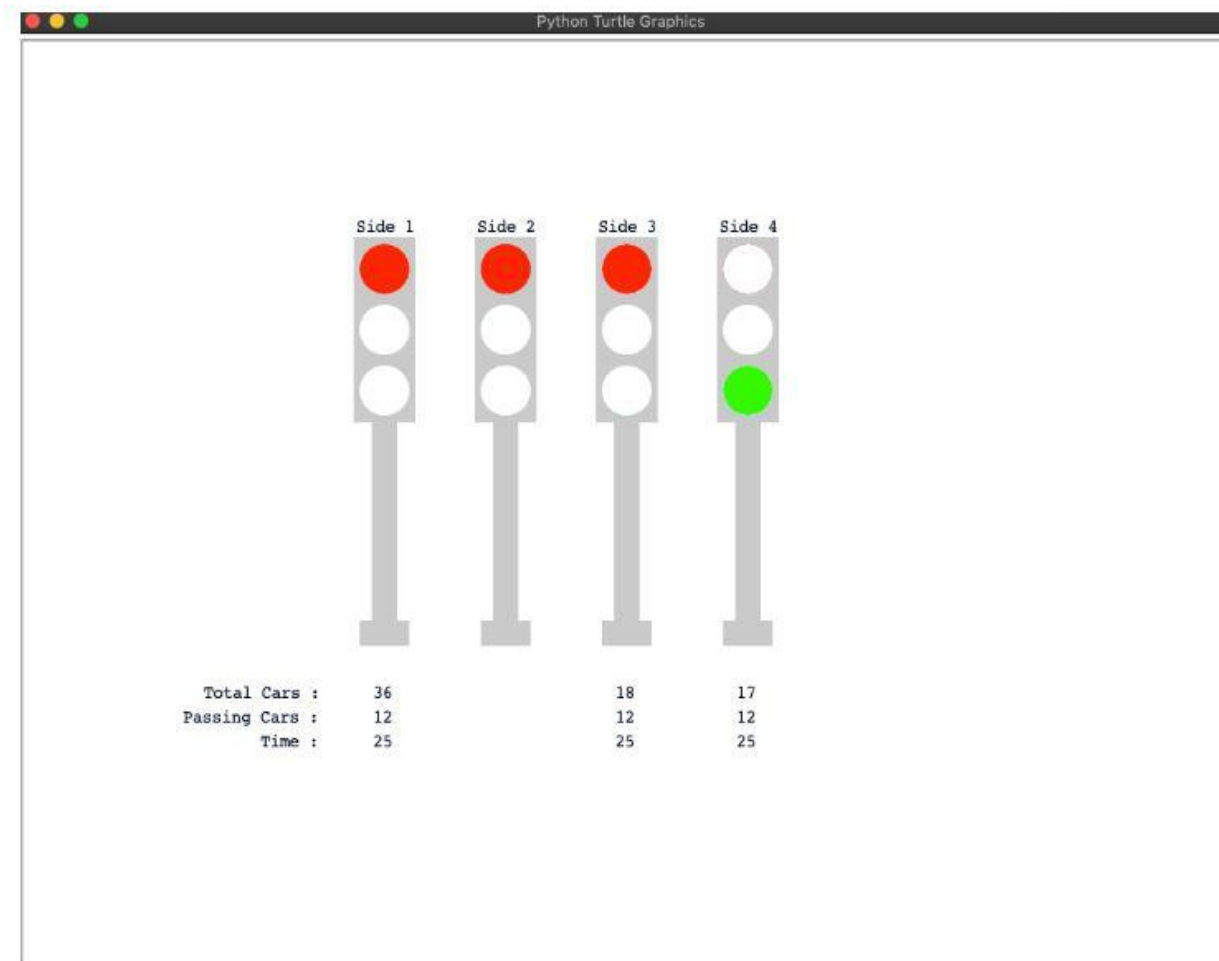
Екі алгоритмды салыстыратын болсақ, айнымалы алгоритм:

- уақытты тізімнің өзгеруіне сәйкес есептейді;
- динамикалық тәсілдегі нәтижелер;

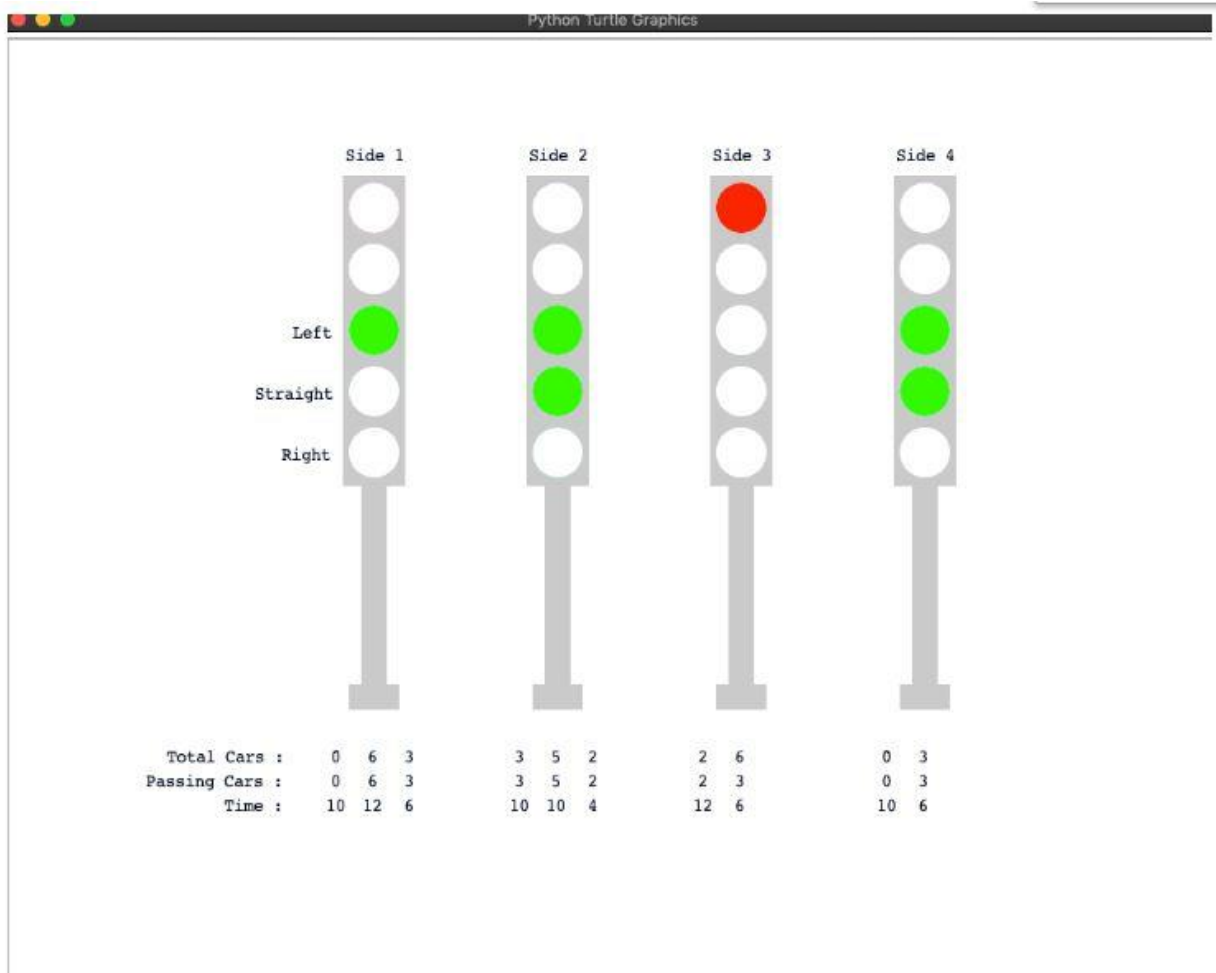
- тиімдірек;

Плиталар-дивизион алгоритмі:

- уақытты бір индекстің мәні бойынша есептейді;
- айнымалы алгоритмге қарағанда аз динамикалық;
- салыстырмалы түрде азырақ;



3.9-сурет – Программаның қалыпты нұсқасын шығару



3.10-сурет – Прототиптің кеңейтілген нұсқасы

4 Экономикалық бөлім

Бұл жоба мегаполистегі бағдарламаларды реттейтін зияткерлік ақпараттық жүйені дамытуды көздейді. Бұл жүйеде мегаполистегі бағдарламаларды реттейтін зияткерлік ақпараттық жүйемен қамтамасыз ету жүйесі талқыланады.

Адам-сағатты үнемдеудің экономикалық тиімділігін талдау үшін ақпараттық жүйеге қабылданған және өңделген қосымшалар санын көбейту үшін деректер қолданылады. Осы мәліметтер негізінде Интернет жүйесін ақпараттық жүйеге орналастырудың экономикалық тиімділігін анықтау қажет.

Экономикалық тиімділікті талдау кезінде ең алдымен бағдарламалық жасақтаманы әзірлеу құнын есептеу қажет. Осы есептеулердің нәтижелерін пайдалана отырып, ақпараттық жүйенің нәтижелерінің экономикалық тиімділігін есептеу қажет.

4.1 Еңбек ресурстары

Жобаға тартылған қызметкерлердің саны және олардың айлық жалақысы 4.1-кестеде келтірілген.

4.1 кесте – Қызметкерлер туралы деректер

Лауазымы	Саны	Жалақы, тг
Программист	1	110000
Веб-дизайнер	1	70000
Жүйелік администратор	1	90000
БАРЛЫҒЫ		270000

Бағдарламалық жасақтаманы іске асырудың негізгі шығындары жұмысшыларға жалақы төлеуге байланысты, олар мына формула бойынша есептеледі:

$$\text{ФОТ} = Z_{\text{осн}} + Z_{\text{доп}}, \quad (4.1)$$

$Z_{\text{осн}}$ – негізгі жалақы.

$Z_{\text{доп}}$ – қосымша ақы.

Күнделікті орташа табыс формула бойынша анықталады:

$$D = \frac{ЗП}{Др}, \quad (4.2)$$

ЗП– 1 айлық жалақы мөлшері;

D_p – бір айдағы жұмыс күндерінің саны (30 күн)

Әзірлеушінің орташа күндік табысы:

$$D = \frac{110000}{30} = 3666 \text{ тг.} \quad (4.3)$$

Веб-дизайнердің орташа күндік табысы:

$$D = \frac{70000}{30} = 2333 \text{ тг.} \quad (4.4)$$

Жүйелік әкімшінің орташа күндік жалақысы:

$$D = \frac{90000}{30} = 3000 \text{ тг.} \quad (4.5)$$

Қызметкердің бір сағаттық жалақысы формула бойынша есептеледі:

$$P = \frac{D}{\text{Ч}_p}, \quad (4.6)$$

D – қызметкердің орташа күндік жалақысы;

Ч_p – жұмыс күнінің сағат саны (8 сағат).

Әзірлеушінің бір сағаттық жалақысы:

$$P = \frac{3666}{8} = 458,25 \text{ тг.} \quad (4.7)$$

Веб-дизайнердің бір сағаттық жалақысы:

$$P = \frac{2333}{8} = 291,6 \text{ тг.} \quad (4.8)$$

Жүйе әкімшісінің бір сағаттық жалақысы:

$$P = \frac{3000}{8} = 375 \text{ тг.} \quad (4.9)$$

4.2 кесте - сайттың кезеңдері және жұмсалған уақыт

№	Атауы	Жұмсалған уақыт, сағ
1	Мәселенің тұжырымы	16
2	Дизайн құру	16
3	Ақпаратты таңдау	8
4	Бағдарламаны орнату және са- тып алу	8
5	Жүйені құрылымдау	56
6	Деректер базасын құру	64
7	Жүйені құру	56
8	Тестілеу	8
9	Тексеру	8
	БАРЛЫҒЫ	240

Осылайша, барлық жұмыс үшін 30 күн қажет.

Қосымша жалақы негізгі жалақының 10% құрайды және формула бойынша есептеледі:

$$Z_{\text{доп}} = Z_{\text{осн}} * 10\% \quad (4.10)$$

$$Z_{\text{доп}} = 270000 * 10\% = 27000 \text{ тг.}$$

Осылайша, еңбек шығындары есептеулер бойынша және сәйкес:

$$\text{ФОТ} = 270000 + 27000 = 297000 \text{ тг.} \quad (4.11)$$

4.2 Техникалық жабдықтар

Жұмыста қолданылатын жабдықтардың сипаттамалары, сондай-ақ олардың құны 4.3 кестеде келтірілген.

4.3 кесте - жабдықтардың тізімі

Аты	Саны	Құны, тг
Ноутбук Acer Predator	3	360000
Xiaomi Mi тышқаны	3	15000
Роутер TP-LINK	1	5000
Принтер HP	1	25000
БАРЛЫҒЫ		380000

4.3 Техникалық бағдарламалық қамтамасыз ету

Ақпараттық жүйені әзірлеу кезінде келесі бағдарламалық қамтамасыздандыру қолданылды:

- Windows 10;
- MS Office 2016;
- Adobe Photoshop;
- MAMP;

Жұмыста қолданылатын және соған байланысты бағдарламалық қамтамасыздандыру құны 4.4 кестеде келтірілген.

4.4 кесте - бағдарламалық қамтамасыздандыру тізімі

Аты	Саны	Құны, тг
Windows 10	1	60000
MS Office 2016	1	Тегін
Adobe Photoshop	1	15000
MAMP	1	6000
БАРЛЫҒЫ		81000

4.4 Салық шығындарын есептеу

Әлеуметтік аударымдар (СО) формула бойынша есептеледі:

$$CO = (ЗП - ОПВ) * 3,5\% . \quad (4.12)$$

ЗП – еңбекақы;

ОПВ – міндетті зейнетақы жарналары;

Міндетті зейнетақы жарнасы жалақының 10% құрайды:

$$ОПВ = 270000 * 10\% = 27000 \text{ тг.} \quad (4.13)$$

Осылайша, СО болып табылады:

$$CO = (270000 - 27000) * 3,5\% = 8505 \text{ тг.} \quad (4.14)$$

Міндетті әлеуметтік медициналық сақтандыру (VOSMS) жарна келесі формула бойынша есептеледі:

$$VOSMS = ЗП * 2\% . \quad (4.15)$$

$$\text{ВОСМС} = 270000 * 2\% = 5400 \text{ тг}$$

Әлеуметтік салықты (СН) формула бойынша есептейміз:

$$\text{СН} = (\text{ЗП} - \text{ОПВ} - \text{ВОСМС}) * 9,5\% - \text{СО}. \quad (4.16)$$

$$\text{СН} = (270000 - 27000 - 5400) * 9,5\% - 8505 = 14067 \text{ тг}$$

4.5 кесте - бағдарламалық қамтамасыздандыру тізімі

Әлеуметтік сақтандыру жарналары (СО)	8505 тг
ВОСМС	5400 тг
Әлеуметтік салық (СН)	14067 тг
БАРЛЫҒЫ	27972

Жалпы салық шығындары 27972 теңгені құрайды.

4.5 Амортизацияны есептеу

Амортизация формула бойынша есептеледі

$$A_j = \frac{N_A * C * N}{100 * 12 * n}, \quad (4.17)$$

N_A - амортизация нормасы;

C - жабдықтың бастапқы құны;

N - жұмысты аяқтауға арналған күндер саны;

n - жұмыс күндерінің саны

Жабдықтардың тозуы:

$$A_1 = \frac{20 * 380000 * 30}{100 * 12 * 30} = 6333 \text{ тг}. \quad (4.18)$$

Бағдарламалық жасақтама үшін амортизациялық аударымдар:

$$A_2 = \frac{33,3 * 81000 * 30}{100 * 12 * 30} = 2247 \text{ тг}. \quad (4.19)$$

Амортизация:

$$A = 6333 + 2247 = 8580 \text{ тг}. \quad (4.20)$$

4.6 Электр энергиясына шығындарды есептеу

Электр жабдықтары өндіріс процесінде қолданылғандықтан, электр энергиясының құнын есептеу керек. Өнеркәсіптік қажеттіліктер үшін электр энергиясының құны жабдықтар мен қосымша қажеттіліктер үшін электр энергиясының құнын қамтиды.

4.6 кесте - Электр энергиясының шығындары

Атауы	Жаб- дықтың қуатты- лығы кВт	Қуат ко- эффици- енті	Жобалық жабдықтың жұмыс уақыты, сағ	Жобаны жасауға ар- налған жаб- дықтың жұмыс уақыты, күндер	Цена электро энергии	Сомасы, тг.
Ноутбук Acer Predator	1.49	0.9	240	30	25	9655,2
Роутер TP-LINK	0.08	0.2	240	30	25	115,2
Принтер HP	1.1	0.5	240	30	25	3960
БАРЛЫҚ электр энергиясы шығындары						13730,4

Энергия шығындарының мөлшері (ЭЭ) келесі формула бойынша есептеледі:

$$Z_{\text{Э}} = Mi \times Ki \times Ti \times Ц, \quad (4.21)$$

Біз әр жабдыққа есептеулер жүргіземіз:

1. Ноутбук Acer Predator: $Z_{\text{Э}} = 1.49 * 0.9 * 240 * 30 = 9655,2$ тг
2. Роутер TP-LINK: $Z_{\text{Э}} = 0.08 * 0.2 * 240 * 30 = 115,2$ тг
3. Принтер HP: $Z_{\text{Э}} = 1.1 * 0.5 * 240 * 30 = 3960$ тг

4.7 Үстеме есептеу

Үстеме шығындар барлық шығындардың 20% -ы ретінде есептеледі және формула бойынша анықталады:

$$H_p = (\text{ФОТ} + \text{СН} + \text{А} + \text{Э}) * 20\%, \quad (4.22)$$

$$H_p = (297000 + 14067 + 8580 + 13730,4) * 20\% = 66675 \text{ тг.}$$

Осылайша, ПП-ны әзірлеу құны болды:

$$C = \text{ФОТ} + \text{СН} + A + \text{Э} + \text{Н}_p \quad (4.23)$$

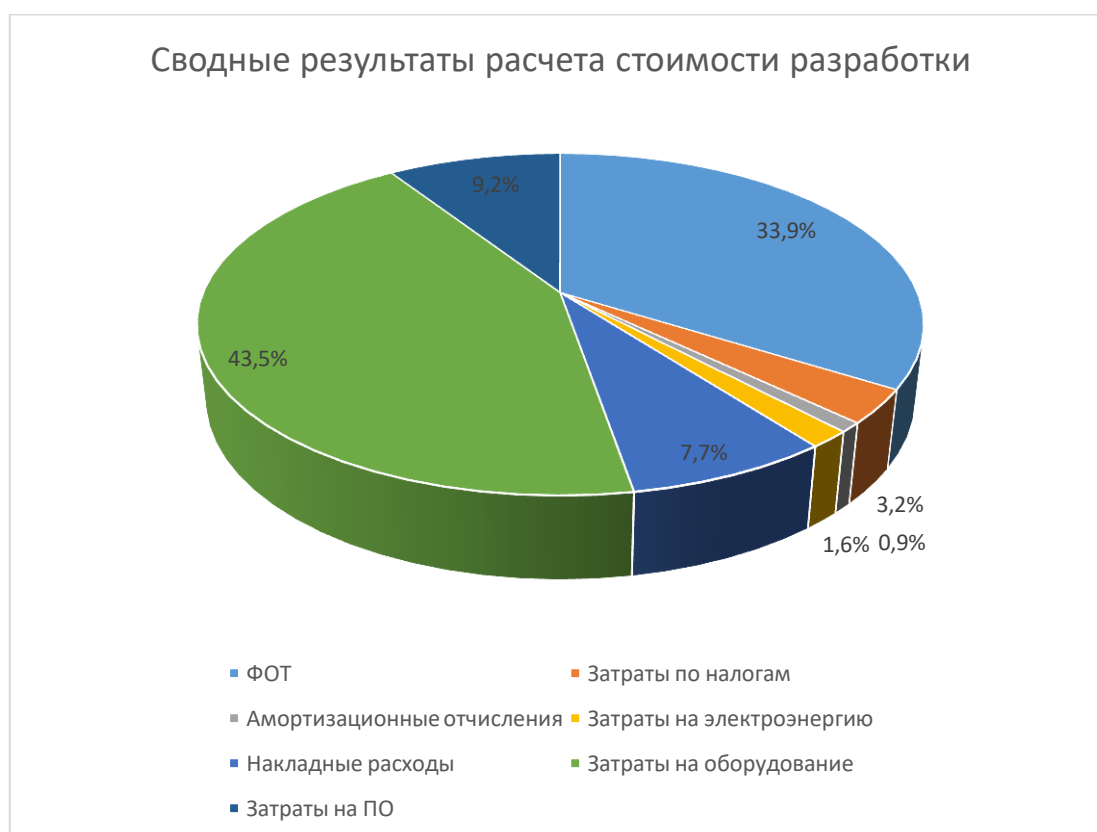
$$C = 297000 + 14067 + 8580 + 13730,4 + 66675 = 400052,4 \text{ тг}$$

4.8 Даму шығындарының сметасы

4.7-кесте - әзірлеудің болжамды шығындары

Аты	Сума, тг	Пайызбен, %
ФОТ	297000	33,9
Салық шығындары	27972	3,2
Амортизациялық аударымдар	8580	0,9
Электр энергиясының шығындары	13730,4	1,6
Есептік шығындар	66675	7,7
Жабдықтың құны	380000	43,5
Бағдарламалық жасақтаманың құны	81000	9,2
БАРЛЫҒЫ	874 957,4	100,00

Төменде есептеулердің жиынтық нәтижелерінің диаграммасы келтірілген



4.1 сурет – Жиынтық нәтижелерінің диаграммасы

Материалдық шығындарды, соның ішінде еңбекақы төлеуге, әлеуметтік қажеттіліктерді шегеруге, негізгі құралдарды амортизациялауға жұмсалған шығындарды есептегеннен кейін 874957,4 теңге аяқталды.

4.9 Жобаны іске асыру бағасы

Бағдарламалық жасақтамаға (ПП) бағаны есептеу, оны бір ұйым екінші ұйымның тапсырысы бойынша әзірлейді және қайталауға арналмаған, формула бойынша жүзеге асырылады.:

$$Ц_{ПП} = Z_{рпр} + П_{п} + НДС \quad (4.24)$$

где $Ц_{ПП}$ – бағдарламалық өнімнің бағасы, теңге;
 $Z_{рпр}$ – дизайнерлік шешімді әзірлеу құны, бұл жағдайда бағдарламалық өнім, теңге;
 $П_{п}$ – жоспарланған пайда, теңге;
НДС – қосылған құн салығы, теңге (12%).

Жоспарланған пайда даму құнының 20% құрайды.

$$П_{п} = C * 20\% = 400052,4 * 0,2 = 80010 \text{ тг} \quad (4.25)$$

НДС, ПР бойынша есептелген сома келесідей анықталады

$$НДС = (Z_{рпр} + П_{п}) * k_{НДС} \quad (4.26)$$

$k_{НДС}$ – қосылған құн салығының ставкасы 12%.

Осылайша, ҚҚ бойынша есептелген ҚҚС болып табылады

$$НДС = (874957,4 + 80010) * 0,12 = 114596 \text{ тг} \quad (4.27)$$

Бағдарламалық жасақтаманың жалпы бағасы:

$$Ц_{ПП} = 874957,4 + 80010 + 114596 = 1\,069\,563,4 \text{ тг} \quad (4.28)$$

Бағдарламалық жасақтаманың жалпы бағасы 1 069 563,4 тг.

Экономикалық тиімділікті есептеу

ІР енгізуден күтілетін жылдық экономикалық эффекттің мәні формула бойынша есептеледі:

$$Э_{г} = Э_{уг} - K \cdot E_{Н} \quad (4.29)$$

\mathcal{E}_r - күтілетін жылдық экономикалық тиімділік, теңге;
 \mathcal{E}_{yr} — күтілетін шартты жылдық жинақ, теңге;
 K — күрделі салымдар, теңге;
 E_H - күрделі салымдардың экономикалық тиімділігінің нормативті коэффициенті.

Күтілетін шартты жылдық жинақ формула бойынша анықталады:

$$\mathcal{E}_{yr} = C_1 - C_2 + \sum \mathcal{E}_i \quad (4.30)$$

Күтілетін шартты жылдық жинақтау:

$$\mathcal{E}_{yr} = 1\,069\,563,4 - 400\,052,4 = 669\,511 \text{ тг} \quad (4.31)$$

C_1 = ПП қолдану шығындары = 1 069 563,4 тг

C_2 = ПП-сыз шығындар = 400052,4 тг

Күрделі салымдардың экономикалық тиімділігінің нормативтік коэффициенті формула бойынша анықталады:

$$E_H = \frac{1}{T_H} \quad (4.32)$$

яғни T_H — капиталды салымдарды өтеудің нормативтік мерзімі, жылдар.

Капиталды салымдардың экономикалық тиімділігінің стандартты коэффициенті болып табылады:

$$E_H = \frac{1}{4} = 0,25 \quad (4.33)$$

Күрделі салымдарды қайтарудың нормативті мерзімі. техникалық құрал-жабдықтардың ескіруі және IP-жобалық шешімдер негізінде қабылданды ($T_H = 1,2,3...n$), бағдарламалық жасақтама өнімдері үшін өтелімді мерзімі 4 жылды құрайды.

ИС енгізудің экономикалық тиімділігі:

$$\mathcal{E}_r = 669\,511 - 400\,052,4 \cdot 0,25 = 569\,498 \text{ тг.} \quad (4.34)$$

Күрделі салымдардың экономикалық тиімділігінің есептік коэффициенті:

$$E_p = \frac{\Delta_{\text{уг}}}{K} \quad (4.35)$$

яғни E_p - күрделі салымдардың экономикалық тиімділігінің есептік коэффициенті;

$\Delta_{\text{уг}}$ — күтілетін шартты жылдық жинақ, теңге;

K — жүйені құру үшін күрделі салымдар, теңге.

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{669511}{400052,4} = 1,6 \quad (4.36)$$

Күрделі салымдарды өтеудің есептік мерзімі:

$$T_p = \frac{1}{E_p} \quad (4.37)$$

яғни E_p - күрделі салымдардың экономикалық тиімділігі коэффициенті.

Күрделі салымдарды өтеудің есептік мерзімі:

$$T_p = \frac{1}{1,6} = 0,6 \text{ жыл} \quad (4.38)$$

Есептеулер нәтижелері төмендегі кестеде келтірілген.:

4.8-кесте - Есептеулер нәтижелері

Көрсеткіштердің атауы	Мәні
Шартты жылдық үнемдеу, теңге	669511
Күрделі салымдардың экономикалық тиімділігі коэффициенті (E_p)	1,6
Күрделі салымдарды қайтару мерзімі (T_p)	0,6

5 ӨМІРТІРШІЛІК ҚАУІПСІЗДІГІ

«Өмір-тіршілік қауіпсіздігі» – адам өміріне төнетін жалпы қауіп-қатерлерді зерттейтін, соған қарсы қоюға болатын шараларды әзірлейтін ғылыми білімнің саласы. Ол нақты машиналардың, құрал-жабдықтардың, механизмдердің, өндірістің қауіпсіздігін қамтамасыз ету мәселелерін шешпейді, онымен арнайы мамандықтар айналысады (атом қауіпсіздігі, электр қауіпсіздігі, жаралыс, өрт қауіпсіздігі және т.с.с). «Өмір-тіршілік қауіпсіздігі» пәнінің мақсаты – қауіпсіздік саласында жалпы біліктілікті көтеру, адамдардың ойлау қабілетін өмірдің қауіпсіздігін қамтамасыз етуге бағыттау -осылар барлық арнайы мамандықтардың қауіпсіздігінің ғылыми-әдістемелік іргесі болып саналады. Осы курсты оқып өткен адамның өз ортасындағы қауіпті және өзі мен өзгеге зиянын тигізетін құбылыстарды тануға, соған қарсы білікті әрекет жасауына мүмкіншілігі болады.

Дипломдық жобаның негізгі мақсаты мегаполис бағдаршамдарын реттеудің зияткерлік ақпараттық жүйесін әзірлеу болып табылады. Бұл дипломдық жобаны жасау барысы компьютерлік технологиялар арқылы жүзеге асырылды. Жұмыс барысында адам өміріне қауіп тудыратын факторлардың алдын алу жағдайлары қарастырылды. Қарастырылатын жұмыс орны ретінде осы жобаны жасау барысында программисттер отырған бөлмені қарастырдым.

5.1 Жұмыс орнының эргономикасы және еңбек жағдайларын талдау

Ғылыми-техникалық прогрестің дамуымен адамдардың өз еңбек міндеттерін қауіпсіз орындау мүмкіндігі маңызды рөл атқарады.

Осыған байланысты адамның еңбек қауіпсіздігі және өмір сүру қауіпсіздігі туралы ғылым құрылып, дамып келеді.

Өмір тіршілігінің қауіпсіздігі (БЖД) - бұл өмір сүру ортасындағы адамның қауіпсіздігін қамтамасыз етуге, оның денсаулығын сақтауға, зиянды және қауіпті факторлардың рұқсат етілген мәндерге дейін әсерін төмендету жолымен қорғау әдістері мен құралдарын әзірлеуге, бейбіт және соғыс уақытындағы төтенше жағдайлардың салдарын жоюдағы залалды шектеу жөніндегі шараларды әзірлеуге бағытталған іс-шаралар кешені.

Еңбекшілердің денсаулығын қорғау, еңбек жағдайларының қауіпсіздігін қамтамасыз ету, кәсіптік аурулар мен өндірістік жарақаттануды жою адам қоғамының басты қамқорлығының бірін құрайды. Еңбекті ғылыми ұйымдастырудың прогрессивті нысандарын кеңінен қолдану қажеттілігіне, қол, біліктілігі аз еңбекті барынша азайтуға, кәсіби аурулар мен өндірістік жарақаттануды болдырмайтын жағдайды жасауға назар аударылады.

Жұмыс орнында өндірістің қауіпті және зиянды факторларының ықтимал әсерінен қорғау шаралары көзделуі тиіс.

Бұл факторлардың деңгейлері құқықтық, техникалық және санитарлық-техникалық нормаларда айтылған шекті мәндерден аспауы тиіс.

Бұл нормативтік құжаттар жұмыс орнында жұмыс істеушілерге қауіпті және зиянды факторлардың әсері жойылған не мүлдем жойылған не рұқсат етілген шектерде болатын еңбек жағдайларын жасауға міндеттейді.

5.1.1 Монитормен жұмыс

Электрондық-сәулелі түтік (ЭЛТ) - бұл электрондық зеңбірек. Бұл дегеніміз, ЭЛТ теріс зарядталған, демек, ЭЛТ-дан тыс оң зарядталған бөлшектердің жиналуы.

Адам қоршаған ортада оң және теріс иондардың арақатынасы бірдей болғанда жақсы сезінеді. Алайда, монитордың экраны алдында оң иондар артық пайда болады. Әрқашанда офистің ауасындағы шағын бөлменің (шаң, темекі түтіні және т. б.), оң зарядталған иондардың ағынымен айдалады және экран алдында отырған программисттің беті мен көздерінде тұрады.

Мұндай "бомбалаудың" нәтижесінде программистте пайда болуы мүмкін:

- бас ауруы, ұйқысыздық;
- көздің шаршауы;
- терінің тітіркенуі.

5.1.2 Көрініс әсері және жұмыс үстелі

Ашық боялған кеңсе жиһазы және үлкен терезелер қосымша жарық көздері болып табылады. Өте жарық бөлмеде монитор экранындағы әріптер мен сандар нашар көрінеді. Бұл бас ауруы, көрудің нашарлауы, концентрацияның төмендеуі, сондай-ақ ақпараттың дұрыс қабылданбауынан жұмыста қателіктерге әкеледі.

5.1.3 Шу

Шуылдың жоғары деңгейі түсті сигналдарды тануда қиындықтар туғызады, түсті сигналдарды қабылдау жылдамдығын төмендетеді, түсті қабылдау жылдамдығын, көру өткірлігін, көру бейімделуін төмендетеді, көрнекі ақпаратты қабылдауды бұзады, үйлестірілген әрекеттерді тез және нақты орындау қабілетін төмендетеді, еңбек өнімділігін 5-10% азайтады. Дыбыс қысымы 90 Дб жоғары шу деңгейінің ұзақ әсері еңбек өнімділігін 30-60% төмендетеді. Программистерді медициналық тексеру еңбек өнімділігінің төмендеуінен басқа жергілікті әрекет кезінде шудың жоғары деңгейі шаршауға, есту қабілетінің нашарлауына әкеп соқтыратынын көрсетті. Сонымен қатар, жалпы әрекет кезінде шудың жоғары деңгейі жүрек қызметінің ырғағының бұзылуына, қан қысымының өзгеруіне, тыныс алу

органдарының нашарлауына әкеледі. Бөлмедегі шу көздері баспа құрылғылары болып табылады.

5.1.4 Артық жылу бөлу

Сыртқы ортаның жоғары температурасы тез шаршауға алып келеді, көру және есту ақпаратын қабылдау жылдамдығын, жүрек қызметінің бұзылуы (жүрек соғуының жылдамдығының артуы), қан қысымының өзгеруі салдарынан адамның жалпы тежелуін төмендетеді.

Көптеген бағдарламашылар ақыл - ойдың шамадан тыс күшеюі, көру және есту аппараттарының шамадан тыс күшеюі, еңбектің монотондылығы, эмоциялық артық жүктеме сияқты психофизикалық факторлардың әсерімен байланысты. Көрсетілген қолайсыз факторлардың әсері дамып келе жатқан шаршаудан туындайтын жұмысқа қабілеттіліктің төмендеуіне әкеледі. Шаршаудың пайда болуы мен дамуы орталық жүйке жүйесінің жұмыс процесінде пайда болатын өзгерістермен, бас миының қабығындағы тежеу процестерімен байланысты.

5.1.5 Жұмыс орнының жарықтандырылуы

Жұмыс орнын ұтымды жарықтандыру адамның еңбек қызметінің тиімділігіне әсер ететін, жарақаттану мен кәсіби аурулардың алдын алатын маңызды факторлардың бірі болып табылады. Дұрыс ұйымдастырылған жарықтандыру қолайлы еңбек жағдайын жасайды, еңбек қабілеттілігі мен өнімділігін арттырады. Программистің жұмыс орнындағы жарықтандыру қызметкер өз жұмысын кернеусіз орындай алатындай болуы тиіс. Көру мүшелерінің шаршауы бірқатар себептерге байланысты:

- 1) жарықтың жеткіліксіздігі;
- 2) шамадан тыс жарықтандыру;
- 3) жарықтың дұрыс емес бағыты.

Жарықтандырудың жеткіліксіздігі көру кернеуіне алып келеді, назар аударуды әлсіретеді, уақытынан бұрын шаршаудың туындауына алып келеді. Жарықтың шамадан тыс ашық болуы көзді соққылау, тітіркену және кесу тудырады.

Жұмыс орнында жарықтың дұрыс емес бағыты күрт көлеңкелер, құймақ жасай алады, жұмыс істеушіге бағдар бере алады. Осы себептердің барлығы жазатайым оқиғаға немесе кәсіби ауруға әкелуі мүмкін, сондықтан жарықтың дұрыс есебі маңызды.

5.1.6 Микроклимат параметрлері

Микроклимат параметрлері кең шектерде өзгеруі мүмкін, ал адамның тіршілік әрекетінің қажетті шарты терморегуляцияның арқасында дене температурасының тұрақтылығын сақтау болып табылады, яғни ағзаның қоршаған ортаға жылу беруді реттеу қабілеті. Микроклиматты нормалау принципі-адам денесінің қоршаған ортамен жылу алмасуына оңтайлы жағдай жасау.

Есептеу техникасы маңызды жылу бөлу көзі болып табылады, ол температураның жоғарылауына және бөлмедегі салыстырмалы ылғалдылықтың төмендеуіне әкелуі мүмкін. Компьютерлер орнатылған үй-жайларда микроклиматтың белгілі бір параметрлері сақталуы тиіс. СН-245-71 санитарлық нормаларында қолайлы жағдай жасайтын микроклимат параметрлерінің шамалары орнатылған. Бұл нормалар жыл уақытына, еңбек процесінің сипатына және өндірістік үй-жайдың сипатына байланысты белгіленеді (4.1 кестеде көрсетілген).

Есептеу орталықтарының қызметкерлері орналасқан үй-жайлардың көлемі ауысымда бір мезгілде жұмыс істейтіндердің ең көп санын ескере отырып, 19, 5м³/адамнан кем болмауы тиіс. Компьютерлер орналасқан үй-жайларға таза ауаны беру нормалары 4.2 кестеде келтірілген.

5.1 кесте. Компьютерлер орнатылған үй-жайларға арналған микроклимат параметрлері

Жыл мезгілі	Микроклимат параметрлері	Шамасы
Суық	Үй-жайдағы ауа температурасы салыстырмалы ылғалдылық Ауа қозғалысының жылдамдығы	22...24°C 40...60% до 0,1м/с
Жылы	Үй-жайдағы ауа температурасы салыстырмалы ылғалдылық Ауа қозғалысының жылдамдығы	23...25°C 40...60% 0,1...0,2м/с

5.2 кесте. Компьютерлер орналасқан үй-жайларға таза ауаны беру нормалары

Үй-жайдың сипаттамасы	Үй-жайға берілетін таза ауаның көлемдік шығыны, сағатына бір адамға м ³
Бір адамға 20м ³ дейін көлем	30-дан кем емес
Бір адамға 20 ... 40м ³	20-дан кем емес
Бір адамға 40м ³ артық	Табиғи желдету

Қолайлы жағдайларды қамтамасыз ету үшін ұйымдастыру әдістері (жыл және тәулік уақытына байланысты жұмыстарды жүргізуді ұтымды ұйымдастыру, еңбек және демалыс кезектесуі), сондай-ақ техникалық құралдар (желдету, ауаны баптау, жылыту жүйесі) қолданылады.

5.1.7 Жұмыс орнының эргономикасы

Ғылыми-техникалық прогресс, әртүрлі технологиялық процестерді механикаландыру және автоматтандыру адамның еңбек сипатының түбегейлі өзгеруіне әкеп соқты. Ол "адам - машина - өмір сүру ортасы" жүйелерінде басқару функциясын жүзеге асыратын оператор ретінде әрекет ете бастады. Адамға дене жүктемелері азайды, оның еңбек жағдайлары жақсарып, жарақат алу мүмкіндігі төмендеді. Алайда, бұл оның жұмысын толық жайлы етпеді, ал жүйенің өзі экологиялық қауіпсіз, өйткені бұл ретте жаңа, деп аталатын психофизиологиялық зиянды факторлар маңызды мәнге ие болды. Адамның психикалық қызметінің мәні айтарлықтай өсті және сонымен бірге – ТЖМСО жүйесін жетілдіру қажеттілігі. Сонымен қатар, барлық жүйе жұмысының сенімділігі мен тиімділігі елеулі түрде тәуелді болатын, оператордың қауіпсіз, авариясыз, жайлы еңбек жағдайларын қамтамасыз ету проблемасы маңызды болды.

Мұндай еңбек жағдайлары бірінші кезекте эргономикалық талаптар тұрғысынан программистің жұмыс орнын дұрыс ұйымдастырумен қамтамасыз етіледі.

Жұмыс орнын жобалау кезінде СанЕЖН-да орнатылған талаптарды ескере отырып, жұмыс орнын таңдау керек. 2.2.2/2.4.1340-03.

ПЭВМ конструкциясы бейне дисплей терминалының (ВДТ) экранын фронтальды бақылауды қамтамасыз ету үшін берілген қалыпта бекітілген көлденең және тік жазықтықта корпусың бұрылу мүмкіндігін қамтамасыз етуі тиіс. ПЭВМ дизайны корпусы тыныш жұмсақ тонға бояуды қарастыру керек. ПЭВМ корпусы, пернетақта және басқа да блоктар мен ПЭВМ құрылғылары 0,4-0,6 шағылысу коэффициенті бар күңгірт беті болуы тиіс және жарықты жасауға қабілетті жылтыр бөлшектері болмауы тиіс.

ВДТ конструкциясы жарықтылық пен контрастылықты реттеуді көздеуі тиіс.

ЭЕМ пайдалануға арналған үй-жайларда қолданыстағы нормативтік құжаттама талаптарына сәйкес келетін табиғи және жасанды жарық болуы тиіс.

Сынаушының жұмыс үстелін ВДТ бүйір жағынан жарық ойықтарына бағытталатындай етіп орналастыру керек және табиғи жарық көбінесе солға құлап кетеді.

Бір жұмыс орнына жалпақ дискретті экрандар (сұйық кристалды, плазмалық) базасында ВДТ бар ПЭВМ қолданушыларының ауданы кемінде 6 м² құрауы тиіс. Қарастырылып отырған үй-жайдың ауданы-12 м².

Бейнемонитор экраны пайдаланушының көзінен 600-700 мм қашықтықта, бірақ алфавиттік-сандық белгілер мен символдардың мөлшерін ескере отырып, 500 мм-ден жақын болмауы тиіс.

Жұмыс үстелінің құрылымы қолданылатын жабдықтың жұмыс бетінде оның саны мен құрылымдық ерекшеліктерін, орындалатын жұмыстың сипатын ескере отырып оңтайлы орналасуын қамтамасыз етуі тиіс.

Бұл ретте эргономиканың заманауи талаптарына жауап беретін әр түрлі конструкциялы жұмыс үстелдерін пайдалануға жол беріледі. Жұмыс үстелінің бетінде 0,5 – 0,7 шағылысу коэффициенті болуы тиіс.

Жұмыс креслоларының конструкциясы ПЭВМ-де жұмыс істеу кезінде тиімді жұмыс қалпын ұстап тұруды қамтамасыз етуі, шаршаудың дамуының алдын алу үшін мойын-иық аймағы мен арқаның бұлшық етінің статикалық кернеуін төмендету мақсатында қалпын өзгертуге мүмкіндік беруі тиіс.

Жұмыс креслосы көтермелі-бұрылмалы, орындық пен Арқаның биіктігі мен көлбеу бұрыштары бойынша реттелетін, сондай-ақ Арқаның орындықтың алдыңғы шетінен арақашықтығы болуы тиіс, бұл ретте әрбір параметрді реттеу тәуелсіз, оңай жүзеге асырылатын және сенімді бекітілуі тиіс.

Орындықтың, арқаның және креслоның басқа элементтерінің беті жартылай жұмсақ, тайғанамайтын, әлсіз электрленетін және ауа өткізбейтін, ластанудан жеңіл тазартуды қамтамасыз ететін жабыны бар болуы тиіс.

Жұмыс бетінің биіктігі 680-800 мм шамасында реттелуі тиіс, мұндай мүмкіндік болмаған жағдайда үстелдің жұмыс бетінің биіктігі 725 мм құрауы тиіс.

Жұмыс үстелінің биіктігі 600 мм – ден кем емес, ені 500 мм – ден кем емес, тізе деңгейіндегі тереңдігі 450 мм – ден кем емес және созылған аяқ деңгейінде-650 мм-ден кем емес аяққа арналған кеңістігі болуы тиіс.

Жұмыс креслосының құрылымы:

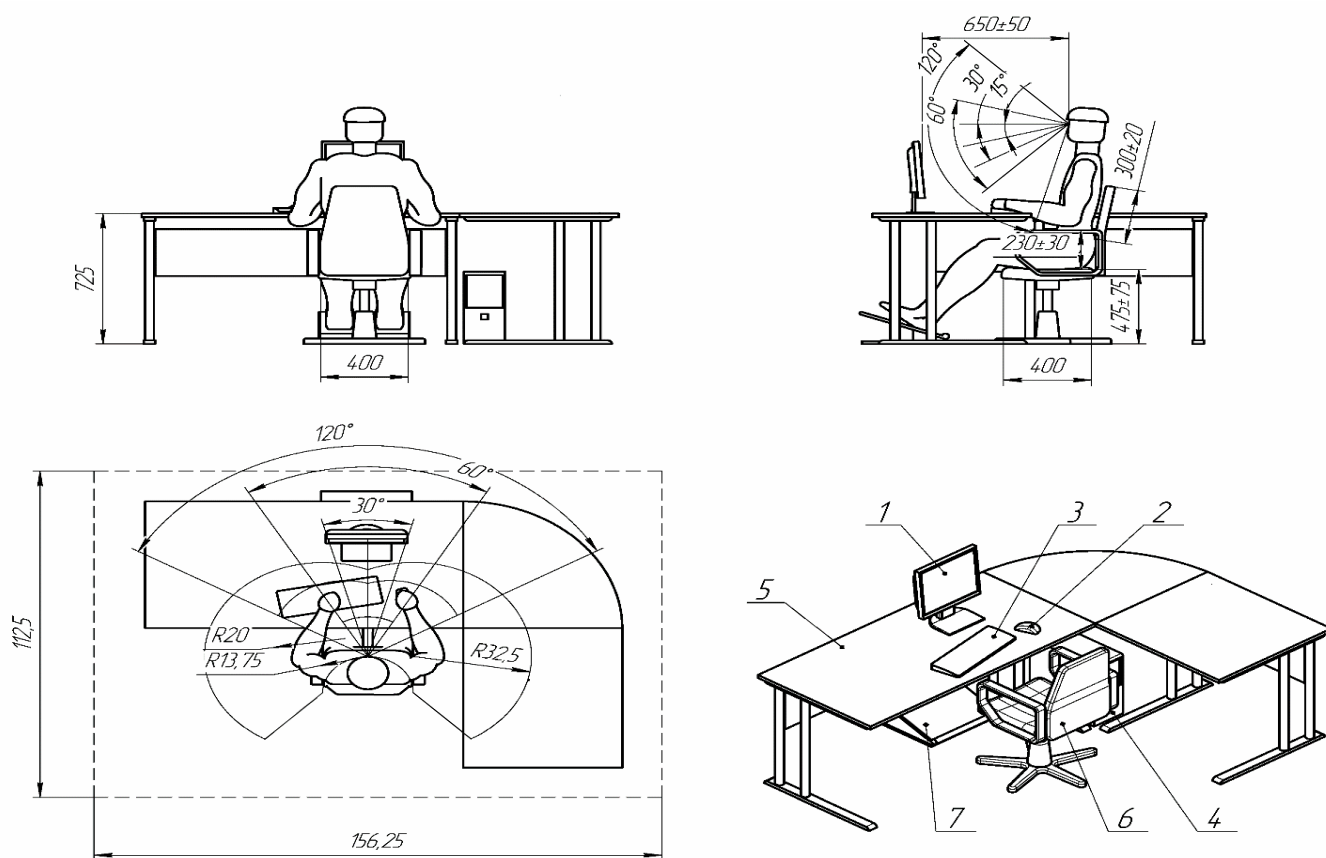
- орындық бетінің ені мен тереңдігі кемінде 400 мм;
- алдыңғы шеті дөңгелектелген орындықтың беті;
- орындық бетінің биіктігін 400-550 мм шегінде және алға қарай көлбеу бұрыштарын 15 градқа дейін реттеу., және артқа 5 град дейін.;
- Арқаның тірек бетінің биіктігі 300 ± 20 мм, ені 380 мм-ден кем емес және көлденең жазықтықтың қисық радиусы-400 мм;
- ± 30 градус шегінде тік жазықтықта Арқалықтың еңіс бұрышы;
- орындық алдыңғы шетінен Арқаның қашықтығын 260-400 мм шегінде реттеу;
- ұзындығы кемінде 250 мм және ені 50-70 мм стационарлық немесе алмалы-салмалы шынтақшалар;
- шынтақшаларды отырғыштың үстіндегі биіктігі бойынша 230 ± 30 мм және шынтақшалар арасындағы ішкі қашықтықты 350-500 мм шегінде реттеу.

ПЭВМ пайдаланушының жұмыс орнын ені 300 мм кем емес, тереңдігі 400 мм кем емес аяққа арналған тіреуішпен, биіктігі бойынша 150 мм дейін және тіреуіш бетінің еңіс бұрышы бойынша 20 градусқа дейін реттеумен жабдықтау қажет. Тұғырықтың беті кедір - бұдырланған және алдыңғы жиегі бойынша биіктігі 10 мм борты болуы тиіс.

Клавиатураны үстелдің бетіне пайдаланушыға қараған шетінен 100 – 300 мм қашықтықта немесе негізгі столшадан бөлінген биіктігі бойынша реттелетін арнайы жұмыс бетіне орналастыру керек.

Көрсетілген талаптарға сәйкес программистің жұмыс орны 4.1 суретте жобаланды.

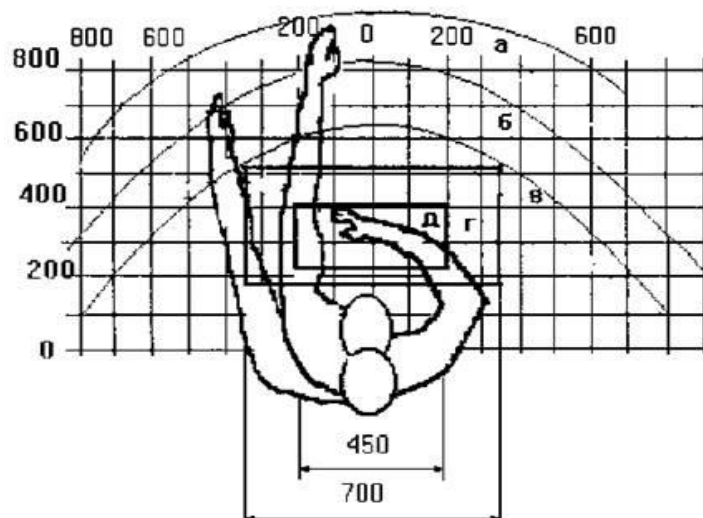
5.1 сурет - Программистің эргономикалық жұмыс орнының сызбасы



мұнда 1 – монитор; 2 – тышқан; 3 – пернетақта; 4 – жүйелік блок; 5 – жұмыс орны; 6 – кресло; 7 – тұғырық;

Қолдардың барынша қол жетімділік аймағы - бұл иық буынында қозғалғанда қолмен барынша созылатын доғалармен шектелген жұмыс орнының моторлы алаңының бөлігі.

Оңтайлы аймақ-шынтақ буындарында шынтақ нүктесінде тіреуі бар және салыстырмалы қозғалмайтын иығы бар қозғалыс кезінде білекпен сипатталатын доғалармен шектелген жұмыс орнының моторлы алаңының бөлігі.



5.1 сурет – Көлденең жазықтықтағы қолдардың қол жетімділік аймағы.

- а - барынша қол жеткізу аймағы;
- б - қол созылғанда саусақтардың қол жетімділік аймағы;
- в - алақанның жеңіл қол жеткізу аймағы;
- г - дөрекі қолмен жұмыс үшін оңтайлы кеңістік;
- д - жұқа қолмен жұмыс істеу үшін оңтайлы кеңістік.

Еңбек жағдайын жақсарту және өндірістегі жұмыс орындарын дұрыс эстетикалық безендіру Еңбекті жеңілдету үшін де, еңбек өнімділігіне оң әсер ететін оның тартымдылығын арттыру үшін де үлкен маңызға ие. Үй-жайлар мен жиһаздарды бояу көру, жақсы көңіл-күй үшін қолайлы жағдай жасауға ықпал етуі тиіс. Айтарлықтай нерв кернеуін және үлкен шоғырлануды талап ететін біркелкі ақыл - ой жұмысы орындалатын қызметтік үй-жайларда түсі тыныш реңктер-суық жасыл немесе көгілдір түстердің аз қанықпаған реңктері болуы тиіс.

Жұмыс орны - программист еңбек қызметін жүзеге асыратын және жұмыс уақытының көп бөлігін өткізетін кеңістіктің бір бөлігі. Инженердің еңбек қызметіне жақсы бейімделген жұмыс орны кеңістікке, формаға, өлшемге қатысты дұрыс және мақсатқа сай ұйымдастырылған, оған жұмыс кезінде ыңғайлы жағдайды және физикалық және психикалық шиеленіскен кезде жоғары еңбек өнімділігін қамтамасыз етеді.

Жұмыс орнын дұрыс ұйымдастыру кезінде инженердің Еңбек өнімділігі 8-20 пайызға өседі.

Осылайша, программист еңбегінің салауатты және қауіпсіз жағдайларын қамтамасыз ету проблемасын шешу - өндірістік қызмет процесінде аса маңызды міндеттердің бірі. Салауатты еңбек жағдайларын

жасау кезінде компьютермен жұмыс істеу кезінде шаршау азаяды, өндірістік жарақаттану қаупі төмендейді. Демек, өндірістік процестің тиімділігі артады.

5.2 Жасанды жарықтандыруды нүктелік әдіспен есептеу

Шамдарды үй-жайда орналастыру келесі параметрлермен анықталады, м.

Бастапқы деректер:

$$A = 10 \text{ м};$$

$$B = 8 \text{ м};$$

$$H = 5 \text{ м};$$

$$h_{pp} = 1,4 \text{ м};$$

$$h_c = 0,5 \text{ м};$$

$$h_p = H - h_c - h_{pp} = 5 - 1,4 - 0,5 = 3,1 \text{ м};$$

мұнда H - үй-жайдың биіктігі;

h_c - шамдардың жабыннан қашықтығы;

h_{pp} - еденнен жұмыс бетінің биіктігі;

$h_p = H - h_c - h_{pp}$ - есептік биіктігі, шам мен жұмыс алаңының ара қашықтығы;

Жұмыс орнында қолайлы көру жағдайларын жасау үшін, жарық көздерінің соқыр әсерімен күресу үшін шамдардың еденнен ең аз биіктігін шектеу талаптары енгізілген.

L - көрші шамдардың немесе қатарлардың арасындағы қашықтық (егер (A) және ені (B) бойынша үй – жай қашықтығы әртүрлі, онда олар LA және LB белгіленеді), 1-шеткі шамдардан немесе қатардан қабырғаға дейінгі қашықтық.

Шамдардың шеткі қатарынан қабырғаға дейінгі оңтайлы қашықтық l (0,3-0,5) L тең қабылдау ұсынылады.

Шамдардың тиімді орналасуының интегралды өлшемі $\lambda = L / h$ шамасы болып табылады, оның азаюы құрылғы мен жарықтандыруға қызмет көрсету қымбаттайды, ал шамадан тыс ұлғаюы жарықтандырудың біркелкі еместігіне әкеледі. Әр түрлі шамдарға λ мәндері үшін 4.1 кесте көрсетілген.

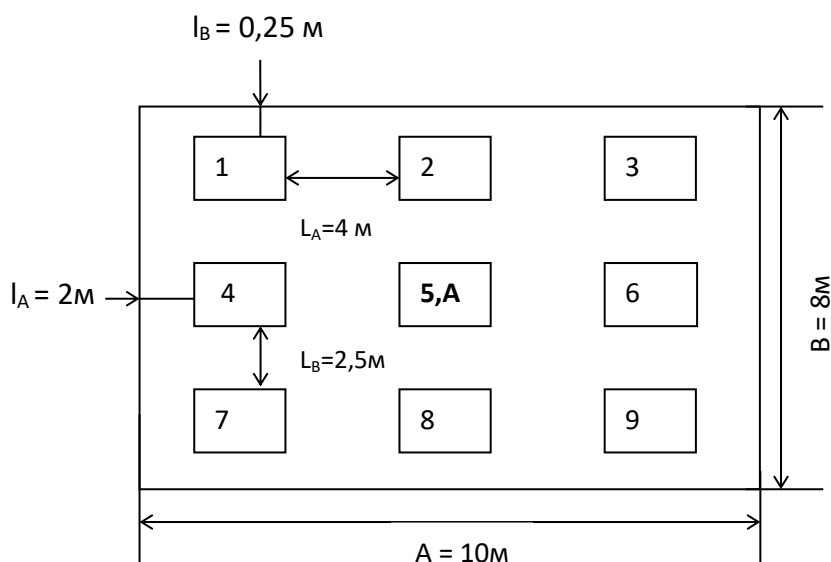
5.1 кесте - шамдардың әртүрлі типтері үшін λ мәндері.

Типтік қисық	λ мәні	
	Ұсынылатын	Ең көп рұқсат етілген
Қойылтылған (К)	0,4 - 0,7	0,9
Терең (Г)	0,8 - 1,2	1,4
Косинусты (Д)	1,2 - 1,6	2,1

Біркелкі (М)	1,8 - 2,6	3,4
Жартылай қалың (Л)	1,4 - 2,0	2,3

L шамдардың арасындағы қашықтық $L = \lambda * h_p$ ретінде анықталады.

Шамдардың орналасу схемасы 4.1 суретте көрсетілген.



5.1 сурет – Шамдардың орналасу схемасы

Қашықтықты есептейміз немесе оны логикалық ойлап анықтаймыз:

Біз 10 м бөлменің ұзындығы, яғни шамдарды олардың арасында 4 м болатындай етіп орналастырамыз, ал қабырғадан шамға дейінгі қашықтық 1 м болатындай етіп орналастырамыз. Демек, $((10 - (4 + 4)) / 2 = 1)$.

Сондай – ақ ені бойынша - ені 8 м, ал шамдардың арасындағы қашықтық 2,5 м, ал қабырғадан 0,25 м. Демек, $((8 - (2,5 + 2,5 + 2,5)) / 2 = 0,25)$.

Есептеу ыңғайлылығы үшін шамдарды нөмірледік.

Ең алдымен ортадағы негізгі шамды есептейміз, ол бізде 5 – ші нөмерлі шам. Және келесі басқа шамдар осы шамнан ара қашықтығынан есептелінеді.

- 1) төбенің және тік d арасындағы бұрышты анықтаймыз, бұл $tg\alpha$ шамның берілген биіктігі бойынша есептелінеді:

$$tg\alpha = \frac{d}{h_p} \quad (5.1)$$

мұнда d шамның осі проекциясынан есептік нүктеге дейінгі қашықтық (d шамасы жоспар бойынша өлшенеді), М;

$$h_p = H - h_c - h_{\text{pp}} = 5 - 0,4 - 1,2 = 3,4 \text{ м};$$

$$\text{tg } \alpha = \frac{1,2}{3,4} = 0,35 \quad (5.2)$$

Енді интернеттен Брадис кестесін қолданамыз.
Егер 0,35 болса, онда бұрыш тең болады: 19° .
Сонда $\cos 19^\circ = 0,94$ тең болады.

5.2 кесте – Тангенстің градус бойынша мәні.

Полная таблица тангенсов для углов от 0° до 360°

Угол в градусах	tg (Тангенс)
0°	0
1°	0.0175
35°	0.7002
36°	0.7265
37°	0.7536
38°	0.7813
39°	0.8098
40°	0.8391
58°	1.6003
59°	1.6643
60°	1.7321

$$e_1 = \frac{132 * \cos^3(19)}{3.4^2} = \frac{132 * 0,94}{11.56} = 10,73 \text{ лк}; \quad (5.3)$$

$I_a = 139$ кд тең 15° болған жағдайда; (4.3 - кесте бойынша)

Және де тангенс мәндерін арктангенске келтіру үшін 4.2 – кестеге қараймыз.

5.3 – кесте. Жарық күші I_ξ кд орналасу бұрышы ξ бойынша.

Светильник түрі	Жарық күші I_ξ кд орналасу бұрышы ξ бойынша										
	0	5	15	25	35	45	55	65	75	85	90
ПВЛМ-1x40	139	135	132	115	104	84	63	44	22	6	0

Келесі шамдар 2 мен 8 шамды аламыз олардын ара қашықтығы үстелден бірдей болғандықтан оларды бірге есептейміз.

2 және 8 шамдар үшін:

$$d_2 = 2,5 \text{ м}; \quad (5.4)$$

$$\alpha = \text{atan}(2,5/3,4) = \text{atan}(0,74) = 36,5^0; \quad (4.2 - \text{кесте бойынша})$$

Егер тангенс 0,74 болса, онда бұрыш тең болады: $36,5^0$. Сонда $\cos^3 36,5 = 0,37$ тең болады.

$$e_2 = \frac{108,2 * \cos^3(36,5)}{3,4^2} = \frac{108,2 * 0,37}{11,56} = 3,46 \text{ лк}; \quad (5.5)$$

$I_a = 108,2 \text{ кд}$ тең $36,5^0$ болған жағдайда; (4.3 – кесте бойынша)

Келесі шам 4 және 6 олардың да ара қашықтығы бірдей болғандықтан оны да бірге есептейміз.

4 пен 6 шамдар үшін:

$$d_3 = 4 \text{ м}; \quad (5.6)$$

$$\alpha = \text{atan}(4/3,4) = \text{atan}(1,18) = 49,7^0; \quad (4.2 - \text{кесте бойынша})$$

Егер тангенс 1,18 болса, онда бұрыш тең болады: $49,7$. Сонда $\cos^3 49,7 = 0,27$ тең болады.

$$e_3 = \frac{74,4 * \cos^3(49,7)}{3,4^2} = \frac{74,4 * 0,27}{11,56} = 1,74 \text{ лк}; \quad (5.7)$$

$I_a = 74,4 \text{ кд}$ тең $49,7^0$ болған жағдайда; (4.3 кесте бойынша)

Енді 1,3,7,9 шамдар бірдей қашықтықта бірдей бұрышпен тұрғандықтан оларды бірге есептейміз. Ен алдымен олардың ара қашықтығын тауып аламыз.

1,3,7,9 шамдар үшін:

$$d_4 = \sqrt{4^2 + 2,5^2} = 4,72 \text{ м}; \quad (5.8)$$

Енді бұрышын есептейміз:

$$\alpha = \text{atan}(4,72/3,4) = \text{atan}(1,39) = 53^0; \text{ (4.2 – кесте бойынша)}$$

Егер тангенс 1,39 болса, онда бұрыш тең болады: 53 . Сонда $\cos^3 53 = 0,6$ тең болады.

$$e_4 = \frac{60 * \cos^3(53)}{3,4^2} = \frac{60 * 0,6}{11,56} = 3,1 \text{ лк}; \quad (5.9)$$

$I_a=60$ кд тең 53^0 болған жағдайда; (4.3 – кесте бойынша).

Жарықтандырудың жиынтық шарты тең болады:

$$\sum E = e_1 * 1 + e_2 * 2 + e_3 * 2 + e_4 * 4 = 10,7 * 1 + 3,46 * 2 + 1,74 * 2 + 3,1 * 4 = 33,5 \quad (5.10)$$

Барлық деректерді қойып мына формуламен анықтаймыз:

$$E_{AG} = \frac{\mu \cdot F_l}{1000 \cdot K_3} \sum_1^n e_{AGn}, \quad (5.11)$$

- мұнда μ -қашықтағы шамдардан жарықтандыруды және қабырғалардан, төбеден және есептік бетінен шағылысқан жарық ағынын ескеретін коэффициент. Бұл коэффициент шамдардың қуатын арттыруға жол бермеу үшін түзету ретінде енгізіледі. Тікелей жарық эмальданған шамдарда $\mu=1,1-1,2$. Айналы кезде $\mu=1,0$. Тікелей жарық шамдарында $\mu=1,3-1,6$.

- Біз $\mu=1,1$ таңдаймыз.
- Жарық ағыны TL-D 58W / 865 типті лампаның $F_l = 5000$ Лм
- K_3 - қор коэффициенті, $K_3 = 1,2$ (4.4 кесте бойынша).

5.4 кесте - Қор коэффициентінің мәні

Үй-жайлар мен аумақтар	Үй-жайлар мысалдары	Қор коэффициенті Кз				
		ЕО және жарық өткізу орналасуы кезінде материал			Жасанды жарықтан-дыру кезінде	
		Тігінен	Көлбеу	Көлденең	Газразрядты шамдар	Қыздыру шамдары
1	2	3	4	5	6	7
1. Жұмыс аймағында бар ауа ортасы бар өндірістік үй-жайлар: а) 5мг/м3 шаң, түтін, тұяқтар	Агломерациялық фабрикалар, цемент зауыттары және құю цехтарының шабу бөлімшелері. Ұсталық, құю, дәнекерлеу, Құрастырмалы темір бетон цехтары.	1,5	1,7	2	2	1,7
б) 1-ден 5 мг/ м3 дейін шаң, түтін, тұяқтар;	Құрал-саймандық, құрастыру, Механикалық, механикалық жинау, тігу цехтары.	1,4	1,5	1,8	1,8	1,5
в) 1 мг/м3 кем шаң, түтін, тұяқтар;	Қышқылдар, сілтілер, күйдіргіш химиялық реактивтер, улы химикаттар, тыңайтқыштар өндіру жөніндегі химия зауыттарының цехтары.	1,3	1,4	1,5	1,5	1,3
г) Ылғалмен жанасу кезінде қышқылдардың әлсіз ерітінділерін түзуге қабілетті булардың, қышқылдардың, сілтілердің, газдардың едәуір шоғырлануы.	Гальваникалық жабындар және гальванопластика цехтары	1,5	1,7	2	1,8	1,5
2. Ауа тазалығы бойынша ерекше режимдегі Үй-жайлар: а) техникалық қабаттан б) үй-жайлардан төменнен		- -	- -	- -	1,3 1,4	1,1 6 1,2
3. Қоғамдық және тұрғын үй ғимараттарының үй-жайлары.	Кабинеттер, оқу үй-жайлары, зертханалар, сауда залдары және т. б.	1,2	1,4	1,5	1,5	1,3

Барлық деректерді формулаға қоямыз:

$$E_{AG} = \frac{5000 \cdot 2 \cdot 1,1 \cdot 33,5}{1000 \cdot 1,2} = 307,8 \text{лк} \quad (5.12)$$

Ал норма бойынша 200лк болу қажет. Демек, өндірістік жайды жарықтандыру жүйесін қайта құру қажет емес.

$$E_{AG} > E_n \quad (5.13)$$

Нормаланған жарықтандыруды қамтамасыз етеді.

Кәсіпорында дұрыс жобаланған және орындалған жарықтандыру қалыпты өндірістік қызмет мүмкіндігін қамтамасыз етеді.

ҚОРЫТЫНДЫ

Дипломдық жұмысты орындау барысында «Мегаполис бағдарламдарын реттеудің ақпараттық жүйесін құру» тақырыбына байланысты Мегаполис бағдарламдарын реттеу үшін Python бағдарламалау тілінде жүйе құрылды. Программа барлық заманауи технологияларды пайдалана отырып жасалды. Адамдардың өмірін жеңілдету үшін барынша ыңғайлы болатын жүйе. Түсінікті әрі функционалды. Трафикті реттеудің немесе басқарудың екі түрлі алгоритмі. Көліктер мен басқада көлік құралдарын тану арқылы.

Программаны жасау үшін барлық заманауи, қажетті және программаны барынша оңтайландыратын технологиялар қолданды. Ол технологиялар программаны жылдамдығын әлдеқайда арттырып, қолданушыларға еш қолайсыздық туғызбайды. Программа барлық стандарттарға сай жасалды.

«Мегаполис бағдарламдарын реттеудің ақпараттық жүйесін құру» тақырыбы бойынша жасалынған программада басқару жүйесімен қозғалатын барлық көліктер оңай анықталатын және әртүрлі көлік құралдарынан алынған деректерді сұрыптауға және көлік құралын бақылауға болатындай етіп, жекеленген құралдардың көмегімен оңай ажыратуға болады. Қазіргі трафикті басқару жүйесін жетілдірумен және оның трафикті басқарудағы тиімділігімен қатар, бұл жұмыс төтенше жағдай кезінде логистиканы жақсартуға және төтенше жағдай орын алған жағдайда қажетсіз кептелістерді азайтуға айтарлықтай ықпал етеді. Сонымен қатар, бұл жүйе жол апаттары болуының ықтималдылығын азайтады және жол ережесін бұзушылар мен қылмыскерлерді іздеу үшін де қолданылуы мүмкін. Сондықтан да, бұл жүйенің тиімділігі өте көп.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Cameron Davidson-Pilon. Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference // Addison-Wesley Data & Analytics, 2015.
- 2 UML – диаграмма классов // Сайттың электрондық нұсқасы
<https://prog-cpp.ru/uml-classes/>
- 3 Python is a programming language // Сайттың электрондық нұсқасы
<https://www.python.org/>
- 4 Хей Дж. Введение в методы байесовского статистического вывода. — М.: Финансы и статистика, 1987
- 5 Cameron Davidson-Pilon. Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference // Addison-Wesley Data & Analytics, 2015
- 6 Max Kuhn, Kjell Johnson. Applied Predictive Modeling // Springer, 2013
- 7 Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep learning
- 8 Christopher Bishop. Pattern Recognition and Machine Learning // Springer, 2006
- 9 Брайан Керниган (Brian W. Kernighan), Деннис Ритчи (Dennis M. Ritchie).
- 10 Mikhail Belyaev, Evgeny Burnaev, Ermek Kapushev, Maxim Panov, Pavel Prikhodko, Dmitry Vetrov, Dmitry Yarotsky. GTApprox: Surrogate modeling for industrial design // Advances in Engineering Software 102 (2016) 29–39