Кіріспе

Ақпараттық қоғам. Смартфон қолданушылардың саны күннен күнге артып барады. Смартфон құрылғылары адамдардың бұл заманда мобильді болуларына көмектеседі. Себебі ол дербес компьютерлермен ноутбуктарға қарағанда жеңіл, қайда барсанда өзінмен бірге алып жүруге болады. Бірақ оларға деген сұраныстың артуының тағы бір себебі интернет қосылу өте оңай. Әрбір адам смартфонмен оңай жұмыс істеп үйрене алады. Себебі қазіргі кездегі жасы үлкен ағаларымыз бен апайларымыз өздерінің күнделікті өмірлерінде компьютермен жұмыс істеуді талап етеді. Компьютермен жұмыс істеуді барлығы біле білмейді. Соңдықтанда смартфондар сұраныста.

Адамдардың жұмысы, күнделікті істейтін шаруалары өте көп. Және

бұл адамдардан оларды жинақы түрде сақтауды талап етеді. Себебі 1 күнде істейтін шаруаны есте сақтап жүрген қиын. Оларды есте сақтайм деп адам сол істейтін шаруаларына дұрыс мән бермей қалуы мүмкін. Соңдықтан көбісі екі варианттың біреуісін қолданады:

1. Шаруаларды қолмен парақ, арнайы бастаулы дәптер, күнделіктерге жазып қойады. Ал командамен жұмыс істегенде барлығына өздерінің істейтін міндеттерін бөліп қою үшін арнайы тақтағаларға да жазып қояды. Көбіне жазған істерін жазылған парағынды көшеде жүрсен немесе автобуста тұрсан алып шығу ыңғайсыз. Ал бітірген ісінді сызу үшін қалам керек.
2. Шаруаларды компьютер жадында сақтау. Смартфон құрылғыларына ондай мобильдік қосымшалар көп. Олар өте ыңғайлы. Үнемі сенімен, қалған істерінді көре салуға жеңіл, өзгерістер еңгізуге де оңай. Бірақ бірінші вариантқа қарағанда олар сызуға, сурет салуға, мүмкіншіліктер бермейді.

Соңдықтан Істер Тізімі мобильдік қосымшасын жасауды таңдадым.

I Тақырыб бойынша аналитикалық зерттеулер және оларды техникалық іске асырудағы негізгі шешімдер зерттемесі

* 1. Delphi жалпы сипаттамсы

Менде екі вариант болды Python немесе Delphi жұмыс істеу ортасында қолданылатын Object Pascal.

1. Delphi ортасы – бағдарламаушының жоғары әрі тиімді жұмысын қамтамасыз ететін күрделі механизм. Ол визуалды түрде бір уақытта

бірнеше ашылған терезелермен жұмыс атқарады. Терезелер экранда толық немесе бөлшектелген түрде бір- бірін жаба отырып жылжи алады, бұл Word тексттік процессор немесе Exsel кестелік процессор ортасының «қатаңдығына» үйренген қолданушының дисккомфорт сезімін оятуы мүмкін.

Delphi-дің бірінші нұсқасы 1994 жылы жарыққа шығып, кейінгі жылдары оның бірте-бірте кеңейтілген 2,3,4,5,6,7 – нұсқалары жарық көрді. 5,6 – нұсқаулардың бір-бірінен айырмашылығы жоқ деуге болады, екеуі де Windows 32 операциялық жүйесінің негізінде дайындалған. Delphi 6-да интернет үшін қосымшаларды дайындау мүмкіндігі кеңейтілген және берілгендер қорымен жұмыс істеуде өзгешеліктер енгізілген.

Delphi-дің негізгі ерекшелігі – онда қосымша құруда компоненттік және объектілік тәсілдер пайдаланылады. Компоненттік тәсілдің мәнісі жеңіл: әр қосымша арнайы іс-әрекеттерді орындайтын компоненттерден жинақталады. Ол жеткіліксіз болса, объектіні өңдеуге арналған үстеме бағдарлама құрылады. Компоненттер визуальды компоненттер кітапханасында (VCL – Visual Component Library) жинақталған. Компоненттер панелінде түрлі кластарға тиісті стандартты компоненттер өте көп. Пайдаланушы жаңа компонент дайындап, оны осы панельге қосуына да болады. Delphi-де программа дайындау, программа мәзірін құру, анимация, мультимедиа үрдістерін ұйымдастыру, OLE технологиясын пайдаланып, басқа офистік қосымшаларды шақыру, олармен жұмыс істеу жене т.б. іс-әрекеттерді орындауға болады. Delphi көптеген есептерді шешуге мүмкіндік береді, оның ішінде:

- Әртүрлі бағыттағы аяқталған қосымшаларды Windows үшін құру;

- Кез келген тілде кез келген қосымшалар үшін профессионалды көрінетін интерфейсті тез құру. Интерфейс Windows- тың барлық талаптарын қанағаттандырады және қолданушы компьютерінде орнатылған жүйеге автоматты түрде ыңғайланады. Өйткені Windows- тың көптеген функцияларын, кітапханаларын қолданады;

- Бағдарламалаудың басқа түрлерінде қолдануға болатын функция, форма, компоненттердің динамикалы біріктірілетін кітапханасын (DDL) құру;

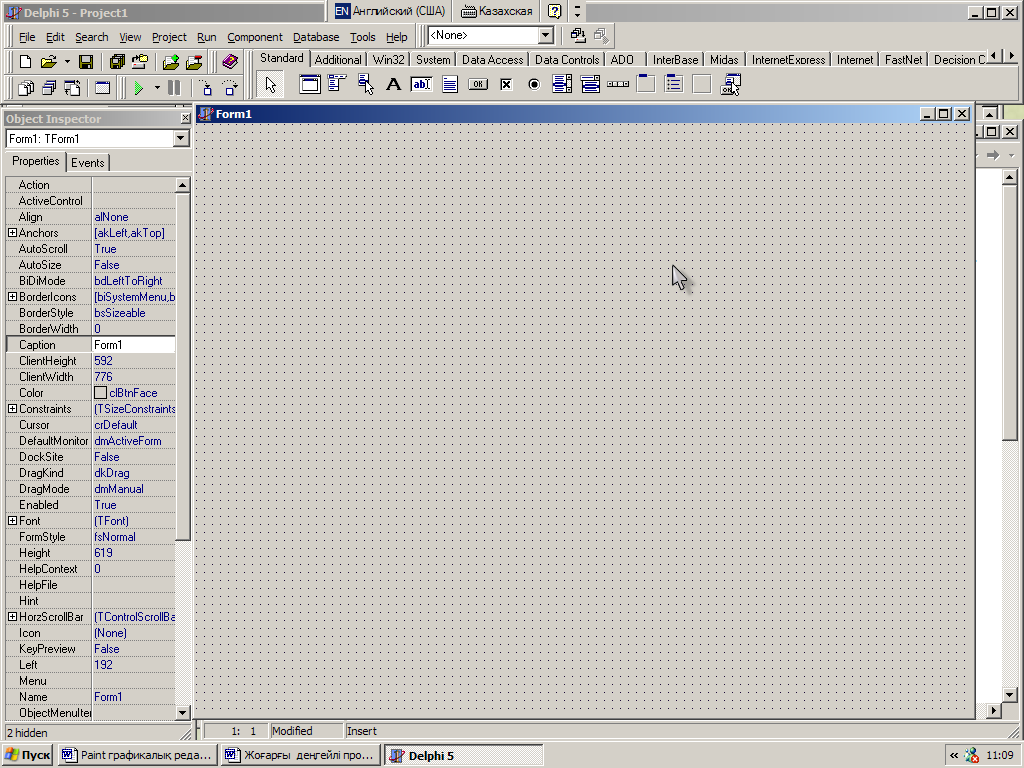
- Кез келген типтегі жергілікті және жойылған мәліметтер базасымен жұмыс істеудің қуатты жүйесін құру;

- Кестелері, графиктері бар күрделі есеп берулерді құру және баспаға шығару;

- Қосымшалардан Windows арқылы жұмыс істеуге болатын өз қосымшалары үшін анықтамалық жүйелерді (hlp файлдары) құру;

- Windows- тың барлық талаптарын және спецификасын ескеретін Windows-тың қосымшалары үшін орнатудың профессионалды бағдарламаларын құру.

Көптеген операторлардың жазылуы Турбо Паскальдағы сияқты. Бірақ, мұнда прогаммалауды үйрену үшін объект, оқиға, қасиет, әдіс, класс ұғымдарымен еркін танысып, компоненттерді пайдалану және түрлі командалардың жазылу түрлерін білу қажет.



1

5

6

2

7

3

4

2-сурет. Delphi ортасы.

1. Delphi 5-тің негізгі терезесі.

2. Форма (Form1).

3. Объект инспекторы терезесі (Object Inspecteor).

4. Модуль терезесі (программалық код терезесі).

5. Негізгі мәзір. 6. Аспаптар панелі.

7. Компоненттер палитрасы.

Delphi-ді Windows арқылы іске қосу командасы:

Іске қосу – Программалар – Borland Delphi 5 – Delphi 5.

Экранда Delphi ортасының үш терезесі көрінеді (2-суретті қара).

Delphi -дің негізгі терезесі құрамына негізгі мәзір (5), аспаптар панелі (6) және компоненттер палитрасы (7) енгізілген (2-суретті қара). Негізгі мәзір пункттеріне Delphi-де жұмыс істеу командалары, аспаптар панеліне мәзір пункттеріне енгізілген негізгі командаларды орындайтын түймелер орналастырылған. Ортада программа құруға арналған төртінші терезе – модуль терезесі (программалық код терезесі) де іске қосылады. Бұл модуль терезесі форма терезесінің астында орналасатындықтан, ол алғашқыда көрінбейді.

Мәзір пункттеріне енгізілген негізгі командаларды

Бұл түймелер – пиктографикалық түймелер деп аталады. Пиктографикалық түймелер бас мәзірдің маңызды опцияларына қол жеткізуге мүмкіндік береді. Пиктографикалық түймелер келесі командаларды жүзеге асырады:

Бағдарламаның алдында жасалынған жобаны ашады (File/Open опциясы);

Файлды ашады (File/OpenFile);

Жобаны дискіде сақтайды (File/Save Project);

Жаңа файлды жобаға қосады (File/SaveFile);

Жобадан файлды алып тастайды (File/AddFile);

Ағымдағы жобаға байланысты модульдер тізімінен модуль таңдайды (View/Units);

Бағдарламаның код терезесі және форма терезесі арасындағы белсенділігін алмастырады (View/Toggle Form/Unit);

Ағымдағы жобоға байланысты формалар тізімінен форма таңдайды (View/Forms);

Жаңа бас форма құрады және оны жобаға қосады (File/New Form);

Бағдарламаны орындайды (Run/Run);

Шақырылатын бағдарламаша жұмысын қадағалауды және бағдарламаның қадамдық із салуын жүзеге асырады (Run/Trace Into); және тағы басқалар.

Delphi-де дайындалатын программа проект (жоба) деп аталады. Форма – программаны дайындау алдында ашылатын программаның сұхбаттық терезесі. Delphi алғашқы рет іске қосылған кезде форма Form1 атауымен (тақырыбымен) көрінеді. Проект құру үшін формаға компоненттер палитрасында орналасқан түрлі компоненттер орнатылады. Кей жағдайларда форма көрінбеуі мүмкін. Оны экранға шығару үшін View Forms командасын беру жеткіліті.

Форманың және формаға енгізілетін компоненттердің түрлі қасиеттері бар. Қасиет (Сипаттама, параметр) – айнымалылардың ерекше түрі. Олар объектінің түрлі мүмкіндіктерін сипаттап, ағымдық күйін анықтайды. Мысалы, форма қасиеттерінің мәндері – форма тақырыбының мәтіні, өлшемі, экранда орналасуы, түсі т.б. Delphi іске қосылған кезде форма қасиеттеріне алғашқы сәйкес мәндер меншіктеліп қойылады.

Программа құру форманы не онда орнатылған компоненттің кейбір қасиеттерінің мәндерін өзгертуден басталады. Объект инспекторы (Object Inspecteor) терезесіне енгізілген қасиеттер тізімін шығару үшін объектіні таңдап алу керек. Инспектор терезесінің жоғары қатарына таңдалған объект (нысан) атауы да жазылып қойылады. Мысалы, Delphi іске қосылған кезде форманың ашылған қасиеттер терезесі 2-суретте көрсетілген. Терезенің екі қосымша беті бар: Properties (Қасиеттер) және Events (Оқиғалар).

Терезе ашылғанда екі бағанаға енгізіген жазулардан оның Properties бөлімі ашылулы тұрады (2-суретті қара). Бірінші бағанда көрінетіндер – қасиет атаулары, екінші бағанға сәйкес жазылғандар – олардың мәндері.

Форманың қасиетттері

*Name (Атау)* – формаға берілген атау. Ол Delphi объектілерінің негізгі қасиеттерінің бірі. Программаның жұмыс істеуі барысында Delphi объектіні осы атау бойынша ажыратып таниды. Delphi-дің формаға автоматты түрде алғашқы рет меншіктеген атауын (Form1) өзгертіп, басқа атау беруге болады. Ол үшін қасиеттер терезесінен Name қасиеті таңдалып, жаңа атау клавиатурадан теріледі.

*Font (Шрифт)* – формаға шығарылатын мәтін шрифтінің қасиеті. Оны таңдап, оң жағында көрінген көп нүкте (…) түймесін шерткен кезде Windows-тың сұхбаттық Қаріпті таңдау терезесі көрінеді. Терезеден, әдеттегідей, қажетті қаріп типін, өлшемін таңдап (мысалы, Time Kaz, 10), OK түймесін шерту керек.

*Caption (Тақырып)* – форма терезесінің тақырыбына енгізілетін мәтін. Алғашқы кезде қасиеттер терезесінің Caption қасиетіне Form1 сөзі енгізіліп қойылады. Оның мәнін өзгерту Name қасиетіне мән меншіктеу сияқты.

*Color(Түс)* – форманың түсін орнату қасиеті. Ол таңдалған кезде оң жағында тілсызық түймесі көрінеді. Тілсызық түймесі – қасиет мәнінің бірнеше екенің белгісі. Тілсызық белгісін шерткен кезде мәндер (түстер) терезесі ашылады. Тізімде көрінген қалаған түс таңдалған соң форма сәйкес түске боялып қойылады.

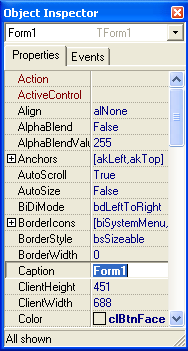
*Width (Ен), Height (Биіктік)* – пиксель өлшем бірлігімен берілген форманың ені мен биіктігін орнату қасиеттері (бұл мәндер форманы қолдан кеңейту не сығу кезінде де автоматты түрде орнатылып қойылады).

Экранда қасиеттер терезесі көрінбесе, оны шығару үшін View-Object Inspector командасын беру жеткілікті. Ол F11 клавиші басылған кезде де көрінеді.

Delphi-де программалар түрлі оқиғалар арқылы басқарылады. Оқиға – программаның жұмыс істеу барысында объект жағдайының белгілі бір әрекетке жауап ретінде өзгеруі. Мысалы, пайдаланушы программа құру үшін алдымен формаға компонент орнатуы, форманы не формада орналастырылған компонентті тышқан арқылы шертуі мүмкін. Оның іс-әрекеті оқиға шақырады.

Delphi-де оқиғаға атау беріліп қойылған. Мысалы, компоненттер палитрасының Button түймесі арқылы формада орнатылған Button1 компонентін шерту Click (Шерту) оқиғасын шақырады.

Әр объектіге байланысты оқиғалар көп. Мысалы, формаға байлансты оқиғалар саны – 35 (3-суретті қара).



Сурет 3. Форма оқиғалары

Олар қасиеттер терезесінің Events қосымша бетіне енгізілген. Терезеде оқиғалар атауының алдына On префиксі (қосымшасы) тіркестіріліп жазылған. Ол – атаудың оқиға екендігін білдіретін белгі. Delphi ортасында жиі кездесетін оқиғалар:

OnClick – тышқан батырмасын бір рет басу;

OnDblClick – тышқан батырмасын екі рет басу;

OnKeyDown – пернені басу;

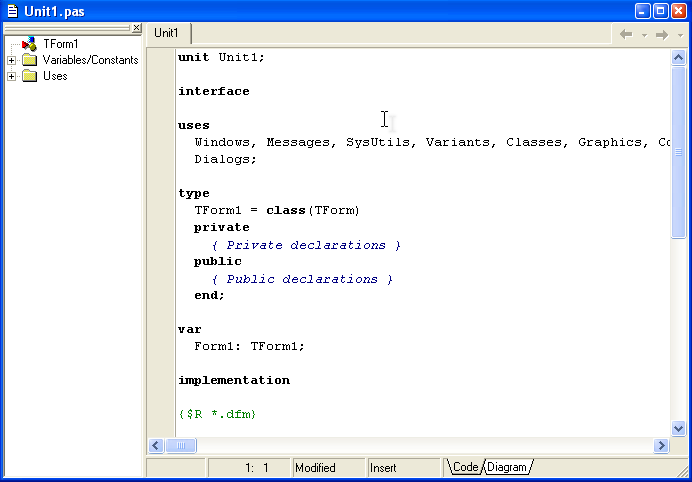
OnCreate – форманы екі рет шерту; т. б.

Проект екі бөлімнен тұрады: автоматты түрде project1 атауы берілген проект файлы (негігі модуль) және unit1, pas атауы берілген модуль. Олар жеке терезелерде орналасқан. Модульге оқиғаларға сәйкес іс-әрекеттерді орындайтын программа мәтіні (процедуралар) енгізіледі. Программа мәтінін программалық код деп, ол енгізілетін терезені программалық код терезесі деп атайды.Формадан код терезесіне өту және код терезесінен формаға өту үшін F12 пернесін басу жеткілікті.

4-суретте форманы екі рет шерткен кезде программмалық код терезесіне енгізілген процедура дайындамасы көрсетілген.

Код терезесі программа мәтінін құру үшін арналған. Бұл мәтін арнайы ережелермен құрастырылады және программа жұмысының алгоритімін сипаттайды. Мәтін жазу ережелерінің жиыны бағдарламалау тілі деп аталады. Delphi жүйесінде Borland корпорациясының Паскальдің кең тараған түрін құрайтын Object Pascal бағдарламалау тілі қолданылады.

Бастапқы кезде код терезесі Windows терезесі ретінде нормальды бос форманың жұмысын қамтамасыз ететін минимальді нәтижелік мәтіннен тұрады. Жобада жұмыс жүргізу кезінде бағдарламалаушы программаға қажетті функционалдықты беру үшін қажетті толықтырулар енгізеді. Жай программа жасау үшін де программалық кодты жасау және өзгерту керек болады.



Сурет 4. Программмалық код терезесі

Оқиғаға байланысты құрылатын процедура оқиғаны өңдеуіш не оқиғаны өңдеу процедурасы делінеді. Терезенің сол бөлігіндегі – браузер терезесі. Онда код терезесінде барлық жарияланулардың құрылымын көріп шығуға болады.

1.2 Python жалпы сипаттамасы

Python – жоғары деңгейлі және әмбебап бағдарламалау тілі

болып табылады. Ол бір мезгілде қарапайым және қуатты объектілі-бағытталған бағдарламалау тілі. Ол, деректер құрылымын қамтамасыз ететін, талғампаздық синтаксисі бар және динамикалық теруді пайдаланады, ол түрлі қосымшалар арқылы бірнеше платформаларында жұмыс істеу үшін арналған тамаша тіл. Python – бүкіл әлем бойынша түрлі мақсаттар -деректер базасын және сөз өңдеу үшін кең таралған әмбебап тіл,ойындарға интерпретатор қосу,және де GUI-ді бағдарламалау және жылдам прототип құру (RAD) үшін арналған тіл. Сонымен қатар Python-INTERNET және WEB-қосымшаларын бағдарламалау үшін пайдаланылады. Python бай стандартты кітапханадан, және модульдердің бай жиынтығынан тұрады. Python мен қосымшалар ең танымал және үлкен фирмалар пайдаланып жазылған, мысалға алып қарайтын болсақ: IBM, Yahoo,

Google.com, Hewlett Packard, Infoseek, НАСА, Red Hat, CBS MarketWatch, Microsoft . Python – бірнеше парадигмалар бағдарламалау тілі, соның ішінде құрылымдық, объекті-бағдарланған, функционалдық, тегеурінді және аспектілі бағытталған. Негізгі архитектуралық ерекшеліктері – динамикалық типтеу, автоматты басқару жады бар, толық сананың тетігі, ерекше жағдайларды өңдеу, ыңғайлы жоғары деңгейлі деректер құрылымы. Python – белсенді дамып келе жатқан программалау тілі, жаңа нұсқасы (қосу/өзгерту тілдік қасиеттерімен) шығады, шамамен екі жарым жыл.

Python-ның айнымалы түрі тек бағдарламаны орындау кезінде анықталады, динамикалық теруді қолдайды. Сондықтан оның орнына «айнымалы тағайындау» шамамен айтуға жақсы «кейбір атымен міндетті күші құндылықтар». Python-ның кірістірілген түрлері: логикалық, жолды, Unicode-жолды, бүтін, еркін-дәлдігін, қалқымалы нүктелі нөмірін, комплекс санды, және басқалар.

Python көптеген бағдарламалау тілдерінің әсерінен, салыстырмалы кеш пайда болды:

* ABC – астыңғы топтастыру үшін операторлар жоғары деңгейлі деректер құрылымы;
* Modula-3 – пакет, модульдер, пайдалану else бірлесіп try және except, аталған дәлелдер функцияларды;
* С, C++ – кейбір синтаксистік конструкциялар;
* Smalltalk – объекті-бағытталған программалау;
* Lisp – жекелеген шегін функционалдық бағдарламалау (lambda, map, reduce, filter және басқа да);
* Fortran – үзіктер массивтер, кешенді арифметика;
* Miranda –тізімдерді білдіру;
* Java – модульдер logging, unittest, threading (бөлігі мүмкіндіктерін түпнұсқалық модулін іске асырылған жоқ), xml.sax стандартты кітапхана бірлесіп пайдалану finally және except өңдеу кезінде қолданылмайтын, пайдалану @ арналған декораторлар;
* Icon – генераторлар.

Модульдің түрін сипаттау мынадай бүтін тұрақты мәндерді анықтайды:

PY\_SOURCE Python тіліндегі бастапқы файл.

PY\_COMPILED Байт компиляторланған файл.

C\_EXTENSION Динамикалық жүктелетін кітапхана.

PY\_RESOURCE Macintosh операциялық жүйесіндегі ресурс.

PKG\_DIRECTORY Пакет каталогы.

C\_BUILTIN Кірістірілген модуль.

PY\_FROZEN Кірістірілген модуль (Python тіліндегі модуль, интерпретаторға қосылған).

Python-ның мүмкіндіктері

Сценарийлер тілі. Сценарий өз бағдарламасын көрсетеді, ол үшін басқару басқа бағдарламасы пайдаланылады . Тілдер сценарий пішімдеу және жылдам өңдеуге пайдаланылады. Сол себепті, олар деректерді бір компоненттен басқа компонентке және бағдарламашыларды осындай күрделі заттардан қорғайды, жадыны басқару секілді операцияларды жақсы орындайды. Бағдарламашылар Python-ды динамикалық программалау тілі деп атайды.

Шегініс үшін топтастыру операторлары. Python-ның бағдарламалау тілі шегініс көмегі кезінде өрнектердің бір топқа тәуелділігін анықтайды. Бұл топ блог коды деп аталады. Бағдарламалаудың басқа тілдерінде тыныс белгілері және сөйлем синтаксисінің басқа белгілері қолданылады. Мысалы, С символ тілінде {командалардың реттілігінің басын білдіреді}. Шегіністің болуы программалаудың басқа тілдерінде жақсы тәжірибе болып табылады. Мәжбүрлі түрде шегінісі сақталатын бағдарламалаудың алғашқы түрлерінің бірі Python болып табылады. Бұл не береді? Ең алдымен шегініс кодты оқу үшін аса ыңғайлы етіп жасалады. Кодтық блоктар басынан аяғына дейін кем белгілерді талап етеді. Ал бұл дегеніміз, кодта аз тыныс белгілері болады, оларды елемеуге болады. Бұның барлығы кодтағы қателердің азаюына әкеледі.

Жоғары деңгейлі деректер типтері. Дербес компьютерлер деректерді нөлдік және бірлікте сақтайды. Алайда, адамдар мәтін секілді ақпаратты сақтаудың неғұрлым күрделі түріне мұқтаж. Егер тіл күрделі деректерді қолдаса, ол жайлы жоғары деңгейі деректер типтерін қолдайды деп айтады. Мұндай типтегі деректерді пайдалану оңай. Мысалы, Python-да жолды біріктіруге болады немесе бөлуге, аударуға болады төменгі регистрға немесе жоғарғы регистрды іздеу және т.б. Сөздіктер және тізімдер секілді деректердің жоғарғы типтері, өзіне басқа да деректерді сақтай алатын мүмкіндіктері бар, функционалдық мүмкіндіктері кең болады.

Кеңейтілуі. Кеңейтілу бағдарламалау тілі, толықтырылуы мүмкін екендігін білдіреді. Кеңейтілген бағдарламалау тілдері өте қуатты болып табылады. Оларды толықтыру көптеген операциялық жүйелерге жарамды етіп жасайды. Кеңейту мүмкін жаңа деректер типтерін қосады, модуль мен модульдерді. Python тіліндегі кеңейту үшін бірнеше тәсіл қарастырылған. Басты бет бағдарламашылар тобы оны жақсарту және өзгерту үшін жұмыс жасайды. Басқа жүздегендері модульдерді нақты мақсаттарды орындау үшін жазады.

Түсіндіру. Түсіндіру тілі тікелей адаммен жазылып орындалады. С++ типі бағдарламалар тілінде құрастырылып жазылған,машина коды тіліне аударылуы тиіс. Әдетте, түсіндіру бағдарламалау тілі баяу жұмыс істейді, өйткені аудармасы лезде жүзеге асырылмайды. Алайда, баптау және жазу өздерінің бағдарламаларын айтарлықтай жылдам жүзеге асырады, өйткені құрастырылған жұмыстың аяқтағанын күту қажет. Түсіндіру бағдарламалау тілдерін оңай ауыстырады түрлі платформаларды.

1.3 Мобильдік қосымшасын жасау үшін арналған фреймворктарға сипаттама

Менде 4 вариант бар:

Kivy, PyObjC, wxWIDGETS, PyQt, PySide.

Kivy

(http://kivy.org/) - даму үшін арналған Python кітапханасы мульти-сенсорлық технологияны қолдайтын мультимедиялық қосымшалар. Kivy белсенді қоғамдастық әзірлейді, BSD стилінде лицензия алады және барлық негізгі платформаларда жұмыс істейді (Linux, OS X, Windows және Android). Kivy Python-да жазылған және құралдар тақтасын терезелермен жұмыс істемейді, SDL2-ден (Simple DirectMedia Layer) тікелей (https://www.libsdl.org/), C деңгейінде жазылған кітапхана төмен деңгейлі кіруді қамтамасыз етеді Пайдаланушы енгізу құрылғыларына және аудио, сондай-ақ көрсетуге рұқсаты бар OpenGL (немесе Windows үшін Direct3D) арқылы 3D кескіндер. Ешқандай, қанша виджет (kivy.uix модулінде орналасқан <https://kivy.org/docs/api-kivy.uix.html>) бірақ Qt және GTK секілді ең танымал баламалар сияқты олардың көпшілігі жоқ. Бизнеске арналған дәстүрлі жұмыс үстелі бағдарламасын дамытатын болсаңыз, QT немесе GTK. Кітапхананы орнату үшін Kivy жүктеу бетіне өтіңіз https://kivy.org/#download, операциялық жүйені табыңыз, ZIP файлын жүктеп алыңыз Python нұсқасы үшін және сіздің операциялық жүйеңізге арналған нұсқауларды орындаңыз. Қосымша код әртүрлі көрсететін он мысалға ие каталогпен келеді API бөліктері.

Qt

Qt (айтылған «кьюте») (http://qt-project.org/) - кросс-платформа графикалық бағдарламалық қамсыздандыруды дамыту үшін кеңінен қолданылады Сондай-ақ, графикалық қосымшаларды әзірлеу үшін де пайдаланылуы мүмкін. Бар- Android үшін Qt5 нұсқасы (http://doc.qt.io/qt-5/android-support.html). Егер сізде бар болса Qt орнатылды (егер Сіз Spyder, Eric IDE, Matplotlib немесе басқа инкубатормен жұмыс істеп жатсаңыз) Qt пайдаланатын жол). Qt екілік файлдарды таратуға мүмкіндік беретін LGPL лицензиясына сәйкес шығарылады. Qt-мен жұмыс істейтін Qt-ны өзгертпесеңіз. Коммерциялық лицензия Бұл деректерді визуализация сияқты құралдарды пайдалану мүмкіндігін береді қолданба. Qt алдын-ала жасалған уақытша платформалармен қамтамасыз етеді түрлі қосымшалар үшін. Qt, PyQt және PySide үшін Python интерфейстері де, өте жақсы құжатталмаған, сондықтан ең жақсы нұсқа - артықшылықты пайдалану Qt-дан C ++-ға құжаттарды (http://doc.qt.io/). Әр интерфейсті қысқаша сипаттаңыз. PyKt Riverbank Computing-дан PySide-ге қарағанда маңызды Qt5 нұсқасы жоқ). Оны орнату үшін құжаттаманы орындаңыз PyQt4 немесе PyQt5 орнату арқылы (<http://pyqt.sourceforge.net/Docs/PyQt4/installation.html>). PyQt4 тек Qt4-де жұмыс істейді, және PyQt4 тек Qt5-мен жұмыс істейді. Riverbank Computing сонымен қатар pyqtdeploy (https://pypi.python.org/

pypi / pyqtdeploy) - бұл PyQt5 ғана жұмыс істейтін графикалық құрал

платформаға тән C ++ кодын жасайды; оны пайдалана аласыз екілік тарату файлдарын жасау.

PySide

PySide (https://wiki.qt.io/PySideDocumentation) Qt болғанда да шыққанNokia, себебі олар компанияны мәжбүрлей алмады Riverside Computing, PyQt-ті құрушылар, GPL-нің өнімдерінің лицензиясын өзгертеді LGPL туралы. Интерфейс PyQt үшін толыққанды ауыстыру болуы керек, бірақ, PyQt дамытуда баяулайды.

wxWidgets

WxWidgets жобасының (https://www.wxwidgets.org/) жобасының философиясы аяқталады бұл бағдарламаға тектік көрініс берудің ең жақсы тәсілі Әр операциялық жүйе үшін жергілікті API параметрін өзгертіңіз. Qt және GTK + енді X11-ді басқа сценарийлерден басқа терезе кітапханаларын қолдануға болады, бірақ Qt- Тікелей және GTK оларға GNOME бағдарламалаудың көрінісін береді. WXWidgets артықшылығы - бұл сіз тікелей өзара әрекеттесу Әрбір платформамен лицензия сізге көбірек мүмкіндік береді. Мәселе, Алайда, әр платформамен әртүрлі тәсілдермен жұмыс істеу керек. Python пайдаланушылары үшін wxWidgets-ті кеңейтетін кеңейтім модулі, wxPython деп аталады (http://wxpython.org/). Ең танымал терезе болғаннан кейін, Python кітапханасы, мүмкін, оның философиясын пайдалану Интерфейспен жұмыс істеудің өзіндік құралдары, бірақ қазір жұмыс жасайды Qt және GTK + -те өте ыңғайлы болды.

PyObjC

Objective-C - Apple компаниясы үшін пайдаланатын арнайы тіл OS X және iOS операциялық жүйелері, сондай-ақ, Cocoa Voka OS X қосымшаларын әзірлеуге арналған. Басқа мүмкіндіктерден айырмашылығы, Objective-C - кросс-платформа емес; ол өнімдерге арналған Apple. PyObjC - OS X және Python үшін Objective-C арасындағы екі жақты көпір, бұл Python тек қана Какаоның құрылымына қол жеткізе алмайды OS X жүйесінде қосымшаларды әзірлеу, сонымен қатар Objective-C бағдарламашыларына қол жетімді болады дейін Python. «Python бағдарламасын орнату» бөлімінде сипатталғандай, Xcode орнату қажет Mac OS X «2-тарауында, себебі PyObjC компилятор қажет. PyObjC жұмыс істейді Тек CPython стандартты үлестірімімен (бірақ мұндай үлестірулермен емес) PyPy немесе Jython) және Python орындалатын файлды пайдалануды ұсынамыз, OS X нұсқасымен қамтамасыз етілген, себебі Python нұсқасының осы нұсқасы өзгертілді Apple компаниясы OS X үшін арнайы конфигурацияланған.

* 1. Мәліметтер базасы

ЭЕМ (электронды есептегіш машиналардың) маңызды ерекшеліктерінің бірі  ақпараттың мол көлемін сақтау және өңдеу, сонымен бірге мәтіндік және графикалық құжаттар (суреттер, сызулар, фотосуретгер, географиялық карталар) ғана емес, жаһандық жүйе, дыбыстық және бейнефайлдардың беттері де жинақталады. Бұл мүмкіндіктер мәліметтер базасының көмегімен іске асады. Мәліметтер базасы (МБ) - белгілі бір арнаулы ауқымға    қатысты мәлтеттерді баяндау, сақтау мен амал-тәсілдермен ұқсата білудің жалпы прииңиптерін қарастыратын белгілі бір ережелер бойынша үйымдастырылған мәлімет-тердің жиынтыгы. Адамның қатысуы ықтимал, автоматтық  құралдармен  өңдеуге  жарамды  түрде ұсынылған ақпарат мәліметтер ретінде түсініледі. Арқаулы ауқым ретінде нақты зерттеуші үшін қызықты нақты әлемнің бөлігі түсініледі.  
Қарапайым МБ мысалы ретінде телефон анықғамалығын, поездар қозғалысы кестесін, кәсіпорын қызметкерлері туралы мәліметгерді, студенттердің сессия тапсыру қорытындылары және т.б. атауға болады.  
Электронды МБ-сының басты қасиеті - ақпаратты тез іздестіру мен сұрыптау (іріктеу), сондай-ақ берілген форма бойынща есепті қарапайым түрлендіру (жасау) мүмкіндігі. Мысалы, сынақ кітапшаларының нөмірлері бойынша студенттердің аты-жөндерін оңай айыруға болады немесе жазушының аты-жөні бойынша шығармаларының тізімін жасауға болады.  
Мәліметтер базасын басқару теориясы дербес пән ретінде шамамен алғанда XX ғасырдың 50-жылдарының басынан бері дами бастады. Осы уақыт ішінде белгілі бір іргелі ұғымдар жүйесі калыптасты. Олардың бірқатарын келтірейік.  
Объект    деп   мағлүматтары   мәліметтер базасында      сақталатын ақпараттың жүйенің элемеиттері аталады.  
Атрибут - объектінің қасиеттерін ақпараттың бейнелеу. Әрбір объект атрибуттардың кейбір жиынтығымен ситатталады.  
Мәлметіпердің басты элементі деп мәлметтердің баска элементтерінің мәндерін аныктауға мүмкіидік беретін осындай атрибут аталады.  
Алғашқы кілт - объектінің (жазбаның) әрбір данасын біреғей түрде жағастыратын атрибут (немесе атрибуттар тобы), Екіншіреттегі кілт - бірнеше жазбалар (объект даналары) үшін мәні қайталанатын атрибут (немесе атрибуттар то¬бы). Ен алдымен, екінгиі реттегі кілттер жазбаларды іздестіру операцияларында қолданылады. Мәліметгер базасы ұғымымен мәліметтер базасын басқару жүйесінің ұғымы тығыз (МББЖ) байланысты.  
МББЖ - жаңа база құрылымын құруға арналған программаның құралдар кешені, оны құрамын толықтыру, құрамында барды редакқиялау және апаратпен көрсетушілік (визуализация). База ақпаратын көрсетушілік ретінде берілген критерийге сәйкес бейнеленетін мәліметгерді іріктеу, оларды ретгеу, сосын шығаратын құрылғыға беру немесе байланыс арналары бойынша беру.  
Кез келген МББЖ мәліметтермен төрт қарапайым операция орындауға мүмкіндік береді:  
■ кестеге бір немесе бірнеше жазбаны қосу;  
■ кестеден бір немесе бірнеше жазбаны жою;  
■ кейбір өрістердің бір немесе бірнеше жазбаларындағы мәндерді жаңарту;  
■ берілген шартты қанағаттандыратын бір немесе бірнеше жазбаларды табу.  
МББЖ-ның тағы бір функңиясы - мәліметтерді басқару. Мәліметтерді басқару ретінде, әдетте, мәліметтерді рұқсат етілмей қол жетуден қорғау, мәліметтермен жұмыс режимін көп мәрте пайдалануды қолдау және мәліметтердің тұтастығы мен үйлесімділігін қамтамасыз ету түсініледі.  
Мәліметтер базасын жіктеу. Мәліметтер арасындағы байланыстарды орнату әдістері бойынша былайша ажыратылады: Реляциялық, иерархиялық және желілік МБ. Реляциялық МБ кесте түрінде мәліметтерді ұсынудың қарапайым және әдеттегі формасы болып саналады. Көптеген теориялар ішінде кестеге қатынас (relation) термині сәйкес келеді және ол МБ-ға осы атты берген. Ол үшін дамыған математикалық аппарат - біріктіру, алу, азайту, қиылысу, қосу және т.б. сияқты математикалық операциялар айқындалатын реляциялық есептеу мен реляциялық алгебра.  
Осы типтегі МБ жасауға американдық ғалым Е. Кодд елеулі үлес қосты.  
Реляциялық МБ-ның жақсы қасиеті - оны қолдаудың аспаптық құралдарының салыстырмалы қарапайымдылығы, кемшілігі - мәліметтер құрылымының қатавдығы (ерікті үзындық кестесінің жолдарын берудің мүмкін еместігі) және оның жұмысы жылдамдығының мәліметтер базасы мөлшеріне тәуелділігі. Мұндай МБ-ға белгіленген көптеген операциялар үшін барлық МБ қарау қажетті болуы мүмкін,  
Иерархиялық және желілік МБ қайсыбір ортақ белгісі бар мәліметтер арасындағы байланыстардың бар болуын көздейді. Иерархиялық МБ-да мұндай байланыстар ағаш-сызық жол (графа) түрінде бейнеленуі мүмкін, мүнда биік үшар бастан төменгі кішілеріне қарай біржақты байланыстар ғана болуы ықтимал. Егер барлық ықтимал сүрақтар ағаш құрылымында бейнеленсе ғана, бұл қажетті ақпаратқа жетуді тездетеді. Ақпарат алудың ешқандай басқа сұрау салулары қанағаттан-дырмайды,  
Аты айтылған кемшілік желілік МБ-дан алып тасталды, онда (тым болмағанда, теориялық жағынан) «бәрімен барлық жағынан» байланыс болуы ықтимал еді. Іс-тәжірибе жүзінде мүны іске асыру мүмкін болмағандыктан, кейбір шектеулерге жүгінуге тура келеді.  
Мәліметтердің әрбір элементі құрамында кейбір басқа элементтердің сілтемесі болуы тиіс. Сол себептен ЭЕМ (электронды есептегіш машинаның) оперативті және дискілік жадындағы едәуір ресурстар талап етіледі.  
Кейбір МББЖ туралы мәліметтер кестеде келтірілген.

MS Accеss

MS Accеss ең кең тараған деректер қоймасын басқару жүйелерінің (ДҚБЖ) бірі болып табылады.

Қазіргі кезде деректер қоймасы (ДҚ) және қоймасын басқару жүйесі (ДҚБЖ) терминдері тек компьютерлерге қатысты қолданылады. Жалпы мағынасында деректер қоймасы терминін белгілі бір нышандарына байланысты біріккен ақпараттардың кез келген жиынына қолдануға болады. Мысалы, деректер қоймасы ретінде поезд қозғалысының кестесін немесе сатып алушылардың тапсырысы және тапсырыстың орындалуы туралы мәліметтер тіркелген кітапты қарастыруға болады. Мұнда деректер қоймасы ретінде анық біріккен мәліметтердің жиыны қарастырылады.

Көптеген ДҚ-да алғашқы деректерді жинақтау кестенің көмегімен іске асырылады, оның жолы – қойманың бір объектісі жайлы толық ақпаратты, ал бағаны – объектінің бір сипаттамасын білдіреді. Кестенің жолы (бір объекті туралы деректер**) жазу** деп, кестенің объектілері сақталған белгілі бір сипаттамалардан тұратын баған **өріс** деп аталады.

ДҚ құруда тиімді жұмыс үшін оның құрылымын әрбір өрісте мазмұны тек бір типтегі деректер болатындай етіп ойлау табу керек, мысалы:

●**өрістің аты: Тегі**

мазмұны : Абдуллаев

●**өрістің аты: Телефон**

мазмұны: 25-89-64

●**өрістің аты: Айы**

мазмұны: сәуір

Соңғы уақытта деректер қоймасының ішіндегі ең көп тарағаны – реляциялық деректер қоймасы. Реляциялық деректер қоймасында ақпарат бір немесе бірнеше кестелерде сақталады. Кестелер арасындағы байланыс бір не бірнеше сәйкес өрістер арқылы жүзеге асырылады.

Sqlite3

«Ендірілген» сөзі SQLite клиент-сервер парадигмасын пайдаланбайды, яғни SQLite қозғалтқышы бағдарлама өзара әрекеттеседі, бірақ бағдарлама байланыстырылған кітапхана және қозғалтқыш бағдарламаның ажырамас бөлігі болып табылады. Осылайша, SQLite кітапханасының функционалдық шақырулары (API) байланыс хаттамасы ретінде пайдаланылады. Мұндай тәсіл шығынды азайтады, жауап беру уақыты мен бағдарламаны жеңілдетеді.

SQLite барлық дерекқорды (анықтамаларды, кестелерді, индекстерді және деректерді қоса) бағдарламаны іске қосатын компьютердегі бір стандартты файлда сақтайды. Іске асырудың қарапайымдылығы жазбаша мәмілені жасамас бұрын, дерекқорды сақтайтын барлық файл бұғатталғандығына байланысты; Сондай-ақ, ACID функциялары журнал файлын жасау арқылы қол жеткізіледі.

Бірнеше үрдістер немесе ақаулар бір уақытта бір дерекқордан ешқандай ақаулықсыз деректерді оқи алады. Дерекқорға жазылу тек басқа сұрауларға қызмет көрсетілмесе ғана орындалуы мүмкін; әйтпесе, жазу әрекеті сәтсіз аяқталады және қате коды бағдарламаға қайтарылады. Оқиғалардың дамуының тағы бір нұсқасы - белгілі бір уақыт аралығында жазу әрекеттерінің автоматты түрде қайталануы.

Жеткізу пакетінде негізгі кітапхананың функцияларын іске асыруды көрсететін орындалатын файл sqlite3 түрінде функционалдық клиент бөлігі бар. Клиент бөлігі - кросс-платформа командалық жолының утилитасы.

Қозғалтқыштың архитектурасы арқасында SQLite-ді ендірілген жүйелерде және гигабайттық деректер массивтері бар арнайы машиналарда пайдалануға болады.

SQLite деректерді динамикалық теруді қолдайды. Мүмкін өріс түрлері:

INTEGER, REAL, TEXT, BLOB.

SQLite кітапхана өзі C жазылған; тіркесімдердің үлкен саны Delphi, C ++, Java, C #, соның ішінде басқа да программалау тілдері, үшін бар, VB.NET, Python, Perl, Node.js, PHP, PureBasic, Tcl (құралдар Tcl жұмыс істеу үшін жеткізу енгізілген SQLite), Ruby, Haskell, Scheme, Smalltalk, Lua және Parser, сондай-ақ басқа да көптеген. Жоба бетте орналастырылған қолданыстағы қаражатты толық тізімі

Қарапайымдылық пен SQLite салу жеңілдігі кітапхана-шолғышта, музыка ойнатқыш және басқа да көптеген бағдарламалар пайдаланылады фактісі әкелді.

Атап айтқанда, SQLite пайдаланылады:

Adobe Біріктірілген Runtime - қосымшалар (ішінара) іске қосу үшін қоршаған ортаны;

Gears;

autoit;

Негіздемелік Qt;

қозғалтқыштың XUL платформа, ықтимал Gecko 1.9+, XULRunner 1.9+ және, осы платформасында негізделген барлық бағдарламалар, оның ішінде:

Mozilla Firefox (3.0 жылдан бастап)

(3.0 жылдан бастап) Mozilla Thunderbird

Songbird

SQLite менеджері

Skype

Viber;

GPS-навигаторлар Garmin кейбір үлгілері

: Көптеген бағдарламалар, соның ішінде (әсіресе Mac OS және Түйініне, Android үшін) сақтау пішімі ретінде SQLite қолдау

1С: (сыртқы компонент арқылы ) Enterprise 7.7;

1С: (журнал жазбаларының сақтау үшін) Enterprise 8.3;

Adobe Photoshop Lightroom;

FlylinkDC ++;

AIMP дәйексөз;

Banshee;

Caliber;

Eserv;

F-Spot;

(3.0 жылдан бері) FAR менеджер;

gajim;

Google Chrome;

Miranda IM (ашасы дерекқор драйвері арқылы);

(Нұсқасы 10.50 бастап) Опера;

qutIM;

Safari;

XnView;

Garena.

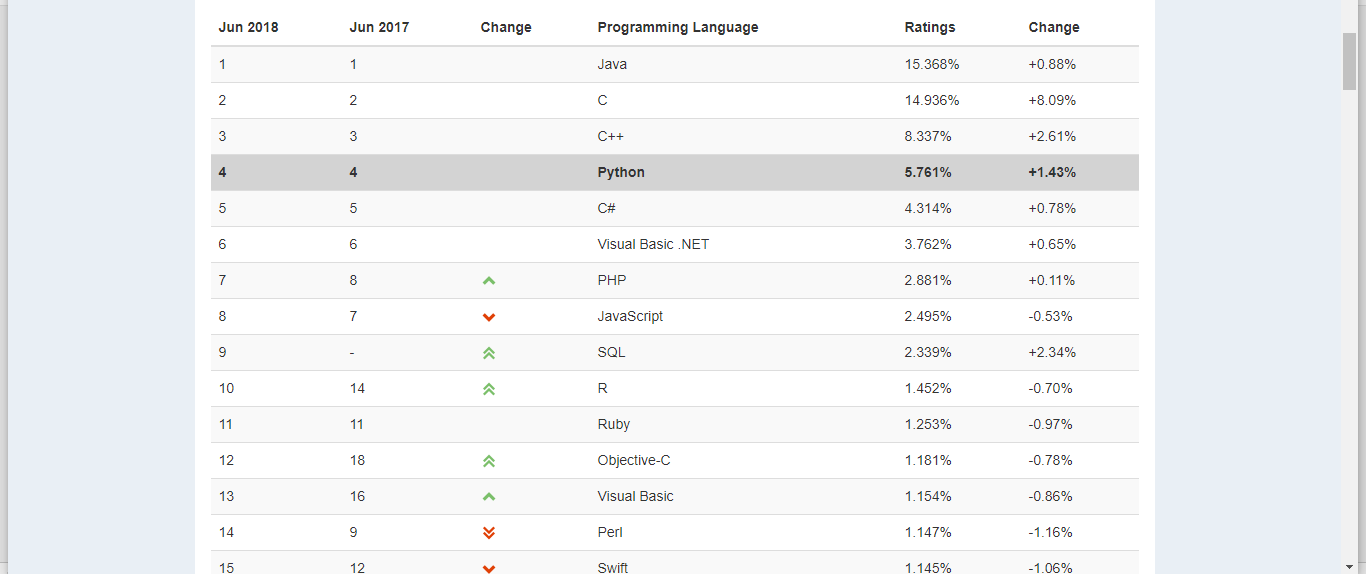
II Ұйымдастырушылық-технологиялық бөлім

2.1 Мобильдік қосымшаны жасау үшін бағдарламалау тілін таңдау

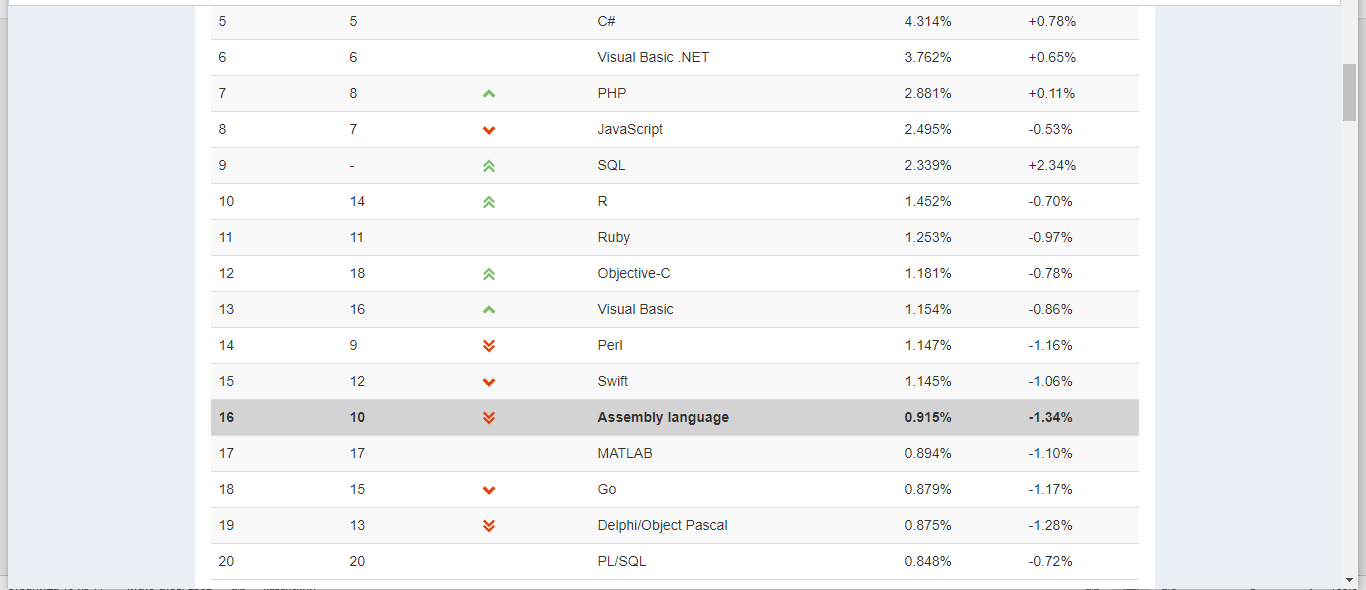
2 тілде және олардың қолданатын орталарыныңда өзіндік ерекшеліктері бар. Python мен Delphi ортасы қолданатын Object Pascal бір – бірімен өте ұқсас. Екеуіде бъектілі-бағытталған бағдарламалау тілі, екеуіде динамикалық теруді пайдаланады. Айырмашылық олардың машина тіліне транляциясында. Мәселен Python интерпретатор арқылы жұмыс істейді. Ол өте ыңғайлы себебі компиляцияны күтпейсін. Көбіне үлкен жобаларда олардың техникалық қиындықтарына қарай бағдарламаларда үлкен болады. Қолданатын ресурстары да көбейеді. Солай компиляция ұзақ уақыт алып кетуі де мүмкін. Интерпретатор Python тіліндегі кодты оқып оны сол машина тіліндегі аналог биттарға айналдырып дереу нәтижесін көрсете алады.

Delphi кодты компиляциядан өткізеді. Бірақ қандайын таңдасанда бұл бағдарламаны жасап шығаруға болады. Себебі Delphi-дің жаңа жаңартулары бұған мүмкіншілік береді. Бұларды салыстыра отырып мен Python тілін таңдадым. Себебі ол өте түсінікті, синтаксисі қатал: ; {} белгілерді талап етпейді. Бұл Мен сияқты үйреніп келе жатқан бағдаламашыларға өте тиімді. Және де Python-ның бағдарламашылар қоғамы өте үлкен.

Ал TIOBE рейтингы бойынша ол 4 орында.



Delphi/Object Pascal 19 орында.



Python қарапайымдылығымен және сұраныс жағына көптеу екенің көрсетті.

Себебі бағдарламаны жасауда қиындықтар туғанда керек Google көмектесе алмай қалуыда мүмкін.

2.2 Мобильдік қосымшасын жасау үшін фреймворк таңдау

5 фреймворкта бір-бірінен өзіндік ерекшеліктері бар. Бірақ мен kivy фреймворкын таңдым. Себебі басқаларына қарағанда ол ауыр емес. Өте жылдам. Және менің жасайтын бағдарламама тура келеді.

2.3 Мәліметтер базасын таңдау

Мен бұл екі таңдауда Sqlite3 таңдадым. Себебі мен жасайтын бағдарлама Linux, Windows, MacOSx, Android, IOS сияқты платформаларға көшірсемде ыңғайлы.