**МАЗМҰНЫ**

[Кіріспе 2](#_Toc163761512)

[1 Балаларға арналған қолданыстағы шешімдерге шолу 3](#_Toc163761513)

[2 Бала оқытудағы педагогикалық технологиялар 7](#_Toc163761514)

[3 Аналогтарды іздеу және салыстыру 9](#_Toc163761515)

[4 Жобалық бөлім 11](#_Toc163761516)

[4.1 Бағдарламалау тілін таңдаудың негіздемесі 11](#_Toc163761517)

[4.2 .NET Framework 13](#_Toc163761518)

[4.3 Windows Forms бағдарламалау интерфейсі 16](#_Toc163761519)

[5 Қолданбаны іске асыру 18](#_Toc163761520)

[5.1 Бағдарламаның құрылымдық көрінісі 18](#_Toc163761521)

[5.2 UML қолдану жағдайларының диаграммаларын құру 18](#_Toc163761522)

[5.3 Visual Studio: шолу және салыстыру 18](#_Toc163761523)

[5.4 Microsoft Access: шолу және салыстыру 20](#_Toc163761524)

[5.5 Бағдарламалық өнім дизайны 21](#_Toc163761525)

[5.6 Бағдарламаның негізгі функциялары: 22](#_Toc163761526)

[Қорытынды 26](#_Toc163761527)

Кіріспе

Қазіргі ақпараттық қоғамда балалар олардың дамуы мен оқуында маңызды рөл атқаратын сандық технологиялардың орасан зор көлеміне ұшырайды. Бұл тек мүмкіндіктерді ғана емес, сонымен қатар олар пайдаланатын бағдарламалық қамтамасыз етудің қауіпсіздігі мен тиімділігіне қойылатын арнайы талаптарды білдіреді. Осыған байланысты балалардың қажеттіліктерін қанағаттандырып қана қоймай, сонымен қатар олардың жас ерекшеліктері мен қауіпсіздік талаптарын ескеретін мамандандырылған бағдарламалық өнімдерді әзірлеу қажеттілігі артып отыр.

Жоба дамуы тежелген балаларды қолдауға бағытталған әртүрлі диагностикалық, оңалту және оқыту құралдарын талдауға бағытталған. Сондай-ақ бұл құралдардың тиімділігін және олардың оқу тәжірибесінде қолданылады. Жобаның негізгі мақсаты – мамандар мен ата-аналарға дамуында дамуы тежелген балалармен олардың өмір сүру сапасын жақсарту және қоғамға интеграциялау үшін жұмыс істеуге көмектесетін әдістер мен құралдар туралы ақпарат беру.

Талдау барысында бағдарламалық жасақтаманың әртүрлі аспектілерін, соның ішінде оның функционалдығын, қол жетімділігін және әртүрлі пайдалану жағдайларында қолданылуы қарастырылады. Талдау нәтижелері мамандар мен ата-аналарға дамуында ауытқуы бар балаларға арналған бағдарламалық жасақтаманы таңдау туралы негізделген шешім қабылдауға көмектеседі. Сонымен қатар, жоба осы саладағы әрі қарайғы зерттеулер мен әзірлемелерді ынталандырады, сайып келгенде, осал балаларды қолдау мен күтудің тиімділігін арттырады.

Соңғы онжылдықтарда білім беру ортасы технологиялық прогрестің әсерінен елеулі өзгерістерге ұшырады. Сандық технологиялардың күнделікті өмірде маңыздылығы артып келе жатқандықтан, олардың білім берудегі рөлі бірдей жылдам қарқынмен өсуде. Оқу процесіне компьютерлерді, интерактивті тақтаны, онлайн оқытуды және басқа да көптеген технологияларды енгізу қазіргі білім беру парадигмасының құрамдас бөлігіне айналды.

Дегенмен, цифрлық технологиялар рөлінің өсуіне байланысты жаңа міндеттер туындайды. Атап айтқанда, білім беруде технологияны тиімді пайдалану тек сәйкес бағдарламалық жасақтаманы жасауды ғана емес, сонымен қатар оны әр түрлі оқушылар топтарына қолдануға назар аударуды талап ететіні белгілі бола бастады. Дамуында кемістігі бар балалар үшін, соның ішінде психо-сөйлеу дамуының кешігуі үшін бұл өте маңызды.

Дамуында ауытқулары бар балаларға қолдау көрсету үшін арнайы әзірленген бағдарламалық қамтамасыз ету олардың оқуы мен дамуында маңызды рөл атқарады. Ол қарым-қатынас дағдыларын жақсартуға, моторикасын дамытуға, когнитивтік функцияларды ынталандыруға және т.б. бағытталған болуы мүмкін. Сонымен қатар, мұндай бағдарламалық қамтамасыз ету дамуында ауытқуы бар балалардың ақпаратты қабылдау және қабылдау ерекшеліктеріне бейімделуі керек.

Дамуында кемістігі бар балаларды қолдау үшін бағдарламалық қамтамасыз етуді сәтті пайдалану өнімнің жоғары функционалдығы мен қолжетімділігін ғана емес, сонымен қатар мұғалімдер мен мамандардың оны әрбір баланың нақты қажеттіліктеріне бейімдей білуін талап ететінін атап өткен жөн.

Жобаның нәтижелері ата-аналарға, тәрбиешілерге және мамандарға дамуында ауытқуы бар балаларға арналған бағдарламалық жасақтаманы таңдауда саналы шешім қабылдауға көмектеседі деп күтілуде.

Балаларға арналған қолданыстағы шешімдерге шолу

Компьютерлік оқыту жүйелері білім беруге қатысты құралдардың елеулі класы ақпараттық технологиялар және педагогикалық шешу үшін қолданылады тапсырмалар. Қазіргі таңда олар оқу үдерісіне өз үлестерін қосуда дәстүрлі оқыту құралдарымен тең дәрежеде. Дегенмен, салыстырғанда компьютерлік оқыту жүйесін оқытудың дәстүрлі құралдары бірегей мүмкіндіктерді және көптеген қолданыстағы функцияларды қамтамасыз етеді жоғары сапалы орындалады. Компьютерлік оқыту жүйелері бағдарламалық модульдер болып табылады оқу процесінде белсенді пайдалануға арналған. Бұл жүйелер когнитивті белсендіруге арналған оқушының іс-әрекеті, оны дамыту және жетілдіру кәсіби білім мен дағды.

Компьютердің пайда болу тарихы оқыту жүйелерін өнеркәсіптік өндірістің басынан бастауға болады бірінші буын компьютерлері және олардың оқудағы пайда болуы институттар мен ұйымдар. Осылайша, жаңа тәсілдер пайда болуда оқыту, атап айтқанда компьютерлік оқыту технологияларын пайдалану. Бірінші оқыту жүйесі 1950 жылдардың аяғында АҚШ-та жасалған. және киді есімі Платон. Платон жүйесі дамыды және жетілдірілді 20 жыл бойы Control Data Corporation қабырғасында, ол компьютерлер мен әртүрлі автоматтандырылған жүйелерді шығарумен айналысады. 1980 жылдардың басы жаппай жасаумен және пайдалануымен ерекшеленді ұқсас оқыту жүйелері, өйткені дербес компьютерлер кең таралған халық арасында тарады. Содан бері компьютерлер үлкен бола бастады математикалық мақсатта емес, білім беру мақсатында пайдаланылады есептеулер. Қазіргі компьютерлердің мультимедиялық мүмкіндіктері оларды жасайды оқыту құралдары ретінде өзіндік ерекшелігімен ерекше. Жеке компьютер әр түрлі пәндерді оқытуда қолданылады және көптеген жаңа ақпараттарды құруға негіз болады оқыту технологиялары.

Дербес компьютердің негізгі айырмашылығы басқа да техникалық оқыту құралдары компьютердің мүмкіндігі болып табылады бір уақытта келесі функцияларды біріктіреді:

- интерактивті жұмыс режимі - адамның әрбір әрекеті компьютерлік жауаппен сүйемелдеуімен;

- тамаша графикалық және иллюстрациялық мүмкіндіктер, бұл мүмкіндік береді оқуды анық және көрнекі түрде тартымды ету;

- компьютердің «тұлғасы» – шағын өлшемдер және салыстырмалы төмен баға бүкіл сыныпты компьютермен қамтамасыз етуге мүмкіндік береді немесе сынып;

- жұмыс істеудің қарапайымдылығы және интуитивті интерфейс;

Егер компьютер оқыту құралы ретінде пайдаланылса, онда ол техникалық мүмкіндіктер мүмкіндік береді:

- материалды берудің анықтығын арттыру;

- оқытуды жекелендіру;

- әр студентке жеке қарқынмен жұмыс жасау;

- оқу процесін белсендіру;

- басты назарды теориялық білімнен практикалық қолдануға аудару пайдалану (және керісінше).

Сонымен қатар, компьютерлік дисплейлердің графикалық мүмкіндіктері компьютерлік оқытуды көрнекі тартымды ету және тиімдірек. Дисплей экранында әртүрлі пішіндерді көрсетуге болады және дизайн, нақты және ойдан шығарылған объектілердің кескіндері, фотосуреттер, сызбалар, gif және 3D кескіндері және т.б. Мұның бәрі болуы мүмкін статикалық және қозғалыста орындалады. Компьютерлік графика мүмкін көрінбейтін процестер мен құбылыстарды көрінетін етіп жасау шындық, іс жүзінде жоқ нәрсенің визуалды бейнесін жасай аласыз сыртқы түрі жоқ (мысалы, дерексіз ұғымдарды, принциптерді бейнелеу салыстырмалылық теориясы және т.б.). Компьютерлердің бұл қабілеті негізделген когнитивтік компьютерлік графика деп аталады – арнайы бағыт мүмкіндігінше ғылыми зерттеулерде компьютерлерді пайдалану үлгілерді зерттеу үшін компьютерлер қолданылады.

Қазіргі заманғы нарықта әртүрлі жастағы балаларға арналған көптеген бағдарламалық өнімдер бар. Бұл өнімдер білім беру қолданбаларынан ойын платформаларына дейін. Төменде кейбір танымал шешімдерге шолу, олардың артықшылықтары мен кемшіліктері берілген.

**Оқыту бағдарламаларының түрлері**

Оқыту бойынша компьютерлік бағдарламаларды төрт түрге бөлуге болады:

1) модельдеу және модельдеу;

2) оқыту және бақылау;

3) ақпараттық;

4) танымдық ойындар.

Оқыту бағдарламаларының негізгі мақсаты шоғырландыру және алған білімдерін одан әрі жетілдіру және қалыптастыру дағдылар, ал бақылаушылар – осы жаңа дағдыларды бақылау. Ақпараттық бағдарламалар қамтамасыз етуге бағытталған жаңа дидактикалық материалды меңгерту. Түрлі тапсырмалар мен сұрақтар бұл бағдарламаларда оқыту барысын басқару үшін пайдаланылады және адам мен машина диалогын ұйымдастыру. Осылайша, егер студент берсе қате жауаптар болса, бағдарлама «артқа» оралып, көрсете алады қайталап оқуға арналған оқу материалы. Модельдеу бағдарламаларының негізі видеографиялық және мүмкіндік беретін компьютердің есептеу мүмкіндіктері компьютерлік эксперимент. Сонымен қатар, бұл бағдарламалар мүмкіндік береді дисплей экранында белгілі бір процесті бақылап, оның барысына әсер ету; пернетақтаны пайдалану.

Оқыту ойындары кейбір қиял түрінде беріледі компьютерлік технологияны қолдану арқылы жүзеге асырылатын орта және тікелей компьютердің өзінде бар. Қолданылуы мүмкіндіктерді іске асыру үшін бағдарламада көзделген қаражат, ойын мен шынайы әлемді зерттеумен тікелей байланысты салыстыру заңдылықтарды тәуелсіз ашуға әкеледі, шындық объектілері арасындағы қарым-қатынас, танымдық дамуы қабілеттері мен оқушы тұлғасын қалыптастыру. Компьютерлік оқыту жүйелерінің кеңінен қолданылуы Оқу-тәрбие процесі кейбір қиындықтармен қатар жүреді әзірлеуге жұмсалған уақытпен, осы әзірлемелердің күрделілігімен байланысты және сонымен қатар қазіргі заманғы барлық мүмкіндіктерді шектеулі пайдаланумен компьютерлер.

Мұның бірнеше себептері бар:

1) мұндай бағдарламаларды білім беру жүйесіне енгізуге қызығушылықтың төмендігі процесс;

2) жобалауға қабілетті мамандардың жетіспеушілігі, содан кейін және жоғары деңгейлі компьютерлік оқыту бағдарламасын әзірлеу;

3) мұндай әзірлемелерді мемлекеттік қаржыландырудың болмауы.

Осы кемшіліктерге қарамастан, компьютер тиімді жаңа мүмкіндіктерді табуға мүмкіндік беретін оқу құралы білім:

- оқушылардың танымдық процестерін белсендіруге мүмкіндік береді және оларды белсенді жұмысқа тарту;

- оқуға деген ынтасын арттырады;

- байланысты ақпаратты ұсыну мүмкіндіктерін арттырады айқындылық пен иллюстрациялық;

- белсенділікті бақылау сапасын өзгертуге және жақсартуға мүмкіндік береді студент;

- оқушылардың өз бетінше ойлау қабілеттерін дамытады әрекеттер.

**Білім беру қолданбалары:**

Артықшылықтары:

- Балаларға оқу, жазу, математика және жаратылыстану ғылымдары сияқты әртүрлі дағдыларды меңгеруге көмектесіңіз.

- Интерактивті формат балаларды қызықтырады және тиімдірек оқуға ықпал етеді.

Кемшіліктер:

Кейбір қолданбалар балалар үшін тым біркелкі немесе қызықсыз болуы мүмкін, бұл оқуға деген қызығушылықты жоғалтуы мүмкін.

- Барлық білім беру қолданбалары балалардың жас ерекшеліктеріне және қауіпсіздік талаптарына бейімделмеген.

**Ойын платформалары:**

Артықшылықтары:

- Ойын және өзара әрекеттесу арқылы балалардың шығармашылық және танымдық қабілеттерін дамытуға мүмкіндік беру.

- Көңіл көтеру және танымдық ойындардың кең ауқымын ұсына алады.

Кемшіліктер:

- Кейбір ойын платформаларында жасына немесе қауіпсіздік себептеріне сәйкес келмейтін мазмұн болуы мүмкін.

- Мазмұн теңдестірілмеген болса, геймплейге тәуелділіктің мүмкін проблемалары.

**Жаңа бағдарламалық өнімге қойылатын негізгі талаптардың анықтамасы:**

Балаларға арналған және талаптарын талдау негізінде жаңа бағдарламалық өнімге келесі негізгі талаптарды анықтауға болады:

- **Қауіпсіздік**: Бағдарламалық жасақтама балалардың пайдалануы үшін мүлдем қауіпсіз, құрамында зиянды мазмұн болмауы және пайдаланушы деректерінің қорғалуын қамтамасыз етуі керек.

- **Оқыту және дамыту:** Өнім қызықты және танымдық тапсырмаларды ұсына отырып, балалардың оқуы мен дамуын ынталандыруы керек, сонымен қатар мақсатты аудиторияның жас ерекшеліктерін ескеруі керек.

- **Интерактивтілік**: Бағдарламалық жасақтаманың интерактивті және көңілді болуы маңызды, бұл балалардың назарын аударуға және олардың қызығушылығын сақтауға көмектеседі.

- **Пайдаланушы интерфейсі**: Интерфейс әртүрлі жас топтарындағы балалар үшін интуитивті және пайдалану оңай болуы керек.

Осы талаптарды ескере отырып, жаңа бағдарламалық қамтамасыз ету қауіпсіздікті, оқу тиімділігін арттыруға арналған.

Бала оқытудағы педагогикалық технологиялар

Балаларды оқытудың педагогикалық технологиялары – бұл білім мен дағдыны барынша тиімді және қызықты меңгеру мақсатында педагогтардың оқу-тәрбие процесін ұйымдастыру және өткізу үшін қолданатын әдістері, тәсілдері және құралдары. Балаларды оқытуда кеңінен қолданылатын бірнеше негізгі педагогикалық технологиялар:

1. Интерактивті оқыту: Бұл технология білім беру үдерісіне оқушылардың белсенді қатысуын көздейді. Ол интерактивті тақтаны, компьютерлік бағдарламаларды, ойындарды және студенттерді өз бетінше әрекет етуге, зерттеуге және білім алмасуға ынталандыратын басқа құралдарды пайдалануды қамтиды.

2. Саралап оқыту: Бұл тәсіл әр оқушының жеке қажеттіліктерін, қабілеттерін және оқу мәнерлерін есепке алуды қамтиды. Мұғалім максималды оқуды қамтамасыз ету үшін оқу материалын, әдістерін және оқу қарқынын әр оқушының ерекшеліктеріне сәйкес келтіруге тырысады.

3. Проблемалық оқыту: Бұл әдіс нақты немесе ойдан шығарылған есептерді шешуге негізделген, оқушылардың сыни ойлауын, аналитикалық және есептерді шешу дағдыларын дамытуға мүмкіндік береді. Студенттер шешімдерді табуға белсенді қатысады, ал мұғалім тәлімгер және көмекші болып табылады.

4. Бірлескен оқыту: Бұл тәсіл оқушылардың ортақ оқу мақсаттарына жету үшін топта бірлесіп жұмыс істеуін және өзара әрекеттесуін қамтиды. Студенттер командада жұмыс істейді, идеялармен алмасады, проблемаларды шешеді және бірлескен білім алады, бұл коммуникация мен топтық жұмыс дағдыларын дамытуға көмектеседі.

5. Ақпараттық-коммуникациялық технологияларды (АКТ) пайдалану: Бұл әдіс оқу процесінде компьютерлерді, интернетті, бағдарламалық қамтамасыз етуді және басқа да заманауи технологияларды қолдануды көздейді. Бұл ақпаратқа қолжетімділікті кеңейтуге және оқу үдерісін студенттер үшін қызықты әрі ынталандыруға мүмкіндік береді.

6. Ойынға негізделген әдістер: Бұл тәсіл балаларды оқыту үшін ойындарды, ойын жағдаяттарын және танымдық ойындарды пайдалануға негізделген. Ойын әдістері оқушылардың белсенділіктерін арттыруға, қиялдарын дамытуға, шығармашылық ойлауға, оқуға деген ынтасын арттыруға көмектеседі.

7. Рефлексиялық оқыту: Бұл әдіс студенттердің өз білімдері, дағдылары мен дағдылары туралы өзіндік рефлексиясын ынталандыруды қамтиды. Студенттер өздерінің жетістіктері мен сәтсіздіктері туралы ойлайды, сонымен қатар олардың оқудағы жетістіктерін жүзеге асырады, бұл өзін-өзі дамытуға және өзін-өзі ұйымдастыруға ықпал етеді.

Бұл педагогикалық технологиялардың барлығын оқыту мақсатына, мұғалімнің қалауы мен оқушының ерекшеліктеріне қарай жеке немесе бір-бірімен біріктіріп қолдануға болады. Олар әртүрлі жастағы және дағдылар деңгейіндегі балалар үшін оқу процесін қызықты, тиімді және қолжетімді етуге көмектеседі.

Қазіргі білім беруде білім беру технологиялары бірнеше себептерге байланысты негізгі рөл атқарады:

Білім беру технологиялары мұғалімдер мен тәрбиешілерге білім беру үдерісінің тиімділігін арттыруға көмектесетін инновациялық құралдармен және әдістермен қамтамасыз етеді. Олар оқу материалын балаларға қолжетімді, қызықты және түсінікті етуге көмектеседі. Педагогикалық технологиялар оқу-тәрбие процесін әр баланың жеке қажеттіліктері мен қабілетіне қарай бейімдеуге мүмкіндік береді. Бұл мұғалімдерге жеке оқыту тәсілдерін ұсынуға және әр оқушыға барынша қолдау көрсетуге көмектеседі. Көптеген білім беру технологиялары балаларда сыни тұрғыдан ойлау, коммуникативті дағдылар, шығармашылық ойлау, проблемаларды шешу және т.б. сияқты әртүрлі дағдылар мен құзыреттердің дамуына ықпал етеді. Бұл балаларды заманауи әлемге сәтті бейімделуге дайындауға көмектеседі. Интерактивті сабақтар, оқу ойындары және онлайн ресурстар сияқты оқытудың әртүрлі технологияларын пайдалану балаларды белсенді қатысуға және оқуда табысқа жетуге ынталандыратын ынталандырушы және қызықты оқу ортасын құруға көмектеседі. Білім беру технологиясы балаларға ақпараттық технологиялармен және цифрлық құралдармен жұмыс істеу дағдыларын дамытуға көмектеседі, бұл оларды цифрлық технология барған сайын маңызды рөл атқаратын заманауи қоғамға дайындаудың маңызды аспектісі болып табылады.

Жалпы алғанда, білім беру технологиялары қазіргі әлемдегі балалардың тиімді оқуы мен дамуына ықпал ететін инновациялық, бейімді және ынталандырушы білім беру ортасын құруға көмектеседі.

Аналогтарды іздеу және салыстыру

Балаларға арналған бағдарламалық жасақтаманы әзірлеу кезінде нарықтағы ұқсас өнімдерге талдау жасап, олардың әзірленіп жатқан өніммен салыстырғанда артықшылықтары мен кемшіліктерін анықтау маңызды. Төменде бірнеше ұқсас бағдарламалық өнімдерді әзірленіп жатқан бағдарламалық жасақтамамен салыстыру берілген:

1. ABCmouse:

**Артықшылықтары:**

- Әртүрлі пәндік салаларды қамтитын оқу іс-әрекеттері мен ойындарының кең ауқымы.

- Балалардың назарын аударатын қызықты және әртүрлі мазмұн.

- Әр түрлі жас санатындағы балаларға бейімделген пайдаланушы интерфейсі.

**Кемшіліктер**:

- Ата-аналар үшін қосымша қаржылық жүктеме болуы мүмкін пайдалануға жазылу.

- Әр баланың жеке қажеттіліктері мен оқу жылдамдығы ескерілмейді.

2. Khan Academy Kids:

**Артықшылықтары**:

- Білім беру мамандары әзірлеген қызықты ойындар, бейне сабақтар мен тапсырмалар.

- үлгерімін қадағалау және материалдарды әр баланың жеке қажеттіліктеріне бейімдеу мүмкіндігі.

**Кемшіліктер**:

- Аудиторияның шектеулі жас диапазоны 8 жасқа дейін.

- Мазмұн әрқашан үлкен балалардың назарын аудару үшін интерактивті бола бермейді.

3. Smarttic:

**Артықшылықтары**:

- Жасанды интеллект арқылы математиканы оқытудың инновациялық тәсілі.

- Әр баланың жеке қабілеті мен оқу қарқынына бейімделген жеке тапсырмалар.

- баланы жақсы нәтижеге жетуге ынталандыратын ынталандыру және марапаттау жүйесі.

**Кемшіліктер:**

- Қаржылық шектеулерге байланысты кейбір отбасылар үшін қол жетімді болмауы мүмкін ақылы жазылым.

Басты екпін – басқа пәндердегі қажеттіліктерді ескермейтін математикаға.

**Жазылған бағдарламалық жасақтамамен салыстыру**:

Бағдарламалық қамтамасыз етудің артықшылықтары:

- Жазылымсыз немесе төлемдерсіз пайдалануға тегін.

- Әртүрлі пәндік салаларды қамтитын оқу іс-әрекеттері мен ойындарының кең ауқымы.

- Балалардың қауіпсіздігі мен қажеттіліктерін ескере отырып жасалған пайдаланушы интерфейсі.

**Жазылған бағдарламалық құралдың кемшіліктері**:

- Даму ресурстарының шектеулілігі және кейбір аналогтармен салыстырғанда аз функционалдылық.

- Аудиторияны тарту үшін қосымша жылжыту және тарату қажеттілігі.

Жобалық бөлім

Әзірленген бағдарлама әртүрлі қойындылары мен компоненттері бар терезе интерфейсі бар қосымша болады, оның ішінде қажетті ақпарат көрсетіледі. Қойындылар тақырыптар бойынша бөлінеді, ал қойындылардың ішінде балалардың негізгі қажеттіліктерін бейнелейтін және дыбыспен жабдықталған PECS карталары болады, олар байланыс пен өзара әрекеттесу құралы ретінде пайдаланылады.

Жобаның мақсаттарына қол жеткізу және дамуында ауытқуы бар балаларды қолдауға мамандандырылған бағдарламалық қамтамасыз ету тұжырымдамасын жүзеге асыру үшін дизайн бөлімі мақсатты аудиторияның нақты қажеттіліктерін ескере отырып, терезе интерфейсі бар қосымшаның тұжырымдамасын әзірледі.

Әзірленген бағдарлама - қажетті ақпаратқа ыңғайлы қол жеткізуді қамтамасыз ететін әртүрлі қойындылар мен компоненттерді қамтитын интуитивті интерфейсі бар қосымша. Қойындылар тақырып бойынша құрылымдалған, осылайша пайдаланушылар қажетті ақпаратты оңай таба алады.

Қолданбаның негізгі элементі – дамуында ауытқуы бар балалардың негізгі қажеттіліктерін көрсететін PECS (Picture Exchange Communication System) карталары. Әрбір картада белгілі бір қажеттілікті немесе әрекетті бейнелейтін сурет, сондай-ақ сыртқы әлеммен байланыс және өзара әрекеттесу құралы ретінде пайдаланылатын дыбыстық сигнал бар.

PECS карталары тақырып бойынша бөлінген және қолданбадағы сәйкес қойындылар арқылы пайдаланушыға оңай қол жетімді. Пайдаланудың қарапайымдылығын және әртүрлі жағдайларға бейімделуін қамтамасыз ету үшін әрбір карта басқарылады: пайдаланушылар баланың ағымдағы қажеттіліктеріне сәйкес карталарды таңдай, жылжыта және пайдалана алады.

Әзірленген қосымша сонымен қатар әрбір баланың жеке ерекшеліктеріне сәйкес интерфейсті теңшеу және жекелендіру мүмкіндігін береді, бұл олардың жеке қажеттіліктері мен қалауларын ескеруге мүмкіндік береді.

Сонымен қатар, қосымшаны тиімді пайдалануды қамтамасыз ету және оның әлеуетін барынша арттыру үшін мұғалімдер мен ата-аналарға нұсқаулар мен ұсынымдар, соның ішінде PECS карталарын пайдалану және балалармен қарым-қатынасты ұйымдастыру бойынша нұсқаулар берілген.

Дизайн бөлімі қосымшаны одан әрі дамыту бойынша жұмысты жалғастыруда, оның ішінде функционалдылықты кеңейту, интерфейсті жақсарту және пайдаланушының нақты қажеттіліктеріне бейімделу.

4.1 Бағдарламалау тілін таңдаудың негіздемесі

Бағдарламаны әзірлеу үшін C# бағдарламалау тілі таңдалды, ол қазіргі заманғы танымал тілдердің бірі болып табылады бағдарламалау. C# («see sharp» деп айтылады) — нысанға бағытталған және типті қауіпсіз бағдарламалау тілі Microsoft арқылы. Ол заманауи стандарттарға сәйкес келеді бағдарламалау және әзірлеуді қолдау үшін арнайы әзірленген .NET Framework технологиялары. Бағдарламалау тілі жылы жасалды 1990 жылдардың соңында Скотт Вилтаумот бастаған инженерлер тобы және Microsoft корпорациясындағы Андерс Хейлсберг негізгі даму тілі ретінде Microsoft .NET платформасына арналған қолданбалар (бағдарламалық құрал платформасы Кәдімгі бағдарламалар мен веб-қосымшаларды жасауға арналған Microsoft компаниясы). Тіл атауындағы # таңбасы (октоторп) бірнеше нұсқаға ие шифрды шешу: бұл таңба пернетақтада Shift+3 ретінде басып шығарылады, ол да C үшінші іске асырылуын білдіруі мүмкін. Сонымен қатар, С++ екі жұп плюс ретінде түсіндіріледі - бұл жаңа қадамды білдіруі мүмкін C++ тілімен салыстырғанда тіл дамытуда. C# құралдар жинағы есептердің кең ауқымын шешуге мүмкіндік береді, сондықтан дамыту барысында қолданылатын тіл:

- веб-қосымшалар;

.NET Framework .NET платформасында веб-қосымшалар мен веб-қызметтерді жасауға арналған құрылым ASP.NET-пен қуатты интеграциясы арқасында веб-әзірлеуде кеңінен қолданылады. Бұл .NET Framework көмегімен веб-қосымшаларды жасау тәсілдерінің бірі. ASP.NET Web Forms құрылымы мен әрекеті бойынша жұмыс үстелі қолданбаларына ұқсайтын веб-қосымшаларды жасауға мүмкіндік беретін оқиғаға негізделген әзірлеу үлгісін ұсынады.

MVC — қолданбаны үш негізгі құрамдас бөлікке бөлуге мүмкіндік беретін архитектуралық үлгі: үлгі, көрініс және контроллер. ASP.NET MVC икемді және басқарылатын әзірлеуге мүмкіндік беретін осы үлгіні пайдаланып веб-қосымшаларды құру үшін негізді қамтамасыз етеді. Веб қызметтері мен API интерфейстерін жасау үшін .NET Framework ASP.NET Web API қамтамасыз етеді. Бұл құрылым HTTP қызметтерін жасауды және орналастыруды жеңілдетеді, RESTful архитектурасына қолдау көрсетеді және .NET қолданбаларының басқа бөліктерімен біріктіреді.

Веб-әзірлеуде қауіпсіздік маңызды рөл атқарады. .NET Framework веб-қосымшалардың аутентификациясы, авторизациясы және қауіпсіздікті өңдеу механизмдерін қамтамасыз етеді. Сондай-ақ жадтың ағып кетуін болдырмауға және веб-қосымшалардың жұмысын жақсартуға көмектесетін жад пен ресурстарды басқаруға қолдау көрсетіледі. .NET Framework Microsoft SQL Server, MySQL, PostgreSQL және т.б. сияқты әртүрлі дерекқорды басқару жүйелерімен интеграцияны қамтамасыз етеді. Бұл дерекқордағы деректермен әрекеттесетін веб-қосымшаларды жасауды жеңілдетеді. Тұтастай алғанда, .NET Framework әзірлеушілерге әртүрлі веб-қосымшалар мен веб-қызметтерді жасау үшін қуатты құралдармен және құрылымдармен қамтамасыз етеді, сонымен бірге өнімділіктің, қауіпсіздіктің және сенімділіктің жоғары деңгейін қамтамасыз етеді.

- әртүрлі ойын бағдарламалары;

.NET Framework ойын әзірлеу үшін ең кең таралған таңдау емес, әсіресе Unity немесе Unreal Engine сияқты платформалармен салыстырғанда. .NET Framework Windows платформасына бағытталған ойындарды әзірлеу үшін пайдаланылуы мүмкін. Бұл үстел ойындарын, Windows дүкені қолданбаларын және Windows құрылғыларына арналған басқа ойын қолданбаларын жасауды қамтуы мүмкін. .NET Framework графиканы, дыбысты, енгізуді және ойынды дамытудың басқа аспектілерін қоса, ойын әзірлеу үшін пайдалануға болатын әртүрлі кітапханалар мен құралдарды ұсынады. Олардың кейбіреулеріне Windows Forms, WPF, DirectX және XNA Framework кіреді (бірақ XNA енді Microsoft қолдамайды). Unity платформасында жасалған көптеген ойындар негізгі бағдарламалау тілі ретінде C# тілін пайдаланады. Unity .NET Framework-ті тікелей қолданбаса да, C# әзірлеуін .NET функционалдығын жанама пайдалану деп санауға болады. Кейбір ойындар Azure бұлттық қызметтері немесе SQL Server дерекқор серверлері сияқты қызметтермен және инфрақұрылыммен біріктіру үшін .NET Framework пайдалана алады. Жалпы, .NET Framework ойынды әзірлеуге арналған негізгі құрал болмаса да, оны белгілі бір жағдайларда, әсіресе әзірлеуші ​​.NET экожүйесімен жұмыс істеу ыңғайлы болса немесе ойын Windows платформасына бағытталған болса, пайдалануға болады.

- Android және IOS үшін мобильді қосымшалар;

Android және iOS платформаларына арналған мобильді қосымшаларды әзірлеу үшін .NET Framework стандартты таңдау емес.

Xamarin — C# тілі мен .NET платформасын пайдаланып кросс-платформалық мобильді қосымшаларды әзірлеуге арналған құрылым. Ол әзірлеушілерге Android және iOS жүйелерінде қолдануға болатын жалпы қолданба кодын жазуға мүмкіндік береді. Xamarin өнімділігі жоғары және бай қолданбаларды жасауға мүмкіндік беретін жергілікті платформа API интерфейстеріне қол жеткізуді қамтамасыз етеді. Бұл жалпы кодты пайдаланып, бірнеше платформалар арқылы пайдаланушы интерфейстерін жасау құралдарын қамтамасыз ететін Xamarin құрылымының бөлігі. Xamarin.Forms әзірлеушілерге бір пайдаланушы тәжірибесі бар кросс-платформалық қолданбаларды жасауға мүмкіндік береді, әзірлеу процесін жеңілдетеді және Android және iOS қолданбаларында дәйекті көрініс пен сезімді қамтамасыз етеді.

MAUI — Microsoft әзірлеген және .NET 6.0 және C# арқылы кросс-платформалық мобильді қолданбаларды жасау мүмкіндігін қамтамасыз ететін Xamarin.Forms бағдарламасының жаңа нұсқасы. Ол пайдаланушы интерфейстерін жасау және платформа API интерфейстерімен біріктіру үшін үлкен мүмкіндіктерді қамтамасыз етеді. .NET Framework Android және iOS үшін мобильді қолданбаларды әзірлеу үшін тікелей таңдау болмаса да, Xamarin немесе MAUI пайдалану әзірлеушілерге сапалы кросс-платформалық қолданбаларды жасау үшін таныс C# бағдарламалау тілін және .NET құралдарын пайдалануға мүмкіндік береді.

- Windows ОЖ-ға арналған бағдарламалар.

Windows операциялық жүйесіне арналған қолданбаларды әзірлеу үшін .NET Framework тамаша таңдау болып табылады.

Windows Forms (WinForms) — .NET Framework көмегімен Windows жүйесіне арналған жұмыс үстелі қолданбаларын жасауға мүмкіндік беретін технология. Ол графикалық пайдаланушы интерфейсін (GUI) құру үшін басқару элементтерінің бай жиынын қамтамасыз етеді және оқиғаға негізделген бағдарламалау үлгісін қолдайды.

WPF — неғұрлым жетілдірілген көрсету мүмкіндіктері бар жұмыс үстелі қолданбаларын жасауға арналған заманауи технология. WPF пайдаланушы интерфейсін сипаттау үшін XAML (eExtensible Application Markup Language) пайдаланады, бұл бай және интерактивті пайдаланушы интерфейстерін жасауды жеңілдетеді.

UWP — компьютерлер, планшеттер, смартфондар, Xbox және басқалар сияқты әртүрлі Windows 10 құрылғыларында жұмыс істей алатын әмбебап қолданбаларды жасауға арналған платформа. UWP қолданбаларын .NET Framework арқылы әзірлеуге болады және жауап беретін дизайнды және құрылғы мүмкіндіктеріне қол жеткізуді қолдайды.

.NET Framework сонымен қатар Windows амалдық жүйесінің пәрмен жолында іске қосу үшін консольдық қолданбаларды жасауды қолдайды. Бұл графикалық интерфейссіз сценарийлерді, утилиталарды және басқа қолданбаларды әзірлеу үшін пайдалы болуы мүмкін.

.NET Framework интерактивті пайдаланушы интерфейсінсіз фондық режимде жұмыс істейтін Windows қызметтерін жасауға мүмкіндік береді. Бұл фондық процестерді жасау, тапсырмаларды автоматтандыру және басқа жүйелік қолданбалар үшін пайдалы болуы мүмкін. .NET Framework әзірлеушілерге жоғары өнімділік пен тиімділікті қамтамасыз ете отырып, Windows операциялық жүйесі үшін қолданбалардың әртүрлі түрлерін жасау үшін қуатты құралдар мен технологияларды ұсынады.

C# тілінде көптеген пайдалы мүмкіндіктер бар:

Полиморфизм – бір интерфейс арқылы әртүрлі типтегі объектілерді өңдеуге мүмкіндік беретін объектіге бағытталған бағдарламалаудың негізгі принциптерінің бірі. C# тілінде полиморфизм әдісті шамадан тыс жүктеу, виртуалды әдістер және интерфейстер арқылы жүзеге асырылады. Бірнеше объектілерге мүмкіндік беретін жүйенің қасиеті бір интерфейске қол жеткізу үшін ортақ сипаттамалар. Жалпы алғанда полиморфизм ұғымын былайша көрсету керек: «бір интерфейс - көптеген әдістер».

Инкапсуляция – объектіге бағытталған бағдарламалаудың негізгі принциптерінің бірі, ол объект ішінде іске асыру мәліметтерін жасыруды және онымен әрекеттесу интерфейсін қамтамасыз етуді білдіреді. C# тілінде инкапсуляция қолжетімді қатынас модификаторлары (жалпыға ортақ, жеке, қорғалған, ішкі) және сипаттар мен әдістерді пайдалану арқылы жүзеге асырылады. Код пен деректерді бір-бірімен байланыстыратын механизм. ол манипуляциялайды. Бұл механизм осы кодты қорғауға мүмкіндік береді және теріс пайдалану және сыртқы қол жеткізу деректері. Яғни, бұл модификаторлардың көмегімен нысан өрістері мен әдістеріне қол жеткізуді басқару қол жеткізу: жеке, қоғамдық және қорғалған.

Мұрагерлік олардың өрістерін, қасиеттерін, әдістерін және басқа мүшелерін қабылдау арқылы бұрыннан барларға негізделген жаңа сыныптарды жасауға мүмкіндік береді. Мұрагерлік пайда болатын класс базалық класс деп аталады, ал негізгі сыныптың мүшелерін иеленетін класс туынды класс деп аталады. Бұл бір объектінің көмегімен мүмкін болатын қасиет басқаның қасиеттеріне ие болады. Яғни, мұрагер болу мүмкін болады кейбір сыныптар басқа сыныптардың қасиеттері мен мінез-құлқына ие. C# тілі C, C++ және Java сияқты бағдарламалау тілдерімен байланысты.

C# тіліндегі оқиғалар сыныптарға кейбір оқиғалардың орын алғаны туралы басқа сыныптарға хабарлауға мүмкіндік береді. Олар көбінесе оқиғаны өңдеу үлгісін жүзеге асыру үшін пайдаланылады, мұнда жариялаушы деп аталатын бір нысан оқиғаны көтереді және жазылушылар деп аталатын басқа нысандар осы оқиғаға әрекет етеді.

Атрибуттар сыныптар, әдістер, сипаттар және т.б. сияқты код элементтеріне тіркелетін сәндік метадеректерді білдіреді. Олар код элементтерінің құрастырылуы, орындалуы немесе сипаттамасы туралы ақпаратты қосу үшін пайдаланылады.

C# тіліндегі итераторлар бүкіл жинақты жадта бөлмей-ақ жинақтардың элементтерін бір-бірден тізімдеуге мүмкіндік береді. Олар тізімдер, массивтер және т.б. сияқты қайталанатын нысандарды жүзеге асыру үшін қолданылады.

Оператордың шамадан тыс жүктелуі пайдаланушы деректер түрлері үшін стандартты операторлардың әрекетін қайта анықтауға мүмкіндік береді. Бұл нысандармен олар кірістірілген деректер түрлері сияқты жұмыс істеуге мүмкіндік береді.

Жалпы типтер мен әдістер компиляция уақыты түрінің қауіпсіздігін сақтай отырып, әртүрлі деректер түрлерінде жұмыс істей алатын құрамдастарды жасауға мүмкіндік береді.

Статикалық теру - бұл деректер түрлері бағдарламаның орындалу уақытында емес, компиляция уақытында анықталатын тіл мүмкіндігі. Бұл түр қателерін әзірлеу кезінде ерте анықтауға көмектеседі және бағдарлама өнімділігін жақсартады.

Бұл контексте сіз C# тілін немесе оның мүмкіндіктерін, мысалы, төмен деңгейлі ресурстармен жұмыс істеу, ерекше жағдайларды өңдеу немесе орындау ағындарын басқару сияқты арнайы пайдалану жағдайларына сілтеме жасайсыз.

XML түсініктемелері код құжаттамасын тікелей кодтың өзінде жасауға мүмкіндік береді. Оларды Sandcastle немесе Visual Studio сияқты құралдарды пайдаланып құжаттаманы автоматты түрде жасау үшін пайдалануға болады.

C# тілінде қоқыс жинау бағдарламасы бұдан былай пайдаланылмайтын нысандар алатын жадты автоматты түрде босатады. Бұл әзірлеушіні жадты нақты басқарудан босатады және жадтың ағып кетуін болдырмауға көмектеседі.

C++ қазір тіл құрастырылған нысан моделін анықтады C#. Си тілімен байланыс одан мұраланған синтаксиске негізделген, операторлар мен түйінді сөздер. C# және Java арасында тікелей байланыс жоқ, бірақ екеуі де тілдер арнайы, портативті кодты жасауға арналған. Бұл код олардың синтаксисі мен нысанын пайдалана отырып, C және C++ тілдеріне негізделген үлгі. Бір немесе басқа тілді меңгеру кез келген тілге ауысуды жеңілдетеді. Олардың ішінен C# бағдарламалау тілінің ең маңызды ерекшелігі болып табылады көптілді ортада жұмыс істеу қабілеті. Бұл арқылы әртүрлі тілдерде жазылған кодтың бірге жұмыс істеу мүмкіндігі, бұл үлкен бағдарламаларды жасау кезінде маңызды қабілет болып табылады.

Тілдің көптеген артықшылықтары бар:

- Microsoft өнімдеріне қолдау көрсету;

- объектіге бағытталған тәсілді қолдану;

- кітапханалар мен шаблондардың үлкен санының болуы;

- деректер типтерінің бекітілген өлшемі бар: 32-биттік int және 64- бит ұзын, бұл тілдің «тасымалдаушылығын» арттырады және жеңілдетеді бағдарламалау;

- тіл синтаксисінің басқа программалау тілдерімен ұқсастығы, ол бағдарламашыға ауысуды жеңілдетеді (мысалы, Java немесе C++ тілінен);

- арнайы дайын құрылымдардың үлкен саны («синтаксистік қант»), кодты түсінуге және жазуға көмектесу үшін жасалған, құрастырылған кезде мағынасы жоқ;

- автоматты «қоқыс жинау». Бұл онымен жұмыс істеуді жеңілдетеді жадты босату, өйткені жалпы тілдің жұмыс уақыты өзі шақырады қоқыс жинағыш және жадты тазалау;

- кросс-платформалық қосымшаларды құру мүмкіндігі. Көмегімен C# тіліндегі Xamarin олар үшін бағдарламалар мен қосымшаларды жаза алады iOS, Android, MacOS және Linux сияқты операциялық жүйелер;

- қосымша функционалды бағдарламалау (F#).

Дегенмен, кемшіліктер де бар:

- негізінен тек .NET-ке назар аударыңыз (Windows жүйесінде платформа);

- тіл тек шағын компаниялар немесе студенттер тобы үшін тегін.

Ірі компания үшін бұл тілдің лицензияланған нұсқасын сатып алу қымбатқа түседі арзан емес.

4.2 .NET Framework

C# бағдарламалау тілі жұмыс істейтін орта деп аталады NET.Framework, сондықтан оларды бірге қарастырған дұрыс. Қосулы мұның бірнеше себептері бар:

1) C# бастапқыда Microsoft корпорациясы жасау үшін әзірлеген

.NET Framework коды;

2) .NET Framework өзі кітапханаларды анықтайды C# пайдаланылады.

.NET Framework әзірлеуге қолдау көрсететін негіз болып табылады және платформаға тәуелсіз қосымшаларды орындау. Платформа мүмкіндік береді қолданбада әртүрлі тілдерді ортақ пайдалану бағдарламалау және жалпы үлгіні де бере алады Windows жүйесіне арналған бағдарламалау және бағдарламаның тасымалдануы. Дегенмен, қоршаған орта NET Framework тек Windows платформасымен шектелмейді, бірақ жазылған бағдарламаларды басқа платформаларға тасымалдауға болады. .NET жұмыс уақыты мен хостты қамтиды C# бағдарламалары пайдаланатын класс кітапханасы. Орындалу уақытының бір бөлігі жалпы тілдің орындалу уақыты деп аталады (әдетте Common Language Runtime үшін CLR ретінде қысқартылған) себебі бұл Орта тек C# тілін ғана емес, сонымен қатар кез келген .NET тілін де қолдайды. Мысалы, Microsoft сонымен қатар C++ үшін Visual Basic, F# және .NET кеңейтімдерін ұсынады. CLR бар бірнеше тілден код алуға мүмкіндік беретін жалпы типті жүйе (CTS). еркін өзара әрекеттесу, яғни .NET кітапханалары әдетте кез келген .NET тілінен пайдалануға болады. Мысалы, F# пайдалануы мүмкін C# және C# тілінде жазылған кітапханалар визуалды кітапханаларды пайдалана алады Негізгі және т.б.

Жұмыс уақытынан басқа, кең сынып кітапханасы бар. Бұл кітапхана бағдарламаға орындау ортасына қол жеткізуді қамтамасыз етеді, мысалы, оны деректерді енгізу және шығару үшін пайдалану. Бағдарлама болуы мүмкін дейін .NET орындалу уақытына қолдау көрсетілетін жерде жұмыс істейді ол сынып кітапханасы анықтаған сипаттамалармен шектеледі .NET. Сынып кітапханасы көптеген негізгі функциялар үшін орауыштарды қамтамасыз етеді операциялық жүйе (OS), бірақ ол сонымен қатар маңызды қамтамасыз етеді сыныптар сияқты жергілікті функциялардың саны жинақтар немесе JSON өңдеу. Көптеген басқа жүйелер өздерінің кітапханаларын қамтамасыз етеді .NET. Мысалы, мүмкіндік беретін кең кітапханалар бар C# бағдарламалары танымал бұлттық қызметтерді пайдаланады. Қажет болғандай Microsoft жұмыс істеу үшін жан-жақты .NET кітапханаларын қамтамасыз етеді деп күтеміз Azure бұлттық платформасындағы қызметтермен. C# әртүрлі ұсыныстарды ұсынады C-стилі API сияқты басқа API түрлерімен жұмыс істеу механизмдері, Win32, macOS және Linux жүйелерінде немесе компонентке негізделген API интерфейсінде қол жетімді Windows жүйесіндегі нысан үлгісі (COM). Сонымен, C# көмегімен тілге енгізілген абстракциялардың күшті жиынтығы алынады, атап айтқанда: қуатты жұмыс уақыты ортасы, көптеген кітапханаларға оңай қол жеткізу және функционалдық платформа. Көптеген жылдар бойы ең кең таралған жұмыс әдісі компилятор бастапқы кодты өңдеп, шығыс деректерін алды процессор тікелей орындай алатын пішін компьютер.

Компиляторлар машина кодын шығарды - инструкциялар сериясы процессордың осы түріне қажетті кез келген екілік пішім, компьютерде болатын. Көптеген компиляторлар әлі де жұмыс істейді осылай, бірақ C# компиляторы мұны істемейді. Оның орнына ол басқарылатын код деп аталатын үлгіні пайдаланады. Басқарылатын кодты пайдаланған кезде компилятор жасамайды процессор орындайтын машина коды. Оның орнына компилятор жасайды Microsoft Intermediate Language деп аталатын екілік кодтың түрі (Microsoft Intermediate Language, MSIL). Орындалатын екілік файл кейінірек, әдетте орындалу уақытында жасалады. MSIL пайдалану қиын немесе тіпті мүмкін емес функцияларды пайдалануға мүмкіндік береді дәстүрлі үлгіде қамтамасыз етіңіз. Басқарылатын модельдің ең маңызды артықшылығы болуы мүмкін компилятордың шығысы біреуіне байланысты емес процессор архитектурасы. Сіз .NET компонентін жаза аласыз үшін компьютерлер пайдаланатын 32-биттік x86 архитектурасында жұмыс істей алады ондаған жылдар, бірақ ол жаңа 64-битте де жақсы жұмыс істейді осы дизайнды жаңарту (x64) және тіпті мүлдем басқа архитектурада, ARM сияқты. (Мысалы, .NET Core іске қосу мүмкіндігін енгізді Raspberry Pi сияқты ARM негізіндегі құрылғылар). .NET көмегімен сіз кез келгенінде жұмыс істей алатын бір компонентті құрастыруға болады олардың ішінде және ол тіпті жоқ платформаларда да жұмыс істей алады егер ол болашақта пайда болса, кодты құрастыру уақытында қолдау көрсетіледі қолайлы орындау ортасы. Жалпы алғанда, кез келген жақсарту

CLR кодын генерациялау - бұл процессордың жаңа архитектурасын қолдау үшін немесе жай ғана барлардың өнімділігін арттыру – бірден пайда әкеледі барлық .NET тілдері. Мысалы, CLR ескі нұсқалары пайдаланылмаған заманауи нұсқада қол жетімді векторлық өңдеу кеңейтімдерінің артықшылықтары x86 және x64 процессорлары, бірақ қазіргі нұсқалар оларды жиі пайдаланады циклдар үшін код жасау. Бұл іске қосылған барлық кодқа пайдалы .NET Core бағдарламасының ағымдағы нұсқалары, соның ішінде бірнеше жыл бұрын жазылған код осы жақсартуды қосу. CLR құрылғының орындалатын файлын жасайтын нақты сәт од әртүрлі болуы мүмкін. Әдетте ол деп аталатын тәсілді пайдаланады Just-in-time Compilation (JIT), онда әрбір жеке жұмыс істейді бірінші іске қосуда құрастырады. Басқарылатын кодтың барлық жерде түрі туралы ақпарат бар. Форматтар CLI белгілеген файлдар мұның болуын талап етеді, себебі ол белгілі бір орындау уақытының мүмкіндіктерін қамтиды. Мысалы, .NET түрлі автоматты сериялау қызметтерін ұсынады, онда нысандарды екілік немесе мәтіндік көріністерге түрлендіруге болады олардың күйлері, содан кейін бұл өкілдіктерді нысандарға қайтаруға болады, мүмкін басқа машинада. .NET Framework қолданбаларды төмендегілермен қамтамасыз етеді: қызметтер (қызметтер):

- есте сақтауды басқару. .NET Framework қолданбаларында Жалпы тілдің орындалу уақыты жад ресурстарын басқару қызметтерін ұсынады автоматты түрде;

- жалпы типтік жүйе. CTS (Common Type System) жүйесі анықтайды .NET Framework-те базалық типтер.Сонымен қатар, бірдей және барлық .NET Framework тілдері үшін бірдей негізгі түрлер;

- нұсқаның үйлесімділігі. Белгілі бір нұсқаның қолданбалары болуы мүмкін кейінгі нұсқада өзгеріссіз іске қосу;

- тілдердің өзара әрекеттесуі. .NET Framework тілінің компиляторлары қолданбаны аралық кодқа, CIL (Common Intermediate Тіл), ол кейін орындалу уақытында құрастырылады CLR ортасы. Сондықтан бір тілде жазылған бағдарламалар басқа тілдерде қолжетімді;

- кеңейтілген сынып кітапханасы. Үлкен кодты жазуды жеңілдетеді оңай қол жетімді пайдалану арқасында стандартты операциялар үшін .NET Framework сынып кітапханасынан тип және мүше кітапханалары;

- параллель орындау. .NET Framework ажыратымдылыққа көмектеседі CLR бірнеше нұсқасын орнатуға рұқсат беру арқылы нұсқа қайшылықтары бір компьютер. C# бағдарламасын жазу кезінде деп аталатын басқарылатын код. Ол әрқашан басқарылатын орындалу уақыты арқылы жүзеге асырылады (CLR) және операциялық жүйемен тікелей орындалмаған. Басқарылатын орындау уақыты әртүрлі қызмет түрлерін ұсынады, мысалы қоқыс жинау, типті тексеру, ерекше жағдайларды өңдеу, шекараларды тексеру және т.б. бағдарламашының араласуынсыз. Ол сонымен қатар бөлектеуді қамтамасыз етеді код үшін жад, теру қауіпсіздігі және т.б. Өтініш ішінде жазылған Java, C#, VB.Net және т.б. тілдері, өйткені олар әрқашан қоршаған орта қызметтеріне бағытталған басқару үшін орындау және осы тілдерде жазылған код, басқарылатын код ретінде белгілі. .NET Framework жағдайында компилятор әрқашан кодты құрастырады аралық тілдегі басқару (MSIL), содан кейін орындалатын файлды жасайды файл. Бағдарламашы орындалатын файлды іске қосқанда, содан кейін Just In Time CLR компиляторы аралық тілді машиналық кодқа құрастырады, бұл негізгі архитектураға тән. Міне, процесс орындалу уақытында басқарылатын орындау ортасында орын алады, сондықтан бұл орта кодты іске қосуға жауапты. Операциялық жүйемен тікелей орындалатын код басқарылмайтын код деп аталады. Ол әрқашан процессорға бағытталған архитектурасы және компьютер архитектурасына байланысты. Бұл код қашан құрастырылған, ол әрқашан нақты архитектура алуға ұмтылады және әрқашан осы платформада жұмыс істейді, басқаша айтқанда, сіз кез келген уақытта басқа архитектура үшін бірдей кодты орындағыңыз келсе, сізге қажет осы архитектураға сәйкес осы кодты қайта құрастырыңыз. Ол әрқашан берілгенге тән машиналық кодқа жинақталады сәулет. Басқарылмайтын кодта, жадты бөлу және қауіпсіздік, соның ішінде оның ішінде түр қауіпсіздігін әзірлеуші ​​бақылайды. Осыған байланысты бар жадқа қатысты бірнеше проблемалар, мысалы, буфердің толып кетуі, жадтың ағуы, көрсеткішті қайта анықтау және т.б. Орындалатын файлдар Басқарылмайтын код әдетте екілік кескіндерде, x86 кодында кездеседі жадқа тікелей жүктеледі. VB 6.0, C тілінде жазылған қолданба, C++ және т.б. әрқашан басқарылмайтын кодта болады.

4.3 Windows Forms бағдарламалау интерфейсі

Қолданба терезелі пайдаланушы ретінде жасалған визуалды әдістерді қолдану арқылы жүзеге асырылған интерфейс бағдарламалау және Windows Forms кітапханасы сыныптары.

Windows Forms – бұл бағдарламалау интерфейсі қолданбаларды жасау үшін қолданылатын қолданбалар (API), графикалық интерфейсімен жабдықталған. Windows Forms бөлігі болып табылады Microsoft .NET Framework. Интерфейс визуалды қол жеткізуді жеңілдетеді үшін орауыштарды жасайтынына байланысты Microsoft Windows жүйесінің құрамдас бөліктері басқарылатын кодта бар Win32 API [19]. Басқарылатын код жоқ ретінде Windows Forms пайдалануға мүмкіндік беретін әзірлеу тіліне байланысты бағдарламалық жасақтаманы C#, C++ тілінде және VB.Net тілінде жазғанда, J# және басқа тілдер.

Windows Forms-тің негізгі класы - пішін, оның даналары негізгі және диалогтық терезелер болып табылады. Форма – экрандық объект, тікбұрышты пішін түрінде бейнеленген, негізінен қолданылады пайдаланушыға ақпаратты беру немесе енгізуді өңдеу үшін пайдаланушы ақпараты. Олар көп құжат нысанын ала алады интерфейс (MDI), стандартты тілқатысу терезесі және де арналған беті графикалық ақпаратты көрсету. Пішіндер кеңістікте анықталған Form класының ұрпақтары болып табылады System.Windows.Forms деп атайды. Windows Forms интерфейсі жұмыс істеуге мүмкіндік береді жобалау режимі, басқару элементтерін қосу (түймелер, енгізу өрістері мәтін, мәтінді көрсету өрістері, мәзірлер және басқа компоненттер) қарапайым пішіннің бетіне «сүйреп апару». Кодта олар өрістер түрінде берілген сыныпты құрайды. Барлық терезе басқару элементтері сыныптардың объектілері болып табылады, System.Windows.Forms ішінде қамтылған және олар базаның ұрпақтары болып табылады сыныпты бақылау. Бағдарламаның нақты әрекетке жауабы (оқиға) басқарумен байланысты - оқиға өңдеушісі - форма әдісі ретінде ресімделеді.

Орналастыруға болатын кейбір басқару элементтері пішіндер:

- Түйме;

- Заттаңба;

- Панель;

- CheckBox;

- ListBox (Тізім);

- RadioButton (қосқыш);

- Мәзір;

- Құралдар тақтасы;

- TabControl (Tab Management);

- DataGrid (Data Grid);

- TreeView (Ағаш);

- MessageBox (Хабарлар терезесі);

- PictureBox (Сурет).

Қолданбаны іске асыру

5.1 Бағдарламаның құрылымдық көрінісі

Қолданбаны жүзеге асыру болашақтың құрылымын анықтаудан басталады бағдарламалық өнім. Ең алдымен, бұл болуы үшін қажет бағдарламаның құрамы туралы түсінік: терезелердің құрылымдық компоненттерінің болуы бағдарламаларды, дамуды жеңілдетуге көмектесетін ішкі жүйелердің болуы, сондай-ақ бағдарламаның әрбір компонентінің функциялары. Құрылымдық ретінде құрамдас бөліктер, қосымша қойындыларды қамтитын қабылданады пайдалану үшін қажетті ақпарат. Блок-схема басқарудың құрамы мен өзара әрекетін көрсетеді әзірленетін бағдарламалық өнімнің бөліктері. Құрылымдық схема қолданбалы интерфейсті құру ерекшеліктерін көрсететін бағдарламалар, 3.1 суретте көрсетілген. Қойындылар арасында ауысу қажет емес жаңа терезені ашу, барлық әрекеттер негізгі терезеде орындалады, ол қолданбаның жылдамдығын айтарлықтай арттырады және оны пайдалануды жеңілдетеді. Интерфейс қарапайым және түсінікті көрінеді.

5.2 UML қолдану жағдайларының диаграммаларын құру

Диаграммалар бағдарламалық жасақтаманы әзірлеуде және жобаны басқару процесінде маңызды рөл атқарады. Диаграммалар абстрактілі ұғымдар мен идеяларды нақты көрнекі көріністерге аударуға көмектеседі, бұл жобаға қатысушылардың түсінуі мен қарым-қатынасын жеңілдетеді. Диаграммалар жүйенің үлгілерін жасау үшін пайдаланылады, әзірлеушілер мен сәулетшілерге жүйе құрамдастарының құрылымын, тәртібін және қарым-қатынастарын сипаттауға мүмкіндік береді. Диаграммалар жүйені оның негізгі құрамдас бөліктерін, деректер ағындарын, тәуелділіктерін және өзара әрекеттесулерін анықтау арқылы талдауға және жобалауға көмектеседі.

Диаграммалар жүйенің құрылымы мен мінез-құлқын, сондай-ақ пайдалану жағдайларын, талаптарды және т.б. бейнелеуді қамтамасыз ететін жобалық құжаттаманың маңызды бөлігі бола алады. Диаграммалар жобадағы ықтимал тәуекелдер мен проблемаларды анықтауға және көрнекі деректерді талдау негізінде негізделген шешімдер қабылдауға көмектеседі. Ддиаграммаларды жаңа топ мүшелерін оқыту, жүйе архитектурасын сипаттау және жоба туралы білімді жеткізу үшін пайдалануға болады.

Диаграммалар жобаның әртүрлі қатысушылары арасында бірлескен жоспарлау және талаптар мен шешімдерді келісу құралы ретінде қызмет ете алады. Тұтастай алғанда, диаграммалар бағдарламалық жасақтаманы әзірлеудің өмірлік циклінің барлық кезеңдерінде, талаптарды талдау мен жобалаудан тестілеуге, енгізуге және жүйені қолдауға дейін негізгі рөл атқарады. Олар бағдарламалық өнімнің түсінігін, жүйелілігін және сапасын жақсартуға көмектеседі.

Деректер ағынының диаграммалары (DFD) жүйедегі деректер ағындарының графикалық көрінісі болып табылады және ақпараттық жүйенің құрылымы мен әрекетін модельдеу және талдау үшін қолданылады. DFD - бұл функцияларды немесе процестерді көрсететін түйіндер желісі және сол функциялар арасындағы деректер ағынын көрсететін көрсеткілер. Олар деректер көздерінен оны өңдеуге және сақтауға дейін жүйе арқылы деректердің қалай қозғалатынын талдауға мүмкіндік береді.

Процестер деректер бойынша нақты операцияларды немесе әрекеттерді орындайтын функционалдық немесе логикалық бірліктер. Олар шеңберлермен немесе сопақшалармен белгіленеді және сәйкестендіру үшін нөмірленеді. Деректер ағындары әртүрлі процестер немесе деректер қоймалары арасында деректерді тасымалдауды білдіреді. Олар көрсеткілермен көрсетілген және деректер түрін анықтау үшін белгілер болуы мүмкін. Деректер қоймалары – деректерді уақытша немесе тұрақты сақтауға болатын орындар. Бұл файлдар, дерекқорлар немесе басқа ақпарат қоймалары болуы мүмкін. Сыртқы нысандар жүйемен өзара әрекеттесетін сыртқы деректер көздерін немесе раковиналарды білдіреді. Бұл пайдаланушылар, басқа жүйелер немесе сыртқы қызметтер болуы мүмкін.

Жүйені егжей-тегжейлі сипаттау үшін DFD детальдарын бірнеше деңгейге бөлуге болады. Жоғары деңгейде ол әдетте жүйенің жалпы құрылымын және оның негізгі құрамдас бөліктері арасындағы деректер ағынын көрсетеді. Төменгі деңгейде негізгі процестердің әрқайсысын құрайтын процестер мен ішкі процестерді егжей-тегжейлі көрсетуге болады.

Деректердің жүйе арқылы қалай қозғалатынын және оның өңделу жолын визуализациялауға мүмкіндік береді. Деректерді өңдеуде кідірістер немесе ақпарат жоғалуы мүмкін жүйенің аймақтарын анықтауға көмектеседі. абстракцияның әртүрлі деңгейлеріндегі жүйенің құрылымы мен әрекетін талдауға және оның процестерін оңтайландыруға мүмкіндік береді. Ақпараттық жүйе талаптары мен архитектурасын құжаттаудың маңызды құралы ретінде қызмет етеді.

DFD – ақпараттық технологиялар, бағдарламалық инженерия және жүйелік талдау салаларында кеңінен қолданылатын ақпараттық жүйелерді талдау және жобалау құралы.

Класс диаграммалары - бағдарламалық жасақтамадағы сыныптардың құрылымын және олардың қарым-қатынастарын визуализациялау үшін қолданылатын UML (Unified Modeling Language) диаграммасының бір түрі. Олар класстардың, интерфейстердің, атрибуттар мен әдістердің, сондай-ақ олардың арасындағы байланыстардың графикалық көрінісі болып табылады.

Класстар – олардың қасиеттерін (өрістерін) және әрекетін (әдістерін) сипаттайтын нысандардың үлгілері. Класстар әдетте сынып атауы, оның атрибуттары және әдістері үшін бөлінген бөлімдері бар тіктөртбұрыштар ретінде ұсынылады.

сыныптың атрибуттары немесе өрістері сол сынып нысандарында сақталған деректерді білдіреді. Олар әдетте сынып бөлімінде name:type жұптары ретінде көрсетіледі, мұндағы атау атрибуттың атауы және тип - төлсипаттың деректер түрі.

Класс әдістері осы класс объектілерінің әрекетін анықтайды. Олар әдетте сынып бөлімінде атау(параметрлер):қайтару\_түрі жұптары ретінде көрсетіледі, мұндағы атау - әдістің атауы, параметрлер - әдіс параметрлерінің тізімі, қайтару\_түрі - әдіс арқылы қайтарылған деректер түрі.

Класс диаграммалары сондай-ақ бірлестіктер, біріктірулер, композициялар, мұра және интерфейсті іске асыру сияқты сыныптар арасындағы қарым-қатынастардың әртүрлі түрлерін көрсете алады. Бұл қатынастар бір сыныптан екіншісіне бағытты көрсететін көрсеткілермен және қатынас түрін сипаттайтын аннотациялармен ұсынылған.

Бағдарламалық жасақтамадағы сыныптардың құрылымын және олардың байланыстарын визуализациялауға мүмкіндік береді. әдістерді және олардың байланыстарын анықтау арқылы бағдарламалық жасақтаманың архитектурасын талдауға және жобалауға көмектеседі. Кодты құжаттаудың, сыныптарды, олардың атрибуттары мен әдістерін сипаттайтын маңызды құрал ретінде қызмет етеді. әзірлеу топтарына бағдарламалық жасақтама дизайны мен архитектурасын талқылауға және келісуге мүмкіндік береді.

Класс диаграммалары бағдарламалық қамтамасыз етуді талдау мен жобалаудың ең кең тараған құралдарының бірі болып табылады және бағдарламалық жасақтаманы әзірлеудің әртүрлі салаларында кеңінен қолданылады.

Күй диаграммалары, сондай-ақ ақырғы күй машиналары ретінде белгілі, әртүрлі күйлердегі нысанның немесе жүйенің әрекетін және осы күйлер арасындағы ауысуды сипаттау үшін пайдаланылатын UML (Unified Modeling Language) диаграммасының түрі.

Күй диаграммаларының негізгі элементтері нысанның немесе жүйенің әртүрлі күйлерін көрсетеді. Әрбір күй белгілі бір уақыт мезетінде объектінің нақты мінез-құлқын немесе сипаттамаларын анықтайды. Шарттар әдетте шартты сипаттайтын аннотациялары бар шеңберлермен немесе сопақшалармен көрсетіледі.

Өтулер объектінің немесе жүйенің күйлері арасындағы мүмкін ауысуларды анықтайды. Олар бір күйден екінші күйге өту жағдайларын көрсетеді және ауысу кезінде орындалатын оқиғалармен және әрекеттермен жабдықталуы мүмкін. Өтулер әдетте күйлер арасындағы көрсеткілермен көрсетіледі.

Оқиғалар күйлер арасында ауысуды тудыруы мүмкін сыртқы немесе ішкі оқиғаларды білдіреді. Мысалы, пайдаланушы енгізуі, хабардың келуі немесе ішкі таймер ауысуларды іске қосатын оқиғалар болуы мүмкін.

Әрекеттер күйлер арасында ауысу кезінде орындалатын әрекеттерді білдіреді. Олар күйге кіру немесе шығу, оқиғаларды өңдеу немесе белгілі бір операцияларды орындаумен байланысты болуы мүмкін. Әрекеттер әдетте ауысулармен байланысты және олардың жанында тізімделеді.

Әртүрлі жағдайлар мен жағдайларда нысанның немесе жүйенің әрекетін визуализациялауға мүмкіндік береді. Объектінің немесе жүйенің күйлерін, ауысуларын және іске асыру үшін қажетті әрекеттерді анықтау арқылы әрекетін талдауға және жобалауға көмектеседі. Күйлерді, ауысуларды және оқиғаларды сипаттау арқылы объектінің немесе жүйенің әрекетін құжаттаудың маңызды құралы ретінде қызмет етеді. Сынақ жағдайларын жасау, жүйенің дұрыс әрекетін тексеру және қателер мен сәйкессіздіктерді анықтау үшін қолданылады.

Статчарттар бағдарламалық қамтамасыз етуді әзірлеудің әртүрлі салаларында кеңінен қолданылады, соның ішінде бизнес-процестерді модельдеу, пайдаланушы интерфейсін жобалау, байланыс хаттамаларын енгізу және басқалар. Олар әзірлеушілер мен талдаушыларға әртүрлі пайдалану жағдайларындағы жүйе әрекетін жақсырақ түсінуге және басқаруға көмектеседі.

Жүйелік диаграммалар — жүйедегі әртүрлі нысандар немесе құрамдас бөліктер арасындағы өзара әрекеттесу ретін визуализациялау үшін пайдаланылатын UML (Unified Modeling Language) диаграммасының түрі. Олар нысандардың уақыт өте келе бір-біріне хабарламалар мен қоңырау әдістерін қалай алмастыратынын көрсетеді.

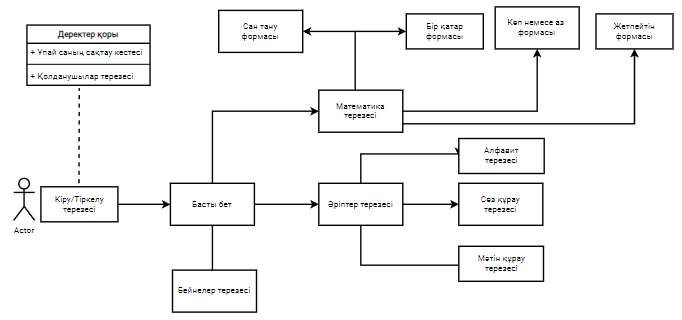
Тізбектік диаграммалардың нысандар өзара әрекеттесуге қатысатын жүйенің объектілерін немесе құрамдастарын білдіреді. Олар диаграмманың жоғарғы жағында орналасқан тік сызықтармен бейнеленеді және әдетте объект немесе сынып атауларымен көрсетіледі. Хабарлар жүйедегі нысандар арасындағы әдіс шақыруларын немесе хабарлама алмасуларын білдіреді. Олар хабарламаны жіберу бағытын көрсететін көрсеткілермен ұсынылған, сонымен қатар хабарлама параметрлері және әдісті шақыру нәтижесі туралы ақпаратты қамтуы мүмкін.

Объектінің өмірлік сызығы жүйедегі нысанның қызмет ету мерзімін көрсетеді. Ол объект жасалған кезде басталып, жойылғанда немесе өзара әрекеттесу аяқталғанда аяқталатын құтқару сызығы деп аталатын тік сызықпен ұсынылған. Операциялар нысандарда шақырылған әдістерді көрсетеді және әдіс атымен анықталады, одан кейін әдіс параметрлерінің тізімі және қайтару түрі (бар болса). Балама немесе қайталанатын өзара әрекеттесу жолдарын сипаттау үшін alts, циклдар және опциондар сияқты басқару фрагменттерін пайдалануға болады.

Жүйедегі нысандар арасындағы хабарлама алмасу және әдісті шақыру ретін визуализациялауға мүмкіндік береді. Жүйедегі тәуелділік пен байланысты анықтай отырып, объектілер арасындағы өзара әрекеттесуді талдауға және жобалауға көмектеседі. Әдістерді шақыру және хабарламаны беру ретін сипаттайтын объектілердің өзара әрекеттесуін құжаттаудың маңызды құралы ретінде қызмет етеді. Сынақ жағдайларын жасау, жүйенің дұрыс әрекеттесуін тексеру және қателер мен сәйкессіздіктерді анықтау үшін қолданылады.

Жүйелік диаграммалар жүйелік талдау мен дизайндағы маңызды құрал болып табылады, бұл дизайнерлерге уақыт өте келе жүйе құрамдастары арасындағы өзара әрекеттесуді жақсы түсінуге және визуализациялауға мүмкіндік береді. Олар бағдарламалық қамтамасыз етуді әзірлеудің әртүрлі салаларында кеңінен қолданылады, соның ішінде бизнес-процестерді модельдеу, жүйе архитектурасын жобалау, пайдаланушы интерфейсін әзірлеу және т.б.

Бағдарламалық өнімді жобалау кезінде оны ескеру қажет ерекшелігі мен мақсаты. Мінез-құлықты графикалық түрде көрсету Жүйе UML деп аталатын тілді пайдаланады. UML (Unified Modeling Language – біртұтас тіл модельдеу) — қолданылатын көрнекі тіл спецификация, визуализация, құрылыс және құжаттама бағдарламалық жүйелер. Use case (немесе use case) диаграммалары – бесеуінің бірі UML-де көріністі модельдеу үшін қолданылатын диаграмма түрлері сыртқы бақылаушы көзқарасы бойынша жүйе. Олар рұқсат етеді үшін жүйенің, ішкі жүйенің немесе сыныптың әрекетін визуализациялау пайдаланушылар оларды қалай пайдалану керектігін түсіндіреді



Актер – жүйемен сырттан әрекеттесетін кез келген нысан. Актер рөлін бағдарламаны пайдаланушы ойнайды. Use case (us case) – реттілік сипаттамасы актерді жетелеу үшін жүйемен орындалатын әрекеттер қалаған нәтиже.

5.3 Visual Studio: шолу және салыстыру

Visual Studio — әртүрлі платформаларда бағдарламалық құрал жасау үшін кеңінен қолданылатын Microsoft корпорациясының біріктірілген әзірлеу ортасы (IDE). Мұнда Visual Studio бағдарламасының негізгі мүмкіндіктері мен мүмкіндіктеріне шолу, сондай-ақ ұқсас әзірлеу орталарымен салыстыру берілген:

1. Интерфейс және қолданудың қарапайымдылығы:

- Visual Studio әзірлеушілерге жылдам жылдамдыққа жетуге және тиімді жұмыс істеуге мүмкіндік беретін пайдаланушыға ыңғайлы және интуитивті интерфейсті ұсынады.

- Біріктірілген әзірлеу ортасы код жазу, жөндеу, тестілеу және қолданбаларды орналастыру үшін құралдардың кең ауқымын қамтамасыз етеді.

2. Бағдарламалау тілдерін қолдау:

- Visual Studio бірнеше бағдарламалау тілдерін қолдайды, соның ішінде C#, Visual Basic, C++, F#, JavaScript және т.б.

- Бірнеше тілдерді біріктіру және жөндеу мүмкіндігі Visual Studio-ны қолданбалардың әртүрлі түрлерін әзірлеуге арналған қуатты құрал етеді.

3. Кеңейту және экожүйе:

- Visual Studio бағдарламасында әзірлеушінің нақты қажеттіліктеріне сәйкес әзірлеу ортасын теңшеуге мүмкіндік беретін кеңейтімдер мен плагиндердің бай экожүйесі бар.

- Жеке кеңейтімдеріңізді жасау және үшінші тарап қызметтерімен біріктіру мүмкіндігі Visual Studio-ны икемді және масштабталатын құрал етеді.

4. Кросс-платформалық және мобильді даму:

- Visual Studio әртүрлі платформалар, соның ішінде Windows, macOS, iOS, Android және Linux үшін қолданбаларды әзірлеуді қолдайды.

- Xamarin-пен интеграция C# тілінде кросс-платформалық мобильді қосымшаларды әзірлеуге мүмкіндік береді.

5. Бұлттық қызметтер және бірлескен даму:

- Visual Studio бағдарламасы Microsoft Azure бұлттық қызметтерімен біріктірілген, бұлттағы қолданбаларды әзірлеу, сынау және орналастыру мүмкіндігін береді.

- Нақты уақыттағы бірлескен даму әзірлеуші ​​топтарға жобалар бойынша тиімді жұмыс істеуге мүмкіндік береді.

Аналогтармен салыстыру:

- Visual Studio коды: Visual Studio-дан айырмашылығы, Visual Studio Code - бұл веб-қосымшаларды әзірлеуге бағытталған және бағдарламалау тілдерінің үлкен санын қолдайтын жеңілірек және кросс-платформалық әзірлеу ортасы.

- Eclipse: Eclipse басқа бағдарламалау тілдерін қолдаса да, негізінен Java әзірлеу үшін пайдаланылатын тағы бір танымал интеграцияланған өңдеу ортасы. Visual Studio-мен салыстырғанда, Eclipse ашық бастапқы кодқа бағытталған және икемді параметрлер мен кеңейтімдерге ие.

Visual Studio кең функционалдығы, әртүрлі бағдарламалау тілдерін қолдауы және бұлттық қызметтермен интеграциясы арқасында ең қуатты және танымал біріктірілген әзірлеу орталарының бірі болып қала береді. Ол Microsoft платформаларында жұмыс істейтін әзірлеушілер үшін, сондай-ақ кросс-платформалық және мобильді қосымшаларды әзірлеу үшін өте қолайлы.

5.4 Microsoft Access: шолу және салыстыру

Microsoft Access — Microsoft корпорациясының жұмыс станцияларында және шағын бизнесте дерекқорларды құруға және басқаруға арналған реляциялық дерекқорды басқару жүйесі (ДҚБЖ). MS Access негізгі мүмкіндіктері мен мүмкіндіктеріне шолу, сондай-ақ ұқсас деректер қорын басқару жүйелерімен салыстыру:

1. Интерфейс және қолданудың қарапайымдылығы:

- Access арнайы дерекқор дағдылары жоқ пайдаланушыларға дерекқорды оңай құруға және басқаруға мүмкіндік беретін интуитивті интерфейсті ұсынады.

- Басқа Microsoft Office қолданбаларымен (мысалы, Excel) интеграция деректерді импорттауды және экспорттауды жеңілдетеді.

2. Мәліметтер қорын әзірлеу және құру мүмкіндіктері:

- Access графикалық интерфейсті пайдаланып, үлгілерді пайдаланып немесе оларды нөлден бастап құруға мүмкіндік береді.

- Пайдаланушылар деректерді ұйымдастыру және басқару үшін кестелерді, сұрауларды, пішіндерді және есептерді жасай алады.

3. Сұрау мүмкіндіктері:

-Access бағдарламасы дерекқордан деректерді таңдауды, жаңартуды, кірістіруді және жоюды қамтитын сұраулардың әртүрлі түрлерін ұсынады.

- SQL (Structured Query Language) көмегімен пайдаланушылар деректерді талдау үшін күрделі сұраулар жасай алады.

4. Есептер мен пішіндерді икемді теңшеу:

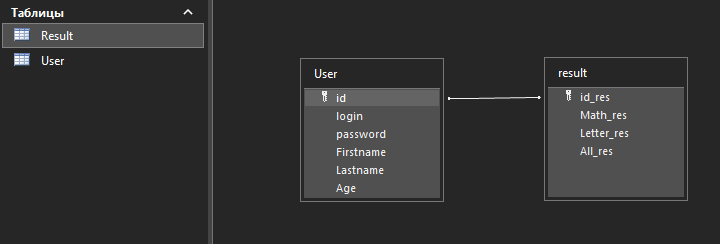
- Пайдаланушылар деректерді ыңғайлы түрде көрсету үшін теңшелетін есептер мен пішіндерді жасай алады.

- Пішімдеу және орналасу опциялары кәсіби көрінетін есептерді жасауға мүмкіндік береді.

5. Шектеулер мен ауқымдылық:

- Access бағдарламасы шағын және орташа өлшемді дерекқорлар үшін ең қолайлы.

- Үлкен көлемдегі деректермен немесе жоғары сервер жүктемесімен жұмыс істегенде, Access өнімділік шектеулеріне тап болуы мүмкін.

****

**Аналогтармен салыстыру:**

- Microsoft SQL Server: Access-тен айырмашылығы, SQL Server корпоративтік деңгейде үлкен және жоғары жүктелген дерекқорлармен жұмыс істеуге арналған толыққанды реляциялық ДҚБЖ болып табылады. SQL Server кең ауқымдылықты, қауіпсіздікті және деректерді басқару мүмкіндіктерін ұсынады.

-MySQL және PostgreSQL: Бұл кең мүмкіндіктерге ие және әр түрлі дерекқор түрлерімен жұмыс істеуге арналған ашық реляциялық ДҚБЖ. Access-пен салыстырғанда, MySQL және PostgreSQL жақсы өнімділік пен ауқымдылықты, сонымен қатар икемді конфигурация мен басқару опцияларын қамтамасыз ете алады.

Microsoft Access шағын және орта өлшемді ұйымдар немесе жұмыс станцияларындағы деректерді басқаруды қажет ететін жеке пайдаланушылар үшін танымал таңдау болып қала береді. Ол деректер қорын құрудың және басқарудың қарапайым және ыңғайлы әдісін ұсынады, бірақ ол үлкен, жоғары жүктемелі жүйелер үшін жеткілікті масштабталмаған болуы мүмкін. ДҚБЖ таңдаған кезде, белгілі бір тапсырма үшін ең қолайлы шешімді таңдау үшін өнімділікті, ауқымдылықты және қауіпсіздік талаптарын ескеру маңызды.

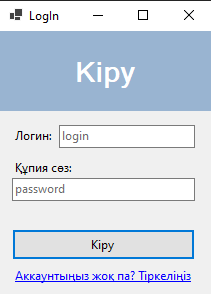
5.5 Бағдарламалық өнім дизайны

Бағдарламалық қамсыздандыру қалыпты windows-тық қаріптермен түстен бөлек, бала назарын аударытын және көзге жағымды түстерден құралатын элементтерді қамтиды.

Берілген түстермен формалар MS Visual Studio платформасында кіріктірілген элементтерден тандалған. Мәтін мен артқы түс ажыратымды болу үшін ашық түспен, қараңғы түстер орнатылды. Элементтер монотонды болмау үшін түрлі түстерге боялды. Көзді ауырпау үшін, түстер рені ашық болып тандалды.

**Рұқсат беру және тіркеу:**

Бағдарлама іске қосылған кезде пайдаланушыға кіру немесе тіркеу пішіні сұралады. Пайдаланушы деректері MS Access деректер базасы арқылы тексеріледі. Сәтті авторизациядан кейін пайдаланушы қосымшаның негізгі пішініне өтеді.



Пайдаланушы өзінің атын, тегін, жасын, логинін және құпия сөзін енгізуге арналған мәтіндік жәшіктерді қамтиды.Ол OleDb көмегімен Microsoft Access дерекқорына қосылым орнатады. «Кіру» түймесін басқан кезде ол пайдаланушы енгізген мәндерді шығарып алады және оларды деректер қорының «Пайдаланушы» кестесіне енгізеді.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Ол SQL инъекциялық шабуылдарын болдырмау үшін параметрленген сұрауларды пайдаланады. Дерекқор жұмысы кезінде орын алуы мүмкін ерекше жағдайларды өңдейді және қажет болған жағдайда қате туралы хабарды көрсетеді.

5.6 Бағдарламаның негізгі функциялары:

Бағдарламаның негізгі функциялары баланың математикалық, логикалық, әріп тану қабілеттерін нығайту негізінде құрылған. Негізгі пішінде әртүрлі ойындар мен тапсырмаларға қол жеткізуге арналған төрт түйме бөлімі бар.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Математика:

Бұл бөлімде әртүрлі математикалық есептер мен ойындар берілген. Тапсырмалар суреттегі заттардың санын болжауды қамтиды.

Пайдаланушыға белгілі бір объектілер саны бар сызба ұсынылады және дұрыс нөмірді көрсету керек.

Изображение выглядит как текст, снимок экрана, Шрифт, логотип

Автоматически созданное описание

Қатарынан ойын:

Бұл бөлімде пайдаланушы ретті болжап, келесі элементті таңдау керек ойын ұсынылады. Пайдаланушыға сандар, әріптер немесе нысандар тізбегі ұсынылады және ол реттіліктегі келесі элементті таңдауы керек.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Аз - көп:

Бұл бөлімде пайдаланушы үлкенірек немесе кішірек санның қай жерде екенін білуі керек ойын ұсынылады. Пайдаланушыға екі сан ұсынылады және қайсысы үлкен немесе аз екенін таңдауы керек.

27

Изображение выглядит как фрукт, Натуральные продукты, снимок экрана, производить

Автоматически созданное описание

Қанша жетіспейді:

Бұл бөлімде пайдаланушыға объектілердің белгілі бір саны ұсынылатын тапсырмалар бар және ол суретте қанша нысан жетіспейтінін болжауы керек. Пайдаланушыға белгілі бір объектілер саны бар кескін ұсынылады және суретте қанша нысан жетіспейтінін көрсетуі керек.

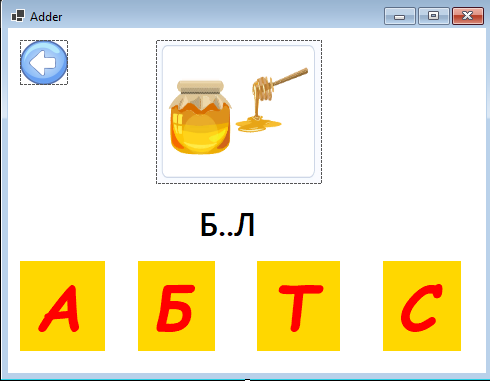
Изображение выглядит как фрукт, клубника, снимок экрана, еда

Автоматически созданное описание

Изображение выглядит как снимок экрана, Графика, логотип, графический дизайн

Автоматически созданное описание

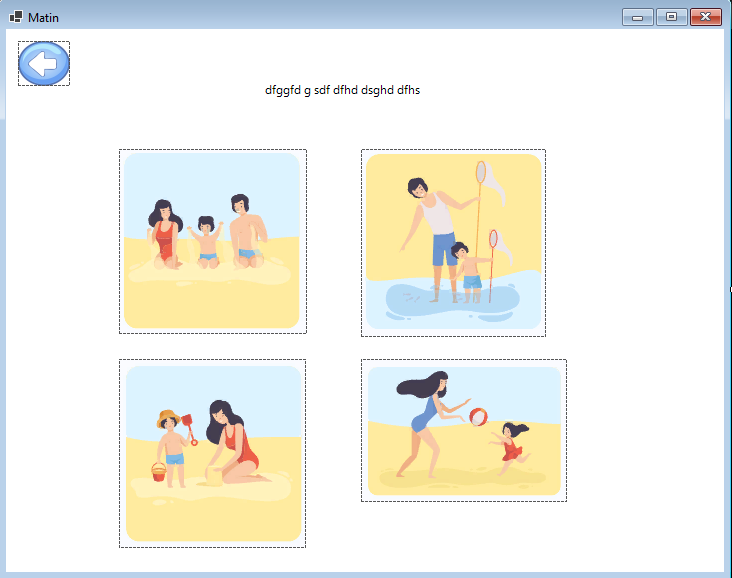
Бұл формада PictureBox суреті берілген және label формасында берілген әріптерді тандау арқылы, тастап кеткен әріпті тандау керек



Бұл формада 42 label элемент берілген және басқан кезде аудио жазба дыбысталады



Мұнда label берілген, және мәтінде берілген сөйлемге сәйкес суреттерді кезекпен тандау керек.



PictureBox және бірнеше панельдер Пішін инициализацияланған кезде, ол панельдер мен сурет жолағы үшін оқиға өңдегіштерін орнатады. Панельдердің кез келгенін басу ағымдағы бейнені ойнатуды тоқтатады, медиа ойнатқыштың URL мекенжайын (MediaPlayer1) белгілі бір бейне файл жолына орнатады және бейнені ойнатуды бастайды.

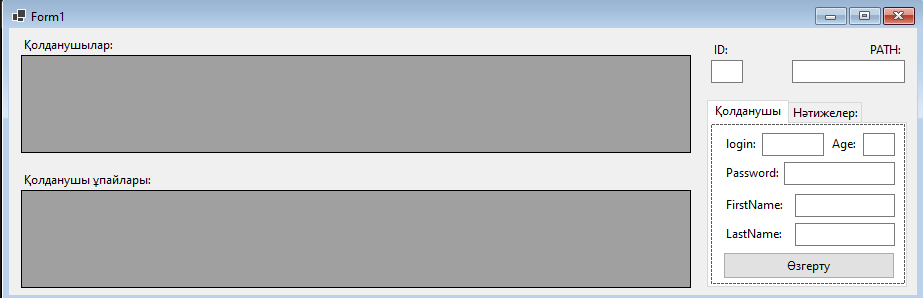
Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

Бұл формада сурет ретінде сандар берілген, әр санды шерткен кезде дыбыстық аудио жазба ойнап сан атауын айтады.



Берілген формада басқарушы админдік панель берілген, мұнда деректер қорымен байланысқан кестені басқару үшін формалар берілген. Екі GridView компоненттері кестедегі деректерді көрсетеді. Идентификатор өрісіне сан жазған кезде өрістерде деректер бейнеленеді. Деректерді өзгерісте енгізіп өзгертуге мүмкіндік береді.

****

* 1. Зерттеудің нәтижелері

Зерттеудің күтілетін нәтижелері жобаның табыстылығын және дамуында ауытқуы бар балаларды оқыту мен дамыту саласына қосқан үлесін анықтауда шешуші рөл атқарады. Олар әзірленген бағдарламалық қамтамасыз етудің тиімділігін және оның мақсатты аудиторияға әсерін бағалауға мүмкіндік беретін нақты көрсеткіштер мен деректерде көрсетіледі.

Бағдарламалық құралды пайдалану дамуында ауытқуы бар балалардың коммуникативті дағдыларын жақсартуға көмектеседі деп күтілуде. Бұл өз ойлары мен тілектерін білдіру қабілетінің жоғарылауында, басқалармен байланыс орнатуда және әлеуметтік дағдыларды дамытуда көрінеді.

Зерттеу нәтижелері балалардың әлеуметтік бейімделу деңгейінің жақсарғанын көрсетуі керек, ол басқалармен қарым-қатынас жасау, топтық іс-шаралар мен іс-шараларға қатысу қабілетінің жоғарылауында көрінеді, сондай-ақ қарым-қатынаста бастамашылдық танытады.

Бағдарламалық қамтамасыз етуді пайдалану дамуында ауытқуы бар балалардың оқу үдерісін жақсартады деп күтілуде, бұл оқуға деген ынтасының жоғарылауында, оқу материалын түсінудің жақсаруында және оқу үлгерімінің жоғарылауында көрінеді.

Зерттеу нәтижелері балалардың және соның ішінде ата-аналар, мұғалімдер және құрдастар арасындағы өзара әрекеттесу сапасының жақсарғанын растауы керек. Бұл әртүрлі жағдайларда неғұрлым жемісті және үйлесімді өзара әрекеттесуде көрінеді.

Алынған деректерді талдау және интерпретациялау үшін сапалы және сандық әдістерді қамтитын кешенді тәсілді қолдану жоспарлануда. Деректерді талдау және түсіндіру жоспары келесі қадамдарды қамтиды:

Сауалнама, бақылау, сұхбат және т.б. сияқты таңдалған зерттеу әдістері мен құралдарын пайдалана отырып, деректер жинауды жүргізу.

Жиналған деректерді статистикалық әдістерді (егер қолданылса), сапалық талдауды және басқа әдістерді қолдану арқылы өңдеу. Бұл кезең әрі қарай талдау үшін деректерді құрылымдау және жүйелеуді қамтиды.

Негізгі тенденцияларды, заңдылықтар мен корреляцияларды анықтау мақсатында алынған мәліметтерге талдау жүргізу. Бұл сипаттамалық талдауды да, деректерді тереңірек статистикалық талдауды да қамтиды.

Жобаның мақсаттары мен міндеттерін ескере отырып, алынған нәтижелерді түсіндіру. Жобаның мақсаттары үшін анықталған заңдылықтар мен қорытындылардың маңыздылығы мен практикалық маңыздылығын талқылау.

Зерттеудің негізгі нәтижелерін, олардың түсіндірмесін және қорытындыларын, сондай-ақ одан әрі іс-әрекеттер мен жобаны дамыту бойынша ұсыныстарды көрсететін есеп беру.

Қорытынды

Балаларға арналған бағдарламалық өнімді әзірлеу нәтижесінде WinForms технологиясы мен MS Access деректер базасын пайдалана отырып, C# тілінде жазылған қауіпсіз, интерактивті және білім беретін қосымшаны құру мақсатына қол жеткізілді.

Бағдарлама пайдаланушыларға білім беру ойындары мен әрекеттеріне қауіпсіз қол жеткізуге мүмкіндік беретін интуитивті интерфейсті қамтамасыз етеді. Балаларға арналған танымал бағдарламалық шешімдерді талдау жаңа өнімге қойылатын негізгі талаптарды анықтауға мүмкіндік берді, атап айтқанда: оқу және дамыту, интерактивтілік, пайдаланушы интерфейсі және математикалық, логикалық амалдарды шешу.

Әртүрлі математикалық есептерді, логикалық ойындарды және оқу тапсырмаларын қамтитын бағдарламалық қамтамасыз етудің негізгі функционалдығы әртүрлі жас санатындағы балалардың танымдық және логикалық дағдыларын дамытуға ықпал етеді.

Бұл жоба балалардың қажеттіліктері мен талаптарын ескеретін мамандандырылған бағдарламалық қамтамасыз етуді жасаудың маңыздылығын растайды. Өнім білім беру мекемелерінде, балалар орталықтарында және үй жағдайында сұранысқа ие болуы мүмкін, бұл балаларға тек дамып қана қоймай, сонымен қатар интерактивті және қызықты тапсырмалар арқылы уақытты пайдалы өткізуге, оқуға және дамытуға көмектеседі.

Тұтастай алғанда, балаларға арналған бағдарламалық қамтамасыз етуді әзірлеу – балалардың дамуы үшін қауіпсіз және білім беру ортасын құру жолындағы маңызды қадам және білім беру мен баланың дамуына ықпал етеді.

Қосымша А

using Project\_for\_kids.data;

using WinApp.frm;

using static Project\_for\_kids.data.Auth;

namespace Project\_for\_kids.frm.Letter

{

public partial class Adder : Form

{

public class Quiz

{

public Dictionary<int, (string imagePaths, string word, List<string> options, string correctAnswer)> DicQuestionValue = new()

{

{ 1, ("bal", "Б..Л", new List<string> {"Б", "А", "С", "Т"}, "А") },

{ 2, ("balyq", "БАЛ..Қ", new List<string> {"О", "Е", "Ы", "У"}, "Ы") },

{ 3, ("gul", "..ҮЛ", new List<string> {"С", "Р", "П", "Г"}, "Г") },

{ 4, ("it", "..Т", new List<string> {"О", "C", "И", "Й"}, "И") },

{ 5, ("keme", "К..М..", new List<string> {"З", "А", "Ж", "Е"}, "Е") },

{ 6, ("kitap", "К..ТАП", new List<string> {"І", "А", "Қ", "E"}, "І") },

{ 7, ("kolik", "КӨ..ІК", new List<string> {"И", "У", "Л", "Ә"}, "Л") },

{ 8, ("kun", "К..Н", new List<string> {"Д", "Ң", "Ұ", "Ү"}, "Ү") },

{ 9, ("mysyq", "М..С..Қ", new List<string> {"Т", "Ы", "І", "Ж"}, "Ы") },

{ 10, ("qaishy", "ҚАЙ..Ы", new List<string> {"Ш", "Х", "У", "Б"}, "Ш") },

{ 11, ("siyr", "..ИЫР", new List<string> {"Т", "С", "В", "М"}, "С") },

{ 12, ("ushaq", "ҰШ..Қ", new List<string> {"Г", "Б", "А", "Ү"}, "А") },

{ 13, ("baqa", "Б..Қ..", new List<string> {"Ы", "Ә", "Д", "А"}, "А") }

};

}

private string imagesFolderPath = @$"{GFold.GPath}resource\\arip\\img\\";

private int counter = 0;

private int currentQuestIndex = 0;

private Quiz quiz = new Quiz();

Auth auth = new Auth();

public Adder()

{

InitializeComponent();

DisplayQuestion(currentQuestIndex);

}

private void DisplayQuestion(int index)

{

var question = quiz.DicQuestionValue[index + 1];

pictureBox2.ImageLocation = Path.Combine(imagesFolderPath, $"{question.imagePaths}.png");

label1.Text = question.word;

label2.Text = question.options[0];

label3.Text = question.options[1];

label4.Text = question.options[2];

label5.Text = question.options[3];

}

private void CheckAnswer(string selectedOption)

{

var question = quiz.DicQuestionValue[currentQuestIndex + 1];

if (selectedOption == question.correctAnswer)

{

counter++;

currentQuestIndex++;

if (currentQuestIndex < quiz.DicQuestionValue.Count)

{

DisplayQuestion(currentQuestIndex);

}

else

{

MessageBox.Show($"Тест завершен. Правильных ответов: {counter} из {quiz.DicQuestionValue.Count}");

auth.SaveTestResults(counter, "Letter\_res");

counter = 0;

currentQuestIndex = 0;

DisplayQuestion(currentQuestIndex);

}

}

else

{

MessageBox.Show($"Попробуйте еще раз.");

}

}

private void label2\_Click(object sender, EventArgs e)

{

CheckAnswer(label2.Text);

}

private void label3\_Click(object sender, EventArgs e)

{

CheckAnswer(label3.Text);

}

private void label4\_Click(object sender, EventArgs e)

{

CheckAnswer(label4.Text);

}

private void label5\_Click(object sender, EventArgs e)

{

CheckAnswer(label5.Text);

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

Form3 form3 = new();

this.Hide();

form3.Show();

}

}

}

using System.Media;

using static Project\_for\_kids.data.Auth;

namespace WinApp.frm

{

public partial class Alphabet : Form

{

private SoundPlayer soundPlayer;

public Alphabet()

{

InitializeComponent();

}

void FunCallSound(string soundName)

{

string soundFilePath = @$"{GFold.GPath}resource\sound\alphabet\{soundName}.wav";

soundPlayer = new SoundPlayer(soundFilePath);

soundPlayer.Play();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

Form3 form3 = new();

this.Hide();

form3.Show();

}

private void label1\_Click(object sender, EventArgs e)

{

FunCallSound("А");

}

private void label2\_Click(object sender, EventArgs e)

{

FunCallSound("Ә");

}

private void label3\_Click(object sender, EventArgs e)

{

FunCallSound("Б");

}

private void label4\_Click(object sender, EventArgs e)

{

FunCallSound("В");

}

private void label5\_Click(object sender, EventArgs e)

{

FunCallSound("Г");

}

private void label6\_Click(object sender, EventArgs e)

{

FunCallSound("Ғ");

}

private void label7\_Click(object sender, EventArgs e)

{

FunCallSound("Д");

}

private void label8\_Click(object sender, EventArgs e)

{

FunCallSound("Е");

}

private void label9\_Click(object sender, EventArgs e)

{

FunCallSound("Ё");

}

private void label10\_Click(object sender, EventArgs e)

{

FunCallSound("Ж");

}

private void label11\_Click(object sender, EventArgs e)

{

FunCallSound("З");

}

private void label12\_Click(object sender, EventArgs e)

{

FunCallSound("И");

}

private void label13\_Click(object sender, EventArgs e)

{

FunCallSound("Й");

}

private void label14\_Click(object sender, EventArgs e)

{

FunCallSound("К");

}

private void label15\_Click(object sender, EventArgs e)

{

FunCallSound("Қ");

}

private void label16\_Click(object sender, EventArgs e)

{

FunCallSound("Л");

}

private void label17\_Click(object sender, EventArgs e)

{

FunCallSound("М");

}

private void label18\_Click(object sender, EventArgs e)

{

FunCallSound("Н");

}

private void label19\_Click(object sender, EventArgs e)

{

FunCallSound("Ң");

}

private void label20\_Click(object sender, EventArgs e)

{

FunCallSound("О");

}

private void label21\_Click(object sender, EventArgs e)

{

FunCallSound("Ө");

}

private void label22\_Click(object sender, EventArgs e)

{

FunCallSound("П");

}

private void label23\_Click(object sender, EventArgs e)

{

FunCallSound("Р");

}

private void label24\_Click(object sender, EventArgs e)

{

FunCallSound("С");

}

private void label25\_Click(object sender, EventArgs e)

{

FunCallSound("Т");

}

private void label26\_Click(object sender, EventArgs e)

{

FunCallSound("У");

}

private void label27\_Click(object sender, EventArgs e)

{

FunCallSound("Ұ");

}

private void label28\_Click(object sender, EventArgs e)

{

FunCallSound("Ү");

}

private void label29\_Click(object sender, EventArgs e)

{

FunCallSound("Ф");

}

private void label30\_Click(object sender, EventArgs e)

{

FunCallSound("Х");

}

private void label31\_Click(object sender, EventArgs e)

{

FunCallSound("Ц");

}

private void label32\_Click(object sender, EventArgs e)

{

FunCallSound("Ч");

}

private void label33\_Click(object sender, EventArgs e)

{

FunCallSound("Ш");

}

private void label34\_Click(object sender, EventArgs e)

{

FunCallSound("Щ");

}

private void label35\_Click(object sender, EventArgs e)

{

FunCallSound("Ы");

}

private void label36\_Click(object sender, EventArgs e)

{

FunCallSound("І");

}

private void label40\_Click(object sender, EventArgs e)

{

FunCallSound("Һ");

}

private void label37\_Click(object sender, EventArgs e)

{

FunCallSound("Э");

}

private void label38\_Click(object sender, EventArgs e)

{

FunCallSound("Ю");

}

private void label39\_Click(object sender, EventArgs e)

{

FunCallSound("Я");

}

}

}

using Project\_for\_kids.data;

using System.Media;

using System.Windows.Forms;

using WinApp.frm;

using static Project\_for\_kids.data.Auth;

namespace Project\_for\_kids.frm.Letter

{

public partial class Matin : Form

{

class Story

{

public Dictionary<int, (string imagePaths, string word, string sounder, List<int> correctanswer)> StorageValue = new()

{

{ 1, ("beach", "word", "А", new List<int> {1,2,3,4 }) },

{ 2, ("cook", "прппары", "Б", new List<int> {1,2,3,4}) },

{ 3, ("evening", "прыкрып вапывапраыв", "В", new List<int> {1,2,3 }) },

{ 4, ("general", "прппары", "Б", new List<int> {1,2,3,4}) },

{ 5, ("girl", "прппары", "Б", new List<int> {1,2,3,4}) },

{ 6, ("kids", "прппары", "Б", new List<int> {1,2,3,4}) },

{ 7, ("nature", "прппары", "Б", new List<int> {1,2,3,4}) },

{ 8, ("play", "прппары", "Б", new List<int> {1,2,3,4}) },

{ 9, ("winnter", "прппары", "Б", new List<int> {1,2,3,4}) }

};

}

private string imagesFolderPath = @$"{GFold.GPath}resource\\arip\\story\\";

private string soundFolderPath = @$"{GFold.GPath}resource\\sound\\alphabet\\";

private int counter = 0;

private int currentQuestIndex = 0;

private Story store = new Story();

private SoundPlayer soundPlayer;

Auth auth = new();

public Matin()

{

InitializeComponent();

DisplayQuestion(currentQuestIndex);

}

private void DisplayQuestion(int index)

{

var question = store.StorageValue[index + 1];

label1.Text = question.word;

pictureBox1.ImageLocation = Path.Combine(imagesFolderPath, $"{question.imagePaths}1.png");

pictureBox2.ImageLocation = Path.Combine(imagesFolderPath, $"{question.imagePaths}2.png");

pictureBox3.ImageLocation = Path.Combine(imagesFolderPath, $"{question.imagePaths}3.png");

pictureBox4.ImageLocation = Path.Combine(imagesFolderPath, $"{question.imagePaths}4.png");

soundPlayer = new SoundPlayer(Path.Combine(soundFolderPath,$"{question.sounder}.wav"));

soundPlayer.Play();

}

private bool allOptionsSelected = false;

int addindex = 0;

private void CheckAnswer(int selectedOption)

{

var question = store.StorageValue[currentQuestIndex + 1];

do

{

if (selectedOption == question.correctanswer[addindex])

{

((PictureBox)Controls.Find($"pictureBox{selectedOption}", true)[0]).Visible = false;

addindex++;

}

else

{

return; // Exit the method if the answer is incorrect

}

} while (addindex < question.correctanswer.Count);

// Check if all options have been selected

if (addindex == question.correctanswer.Count)

{

allOptionsSelected = true;

addindex = 0;

}

else

{

allOptionsSelected = false;

return; // Exit the method if not all options are selected

}

foreach (Control control in Controls)

{

if (control is PictureBox pictureBox)

{

pictureBox.Visible = true;

}

}

counter++;

currentQuestIndex++;

if (allOptionsSelected)

{

if (currentQuestIndex < store.StorageValue.Count)

{

DisplayQuestion(currentQuestIndex);

}

else

{

MessageBox.Show($"Тест завершен. Правильных ответов: {counter} из {store.StorageValue.Count}");

auth.SaveTestResults(counter, "Letter\_res");

counter = 0;

currentQuestIndex = 0;

DisplayQuestion(currentQuestIndex);

}

}

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

CheckAnswer(1);

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

CheckAnswer(2);

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

CheckAnswer(3);

}

private void pictureBox4\_Click(object sender, EventArgs e)

{

CheckAnswer(4);

}

private void backbutton\_Click(object sender, EventArgs e)

{

Form3 form3 = new();

this.Hide();

form3.Show();

}

}

}

using Project\_for\_kids;

using Project\_for\_kids.frm.Letter;

using Project\_for\_kids.frm.Math;

namespace WinApp.frm

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Alphabet secwin2 = new();

this.Hide();

secwin2.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

Adder adder = new();

this.Hide();

adder.Show();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

MainForm form1 = new();

this.Hide();

form1.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

Matin matin = new();

this.Hide();

matin.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

MathNumber number = new();

this.Hide();

number.Show();

}

}

}

using Project\_for\_kids;

using Project\_for\_kids.data;

using System.Data.OleDb;

using WinApp.frm.panel;

namespace WinApp.frm.Login

{

public partial class LogIn : Form

{

public LogIn()

{

InitializeComponent();

}

private void AuthenticateUser(string username, string password)

{

string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=base.accdb";

using (OleDbConnection connection = new OleDbConnection(connectionString))

{

try

{

connection.Open();

string selectQuery = "SELECT [id] FROM [User] WHERE Login = ? AND Password = ?";

using (OleDbCommand command = new OleDbCommand(selectQuery, connection))

{

command.Parameters.AddWithValue("@Login", username);

command.Parameters.AddWithValue("@Password", password);

OleDbDataReader reader = command.ExecuteReader();

if (reader.Read())

{

int userId = Convert.ToInt32(reader["id"]);

Auth.Id = userId;

Auth.Username = username;

Auth.Password = password;

MainForm mainForm = new MainForm();

mainForm.Show();

this.Hide();

}

else if (username == "admin" && password == "admin")

{

AdminPanel panel = new AdminPanel();

panel.Show();

this.Hide();

}

else

{

MessageBox.Show("Логин немесе құпиясөз қате!");

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

string username = textBox\_Username.Text;

string password = textBox\_Password.Text;

AuthenticateUser(username, password);

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

SigIn sigin = new();

this.Hide();

sigin.Show();

}

}

}

using System.Data.OleDb;

namespace WinApp.frm.Login

{

public partial class SigIn : Form

{

public SigIn()

{

InitializeComponent();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

LogIn login = new();

this.Hide();

login.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

string firstname = fname.Text;

string lastname = lname.Text;

int userage;

if (!int.TryParse(age.Text, out userage))

{

MessageBox.Show("Тіркелуші жасын дұрыс енгізіңіз.");

return;

}

string userlogin = login.Text;

string userpassword = password.Text;

string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=base.accdb";

using (OleDbConnection connection = new OleDbConnection(connectionString))

{

try

{

connection.Open();

string insertQuery = @"INSERT INTO [User] ([Login], [FirstName], [LastName], [Age], [Password])

VALUES (?, ?, ?, ?, ?)";

string insertResult = @"INSERT INTO [Result] ([id\_res], [Math\_res], [Letter\_res])

VALUES ((SELECT TOP 1 [id] FROM [User] WHERE [Login] = ?), 0, 0)";

using (OleDbConnection dbConnection = new OleDbConnection(connectionString))

{

try

{

connection.Open();

using (OleDbCommand command = new OleDbCommand(insertQuery, dbConnection))

{

command.Parameters.AddWithValue("@p1", userlogin);

command.Parameters.AddWithValue("@p2", firstname);

command.Parameters.AddWithValue("@p3", lastname);

command.Parameters.AddWithValue("@p4", userage);

command.Parameters.AddWithValue("@p5", userpassword);

command.ExecuteNonQuery();

MessageBox.Show("Қолданушы сәтті тіркелді.");

}

using (OleDbCommand command = new OleDbCommand(insertResult, dbConnection))

{

command.Parameters.AddWithValue("@p1", userlogin);

command.ExecuteNonQuery();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

}

using Project\_for\_kids.data;

using WinFormsApp1;

using static Project\_for\_kids.data.Auth;

namespace Uniclo.frm

{

public partial class MathCount : Form

{

public class Question

{

public int CorrectOptionIndex { get; set; }

public string? ImageName { get; set; }

}

Auth auth = new Auth();

List<Question> questions = new List<Question>

{

new Question { CorrectOptionIndex = 4, ImageName = "balyq4"},

new Question { CorrectOptionIndex = 3, ImageName = "alma3"},

new Question { CorrectOptionIndex = 1, ImageName = "ananas1"},

new Question { CorrectOptionIndex = 5, ImageName = "apelsin5"},

new Question { CorrectOptionIndex = 1, ImageName = "arbuz1"},

new Question { CorrectOptionIndex = 2, ImageName = "avacado2"},

new Question { CorrectOptionIndex = 1, ImageName = "balyq1"},

new Question { CorrectOptionIndex = 2, ImageName = "cocos2"},

new Question { CorrectOptionIndex = 5, ImageName = "japyraq5"},

new Question { CorrectOptionIndex = 5, ImageName = "jidek5"},

new Question { CorrectOptionIndex = 2, ImageName = "japyraq2"},

new Question { CorrectOptionIndex = 1, ImageName = "rakushka1"},

new Question { CorrectOptionIndex = 5, ImageName = "sliva5"},

new Question { CorrectOptionIndex = 3, ImageName = "balyq3"},

new Question { CorrectOptionIndex = 2, ImageName = "rakushka2"},

new Question { CorrectOptionIndex = 5, ImageName = "kivi5"},

new Question { CorrectOptionIndex = 2, ImageName = "balyq2"},

new Question { CorrectOptionIndex = 3, ImageName = "rakushka3"}

};

private string imagesFolderPath = @$"{GFold.GPath}resource\\san\\counter\\";

private int counter = 0;

private int currentImageIndex = 0;

public MathCount()

{

InitializeComponent();

}

private void CheckAnswer(int selectedOption)

{

Question currentQuestion = questions[currentImageIndex];

if (selectedOption == currentQuestion.CorrectOptionIndex)

{

counter++;

currentImageIndex = (currentImageIndex + 1) % questions.Count;

if (currentImageIndex == 17)

{

MessageBox.Show($"Тест завершен. Правильных ответов: {counter}");

auth.SaveTestResults(counter,"Math\_res");

}

else

{

pictureBox1.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}.png");

}

}

else

{

MessageBox.Show("Неправильный ответ. Попробуйте еще раз.");

}

}

private void pictureBox7\_Click(object sender, EventArgs e)

{

CheckAnswer(0);

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

CheckAnswer(1);

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

CheckAnswer(2);

}

private void pictureBox4\_Click(object sender, EventArgs e)

{

CheckAnswer(3);

}

private void pictureBox2\_Click\_1(object sender, EventArgs e)

{

CheckAnswer(4);

}

private void pictureBox6\_Click(object sender, EventArgs e)

{

CheckAnswer(5);

}

private void MathCount\_Load(object sender, EventArgs e)

{

pictureBox1.ImageLocation = Path.Combine(imagesFolderPath, $"balyq4.png");

}

private void pictureBox8\_Click(object sender, EventArgs e)

{

Form2 form2 = new Form2();

this.Hide();

form2.Show();

}

}

}

using Project\_for\_kids.data;

using WinFormsApp1;

using static Project\_for\_kids.data.Auth;

namespace WinApp.frm

{

public partial class MathLess : Form

{

public class Question

{

public int CorrectOptionIndex { get; set; }

public string? ImageName { get; set; }

}

List<Question> questions = new List<Question>

{

new Question { CorrectOptionIndex = 2, ImageName = "1"},

new Question { CorrectOptionIndex = 2, ImageName = "2"},

new Question { CorrectOptionIndex = 1, ImageName = "3"},

new Question { CorrectOptionIndex = 2, ImageName = "4"},

new Question { CorrectOptionIndex = 2, ImageName = "5"},

new Question { CorrectOptionIndex = 2, ImageName = "6"}

};

private string imagesFolderPath = @$"{GFold.GPath}resource\\san\\less\\";

private int counter = 0;

private int currentImageIndex = 0; // Add this line to track the current image index

Auth auth = new Auth();

public MathLess()

{

InitializeComponent();

}

private void CheckAnswer(int selectedOption)

{

Question currentQuestion = questions[currentImageIndex];

if (selectedOption == currentQuestion.CorrectOptionIndex)

{

counter++;

currentImageIndex = (currentImageIndex + 1) % questions.Count;

if (counter == 6)

{

MessageBox.Show($"Тест завершен. Правильных ответов: {counter}");

auth.SaveTestResults(counter, "Math\_res");

}

else

{

pictureBox1.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}0.png");

pictureBox2.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}1.png");

pictureBox3.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}2.png");

pictureBox4.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}3.png");

}

}

else

{

MessageBox.Show("Неправильный ответ. Попробуйте еще раз.");

}

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

CheckAnswer(1);

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

CheckAnswer(2);

}

private void pictureBox4\_Click(object sender, EventArgs e)

{

CheckAnswer(3);

}

private void MathLess\_Load(object sender, EventArgs e)

{

pictureBox1.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}0.png");

pictureBox2.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}1.png");

pictureBox3.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}2.png");

pictureBox4.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}3.png");

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

Form2 form2 = new();

this.Hide();

form2.Show();

}

}

}

using Project\_for\_kids.data;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using WinFormsApp1;

using static Project\_for\_kids.data.Auth;

namespace Program\_for\_kids.frm

{

public partial class MathMore : Form

{

public class Question

{

public int CorrectOptionIndex { get; set; }

public string? ImageName { get; set; }

public string? labeltext { get; set; }

}

List<Question> questions = new List<Question>

{

new Question { CorrectOptionIndex = 1, ImageName = "1", labeltext = "Қай жерде чемодан аз?"},

new Question { CorrectOptionIndex = 0, ImageName = "2", labeltext = "Қайда алма көп?"},

new Question { CorrectOptionIndex = 1, ImageName = "3", labeltext = "Қайда қолғап көп?"},

new Question { CorrectOptionIndex = 1, ImageName = "4", labeltext = "Қай жерде көлік аз?"},

new Question { CorrectOptionIndex = 1, ImageName = "5", labeltext = "Қай жерде кесе көп?"},

new Question { CorrectOptionIndex = 0, ImageName = "6", labeltext = "Жидек қай жерде аз?"},

new Question { CorrectOptionIndex = 0, ImageName = "7", labeltext = "Қай жерде айғұлақ көп?"},

new Question { CorrectOptionIndex = 0, ImageName = "8", labeltext = "Бас киім қайда көп?"},

new Question { CorrectOptionIndex = 1, ImageName = "9", labeltext = "Гүлдер қай жерде көп?"},

new Question { CorrectOptionIndex = 1, ImageName = "10", labeltext = "Қай тарелкеде алма көп?"}

};

private string imagesFolderPath = @$"{GFold.GPath}resource\\san\\more\\";

private int counter = 0;

private int currentImageIndex = 0;

Auth auth = new Auth();

private void CheckAnswer(int selectedOption)

{

Question currentQuestion = questions[currentImageIndex];

if (selectedOption == currentQuestion.CorrectOptionIndex)

{

counter++;

currentImageIndex = (currentImageIndex + 1) % questions.Count;

if (counter == 10)

{

MessageBox.Show($"Тест завершен. Правильных ответов: {counter}");

auth.SaveTestResults(counter, "Math\_res");

}

else

{

label1.Text = questions[currentImageIndex].labeltext;

pictureBox1.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}1.png");

pictureBox2.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}2.png");

}

}

else

{

MessageBox.Show("Неправильный ответ. Попробуйте еще раз.");

}

}

public MathMore()

{

InitializeComponent();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

CheckAnswer(0);

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

CheckAnswer(1);

}

private void MathMore\_Load(object sender, EventArgs e)

{

label1.Text = questions[currentImageIndex].labeltext;

pictureBox1.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}1.png");

pictureBox2.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}2.png");

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

Form2 form2 = new();

this.Hide();

form2.Show();

}

}

}

using System.Media;

using System.Windows.Forms;

using WinApp.frm;

using static Project\_for\_kids.data.Auth;

namespace Project\_for\_kids.frm.Math

{

public partial class MathNumber : Form

{

private SoundPlayer soundPlayer;

public MathNumber()

{

InitializeComponent();

SetTransparentPictures(this);

}

private void SetTransparentPictures(Control control)

{

foreach (Control ctrl in control.Controls)

{

if (ctrl is PictureBox)

{

PictureBox picture = (PictureBox)ctrl;

picture.BackColor = Color.Transparent;

picture.BorderStyle = BorderStyle.None;

}

}

}

void FunCallSound(string soundName)

{

string soundFilePath = @$"{GFold.GPath}resource\sound\san\{soundName}.wav";

soundPlayer = new SoundPlayer(soundFilePath);

soundPlayer.Play();

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

Form3 secwin = new();

this.Hide();

secwin.Show();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

FunCallSound("1");

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

FunCallSound("2");

}

private void pictureBox4\_Click(object sender, EventArgs e)

{

FunCallSound("3");

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

FunCallSound("4");

}

private void pictureBox6\_Click(object sender, EventArgs e)

{

FunCallSound("5");

}

private void pictureBox11\_Click(object sender, EventArgs e)

{

FunCallSound("6");

}

private void pictureBox10\_Click(object sender, EventArgs e)

{

FunCallSound("7");

}

private void pictureBox9\_Click(object sender, EventArgs e)

{

FunCallSound("8");

}

private void pictureBox8\_Click(object sender, EventArgs e)

{

FunCallSound("9");

}

private void pictureBox7\_Click(object sender, EventArgs e)

{

FunCallSound("10");

}

}

}

using Project\_for\_kids.data;

using WinFormsApp1;

using static Project\_for\_kids.data.Auth;

namespace Program\_for\_kids.frm

{

public partial class MathRow : Form

{

public class Question

{

public int CorrectOptionIndex { get; set; }

public string? ImageName { get; set; }

public string? labeltext { get; set; }

}

List<Question> questions = new List<Question>

{

new Question { CorrectOptionIndex = 0, ImageName = "1"},

new Question { CorrectOptionIndex = 1, ImageName = "2"},

new Question { CorrectOptionIndex = 0, ImageName = "3"},

new Question { CorrectOptionIndex = 0, ImageName = "4"},

new Question { CorrectOptionIndex = 0, ImageName = "5"},

new Question { CorrectOptionIndex = 0, ImageName = "6"},

new Question { CorrectOptionIndex = 1, ImageName = "7"},

new Question { CorrectOptionIndex = 0, ImageName = "8" }

};

private string imagesFolderPath = @$"{GFold.GPath}resource\\san\\row\\";

private int counter = 0;

private int currentImageIndex = 0;

Auth auth = new();

private void CheckAnswer(int selectedOption)

{

Question currentQuestion = questions[currentImageIndex];

if (selectedOption == currentQuestion.CorrectOptionIndex)

{

counter++;

currentImageIndex = (currentImageIndex + 1) % questions.Count;

if (counter == 8)

{

MessageBox.Show($"Тест завершен. Правильных ответов: {counter}");

auth.SaveTestResults(counter, "Math\_res");

}

else

{

//label1.Text = questions[currentImageIndex].labeltext;

pictureBox1.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}0.png");

pictureBox2.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}1.png");

pictureBox3.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}2.png");

}

}

else

{

MessageBox.Show("Неправильный ответ. Попробуйте еще раз.");

}

}

public MathRow()

{

InitializeComponent();

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

CheckAnswer(0);

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

CheckAnswer(1);

}

private void MathRow\_Load(object sender, EventArgs e)

{

pictureBox1.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}0.png");

pictureBox2.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}1.png");

pictureBox3.ImageLocation = Path.Combine(imagesFolderPath, $"{questions[currentImageIndex].ImageName}2.png");

}

private void pictureBox4\_Click(object sender, EventArgs e)

{

Form2 form2 = new();

this.Hide();

form2.Show();

}

}

}

using Project\_for\_kids;

using System.Data;

using System.Data.OleDb;

using static Project\_for\_kids.data.Auth;

namespace WinApp.frm.panel

{

public partial class AdminPanel : Form

{

private readonly string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=base.accdb";

private OleDbCommand userSelectCommand;

private OleDbCommand resultSelectCommand;

private DataSet userDataSet;

private DataSet resultDataSet;

public AdminPanel() => InitializeComponent();

public void LoadDataTable()

{

using (OleDbConnection connection = new OleDbConnection(connectionString))

{

try

{

connection.Open();

userSelectCommand = new OleDbCommand("SELECT \* FROM [User]", connection);

resultSelectCommand = new OleDbCommand("SELECT \* FROM [Result]", connection);

OleDbDataAdapter userAdapter = new OleDbDataAdapter(userSelectCommand);

OleDbDataAdapter resultAdapter = new OleDbDataAdapter(resultSelectCommand);

userDataSet = new DataSet();

resultDataSet = new DataSet();

userAdapter.Fill(userDataSet, "UserTable");

resultAdapter.Fill(resultDataSet, "ResultTable");

dataGridView1.DataSource = userDataSet.Tables["UserTable"];

dataGridView2.DataSource = resultDataSet.Tables["ResultTable"];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

private void AdminPanel\_Load(object sender, EventArgs e)

{

LoadDataTable();

textBox1.Text = GFold.GPath;

}

public void button1\_Click(object sender, EventArgs e)

{

int id = Convert.ToInt32(idTextBox.Text);

string login = loginTextBox.Text;

string password = passwordTextBox.Text;

string firstname = firstnameTextBox.Text;

string lastname = lastnameTextBox.Text;

int age = Convert.ToInt32(ageTextBox.Text);

try

{

using (OleDbConnection connection = new OleDbConnection(connectionString))

{

connection.Open();

string query = "UPDATE [User] SET [Login] = ?, [Password] = ?, [Firstname] = ?, [Lastname] = ?, [Age] = ? WHERE [id] = ?";

using (OleDbCommand command = new OleDbCommand(query, connection))

{

command.Parameters.AddWithValue("@p1", login);

command.Parameters.AddWithValue("@p2", password);

command.Parameters.AddWithValue("@p3", firstname);

command.Parameters.AddWithValue("@p4", lastname);

command.Parameters.AddWithValue("@p5", age);

command.Parameters.AddWithValue("@p6", id);

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Данные успешно обновлены");

LoadDataTable();

}

else

{

MessageBox.Show("Не удалось обновить данные");

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

private void button3\_Click(object sender, EventArgs e)

{

int id = Convert.ToInt32(idTextBox.Text);

int math\_res = Convert.ToInt32(MathTextBox.Text);

int letter\_res = Convert.ToInt32(LetterTextBox.Text);

try

{

using (OleDbConnection connection = new OleDbConnection(connectionString))

{

connection.Open();

string query = "UPDATE [Result] SET [Math\_res] = ?, [Letter\_res] = ? WHERE [id\_res] = ?";

using (OleDbCommand command = new OleDbCommand(query, connection))

{

command.Parameters.AddWithValue("@p1", math\_res);

command.Parameters.AddWithValue("@p2", letter\_res);

command.Parameters.AddWithValue("@p3", id);

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Данные успешно обновлены");

LoadDataTable();

}

else

{

MessageBox.Show("Не удалось обновить данные");

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

private void idTextBox\_TextChanged(object sender, EventArgs e)

{

using (OleDbConnection connection = new OleDbConnection(connectionString))

{

try

{

connection.Open();

if (idTextBox.Text == "")

{

loginTextBox.Text = "";

passwordTextBox.Text = "";

firstnameTextBox.Text = "";

lastnameTextBox.Text = "";

ageTextBox.Text = "";

idUserTextBox.Text = "";

MathTextBox.Text = "";

LetterTextBox.Text = "";

}

else

{

string query = "SELECT \* FROM [User] INNER JOIN [Result] ON [User].[id] = [Result].[id\_res] WHERE [User].[id] = ?";

using (OleDbCommand command = new OleDbCommand(query, connection))

{

command.Parameters.AddWithValue("@p1", idTextBox.Text);

OleDbDataReader reader = command.ExecuteReader();

if (reader.Read())

{

loginTextBox.Text = reader["Login"].ToString();

passwordTextBox.Text = reader["Password"].ToString();

firstnameTextBox.Text = reader["Firstname"].ToString();

lastnameTextBox.Text = reader["Lastname"].ToString();

ageTextBox.Text = reader["Age"].ToString();

idUserTextBox.Text = reader["id\_res"].ToString();

MathTextBox.Text = reader["Math\_res"].ToString();

LetterTextBox.Text = reader["Letter\_res"].ToString();

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

textBox1.TextChanged -= textBox1\_TextChanged;

GFold.GPath = textBox1.Text;

textBox1.TextChanged += textBox1\_TextChanged;

}

}

}

using Project\_for\_kids.data;

using Project\_for\_kids.frm.Video;

using WinApp.frm;

using WinApp.frm.Login;

using WinApp.frm.panel;

using WinFormsApp1;

namespace Project\_for\_kids

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

label1.Text = @$"Ұпай: {Auth.LetterBall+ Auth.MathBall}";

}

private void button1\_Click(object sender, EventArgs e)

{

Form2 matform = new();

this.Hide();

matform.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

Form3 secWin = new();

this.Hide();

secWin.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

AdminPanel panel = new();

panel.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

LogIn logiform = new();

logiform.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

VideoPanel videopanel = new();

this.Hide();

videopanel.Show();

}

private void MainForm\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

}

}

using static Project\_for\_kids.data.Auth;

namespace Project\_for\_kids.frm.Video

{

public partial class VideoPanel : Form

{

public VideoPanel()

{

InitializeComponent();

}

private void pictureBox6\_Click(object sender, EventArgs e)

{

MainForm main = new();

MediaPlayer1.close();

this.Hide();

main.Show();

}

private void soundVideoPanel(string val)

{

MediaPlayer1.Ctlcontrols.stop();

MediaPlayer1.URL = @$"{GFold.GPath}resource\\video\\{val}.mp4";

MediaPlayer1.Ctlcontrols.play();

}

private void panel2\_Click(object sender, EventArgs e)

{

soundVideoPanel("1");

}

private void panel3\_Click(object sender, EventArgs e)

{

soundVideoPanel("2");

}

private void panel4\_Click(object sender, EventArgs e)

{

soundVideoPanel("3");

}

private void panel5\_Click(object sender, EventArgs e)

{

soundVideoPanel("4");

}

}

}