Зміст

[Вступ 2](#_Toc265584028)

[1. Характеристика предметної області і постановка задачі. 4](#_Toc265584029)

[1.1 Аналіз предметної області 4](#_Toc265584030)

[1.2 Аналіз інформаційного забезпечення предметної області 10](#_Toc265584031)

[1.3 Постановка задачі 16](#_Toc265584032)

[2. Проектування структури інформаційної системи 18](#_Toc265584033)

[2.1 Аналіз та автоматизація обробки інформаційних потоків 18](#_Toc265584034)

[2.2 Розробка структури інформаційної системи 18](#_Toc265584035)

[2.2.1 Visual Studio .NET 2008 18](#_Toc265584036)

[2.2.2 Delphi 7 21](#_Toc265584037)

[2.2.3 Бази даних 23](#_Toc265584038)

[2.3. Вибір засобів розробки інформаційної системи 28](#_Toc265584039)

[3. Програмна реалізація 30](#_Toc265584040)

[3.1 Структура і функціональне призначення модулів системи, їх взаємозв’язок 30](#_Toc265584041)

[3.2 Розробка програмних модулів 31](#_Toc265584042)

[3.3 Інструкція користувача 51](#_Toc265584043)

[3.4 Вимоги до технічних засобів 53](#_Toc265584044)

[Висновки 55](#_Toc265584045)

[Література 56](#_Toc265584046)

# Вступ

На відміну від вищих учбових закладів школа має свою специфіку, яка дозволяє реалізувати важливі ідеї електронних журналів і переосмислити їх роль в освітньому процесі. Основний аргумент, який наводять в школах противники подібних нововведень, – додаткове навантаження для вчителів за яке додатково не сплачується. В першу чергу це пов'язано з тим, що в школах виставляється на порядок більше поточних оцінок, а доручити учням роботу по внесенню інформації в електронний журнал не завжди можливо. В сукупності із хронічною перевантаженістю вчителів в більшості шкіл і низьким рівнем зарплат це дійсно створює в педколективі атмосферу неприйняття нової ідеї. Окрім цього можуть накладатися проблеми технічного оснащення школи (недостатня кількість ПК, недостатні характеристики ПК), низький рівень володіння ПК у педагогів, складність установки і налаштування програмного забезпечення, психологічні проблеми (занепокоєння з приводу несанкціонованого втручання з боку учнів, втрати інформації у наслідок технічних проблем та інше). Але практиці впровадження елктронного журналу означає виконання автоматизованої обробки аналізу успішності школярів; звільнення співробітників від ряду рутинних робіт з підготовки звітної документації; створення бази даних про учнів; здійснення операцій по створенню архівів успішності учнів. Завдяки автоматизації можна виконувати пошук інформації за запитами викладачів.

Достатній мінімум – один комп'ютер, що має мінімальні системні вимоги, за допомогою якого будуть створюватися електронні журнали і вноситися оцінки. Найкращий варіант – в кожній аудиторії комп'ютер з постійним підключенням до локальної мережі, а база даних журналу знаходиться на головному комп’ютері, кожен викладач володіє комп'ютером на рівні досвідченого користувача, всі оцінки вносяться до електронного журналу безпосередньо на заняттях. Між найпростішим і найкращим можливо безліч інших варіантів. Жодних особливих вимог до операційної системи і програмного забезпечення комп'ютера немає. Враховуючи сказане, можна стверджувати, що кожний учбовий заклад, який має якесь комп’ютерне забезпечення технічно готовий до впровадження електронних журналів успішності.

Повною мірою відчути певний результат у наслідок такого підходу можна буде, коли база оцінок набере свою «критичну масу» і пройде деякий час, щоб педагоги змогли оцінити нововедення і свій зекономлений час.

З урахуванням викладеного, можна запропонувати просте рішення питання про додаткове навантаження вчителів, але і звільнення їх від ряду рутинних робіт. Це допоможе уникнути більшості проблем, але в той же час дозволить досягти основних цілей впровадження.

# 1. Характеристика предметної області і постановка задачі

## 

## 1.1 Аналіз предметної області

Процес навчання в школі спрямований на вирішення навчально-виховних завдань, кожне з яких характеризується дидактичною завершеністю. Обов'язковим компонентом цього процесу є контроль знань, умінь та навичок, тобто перевірка його результативності.

Головна мета контролю як дидактичного засобу управління навчанням — забезпечення його ефективності приведенням до системи знань, умінь, навичок учнів, самостійного застосування здобутих знань на практиці, стимулювання навчальної діяльності учнів, формування у них прагнення до самоосвіти.

Контроль знань учнів складається з:

*перевірки* — виявлення рівня знань, умінь та навичок;

*оцінки* — вимірювання рівня знань, умінь і навичок;

*обліку* — фіксування результатів у вигляді оцінок у класному журналі, щоденнику учня, відомостях.

За допомогою контролю в процесі навчання розв'язують низку завдань: виявлення готовності учнів до сприйняття, усвідомлення і засвоєння нових знань; отримання Інформації про характер самостійної роботи у процесі навчання; визначення ефективності організаційних форм, методів і засобів навчання; виявлення ступеня правильності, обсягу і глибини засвоєних учнями знань, умінь та навичок. Ці та інші завдання визначають зміст контролю, який змінюється із зміною дидактичних завдань.

Контроль виконує такі функції:

*а) освітню (навчальну)* — суть якої в тому, що вона є корисною для всього класу. Слухаючи змістовну відповідь товариша, учні звіряють з нею свої знання, ставлять запитання, доповнюють її, що сприяє повторенню та систематизації знань усіх учнів. Освітнє значення для учнів має й те, що вони слухають доповнення вчителем неповних відповідей учнів. Освітня функція чітко виявляється також у тому, що учень, який відповідає перед класом, повторює вивчений матеріал, виконує практичні завдання, закріплюючи знання, вміння і навички, краще усвідомлює навчальний матеріал;

*б) виховну* — яка виходить з того, що очікування перевірки спонукає учня регулярно готувати уроки. Крім того, перевірка і оцінювання знань допомагають учневі самому оцінити свої знання і здібності: одні позбуваються зайвої самовпевненості та самозаспокоєння, другі починають усвідомлювати свої труднощі, треті починають відчувати впевненість у власних силах. Все це сприяє формуванню позитивного ставлення учнів до навчання;

*в) розвиваючу* — в процесі навчання в учнів розвивається логічне мислення, зокрема вміння аналізу і синтезу, порівняння І узагальнення, абстрагування і конкретизації, класифікації та систематизації, мислительна діяльність, мовлення, пам'ять, уява, увага;

*г) діагностичну* — у процесі контролю виявляють успіхи та недоліки в знаннях, уміннях і навичках учнів, встановлюють причини і шляхи їх усунення, визначають заходи, спрямовані на поліпшення успішності;

*ґ) стимулюючу* — схвалення успіхів учня сприяє розвитку в нього спонукальних мотивів до навчання;

*д) оцінювальну* — об'єктивна оцінка знань, умінь і навичок учнів сприяє кращому навчанню;

*е) управлінську* — на основі контролю визначається стан успішності учнів, що дає змогу запобігти неуспішності або подолати її. В цьому разі вчитель коригує і свою власну діяльність — змінює методику викладання, вдосконалює навчальну діяльність учнів.

За місцем у навчальному процесі розрізняють попередній, поточний, періодичний, підсумковий види контролю.

Попередній контроль здійснюють переважно з діагностичною метою перед вивченням нової теми або на початку уроку, семестру для з'ясування загального рівня підготовки учнів з предмета, щоб намітити організацію їх навчально-пізнавальної діяльності.

Поточний контроль використовують у повсякденній навчальній роботі. Він полягає в систематичному спостереженні вчителя за навчальною діяльністю учнів на уроці. Мета такого контролю — отримання оперативних даних про рівень знань учнів і якість навчальної роботи на уроці та вирішення завдань управління навчальним процесом.

Періодичний (тематичний) контроль передбачає виявлення й оцінку знань та умінь учнів, засвоєних на кількох попередніх уроках, з метою визначення, наскільки успішно вони володіють системою знань, чи відповідають ці знання програмі. Різновидом періодичного є тематичний контроль, що полягає у перевірці та оцінюванні знань учнів з кожної теми і спрямований на те, щоб кожен учень належно засвоїв кожну тему. Така система дає змогу усунути елементи випадковості при виведенні підсумкових оцінок, що трапляється, коли вчитель орієнтується лише на поточний контроль. При тематичному опитуванні перед початком вивчення чергової теми усі учні мають бути ознайомлені з тривалістю вивчення теми (кількість занять); кількістю і тематикою обов'язкових робіт і термінами їх проведення; питаннями, що виносяться на атестацію, якщо атестація проводиться в усно-письмовій формі, або орієнтованим завданням (задачами) тощо; терміном і формою проведення тематичної атестації; умовами оцінювання. Якщо темою передбачено виконання учнями практичних, лабораторних робіт та інших обов'язкових практичних завдань, то їхнє виконання є обов'язковою умовою допуску учнів до тематичної атестації.

Перед учнями, які не засвоїли матеріалу теми чи отримали бали на початковому рівні, ставиться вимога обов'язкового його доопрацювання; їм надається для цього необхідна допомога, визначається термін повторної атестації. Учень має право на переатестацію також для підвищення атестаційного балу.

Підсумкове оцінювання здійснюється наприкінці семестру або навчального року. Підсумкова оцінка за семестр виставляється за результатами тематичного оцінювання, а за рік — на основі семестрових оцінок.

Змістом перевірки рівня навчальних досягнень учня має бути не тільки виявлення суми засвоєних знань, умінь та навичок, а й сформованості й компетенції, тобто загальної здатності, що базується на знаннях, досвіді, цінностях, здібностях, набутих завдяки навчанню.

Основними групами компетенцій, яких потребує сучасне життя, є:

*— соціальні* — пов'язані з готовністю брати на себе відповідальність, бути активним у прийнятті рішень, у суспільному житті, у врегулюванні конфліктів ненасильницьким шляхом, у функціонуванні і розвитку демократичних інститутів суспільства;

*— полікультурні* — стосуються розуміння несхожості людей, взаємоповаги до їхньої мови, релігії, культури тощо;

*— комунікативні* — передбачають опанування важливими у роботі і суспільному житті усним і писемним спілкуванням, оволодіння кількома мовами;

*— інформаційні* — зумовлені зростанням ролі інформації у сучасному суспільстві і передбачають оволодіння інформаційними технлогіями, умінням здобувати, критично осмислювати і використовувати різноманітну інформацію;

*— саморозвитку* — та самоосвіти, пов'язані з потребою і готовністю постійно навчатися як у професійному відношенні, так і в особистому та суспільному житті;

*— компетенції* — що реалізується у прагнені і здатності до раціональної продуктивної, творчої діяльності.

Ефективність контролю залежить від його організації: часу проведення контрольних занять, їх частоти й послідовності; характеру й форм самостійної роботи учнів (індивідуальна, групова, фронтальна); використання дидактичних і технічних засобів навчання; поєднання методів контролю і самоконтролю (усна, письмова, графічна, практична, тестова, програмована перевірка); фіксування й оформлення даних контролю процесу навчання.

Ефективність контролю забезпечується дотриманням певних вимог, серед яких:

*— індивідуальний характер контролю успішності*. Виявлення знань кожного учня, його успіхів чи невдач; знання рівня самостійності учня в пізнавальному процесі, характеру труднощів, як він їх долає і якої допомоги потребує; особлива увага до учнів з фізичними вадами; вміле формулювання запитань, використання додаткових запитань під час опитування слабших учнів та ін.

*— систематичність контролю*. Привчання учнів систематично виконувати уроки, створення в класі відповідного морально-психологічного клімату, коли «не знати — соромно»; систематичне опитування за допомогою самостійних міні-завдань; розробка системи опитування, за якої оцінки виставляють і учням, які доповнювали відповіді інших, були активними на уроці; приділення особливої уваги слабшим учням, спонукання їх до пізнавальної діяльності на всіх етапах уроку.

Відсутність системи в опитуванні учнів призводить до того, що вони вчать матеріал тільки тоді, коли відчувають, що їх можуть викликати, і намагаються визначити наперед, коли і з якого предмета будуть опитувати. Щоб запобігти цьому, учитель повинен добре продумати систему опитування.

Достатня кількість даних для оцінки: при виставленні оцінки враховуються насамперед відповіді учня на поставлені запитання, а також його доповнення відповідей інших учнів на цьому й попередніх уроках.

Дотримання об'єктивності під час оцінювання знань: оцінка виставляється тільки за знання учня, а не за те, що він «забув щоденник» чи «крутився на уроці»; на оцінку не повинні впливати суб'єктивні чинники, особисті мотиви ставлення педагога до учня; кожну виставлену оцінку педагог обґрунтовує, щоб запобігти невдоволенню учнів, схильних до переоцінки своїх знань.

*— єдність вимог до оцінювання знань учнів.* Дотримання єдиних розумних норм оцінок з кожного предмета, подолання крайнощів в оцінюванні — надмірної вимогливості або поблажливості.

*— оптимізація контролю успішності учнів.* Розробка методики контролю, яка б передбачала мінімальні затрати зусиль і часу педагогів та учнів для отримання обов'язкових відомостей, запобігала переобтяженню їх виконанням зайвих завдань.

*— гласність контролю*. Повідомлення учневі результатів перевірки рівня його знань, обґрунтування виставленої йому оцінки, позитивного і недоліків у його відповіді.

*— всебічність контролю*. Перевірка та оцінювання теоретичних знань, умінь та навичок, застосування їх на практиці, навичок самостійної роботи учня.

Тематична спрямованість контролю: здійснюючи опитування, проводячи контрольні роботи, вчитель повинен визначити, який саме розділ програми, тема, вид знань, умінь та навичок оцінюються.

*— дотримання етичних норм*. Сюди входить:

а) віра педагога в можливість учнів навчатися (вміння переконати їх у тому, що вони здатні навчатися); розповідь про можливі труднощі перед вивченням матеріалу, висловлення сподівання, що вони будуть подолані; оптимістичне ставлення до навчальних успіхів і невдач учнів;

б) педагогічний такт: доброзичливість і делікатність у ставленні до учнів; відчуття міри в заохоченні й покаранні; вміння визнати свої помилки.

Оцінювання у цілому має ґрунтуватися на позитивному принципі, що передусім передбачає врахування рівня досягнень учня, а не ступеня його невдач.

Електронний Журнал, успішніше вирішує ряд завдань, що стоять перед вчителями і класними керівниками, чим традиційний паперовий. Оскільки зараз спостерігається зростання реалізації інформаційних технології в питаннях автоматизації проблем управління освітою, те створення електронного журналу успішності є актуальним завданням.

Кажучи про плюси електронного журналу, при його введенні виникають обставини, які ставлять нові завдання: необхідність навчання технології роботи з електронним журналом; необхідність доступних технічних засобів: вільний комп'ютер, працююча мережа; скільки б не було комп'ютерів в школі, їх недолік або будь-які технічні причини завжди називатимуться як головна проблема роботи з журналом, хоча єдиний паперовий журнал, безумовно, менш доступний; необхідність технічних засобів: комп'ютери (персональні і сервери), програмне забезпечення, мережі; необхідність технічного обслуговування: люди, інструменти, запчастини.

## 1.2 Аналіз інформаційного забезпечення предметної області

Ідея створити альтернативу паперовому шкільному журналу та зберігати дані про уроки, оцінки в комп'ютері не нова. Його використання полегшило б доступ до інформації про поточну успішність учнів, дозволило виконувати різні статистичні запити для наступного аналізу діяльності на педагогічній раді. Роботи зі створення електронних журналів ведуться як професійними компаніями, так і вчителями-ентузіастами.

Для своїх перших електронних журналів, використовувались електронні таблиці MS Excel. Такий журнал просто створювати та заповнювати, його структура й зовнішній вигляд нагадують свого паперового побратима. Разом із тим головним недоліком такого способу є орієнтованість електронних таблиць на одного користувача. Такий журнал не можуть одночасно заповнювати кілька вчителів. Дані з нього важко викласти у відкритий доступ, тому що інформація в електронних таблицях погано структурована, що ускладнює одержання різної статистичної інформації.

Є велика кількість професійних продуктів, як українських так і закордоних наприклад які поділяються на веб-сервіси для роботи з якими окрім комп’ютера потрібний доступ до мережі інтернет це такі як:

«МІОП» *—* (Міжнародний інформаційно-освітній портал) це інформаційне середовище, що бурхливо розвивається, в якому користувачі дістають цілодобовий і швидкий доступ до необхідних, постійно оновлюваних даних (дані про свій учбовий заклад; діючі підходів в системах освіти; програм і методів навчання з різних предметів; дані про успішність та відвідуваність потрібних учнів)



Рисунок 1.1 — Інтерфейс «МІОП»

«My-School.com.ua» *—* організація та ведення електронних журналів успішності. Будь-який навчальний заклад або окремий викладач можуть створити і вести в Системі будь-яку кількість електронних журналів для різних навчальних груп (класів), використовувати різні сістемми оцінювання (5-ти, 12-ти або 100 бальну), самостійно визначати види оцінок (поточні, атестаційні, підсумкові, екзаменаційні і т.д.), враховувати пропуски занять студентами (учнями) з поважної і без поважної причини, використовувати зведені відомості, угруповання оцінок з предметів або студентам (учням), фільтри оцінок і багато іншого.



Рисунок 1.2 — Інтерфейс «My-School.com.ua»

На даний момент «My-School.com.ua» є не дуже стабільним сервісом через низьку увагу до нього розробників та керівників, також на даний момент є непрацюючим.

«Щоденник» — новий перпекстивний сервіс з приємним, продуманим інтерфейсом та багатьма, як корисними так і не дуже функціями.

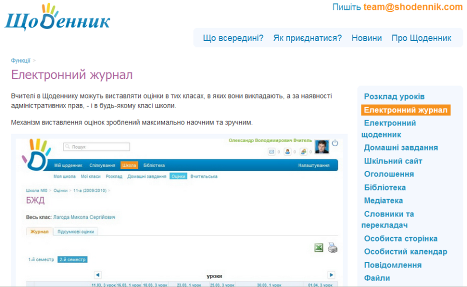


Рисунок 1.3 — Інтерфейс «Щоденник»

Основні функції сервісу «Щоденник»:

розклад уроків, шкільний сайт, словники та перекладач, файли, електронний журнал, оголошення, особиста сторінка, групи та події, електронний щоденник, бібліотека, мій календар.

«Електронна Школа» є новою і включає в себе комплекс послуг і рішень в галузі інформаційних технологій з контролю доступу відвідувачів до навчального закладу, а також з можливістю віддаленого перегляду журналу подій входу / виходу та відстеження успішності учнів.

Сервіс включає в себе кілька напрямків: контроль відвідуваності; збір та висновок оцінок на сайті, по електронній пошті повідомлення про події у вигляді sms повідомлень; статистика успішності; сайт для навчального закладу; встановлення інформаційних кіосків для учнів і вчителів;



Рисунок 1.4 — Інтерфейс «Електронна Школа»

Система «Електронна школа» не є програмою, яку необхідно встановлювати на персональні комп'ютери для перегляду або занесення оцінок, цим виключається можливість шкідливого програмного забезпечення, проблеми пов'язані з встановленням та налаштуванням програми. Доступ до системи можна отримати через будь-який комп'ютер, що має вихід в Інтернет.

Суттєвим недоліком вищеперечислених сервісів з функціями електроних журналів є те, що для роботи з ними потрібний обовязковий доступ до мережі інтернет, що далеко не у кожній школі є можливим, і до того всього такі сервіси є або комерційними проектами, або розробкою ентузіастів які не завжди працюють.

Також є декілька стаціонарних програм для ведення обліку успішності школярів:

Методичний посібник «Електронний журнал. На допомогу класному керівнику» до складу якого входить програми:

«Табель успішності» — може миттєво створити табель успішності потрібного учня, вам лишається тільки його пустити на друк;

«Сітка оцінок» — полегшить процес заповнення додатка до свідоцтва чи атестата, покаже оцінки та інформацію про окремого учня;

«Щоденник класного керівника» — необхідна для ведення документації класного керівника: планування, діагностика, відомості про учнів і ін.

Ще одина потужна програма, яка за наявності компютера може перенести всю паперову роботу пов’язану з обліком упішності на комп’ютер це «3Т:ХроноГраф Журнал».

«3Т: хронограф Журнал» Програмний комплекс інтегрується в адміністративну систему управління освітніми установами "1С: хронограф Школа 2.5 ПРОФ", поставлену в усі школи Росії в рамках пріоритетного національного проекту «Освіта» у складі СБППО "Перша допомога 1.0". Також «3Т: хронограф Журнал» входить до складу Програмно-технологічного комплексу «Електронна вчительська», що зайняв перше місце в номінації «Кращий проект для освіти» на конкурсі «Кращі 10 ІТ-проектів для держсектора» 2009 року. Програмний комплекс "3Т: хронограф Журнал" включає в себе:

— *програма «Журнал»*, що входить до складу програмного комплексу, являє собою персональний інструментарій вчителя-предметника, призначений для оперативного збору, відображення та аналізу результатів навчальної діяльності учня, включаючи успішність і відвідуваність, а також аналізу виконання навчального плану та тематичного планування. Програма «Журнал» може бути запущена практично на всіх видах комп'ютерної техніки (ПК, ноутбуках, нетбуках і КПК), також для зручності роботи, може бути розміщена на картах флеш-пам'яті. — *серверний компонент*, що виробляє збір та зберігання даних. Модуль шкільного щоденника, що входить до складу програмного комплексу формує Web сторінки електронних щоденників учнів відповідно до чинного законодавства про захист персональних даних. — *модуль синхронізації даних з програмою* "1С: хронограф Школа 2.5 ПРОФ".

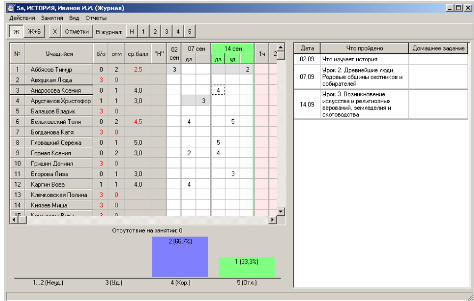


Рисунок 1.5 — Інтерфейс «3Т:ХроноГраф Журнал»

Дуже цікава і потужна програма для обліку успішності, але є платною і використовується на даний момент у деяких школах Росії.

На сучасному просторі освіти представлена достатня кількість різних реалізацій електронних журналів вчителя і щоденників учня. Відмінною особливістю таких рішень є те, що вони виконуються на web технологіях і, як правило, вимагають обов'язкової зв'язку комп'ютера вчителя зі шкільним сервером або сервером, розташованим в Інтернет, у момент безпосередньої роботи: вказівки тим уроків, фіксації пропусків, виставлення оцінок і т. д. На жаль, далеко не кожна звичайна відчизняна школа може забезпечити як створення робочих місць кожному викладачеві, так і підключення їх до комп'ютерної мережі, тому більш доречно створювати стаціонарні програмні продукти з підтримкою мережевих технологій.

## 1.3 Постановка задачі

Будь-яка освітня установа стикається з великим потоком документообігу. Вчителі щодня виконують десятки маніпуляцій по відбору потрібної інформації, заповнюють класний журнал.

На сьогоднішній день рівень інформаційних технологій достатньо високий, і все більша кількість електронної документації набуває офіційний статус, замінюючи фізичні джерела. Таким чином, інформаційні технології дозволяють реалізувати систему контролю успішності, автоматизуючи роботу з класним журналом і розвантажуючи при цьому, роботу вчителів.

При роботі з класним журналом, в обов'язки класного керівника входить, заповнення наступної інформації:

прізвище, ім'я, по батькові учнів (прізвище ім'я на кожній сторінці в алфавітному порядку);

відомості про батьків учнів;

місце проживання учнів;

дата народження учнів;

відомості про здоров'я учнів;

номер класу;

прізвище, ім'я, по батькові вчителів;

найменування предметів;

відомості про заняття у кружках (клубах, секціях);

відомості про заняття у факультативах;

інформація про кількість пропущених уроків;

У обов'язки вчителя входить:

перевіряти і оцінювати знання учнів;

відзначати відвідуваність;

виставляти відмітки за кожну чверть і за рік по своєму предмету;

записувати тему, вивчену на уроці і завдання додому.

Всі перераховані дії класного керівника і вчителів можуть бути автоматизовані при використанні електронного журналу. Оскільки електронний журнал - це інформаційна система, то список учнів досить заповнити один раз, звільнивши тим самим класного керівника від рутинної роботи, заповнювати список учнів на кожній сторінці журналу. Заповнивши всі дані про учнів один раз, не потрібно буде заповнювати їх щорічно, а лише доповнювати, або виправляти при необхідності. Основною перевагою електронного журналу є автоматизований процес виставляння четвертних і річних відміток, складання відомостей відвідуваності і успішності що вчаться, а так само відомостей про те які конкретно уроки були пропущені.

При розробці електронного журналу хотілося зробити його максимально простим у використанні, зробити так, щоб ним могли користуватись усі вчителів (в ідеалі весь педагогічний колектив), також інформація, збережена в журналі, повинна бути захищена від несанкціонованого доступу.

Будь-яку інформацію, введену в електронний журнал, у подальшому можна змінити. Зроблено це для того, щоб учитель міг виправити допущені помилки чи змінити оцінку учня, наприклад, при перездачі. Усі розуміють, що рано чи пізно освіта прийде до використання таких сервісів, і вони стануть настільки ж природними, як електронні підручники чи мобільні телефони.

Від учителя потрібні невелика підготовка й самоорганізація, але зусилля на заповнення ще одного журналу окупаються, коли вчитель розуміє, що він одержує в підсумку зручний інструмент для роботи. Використання сервісу в жодному разі не повинно насаджуватися адміністративним шляхом. Дуже важливо, що проект може стати своєрідним інструментом для підвищення інформаційної культури вчителів, стати свого роду локомотивом, що витягає освітню установу на новий рівень розвитку інформатизації.

# 2. Проектування структури інформаційної системи

## 

## 2.1 Аналіз та автоматизація обробки інформаційних потоків

Під час використання програми персоналом відбувається інтенсивна та постійна взаємодію проекту з базою даних. В такому процесі проводиться майже непереривний обмін інформацією між модулями. На рисунку 2.1 представлена схема інформаційних потоків у проекті.

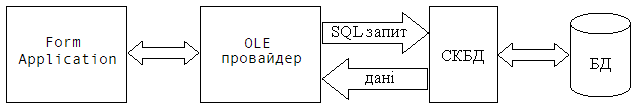


Рисунок 2.1 – Схема інформаційних потоків

Користувач взаємодіючи з інтерфейсом програми взаємодіє з базою даних і за допомогою SQL запитів відбувається керування передачею даних між OLE провайдером та базою даних. OLE провайдер забезпечує обмін даними між СКБД та елементами інтерфейсу проекту. Такий спосіб забезпечує зручну роботу користувача з базою даних та ефективну роботу самої бази.

## 2.2 Розробка структури інформаційної системи

### 

### 2.2.1 Visual Studio .NET 2008

Можна сказати, що життєвий шлях платформи. NET по-справжньому почався з появою на ринку комплексу засобів розробки нового покоління Visual Studio. NET. Його офіційне представлення і продаж перших коробкових версій відбулися в лютому 2002 р. При цьому бажаючі могли отримати бета-версії чудового продукту ще раніше. Важливість Visual Studio. NET для всієї концепції Microsoft. NET зрозуміла: успішне просування платформи безпосередньо залежить від наявності широкого кола прикладних програм, що дозволяють працювати з. NET-технологією. Навіть якщо ви вже працювали з якою-небудь середовищем розробки від компанії Microsoft, то будете приємно здивовані появою нових можливостей в середовищі Visual Studio. NET. Якщо ні, то вам належить багато що освоїти. Нова середовище містить величезний набір «корисностей», покликаних Visual Studio 2008 містить нові можливості, такі як наочні засоби розробки для більш швидкої роботи з. NET Framework 3.5, вдосконалені засоби розробки веб-додатків і поліпшення мов, що прискорюють роботу з усіма типами даних. У Visual Studio 2008 є всі необхідні засоби для створення яскравих веб-додатків з підтримкою AJAX.

Можливість скористатися потужними клієнтськими і серверними платформами для створення клієнтських веб-додатків, що виконуються у всіх сучасних веб-браузерах, легко інтегрується з серверними компонентами і мають повний доступ до служб додатків ASP.NET і платформі Microsoft.

За допомогою засобів, включених до складу Visual Studio 2008, можна створювати додатки для широкого спектру платформ - від так званих розумних пристроїв (smart personal objects) до стільникових телефонів, планшетних комп'ютерів і настільних комп'ютерів і серверів.

Для більш швидкої розробки в Visual Studio 2008 поліпшені функції роботи з мовами і даними. Наприклад, за допомогою LINQ ви, як індивідуальні розробники можете створювати рішення, що виконують аналіз даних.

У Visual Studio 2008 можливо орієнтуватися на різні версії. NET Framework в одній і тій же середовищі розробки. Можна створювати додатки для. NET Framework 2.0, 3.0 або 3.5, тобто підтримувати безліч проектів в одному середовищі.

Visual Studio 2008 містить нові засоби розробки, які полегшують створення підключення додатків на найсучасніших платформах, включаючи веб-додатки, додатки для Windows Vista, Office 2007, SQL Server 2008 і Windows Server 2008. В області веб ASP.NET AJAX та інші нові технології Visual Studio 2008 дозволять швидко створювати високоефективні інтерактивні веб-інтерфейси. Платформа. NET забезпечує швидке створення підключення додатків з зручним призначеним для користувача інтерфейсом завдяки набору готових програмних модулів для вирішення типових завдань програмування. Підключаються програми, побудовані на платформі. NET Framework, забезпечує моделювання бізнес-процесів та інтеграцію систем в різні середовища. Разом Visual Studio і. NET Framework знижують потребу в написанні загального сполучного коду, скорочуючи час розробки і дозволяючи зосередити зусилля на вирішенні бізнес-завдань.

**Мова програмування С #**

Простота, об'єктна орієнтованість, типова захищеність, «прибирання сміття», підтримка сумісності версій і багато іншого ось основні переваги C#. Дані можливості дозволяють швидко і легко розробляти програми.

При створенні С# її автори враховували досягнення багатьох інших мов програмування: C + +, С, Java, Visual Basic і т.д. Потрібно відзначити, що оскільки С# розроблявся що називається «з нуля», у його авторів була можливість не переносити в нього всі невдалі особливості будь-якої з попередніх мов. Особливо це стосується проблеми сумсності з попередніми версіями. В результаті вийша дійсно проста, зручна і сучасна мова, яка за потужністю не поступається C ++, але істотно підвищує продуктивність розробок.

Зважаючи на високу об'єктної орієнтованості, мову С # чудово підходить для швидкого конструювання різних компонентів - від високорівневої бізнес-логіки до системних додатків, що використовують низькорівневий код.

Андерс Хіджісберг, який очолив у Microsoft роботу зі створення мови С #, наступним чином визначив цілі які стояли перед розробникми: • створити першу компонентно-орієнтовану мову програмування сімейства C/C++; • створити об'єктно-орієнтовану мову, в якії будь-яка сутність представлялася об'єктом; • зпростити C + +, зберігши його міць і основні конструкції.

Головне нововведення пов'язане із заявленою компонентною орієнтованістю мови. Компоненти дозволяють вирішувати проблему модульної побудови додатків на новому рівні. Побудова компонентів зазвичай визначається не тільки мовою, але і платформою, на якій ця мова реалізована. Найпривабливішою платформою заведено рахувати NET оскільки вона є багатомовною, відкрита для вільного підключення нових мов, створюваних не тільки Microsoft, але й іншими фірмами. Всі мови, що включаються у платформу. NET, повинні спиратися на єдиний каркас, роль якого грає. NET Framework. Це серйозне обмеження, одночасно є й найважливішим достоїнством.

### 2.2.2 Delphi 7

У Borland Delphi 7, можна відзначити, як мінімум, три важливі тенденції.

Перша - згортання власних технологій, таких як Borland Database Engine (BDE) і SQL Links. Здається ймовірним відмову від набору компонент Quick Reports, успішно поставлялися з кількома останніми версіями Delphi.

Друга - що стала вже традиційною підтримка написання додатків для Linux.

Третя - поступовий відхід провідних засобів розробки від платформи Win32 і їх міграція в бік платформи. Net. І хоча, як сказано в прес-релізі Borland, за одну ніч нова платформа не змінить технологію розробки і поширення додатків, однак платформа Net - наступна, еволюційна ступінь Windows, прихід якої неминучий. Беручи до уваги той факт, що розробники в основній своїй масі люди інертні, а також те, що на вивчення нових технологій необхідно досить багато часу і коштів, Borland зробила ставку на необхідність еволюційного розвитку Delphi і не помилилася. Включення в цю версію підказок і попереджень компілятора, що стосуються сумісності з Microsoft. Net, можливості імпорту вузлів (assembly). Net в Delphi та експорту COM об'єктів у додатки. Net, підтверджує сказане вище. Повноцінна ж підтримка цієї багатообіцяючої платформи напевно буде реалізована в наступній версії Delphi - Delphi. Net, яку і будемо з нетерпінням чекати.

**Мова програмування Pascal**

Паскаль (англ. Pascal) - високорівнева мова програмування загального призначення. Один з найбільш відомих мов програмування, широко застосовується в промисловому програмуванні, навчанні програмуванню у вищій школі, є базою для великої кількості інших мов. Був створений Ніклаус Вірт в 1968-69 роках (опубліковано в 1970 році) після його участі в роботі комітету розробки стандарту мови Алгол-68.

Паскаль був створений як мова для навчання процедурного програмування (хоча, за словами Вірта, мову не можна вважати тільки навчальним, оскільки мова, непридатний для написання реальних програм, для навчання використовуватися не повинен). Назва мови дано на честь видатного французького математика, фізика, літератора і філософа Блеза Паскаля.

Важливим кроком у розвитку мови є поява вільних реалізацій мови Паскаль Free Pascal та GNU Pascal, які не тільки увібрали в себе риси безлічі інших діалектів мови, а й забезпечили надзвичайно широку переносимість написаних на ньому програм (наприклад GNU Pascal підтримує більше 20 різних платформ, під більше ніж 10 різними операційними системами, Free Pascal забезпечує спеціальні режими сумісності з різними поширеними діалектами мови (Turbo Pascal (повна сумісність), Delphi і іншими).

Особливостями мови є строга типізація та наявність коштів структурног (процедурного) програмування. Паскаль був одним з перших таких мов. На думку М. Вірта, мова повинна сприяти дисциплінування програмування, тому, поряд зі строгою типізацією, у Паскалі зведені до мінімуму можливі синтаксичні неоднозначності, а сам синтаксис автор постарався зробити інтуїтивно зрозумілим навіть при першому знайомстві з мовою.

### 2.2.3 Бази даних

Система керування базами даних MS Access

Основні переваги:

Використання Access доцільне для зберігання й пошуку даних, подання інформації в зручному виді й автоматизації виконання повторюваних завдань розробки простих і зручних форм уведення даних, обробки даних і генерації складних звітів.

Застосування MS Access для розміщення форм Access у вигляді документів HTML на Web-Сторінках і обміну даними з вузлами Internet/Intranet.

Підтримка в Access механізму Запиту за зразком (Query by Example - QBE), що забезпечує вибірку, сортування й пошук даних.

Створення додатків без програмування з використанням макросів Access.

Access - це, насамперед, система керування базами даних (СКБД). Як і інші продукти цієї категорії, вона призначена для зберігання й пошуку даних, подання інформації в зручному виді й автоматизації часто повторюваних операцій (таких, як ведення рахунків, облік, планування й т.п.). За допомогою Access можна розробляти прості й зручні форми введення даних, а також здійснювати обробку даних і генерацію складних звітів.

Access – потужний додаток Windows. Вперше продуктивність СКБД органічно сполучається з тими зручностями, які доступні користувачам Microsoft Windows. Оскільки обидва ці продукти – розроблені компанією Microsoft, вони прекрасно взаємодіють між собою. Система Access працює під керуванням систем сімейства Windows, тому при роботі з нею користувачеві доступні всі переваги Windows. Можна вирізати, копіювати й вставляти дані з будь-якого додатка Windows в Access і навпаки; можна створити проект форми в Access і вставити його в конструктор форм.[6]

Система Access - це набір інструментів кінцевого користувача для керування базами даних. У її склад входять конструктори таблиць, форм, запитів і звітів. Цю систему можна розглядати і як середовище розробки додатків. Використовуючи макроси або модулі для автоматизації рішення завдань, можна створювати орієнтовані на користувача додатки такої ж потужності, як і додатки, написані безпосередньо на мовах програмування. При цьому їхній інтерфейс буде включати кнопки, меню й діалогові вікна. Програмуючи мовою VBA, можна створювати такі потужні програми, як сама система Access. По суті, багато інструментальних засобів Access (наприклад, майстри й конструктори) написані саме на VBA.

Засоби які пропонує Access:

Справжня реляційна модель баз даних

Контекстно-контекстно-залежна довідка й Office Assistant

Прості у використанні майстра й конструктори

Імпортування, експортування й зв'язування зовнішніх файлів

Форми й звіти WYSIWYG

Графіки й діаграми

Можливості DDE і OLE

Доступ до Internet

Убудовані функції

Макроси: програмування без програмування

Модулі: Visual Basic for Applications програмування баз даних

Рекомендації для користувачів

Інформація для користувачів електронних таблиць

MySQL — вільна система керування реляційними базами даних.

Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL — компактний сервер баз даних. Характеризується великою швидкістю, стійкістю і простотою використання. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. MySQL вважається гарним рішенням для малих і середніх застосунків. Вихідні коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNІХ-системах. Для некомерційного використання MySQL є безкоштовним. Можливості сервера MySQL:

- простота у встановленні та використанні;

- підтримується необмежена кількість користувачів, що одночасно працюють із БД;

- кількість рядків у таблицях може досягати 50 млн.;

- висока швидкість виконання команд;

- наявність простої і ефективної системи безпеки.

MySQL виникла як спроба застосувати mSQL до власних розробок компанії: таблицям, для яких використовувалися ISAM — підпрограми низького рівня. У результаті був вироблений новий SQL-інтерфейс, але API-інтерфейс залишився в спадок від mSQL. Звідки походить назва «MySQL» — достеменно не відомо. Розробники дають два варіанти: або тому, що практично всі напрацювання компанії починалися з префікса My, або на честь дівчинки на ім'я My, дочки Майкла Монті Віденіуса, одного з розробників системи. Логотип MySQL у вигляді дельфіна носить ім'я «Sakila». Він був обраний з великого списку запропонованих користувачами «імен дельфіна». Ім'я «Sakila» було відправлено Open Source-розробником Ambrose Twebaze.

В січні-лютому 2008 Sun Microsystems придбала виробника системи керування базами даних MySQL за $1 млрд. Після поглинання у 2009 році Sun Microsystems з боку Oracle Corporation MySQL стала власністю Oracle.

Основні переваги пакету MySQL:

1) багато поточність;

2) підтримка декількох одночасних запитів;

3) оптимізація зв'язків з приєднанням багатьох даних за один прохід;

4) записи фіксованої і змінної довжини;

5) ODBC драйвер в комплекті;

6) гнучка система привілеїв і паролів;

7) до 16 ключів в таблиці, кожен ключ може мати до 15 полів;

8) основана на потоках, швидка система пам'яті;

9) всі операції роботи з рядками не звертають уваги на регістр символів в оброблюваних рядках;

10) псевдоніми застосовні як до таблиць, так і до окремих стовпців в таблиці;

11) всі поля мають значення за замовчуванням;

12) легкість управління таблицею, включаючи додавання і видалення ключів і полів.

Реляційна система керування базами даних Microsoft SQL Server.

Microsoft SQL Server — реляційна система керування базами даних (СКБД), розроблена корпорацією Microsoft. Основна використовувана мова запитів — Transact-SQL, створений спільно Microsoft й Sybase. Transact-SQL являється реалізацією стандарту ANSI/ISO по структурованій мові запитів (SQL) с розширеннями. Використається для невеликих і середніх по розміру баз даних, і в останні 5 років - для великих баз даних масштабів підприємства, конкурує з іншими СКБД у цьому сегменті ринку.

Вихідний код MS SQL Server (до версії 7.0) ґрунтувався на коді Sybase SQL Server, і це дозволило Microsoft вийти на ринок баз даних для підприємств, де конкурували Oracle, IBM, і, пізніше, сама Sybase. Microsoft, Sybase й Ashton-Tate спочатку об'єдналися для створення й випуску на ринок першої версії програми, що одержала назва SQL Server 1.0 для OS/2, яка фактично була еквівалентом Sybase SQL Server 3.0 для Unix, VMS і ін. Microsoft SQL Server 4.2 був випущений в 1992 році і входив до складу операційної системи Microsoft OS/2 версії 1.3. Офіційний реліз Microsoft SQL Server версії 4.21 для ОС Windows NT відбувся одночасно з релізом самої Windows NT (версії 3.1). Microsoft SQL Server 6.0 був першою версією SQL Server, створеної винятково для архітектури NT і без участі в процесі розробки Sybase.

Поточна версія — Microsoft SQL Server 2005 — була представлена в листопаді 2005 року. Запуск версії відбувався паралельно запуску Visual Studio 2005. Існує також «врізана» версія Microsoft SQL Server - Microsoft SQL Server Express; вона доступна й вільно поширюється з її програмним забезпеченням.

З моменту випуску попередньої версії SQL Server (SQL Server 2000) був здійснений розвиток інтегрованого середовища розробки й ряду додаткових підсистем, що входять до складу SQL Server 2005. Зміни торкнулися реалізації технології ETL, що входить до складу компонента SQL Server Integration Services (SSIS), сервера оповіщення, засобів аналітичної обробки багатомірних моделей даних (OLAP) і збору релевантної інформації (обидві служби входять до складу Microsoft Analysis Services), а також декількох служб повідомлень, а саме Service Broker й Notification Services. Крім цього, зроблені поліпшення в продуктивності.

Функціональність

Microsoft SQL Server як мова запитів використовує версію SQL, яка отримала назву Transact-SQL (скорочено T-SQL), що є реалізацією SQL-92 (стандарт ISO для SQL) із множинними розширеннями. T-SQL дозволяє використати додатковий синтаксис для процедур збереження і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим додатком). Microsoft SQL Server й Sybase ASE для взаємодії з мережею використовують протокол рівня додатка за назвою Tabular Data Stream (TDS, протокол передачі табличних даних). Протокол TDS також був реалізований у проекті FreeTDS з метою забезпечити різним додаткам можливість взаємодії з базами даних Microsoft SQL Server й Sybase.

SQL Server підтримує надлишкове дублювання даних по трьох сценаріях:

- знімок – виробляється «знімок» бази даних, що сервер відправляє одержувачам;

- історія змін – всі зміни бази даних безупинно передаються користувачам.

- синхронізація з іншими серверами – бази даних декількох серверів синхронізуються між собою.

Зміни всіх баз даних відбуваються незалежно один від одного на кожному сервері, а при синхронізації відбувається звірення даних. Даний тип дублювання передбачає можливість розв’язання протиріч між БД.

В SQL Server 2005 вбудована підтримка .NET Framework. Завдяки цьому, збережені процедури БД можуть бути написані на будь-якій мові платформи .NET, використовуючи повний набір бібліотек, доступних для .NET Framework, включаючи Common Type System (система обігу з типами даних в Microsoft .NET Framework). Однак, на відміну від інших процесів, .NET Framework, будучи базисною системою для SQL Server 2005, виділяє додаткову пам'ять і вибудовує засоби керування SQL Server замість того, щоб використовувати вбудовані засоби Windows. Це підвищує продуктивність у порівнянні із загальними алгоритмами Windows, тому що алгоритми розподілу ресурсів спеціально налаштовані для використання в структурах SQL Server.

## 2.3 Вибір засобів розробки інформаційної системи

Розробку журналу успішності системи планується проводити з використанням сучасних засобів розробки програмного забезпечення для мережевих інформаційних систем. Вибрані інструментальні засоби являють собою універсальне інструментальне програмне забезпечення, за допомогою якого можна розробляти різні за складністю та функціональним призначенням інформаційні системи.

Переваги C# над Delphi:

простіше працювати з динамічною пам'яттю;

більше розвинене ООП в порівнянні з процедурним;

сама платформа .NET більше орієнтована на C# ніж на Delphi.

Оскільки вибрано C#.NET немає особливої різниці у виборі ситеми керування базами даних тому вибрано MS Access через зручний інтуєтивно зрозумілий інтерфейс для створення та редагування бази даних на початку проектування проекту.

# 3. Програмна реалізація

## 

## 3.1 Структура і функціональне призначення модулів системи, їх взаємозв’язок

За своєю структурою програмне забезпечення для ведення успішності школярів складається з чотирьох блоків. Кожен із блоків має своє чітко визначене призначення, яке полягає в реалізації певної функцій системи.

Перелі модулів:

1) uchni.cs – модуль, який відповідає за роботу форми відображення усіх учнів школи;

2) uchеn.cs – модуль, який призначений для роботи форми відображення успішності певного учня;

3) edit.cs – модуль який відповідає за редагування даних;

4) zhu.cs – модуль в якому описані деякі функції для роботи з даними.

Модуль edit.cs являє собою основний функціональний модуль, який включає процедури та функції, які відповідають за:

- редагування даних успішності (додання, видалення);

- редагування предметів (додання, видалення);

- редагування оцінок успішності;

- зміна окремих даних;

- відбір за заданими параметрами;

Модуль edit.cs являє собою основний функціональний модуль, який включає процедури та функції, які відповідають за:

- редагування даних успішності (додання, видалення);

- редагування предметів (додання, видалення);

- редагування оцінок успішності;

- зміна окремих даних;

- відбір за заданими параметрами;

Допоміжний функціональний модуль zhu.cs являє включає процедури та функції, які відповідають за:

- заповнення даними головних таблиць;

- заповнення даними випадаючих списків;

- коректне відображення даних.

Модуль uchni.cs створений для відображення всіх учнів школи з можливістю відбору учнів які навчаються в певному вказаному класі, також для фомування вхідних даних модуля uchеn.cs який в свою чергу керує формою, яка відображає оцінки певного учня.

Програмний продукт

uchni.cs – модуль головної форми

edit.cs – модуль для редагування

uchеn.cs – модуль конкретного учня

zhu.cs – модуль з допоміжними функціями

Рисунок 3.1 – Модулі програмного продукту

## 3.2 Розробка програмних модулів

При розробці структури даний програмний продукт був поділений на модулі Для того, щоб програма відповідала поставленим вимогам, були розроблені окремі блоки для реалізації завдань.

Блок завантаження головної форми який знаходиться у файлі ініціалізації програми:

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new uchni());

}

Підключення бази даних яке знаходиться у кожному модулі:

OleDbConnection connection = new System.Data.OleDb.OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=dbase.mdb");

Подія завантаження форми при якій відбувається завантаження даних з бази у DataGridViev при запуску програми:

private void Form2\_Load(object sender, EventArgs e)

{

// Uchni();

zaput1 = "SELECT uchni.fkuchni, uchni.fkklasu, klasu.klas AS [Клас], uchni.prizv AS [Прізвище], uchni.imya AS [Ім'я], uchni.pobat AS [Побатькові] FROM (uchni INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) WHERE (klasu.klas = " + strkl + ") ORDER BY uchni.prizv";

ss.MainSel(dataGridView1, zaput1);

ss.VisiF(dataGridView1, 1);

ss.combSel(combklasu, "SELECT klas, fkklasu, rikpost FROM klasu ORDER BY rikpost;",false);

}

Блок відбору однокласників за заданим класом:

if (combklasu.SelectedItem.ToString() == "\*") strkl = "klasu.klas";

else

strkl = "'" + combklasu.SelectedItem.ToString() + "'";

zaput1 = "SELECT uchni.fkuchni, uchni.fkklasu, klasu.klas AS [Клас], uchni.prizv AS [Прізвище], uchni.imya AS [Ім'я], uchni.pobat AS [Побатькові] FROM (uchni INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) WHERE (klasu.klas = " + strkl + ") ORDER BY uchni.prizv";

ss.MainSel(dataGridView1, zaput1);

ss.VisiF(dataGridView1, 1);

Запуск форми на якій відображається успішність вибраного учня з переданням параметрів для його вибору:

uchen fo3 = new uchen(dataGridView1.CurrentRow.Cells[0].Value.ToString(),"ІДЗ");

fo3.Show();

Запуск форми редагування з передачею відомостей про те який клас потрібно відкрити:

edit ed = new edit(strkl);

ed.Show();

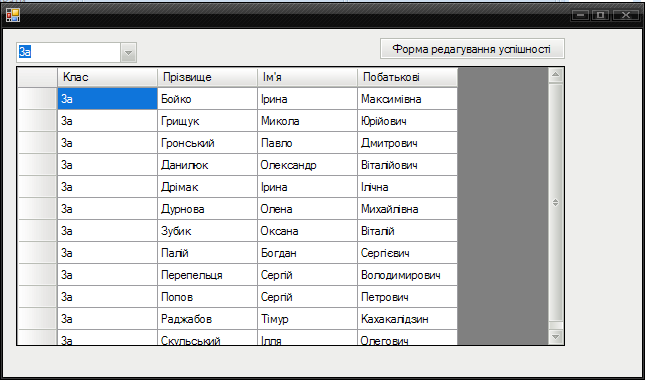


Рисунок 3.2 – Форма головної сторінки.

Блок інізіалізації форми що відображає інформацію про учня з прийомом даних які передаються з іншої форми:

public uchen(string idUhc,string tup)

{

fkUch = idUhc;

zaput1 = "SELECT idz.ocinka AS [Оцінка], idznazva.nazva AS [Назва ІДЗ], predmetu.nazva AS [Предмет] FROM (((predmetu INNER JOIN idz ON predmetu.fkpredm = idz.idzfkpre) INNER JOIN((uchni INNER JOIN semestru ON uchni.fkuchni = semestru.semfkuchen) INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) ON idz.idzfksem = semestru.fksem) INNER JOIN idznazva ON idz.nazva = idznazva.fkidznazva) WHERE (uchni.fkuchni = " + fkUch + ")";

InitializeComponent();

}

Процедура події яка відповідає за завантаження форми з блоком заповнення даними учня таблиці DataGridViev:

private void Form3\_Load(object sender, EventArgs e)

{

zhusl.MainSel(dataGridView1, zaput1);

connection.Open();

OleDbCommand thisCommandp = new OleDbCommand("SELECT uchni.prizv FROM (uchni INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) WHERE (uchni.fkuchni = " + fkUch + ")", connection);

OleDbDataReader thisReader = thisCommandp.ExecuteReader();

thisReader.ToString();

thisReader.Read();

ipp.Text += thisReader[0].ToString()+" ";

thisCommandp = new OleDbCommand("SELECT uchni.imya FROM (uchni INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) WHERE (uchni.fkuchni = " + fkUch + ")", connection);

thisReader = thisCommandp.ExecuteReader();

thisReader.ToString();

thisReader.Read();

ipp.Text += thisReader[0].ToString() + " ";

thisCommandp = new OleDbCommand("SELECT uchni.pobat FROM (uchni INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) WHERE (uchni.fkuchni = " + fkUch + ")", connection);

thisReader = thisCommandp.ExecuteReader();

thisReader.ToString();

thisReader.Read();

ipp.Text += thisReader[0].ToString() + " ";

thisReader.Close();

connection.Close();

}

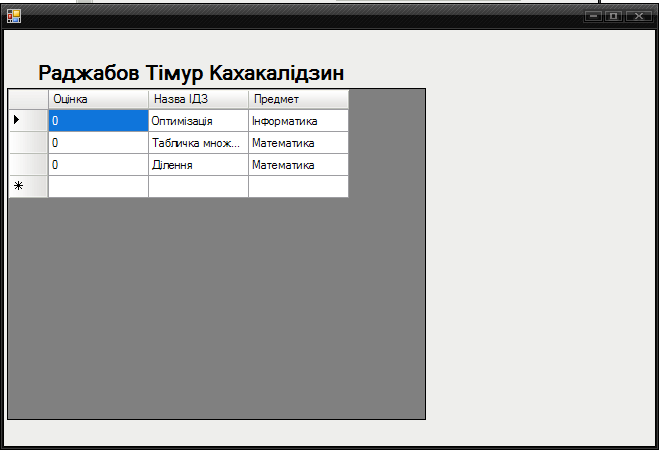


Рисунок 3.3 – Форма яка відображає успішність вибраного учня.

Прийом даних які передає головна форма – формі редагування:

public edit(string klos)

{

strkl = klos;

InitializeComponent();

}

Зміні необхідні для вибірки даних:

public static string strkl = "klasu.klas", strpredm = "predmetu.nazva", stridznazva = "idznazva.nazva";

Заповнення усіх об’єктів на формі даними:

zhusl.combSel(combklasu, "SELECT klas, fkklasu, rikpost FROM klasu ORDER BY rikpost", false);

zhusl.combSel(combpredm, "SELECT nazva FROM predmetu ORDER BY nazva", true);

zhusl.combSel(combidz, "SELECT nazva FROM idznazva ORDER BY nazva", true);

combpredm.SelectedIndex = 0;

//combklasu.SelectedIndex = 11;

// combimya.SelectedIndex = 0;

//FormLoad

zhusl.MainSel(dataGridView1, "SELECT idz.fkidz AS [FKIDZ], idz.idzfksem AS [FKSEMIDZ], idz.idzfkpre, idz.nazva AS idznazva, klasu.fkklasu, uchni.fkuchni, klasu.klas AS [Клас], uchni.prizv AS [Прізвище], uchni.imya AS [Ім'я], uchni.pobat AS [Побатькові], idznazva.nazva AS [Назва ІДЗ], idz.ocinka AS [Оцінка], predmetu.nazva AS [Предмет] FROM (predmetu INNER JOIN (((uchni INNER JOIN semestru ON uchni.fkuchni = semestru.semfkuchen) INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) INNER JOIN(idz INNER JOIN idznazva ON idz.nazva = idznazva.fkidznazva) ON semestru.fksem = idz.idzfksem) ON predmetu.fkpredm = idz.idzfkpre) WHERE (klasu.klas = " + strkl + " AND predmetu.nazva = " + strpredm + " AND idznazva.nazva = " + stridznazva + ") ORDER BY uchni.prizv");

zhusl.VisiF(dataGridView1, 5);

Вибірка за предметами:

if (combpredm.SelectedItem.ToString() == "\*") strpredm = "predmetu.nazva";

else strpredm = "'" + combpredm.SelectedItem.ToString() + "'";

zhusl.MainSel(dataGridView1, "SELECT idz.fkidz AS [FKIDZ], idz.idzfksem AS [FKSEMIDZ], idz.idzfkpre, idz.nazva AS idznazva, klasu.fkklasu, uchni.fkuchni, klasu.klas AS [Клас], uchni.prizv AS [Прізвище], uchni.imya AS [Ім'я], uchni.pobat AS [Побатькові], idznazva.nazva AS [Назва ІДЗ], idz.ocinka AS [Оцінка], predmetu.nazva AS [Предмет] FROM (predmetu INNER JOIN (((uchni INNER JOIN semestru ON uchni.fkuchni = semestru.semfkuchen) INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) INNER JOIN(idz INNER JOIN idznazva ON idz.nazva = idznazva.fkidznazva) ON semestru.fksem = idz.idzfksem) ON predmetu.fkpredm = idz.idzfkpre) WHERE (klasu.klas = " + strkl + " AND predmetu.nazva = " + strpredm + " AND idznazva.nazva = " + stridznazva + ") ORDER BY uchni.prizv");

zhusl.VisiF(dataGridView1, 5);

Вибірка за класами:

if (combklasu.SelectedItem.ToString() == "\*") strkl = "klasu.klas"; else

strkl = "'" +combklasu.SelectedItem.ToString()+"'";

zhusl.MainSel(dataGridView1, "SELECT idz.fkidz AS [FKIDZ], idz.idzfksem AS [FKSEMIDZ], idz.idzfkpre, idz.nazva AS idznazva, klasu.fkklasu, uchni.fkuchni, klasu.klas AS [Клас], uchni.prizv AS [Прізвище], uchni.imya AS [Ім'я], uchni.pobat AS [Побатькові], idznazva.nazva AS [Назва ІДЗ], idz.ocinka AS [Оцінка], predmetu.nazva AS [Предмет] FROM (predmetu INNER JOIN (((uchni INNER JOIN semestru ON uchni.fkuchni = semestru.semfkuchen) INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) INNER JOIN(idz INNER JOIN idznazva ON idz.nazva = idznazva.fkidznazva) ON semestru.fksem = idz.idzfksem) ON predmetu.fkpredm = idz.idzfkpre) WHERE (klasu.klas = " + strkl + " AND predmetu.nazva = " + strpredm + " AND idznazva.nazva = " + stridznazva + ") ORDER BY uchni.prizv");

zhusl.VisiF(dataGridView1, 5);

Вибірка за назвою оцінки:

if (combidz.SelectedItem.ToString() == "\*") stridznazva = "idznazva.nazva";

else

stridznazva = "'" + combidz.SelectedItem.ToString() + "'";

zhusl.MainSel(dataGridView1, "SELECT idz.fkidz AS [FKIDZ], idz.idzfksem AS [FKSEMIDZ], idz.idzfkpre, idz.nazva AS idznazva, klasu.fkklasu, uchni.fkuchni, klasu.klas AS [Клас], uchni.prizv AS [Прізвище], uchni.imya AS [Ім'я], uchni.pobat AS [Побатькові], idznazva.nazva AS [Назва ІДЗ], idz.ocinka AS [Оцінка], predmetu.nazva AS [Предмет] FROM (predmetu INNER JOIN (((uchni INNER JOIN semestru ON uchni.fkuchni = semestru.semfkuchen) INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) INNER JOIN(idz INNER JOIN idznazva ON idz.nazva = idznazva.fkidznazva) ON semestru.fksem = idz.idzfksem) ON predmetu.fkpredm = idz.idzfkpre) WHERE (klasu.klas = " + strkl + " AND predmetu.nazva = " + strpredm + " AND idznazva.nazva = " + stridznazva + ") ORDER BY uchni.prizv");

zhusl.VisiF(dataGridView1, 5);

Функція яка дістає ключове поле таблиці за назвою інших полів:

string getFK(string stre)

{

string tem = "";

OleDbCommand thisCommand = new OleDbCommand(stre, connection);

OleDbDataReader thisReader = thisCommand.ExecuteReader();

thisReader.ToString();

thisReader.Read();

tem = thisReader[0].ToString();

thisReader.Close();

return tem;

}

Динамічна зміна предмету та назви оцінки:

switch (dataGridView1.CurrentCell.ColumnIndex)

{

case 10:

zhusl.combSel(combfrog, "SELECT nazva FROM idznazva ORDER BY nazva;",false);

panel1.Visible = true;

panel1.Top = e.Y + dataGridView1.Top;

panel1.Left = e.X + dataGridView1.Left;

break;

case 12:

zhusl.combSel(combfrog, "SELECT nazva FROM predmetu ORDER BY nazva;",false);

panel1.Visible = true;

panel1.Top = e.Y + dataGridView1.Top;

panel1.Left = e.X + dataGridView1.Left

break;

default:

break;}

Процедура динамічної зміни:

private void combfrog\_SelectedIndexChanged(object sender, EventArgs e)

{

connection.Open();

OleDbCommand thisCommand = new OleDbCommand();

thisCommand.Connection = connection;

thisCommand.CommandText = "SELECT nazva FROM idznazva ORDER BY nazva;";

OleDbDataReader thisReader = thisCommand.ExecuteReader();

thisReader.Close();

switch (dataGridView1.CurrentCell.ColumnIndex)

{

case 10:

thisCommand.CommandText = "UPDATE idz SET nazva = ? WHERE (fkidz= ?);";

thisCommand.Parameters.Add("@nazva", OleDbType.VarChar).Value = getFK("SELECT fkidznazva FROM idznazva WHERE (nazva = '" + combfrog.SelectedItem.ToString() + "');");

thisCommand.Parameters.Add("@fkidz", OleDbType.VarChar).Value = dataGridView1.CurrentRow.Cells[0].Value.ToString();

thisReader = thisCommand.ExecuteReader();

thisReader.ToString();

thisReader.Close();

break;

case 12:

thisCommand.CommandText = "UPDATE idz SET idzfkpre = ? WHERE (fkidz= ?);";

thisCommand.Parameters.Add("@idzfkpre", OleDbType.VarChar).Value = getFK("SELECT fkpredm FROM predmetu WHERE (nazva = '" + combfrog.SelectedItem.ToString() + "');");

thisCommand.Parameters.Add("@fkidz", OleDbType.VarChar).Value = dataGridView1.CurrentRow.Cells[0].Value.ToString();

thisReader = thisCommand.ExecuteReader();

thisReader.ToString();

thisReader.Close();

break;

default:

break;

}

panel1.Visible = false;

connection.Close();

zhusl.MainSel(dataGridView1, "SELECT idz.fkidz AS [FKIDZ], idz.idzfksem AS [FKSEMIDZ], idz.idzfkpre, idz.nazva AS idznazva, klasu.fkklasu, uchni.fkuchni, klasu.klas AS [Клас], uchni.prizv AS [Прізвище], uchni.imya AS [Ім'я], uchni.pobat AS [Побатькові], idznazva.nazva AS [Назва ІДЗ], idz.ocinka AS [Оцінка], predmetu.nazva AS [Предмет] FROM (predmetu INNER JOIN (((uchni INNER JOIN semestru ON uchni.fkuchni = semestru.semfkuchen) INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) INNER JOIN(idz INNER JOIN idznazva ON idz.nazva = idznazva.fkidznazva) ON semestru.fksem = idz.idzfksem) ON predmetu.fkpredm = idz.idzfkpre) WHERE (klasu.klas = " + strkl + " AND predmetu.nazva = " + strpredm + " AND idznazva.nazva = " + stridznazva + ")");

zhusl.VisiF(dataGridView1, 5);

}

Процедура створення нової оцінки для динамічної процедури:

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

{

//panel1.Height = 70;

MessageBox.Show(combfrog.Text);

connection.Open();

OleDbCommand thisCommand = new OleDbCommand();

thisCommand.Connection = connection;

thisCommand.CommandText = "SELECT nazva FROM idznazva ORDER BY nazva;";

OleDbDataReader thisReader = thisCommand.ExecuteReader();

thisReader.Close();

switch (dataGridView1.CurrentCell.ColumnIndex)

{

case 10:

thisCommand.CommandText = "INSERT INTO idznazva (nazva) VALUES(@Nazva);";

thisCommand.Parameters.Add("@nazva", OleDbType.VarChar).Value = combfrog.Text;

thisReader = thisCommand.ExecuteReader();

thisReader.ToString();

thisReader.Close();

connection.Close();

zhusl.combSel(combfrog, "SELECT nazva FROM idznazva ORDER BY nazva", false);

break;

case 12:

thisCommand.CommandText = "INSERT INTO predmetu (nazva) VALUES(@Nazva);";

thisCommand.Parameters.Add("@nazva", OleDbType.VarChar).Value = combfrog.Text;

thisReader = thisCommand.ExecuteReader();

thisReader.ToString();

thisReader.Close();

connection.Close();

zhusl.combSel(combfrog, "SELECT nazva FROM idznazva ORDER BY nazva", false);

break;

default:

break;

}

}

Блок відкриття, закриття панелі для додавання нової оцінки:

if (panel2.Visible) panel2.Visible = false; else

panel2.Visible = true;

Додання нової оцінки:

OleDbCommand thisCommanduch = new OleDbCommand("SELECT uchni.fkuchni FROM (uchni INNER JOIN klasu ON uchni.fkklasu = klasu.fkklasu) WHERE (klasu.klas = " + strkl + ")", connection);

OleDbCommand thisCommandsem = new OleDbCommand();

OleDbCommand thisCommand = new OleDbCommand();

OleDbCommand thisCommandi = new OleDbCommand();

connection.Open();

thisCommandi = new OleDbCommand("INSERT INTO idznazva (nazva) VALUES (@nazva)", connection);

thisCommandi.Parameters.Add("@nazva", OleDbType.VarChar).Value = textBox1.Text;

OleDbDataReader thisReaderuch2 = thisCommandi.ExecuteReader();

OleDbDataReader thisReaderuch = thisCommanduch.ExecuteReader();

string uch = "",uch2 = "";

bool kt = false;

thisReaderuch.ToString();

string stre = "",stre2 = "";

while (thisReaderuch.Read())

{

string rea = thisReaderuch[0].ToString();

thisCommandsem = new OleDbCommand("SELECT semestru.fksem FROM (semestru INNER JOIN uchni ON semestru.semfkuchen = uchni.fkuchni) WHERE (uchni.fkuchni = " + rea + ")", connection);

OleDbDataReader thisReadersem = thisCommandsem.ExecuteReader();

thisReadersem.ToString();

while (thisReadersem.Read())

{

if ((sem1.Checked) && (uch!=rea))

{

string reasem = thisReadersem[0].ToString();

thisCommand = new OleDbCommand("INSERT INTO idz (nazva, ocinka, idzfkpre, idzfksem) VALUES (@nazva, @ocinka, @idzfkpre, @idzfksem);", connection);

thisCommand.Parameters.Add("@nazva", OleDbType.VarChar).Value = getFK("SELECT fkidznazva FROM idznazva WHERE (nazva = '" + textBox1.Text + "')");

thisCommand.Parameters.Add("@ocinka", OleDbType.VarChar).Value = "0";

thisCommand.Parameters.Add("@idzfkpre", OleDbType.VarChar).Value = getFK("SELECT fkpredm FROM predmetu WHERE (nazva = " + strpredm + ")");

thisCommand.Parameters.Add("@idzfksem", OleDbType.VarChar).Value = reasem;

OleDbDataReader thisReader = thisCommand.ExecuteReader();

uch = rea;

kt = true;

}

else { kt = false; uch = rea; }

if ((sem2.Checked) && (uch == rea) && !kt)

{

kt = false;

}

else { }

}

}

connection.Close();

textBox1.Text = "";

panel2.Visible = false;

zhusl.combSel(combidz, "SELECT nazva FROM idznazva ORDER BY nazva", true);

Відкриття, закриття панелі для додання нового предмету:

if (panel3.Visible) panel3.Visible = false; else panel3.Visible = true;

Процедура виставлення оцінок із збереженням іх у базу даних:

void updOcinku()

{

connection.Open();

if (dataGridView1.CurrentRow.Cells[11].Value.ToString() == "") dataGridView1.CurrentRow.Cells[11].Value = "0";

OleDbCommand thisCommand = new OleDbCommand("UPDATE idz SET ocinka = ? WHERE (fkidz= ?);", connection);

thisCommand.Parameters.Add("@ocinka", OleDbType.VarChar).Value = dataGridView1.CurrentRow.Cells[11].Value.ToString();

thisCommand.Parameters.Add("@fkidz", OleDbType.VarChar).Value = dataGridView1.CurrentRow.Cells[0].Value.ToString();

OleDbDataReader reader = thisCommand.ExecuteReader();

reader.Close();

connection.Close();

}

Відалення вибраної оцінки:

OleDbCommand thisCommandsem = new OleDbCommand();

OleDbCommand thisCommand = new OleDbCommand();

OleDbCommand thisCommandi = new OleDbCommand();

OleDbDataReader thisReader;

connection.Open();

thisCommandsem = new OleDbCommand("SELECT idz.fkidz FROM (idz INNER JOIN idznazva ON idz.nazva = idznazva.fkidznazva) WHERE (idznazva.nazva = " + stridznazva + ")", connection);

OleDbDataReader thisReadersem = thisCommandsem.ExecuteReader();

thisReadersem.ToString();

while (thisReadersem.Read())

{

string rea = thisReadersem[0].ToString();

thisCommand = new OleDbCommand("DELETE FROM idz WHERE (fkidz = " + rea + ")", connection);

thisReader = thisCommand.ExecuteReader();

}

thisCommandi = new OleDbCommand("DELETE FROM idznazva WHERE (nazva = " + stridznazva + ")", connection);

thisReader = thisCommandi.ExecuteReader();

connection.Close();

zhusl.combSel(combidz, "SELECT nazva FROM idznazva ORDER BY nazva", true);

combidz.SelectedIndex = 0;

Добавлення нового предмету:

connection.Open();

OleDbCommand thisCommand = new OleDbCommand("INSERT INTO predmetu (nazva) VALUES(@nazva);", connection);

thisCommand.Parameters.Add("@nazva", OleDbType.VarChar).Value = textBox2.Text;

OleDbDataReader thisReader = thisCommand.ExecuteReader();

thisReader.ToString();

connection.Close();

zhusl.combSel(combpredm, "SELECT nazva FROM predmetu ORDER BY nazva", true);

textBox2.Text = "";

panel3.Visible = false;

Видалення предмету і всіх оцінок пов’язаних з ним:

OleDbCommand thisCommandsem = new OleDbCommand();

OleDbCommand thisCommand = new OleDbCommand();

OleDbCommand thisCommandi = new OleDbCommand();

OleDbDataReader thisReader;

connection.Open();

thisCommandsem = new OleDbCommand("SELECT idz.fkidz FROM (idz INNER JOIN predmetu ON idz.idzfkpre = predmetu.fkpredm) WHERE (predmetu.nazva = " + strpredm + ")", connection);

OleDbDataReader thisReadersem = thisCommandsem.ExecuteReader();

thisReadersem.ToString();

while (thisReadersem.Read())

{

string rea = thisReadersem[0].ToString();

thisCommand = new OleDbCommand("DELETE FROM idz WHERE (fkidz = " + rea + ")", connection);

thisReader = thisCommand.ExecuteReader();

}

thisCommandi = new OleDbCommand("DELETE FROM predmetu WHERE (nazva = " + strpredm + ")", connection);

thisReader = thisCommandi.ExecuteReader();

connection.Close();

zhusl.combSel(combpredm, "SELECT nazva FROM predmetu ORDER BY nazva", true);

zhusl.combSel(combidz, "SELECT nazva FROM idznazva ORDER BY nazva", true);

combpredm.SelectedIndex = 0;

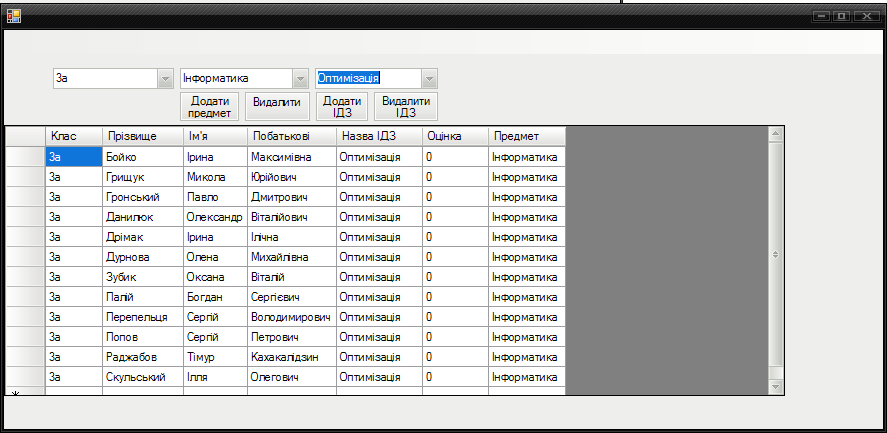


Рисунок 3.4 – Форма редагування даних.

Функція для заповнення даними об’єктів DataGridView модуля «zhu.cs»:

public void MainSel(DataGridView datta,string zaput1)

{

connection.Open();

OleDbDataAdapter adapter = new System.Data.OleDb.OleDbDataAdapter(zaput1, connection);

DataSet dSet = new DataSet();

adapter.Fill(dSet);

datta.DataSource = dSet.Tables[0];

connection.Close();

}

Функція для заповнення даними об’єктів ComboBox модуля «zhu.cs»:

public void combSel(ComboBox combi, string stre,bool resh)

{

// OleDbDataAdapter adapter;

combi.Items.Clear();

OleDbCommand thisCommand = new OleDbCommand(stre, connection);

connection.Open();

OleDbDataReader thisReader = thisCommand.ExecuteReader();

thisReader.ToString();

if (resh) combi.Items.Add("\*");

while (thisReader.Read())

combi.Items.Add(thisReader[0].ToString());

thisReader.Close();

connection.Close();

}

Функція для коректного відображеня даних в об’єктах DataGridView модуля «zhu.cs»:

public void VisiF(DataGridView dataGridParent, int county)

{

CurrencyManager cmgr = (CurrencyManager)dataGridParent.BindingContext[dataGridParent.DataSource];

cmgr.SuspendBinding();

for (int i = 0; i <= county; i++)

dataGridParent.Columns[i].Visible = false;

cmgr.ResumeBinding();

}

## 3.3 Інструкція користувача

При запуску програми з’являється форма зі списком усіх учнів школи:

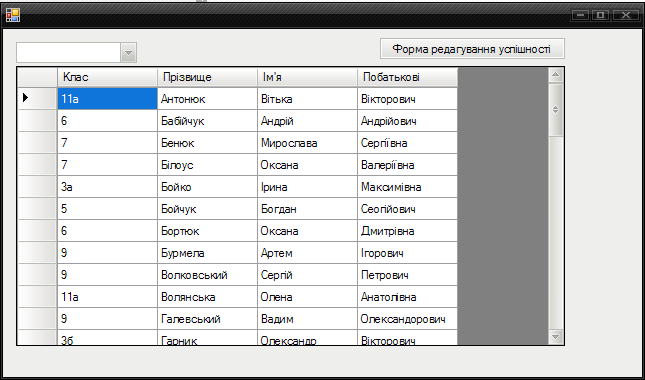


Рисунок 3.5 – Головна форма.

У ній можна вибрати клас який нас цікавить у випадаючому списку, який знаходиться у верхньому лівому куті:

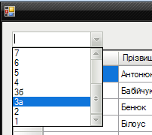


Рисунок 3.6 – Вибір класу за яким проведеться вибірка.

Якщо ж двічі клацнути по якомусь з учнів, то з’виться форма з його успішністтю:

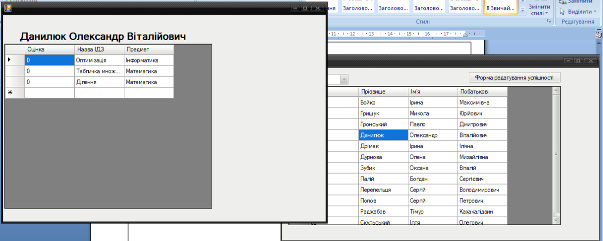


Рисунок 3.7 – Успішність вибраного учня.

Для переходу на форму редагування потрібно натиснути кнопку «Форма редагування успішності», яка знаходиться у верхньому правому куті головної форми.

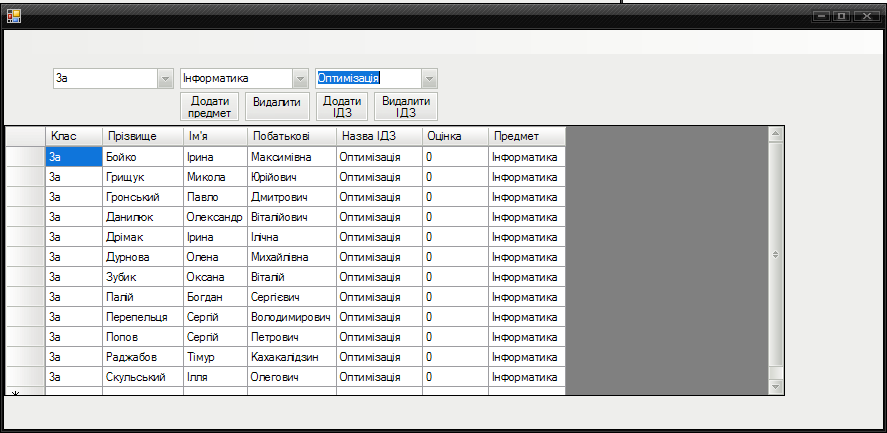


Рисунок 3.7 – Форма редагування.

На формі редагування можна додати, або видалити предмети, та оцінки з назвами тем, також можна виставляти оцінки редагуючи стовпчик оцінки. Всі змінювані дані будуть заноситись в базу даних.

## 3.4 Вимоги до технічних засобів

Вимоги до апаратних засобів значною мірою залежать не від вимог розробленого програмного забезпечення, а від вимог операційної системи. Однак рекомендованими мінімальними вимогами до апаратного забезпечення розрахункової програми можна вважати такі:

процесор - AMD Athlon 1.8 GHz або його аналог іншого виробника;

обсяг оперативної пам’яті - 256 MB Мбайт;

обсяг вільного місця на диску - 20 MB.

Зазначені мінімальними вимоги до апаратного забезпечення стосуються лише IBM-сумісних персональних комп’ютерів, оскільки для комп’ютерів сімейства Apple Macintosh тестування розробленого програмного забезпечення не проводилося.

До зазначених апаратних вимог необхідно також додати наявність мінімальної розширювальної здатності монітору 800\*600.

Вимоги до програмного забезпечення викладені нижче.

У якості операційної системи комп’ютера можуть виступати: Windows XP/Vista/7 x86 - x64. Тестування з іншими операційними системами не проводилось.

# Висновки

Як висновок, можна відзначити, що електронні журнали необхідно розглядати не як новомодний спосіб відображення оцінок, а як потужний засіб підвищення успішності і рівня освіти в цілому на рівні школи.

Для кращої реалізації власного проекту було проведено огляд новітніх інформаційних технологій, та програмного забезпечення, що могли б бути використані для розробки сучасного, ефективного та головне конкуренто здатного програмного продукту. Для роботи обрано середовище розробки програмного забезпечення Microsoft Visual Studio 2008.

# Література

1. А. Климов C#. Советы программистам,2008 – 658с.
2. Чарльз Петцольд Программирование с использованием Microsoft Windows Forms Русская Редакция, -Питер,2006 - 432с
3. Павловская Т.А. C#. Программирование на языке высокого уровня,2009 – 453с
4. Фаронов В.В. Программирование на языке С#,2007 – 240с.
5. Ульман Л. MySQL: Пер. с англ. Слинкина А.А — М.: ДМК Пресс; СПб.: Питер, 2004. — 352 с.
6. Интернет-Университет Информационных Технологий [Електронний ресурс]. – Режим доступу: http://www.INTUIT.ru
7. Библиотека класов С# [Електронний ресурс]. – Режим доступу: http://msdn.microsoft.com/
8. Форум программистов .NET [Електронний ресурс]. – Режим доступу: http://forum.vingrad.ru/forum/