

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Олександр Коваль

«___» _____ 2021 р.

Дипломна робота

на здобуття ступеня бакалавра

спеціальності 122 «Комп'ютерні науки»

**освітня програма «Комп'ютерний моніторинг та геометричне
моделювання процесів і систем»**

на тему: «Система автоматизованого тестування»

Виконав (-ла):

студент (-ка) IV курсу, групи ТР-72

Мельник Валентина Сергіївна _____

Керівник:

старший викладач,

Колумбет Вадим Петрович _____

Консультант з назва розділу: _____

Рецензент:

Кандидат наук, старший викладач,

Баранюк Олександр Володимирович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент (-ка) _____

Київ – 2021 року

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерний моніторинг та геометричне моделювання процесів і систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр Коваль

(підпис)

” ” _____ 2021р.

ЗАВДАННЯ

на дипломну роботу студенту

Мельник Валентині Сергіївні

(прізвище, ім'я, по батькові)

1. Тема роботи : _____ „Система автоматизованого тестування”

керівник роботи _____ Колумбет Вадим Петрович , старший викладач

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”25” травня 2020р. №
1168-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи програмний застосунок розроблено в середовищі DE Visual Studio на платформі .NET Framework з використанням мови #C

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) розробити модель елемента системи для інформаційної підтримки процесу контролю та тестування знань; на основі моделі, спроектувати і реалізувати практично окремий модуль елемент інформаційної системи освітньої установи для автоматизації процесів тестування знань; провести тестування і перевірити працездатність розробленого модуля

5. Перелік ілюстративного матеріалу

«Постановка задачі», «Діаграма прецедентів для адміністраторської системи», «Діаграма прецедентів для користувача», «Функціональна схема проекту», «Модульний склад проекту», «Модель даних схеми», «Дерево діалогів», «Дерево модулів», «Приклад роботи системи 1», «Приклад роботи системи 2», «Приклад роботи системи 3», «Висновки»

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання "10" жовтня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	2021р	
2.	Вивчення та аналіз задачі	15.10.21 - 01.02.21	
3.	Розробка архітектури та загальної структури системи	01.02.21 - 21.02.21	
4.	Розробка структур окремих підсистем	21.02.21 - 17.03.21	
5.	Програмна реалізація системи	17.03.21 - 15.04.21	
6.	Оформлення пояснювальної записки	15.04.21 - 08.05.21	
7.	Захист програмного продукту	11.05.21	
8.	Передзахист	25.05.21	
9.	Захист	14.06.21	

Студентка

(підпис)

Мельник Валентина Сергіївна

(прізвище та ініціали,)

Керівник роботи

(підпис)

Колумбет Вадим Петрович

(прізвище та ініціали,)

АНОТАЦІЯ

Метою дипломної роботи є створення системи автоматизованого тестування та розробка програмного коду що надає змогу вносити інформаційні ресурси до бази даних.

Об'єктом дослідження є спеціальні інформаційні технології в освіті, що дозволяють поліпшити, як протікання освітнього процесу та і підвищити рівень кінцевого результату рівень освіти учнів. Було виконано огляд існуючих інформаційних ресурсів та зроблено висновки щодо їх переваг та недоліків. Розроблено програмний комплекс для тестування. Продукт можна використовувати для контролю знань студентів та учнів.

Загальний обсяг роботи: 88 сторінки, 35 ілюстрацій та 56 бібліографічних найменувань.

Ключові слова: Microsoft Visual Studio, Windows Communication Foundation, .Net Framework, C#, , бази даних, клієнт-серверна система.

ABSTRACT

The purpose of the thesis is to create a system of automated testing and development of software code that allows you to enter information resources into the database.

The object of research is special information technologies in education, which allow to improve the course of the educational process and increase the level of the end result of the level of education of students.

Total volume of work: 88 pages, 35 illustrations and 56 bibliographic titles.

Keywords: Microsoft Visual Studio, Windows Communication Foundation, .Net Framework, C #,, databases, client-server system.

ЗМІСТ

Перелік скорочень, умовних позначень і термінів.....	6
ВСТУП.....	7
1 ЗАДАЧА СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ТЕСТУВАННЯ.....	10
1.1 Характеристика предметної області	10
1.2 Огляд літературних джерел за темою дослідження	17
1.3 Тестування як засіб перевірки рівня знань	23
1.4 Характеристика програмного комплексу	26
2 АНАЛІЗ СИСТЕМИ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ	29
2.1 Коротка характеристика інструментарію	29
2.2 Технічне та програмне забезпечення проекту	33
2.3 Сучасні технології та моделі тестування	35
3 ОБГРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ СИСТЕМИ	44
3.1 Модель програмного комплексу	44
3.2 Функціональна схема проекту. Складові системи	47
3.3 Модель даних	49
3.4 Фізична схема бази даних	50
3.5 Програмна реалізація	53
3.5.1 Загальні положення (дерево функцій і сценарій діалогу)	53
3.5.2 Опис програмних модулів	55
3.5.3 Реалізація доступу до бази даних.....	57
3.6 Організація збору, передачі, обробки інформації	59
3.7 Проектування форм, інтерфейс	62
4 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ	64
4.1 Планування і реалізація тестів по проекту.....	64
4.2 Загальні характеристики розробленого комплексу	72
4.3 Інструкція для користувача	74
ВИСНОВКИ.....	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	83
ДОДАТКИ.....	88
Додаток А Текст програми (фрагменти).....	88
Додаток Б Акт впровадження.....	104

Перелік скорочень, умовних позначень і термінів

1) СШарп — мова, яка об'єднує в собі багато зручні конструкції і потужні інструменти багатьох мов, в першу чергу варто згадати С ++, Java, VisualBasic.

2) Node.js — платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript.

3) Microsoft Visual Studio — лінійка продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів.

4) Windows Communication Foundation — ім'я платформи наступного покоління, для мережевих систем з використанням розширення API .NET,

5) СУБД Access — універсальний засобі для створення і обслуговування баз даних, забезпечення доступу до даних і їх обробки.

6) База даних – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами.

ВСТУП

Суспільно-політичні і соціально-економічні перетворення, які відбуваються в країні, інтенсивно впливають на діючу освітню систему: змінюються ціннісні орієнтири, застосовуються нові педагогічні технології, технології управління, що сприяють модернізації змісту освіти, інтенсивному розвитку і впровадженню інноваційних процесів, адекватних сучасним вимогам і потребам часу.

Роль засобів нових інформаційних технологій в освіті сьогодні важко переоцінити. Прискорення науково - технічного прогресу, засноване на впровадженні у виробництво гнучких автоматизованих систем, мікропроцесорних засобів і пристроїв програмного управління, роботів і обробних центрів, поставило перед сучасною педагогічною наукою важливе завдання виховати і підготувати підростаюче покоління, здатне активно включитися в якісно новий етап розвитку сучасного суспільства, пов'язаний з інформатизацією.

Масове впровадження інноваційних технологій, методик з використанням комп'ютерних технологій і програм спостерігається як в секторі середньої освіти так і в ВУЗах.

Впровадження в діяльність навчального закладу інформаційних технологій на сьогодні є пріоритетним напрямком реалізації державної освітньої політики. В рамках виконання Закону України "Про освіту" і, зокрема, статті 48 [1], а також в ряді статей розділу IX, визначено необхідність впровадження технологій, зокрема і систем тестування знань. Кроки, щодо впровадження інформаційних технологій, сприяють активній модернізації освітньої галузі та зміні підходів в реалізації процесів і здійсненні професійної діяльності.

Саме сучасні інформаційні технології зробили можливим доступ кожного фахівця до величезної кількості різних видів інформації. В умовах

розвитку національної системи освіти важливого значення набуває інноваційна діяльність загальноосвітніх навчальних закладів, яка характеризується системним експериментуванням інновацій в освітньому процесі.

Новітня педагогічна технологія це строго обґрунтована система педагогічних засобів, форм і методів, їх поетапна націленість на вирішення конкретних навчально-виховних завдань сучасності.

Застосування інформаційних технологій в навчальній діяльності особливо необхідно і актуально, оскільки саме слідування сучасним потребам і своєчасне впровадження ІТ здатні змінити всю систему в цілому, а від їх правильності і своєчасності залежить ефективність системи освіти.

Сучасні технології дозволяють оптимізувати процес обміну інформацією, зменшити обсяг роботи учасників системи освіти, дозволяє приймати ефективні управлінські рішення і рішення пов'язані з навчальним процесом безпосередньо.

Проблемі використання інформаційних технологій в освіті присвячені праці В.Ю. Бикова, І. Жалдака, С.А. Ракова, В.Н. Кухаренко, А.М. Гуржій, Ю.А. Жука та ін. Дослідник А.Є. Кравчина розглядає проблему інформатизації організаційно - управлінської діяльності в ОУ (освітні установи). В.В. Дивак досліджує інформаційно-комунікаційні технології в професійній діяльності менеджера в сфері освіти. Дослідження вчених Н.В. Морзе і А.Г. Глазунової присвячені розробці моделі ефективного використання інформаційно-комунікаційних та дистанційних технологій навчання у вищому навчальному закладі. Н.А. Зінчук досліджує інформаційно-аналітичну компетентність, значення в професійній управлінській діяльності та передумови формування в вузі.

Об'єктом дослідження є спеціальні інформаційні технології в освіті, що дозволяють поліпшити, як протікання освітнього процесу та і підвищити рівень кінцевого результату - рівень освіти учнів. У той же час, як об'єкт варто

розглядати і освітню установу (ОУ), як цільову організацію, для якої проектується програмне забезпечення.

Предметом дослідження є сектор програмних і програмно-технічних засобів для використання в сфері освітньої діяльності, зокрема програми контролю та оцінки знань, як елемент єдиної інформаційної сфери НЗ (навчальний заклад).

Завдання дипломного проекту:

1. коротко охарактеризувати ВНЗ, як майданчик, на якому будуть впроваджуватися запропоновані технології, описати структуру навчального закладу;
2. розглянути загальну структуру інформаційного середовища освітньої установи, визначити напрямки, які не автоматизовані або рівень автоматизації міг би бути поліпшеним;
3. провести аналіз існуючих розробок і підходів до автоматизації контролю знань з використанням комп'ютерних програм;
4. запропонувати шляхи розвитку існуючої системи інформаційної підтримки та засобів автоматизації в цілому;
5. розробити модель елемента системи для інформаційної підтримки процесу контролю та тестування знань;
6. на основі моделі, спроектувати і реалізувати практично окремий модуль - елемент інформаційної системи освітньої установи для автоматизації процесів тестування знань;
7. провести тестування і перевірити працездатність розробленого модуля.
8. оформити документацію по проекту у вигляді пояснювальної записки по дипломній роботі.

1 ЗАДАЧА СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ТЕСТУВАННЯ

1.1 Характеристика предметної області

Ідеєю дослідження і практичної реалізації в роботі є підсистема тестування знань. Розробка цієї системи повинно передбачати відповідний рівень універсальності, така система не повинна бути прив'язана до конкретного ВНЗ або іншого НЗ (навчального закладу). Виходячи з цього немає сенсу аналізувати конкретний ВНЗ, варто розглянути структуру абстрактного навчального закладу і визначити місце і роль проекрованої в роботі системи.

Розглянемо структуру узагальненого вищого навчального закладу.

ВНЗ це організація, призначення якої - забезпечити навчальний процес для студентів з метою здобуття вищої освіти в конкретній спеціальності. ВНЗ це величезний комплекс, який включає в себе набір різних компонентів і бізнес-процесів. З точки зору тематики даної роботи варто розглянути структуру ВНЗ з точки зору забезпечення навчального процесу і наукової діяльності.

Процес підготовки фахівця є композицією цілого класу бізнес процесів організації, в які залучені такі базові категорії персон і підрозділів, прийняті в більшості вузів країни.

Сьогодні пропонуються і альтернативні (щодо класичної ієрархічної структури) структури, розглядаються їх переваги та недоліки, пропонуються різні проекти з реструктуризації і реорганізації. Наприклад: ряд авторів монографії [54] пропонують альтернативні структури. Однією з найбільш пріоритетних, виходячи з їхніх досліджень, є матрична структура (рис. 1.1)

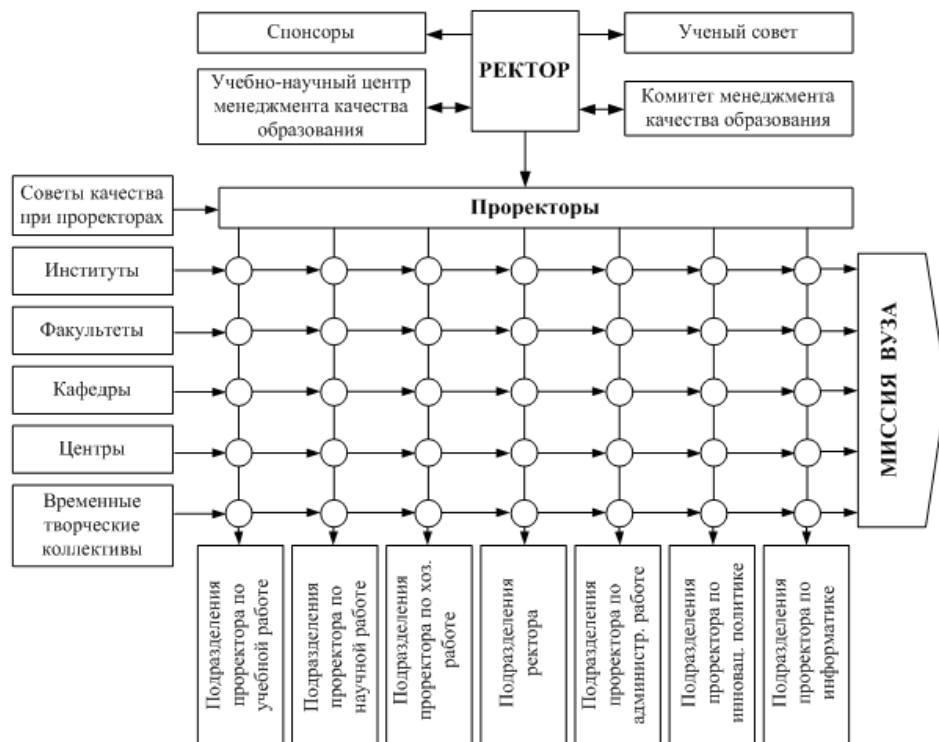


Рисунок 1.1 - Приклад альтернативної організації структури

У той же час, на даному етапі, більшість ВНЗ України сьогодні все ж дотримуються ієрархічної структури (рис. 1.2)

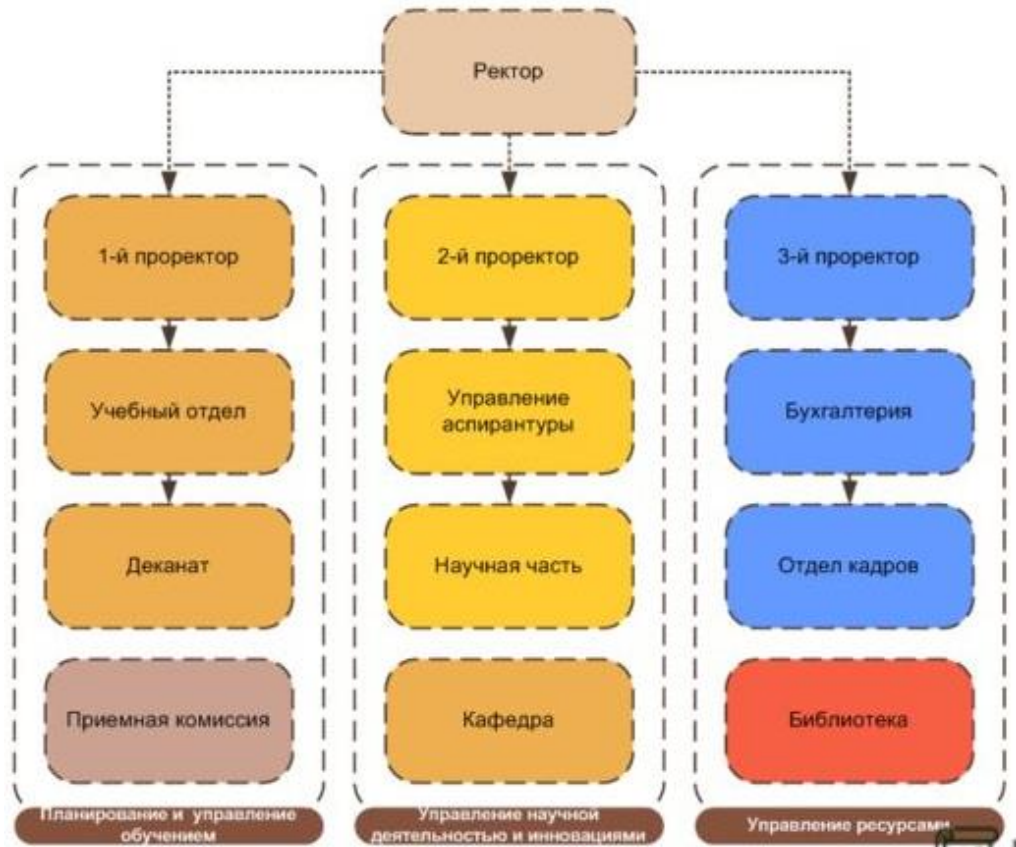


Рисунок 1.2 - Загальна ієрархічна структура

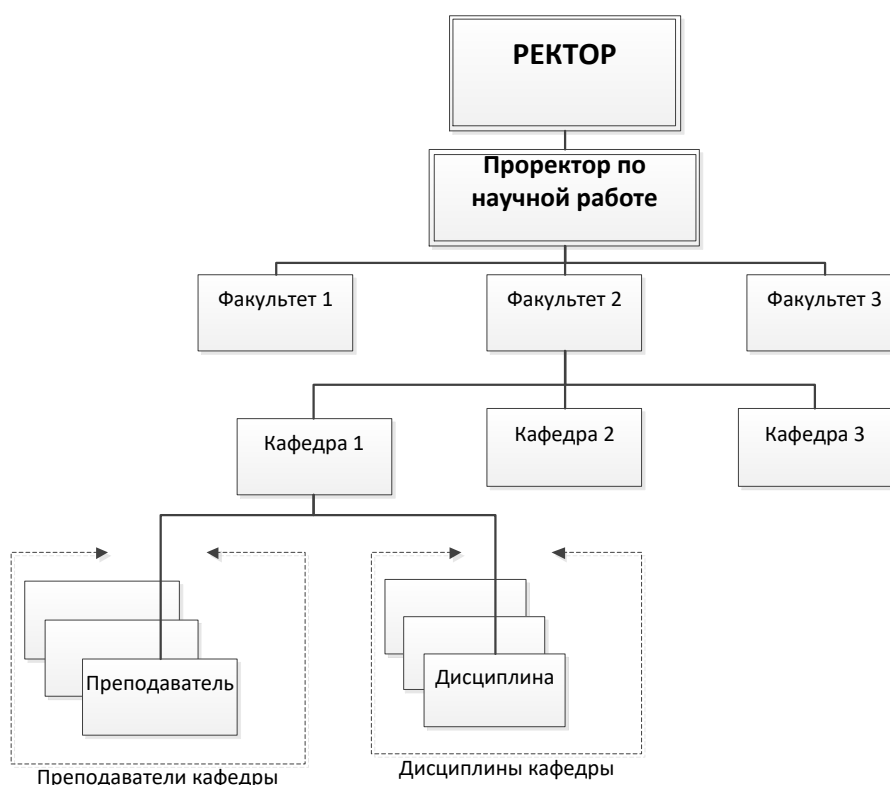


Рисунок 1.3 - Ієрархічна організаційна структура навчальної гілки ВНЗ

Далі визначимо основні поняття.

Ректор - керівник навчального закладу йому підпорядковані прямо або побічно всі структурні підрозділи і співробітники. Ректор підпорядкований безпосередньо міністерству освіти.

Проректор з наукової роботи - заступник ректора в питаннях забезпечення навчальної та наукової діяльності.

Факультет навчально-науковий і адміністративний структурний підрозділ вищого навчального закладу, що здійснює підготовку студентів та аспірантів за однією або кількома спорідненими напрямками підготовки.

Кафедра підрозділ вузу, що здійснює виконання навчальних курсів в рамках спеціалізацій кафедри, що здійснює підготовку студентів в рамках своїх спеціалізацій. Кафедра структурно відноситься до певного факультету. Існують так звані загально – факультетні (міжфакультетні) кафедри, наприклад в технічних вузах це можуть бути кафедри, що забезпечують соціальні дисципліни (соціологія, філософія, культура і т.д.). Такі кафедри забезпечують дисципліни на різних кафедрах.

Основними клієнтами (споживачами освітніх послуг) є студенти, студентом стає випускник середньої школи, пройшовши етап «Абітурієнт».

Абітурієнти особи, що беруть участь в конкурсі для вступу до ВНЗ.

Студенти особи, що проходять навчання за певним напрямом підготовки, що здійснюється в рамках факультету ВНЗ.

Професорсько-викладацький склад (ПВС) викладачі, які безпосередньо проводять навчальні курси в рамках напрямків підготовки, а також здійснюють проміжну і підсумкову атестацію студентів. Викладачі можуть бути співробітниками кафедр, силами якої виконується деякий набір навчальних курсів в рамках спеціалізації кафедр.

Навіть такий побіжний аналіз дає можливість розділити досліджуваний об'єкт на: організаційні особливості (глобально ВНЗ), складові структурні елементи організації, та особи (актори) - люди, які беруть безпосередню активну участь в процесі.

Базову модель структури відносин, з точки зору проектування якоїс інформаційної системи, можна представити таким чином (рис. 1.4).

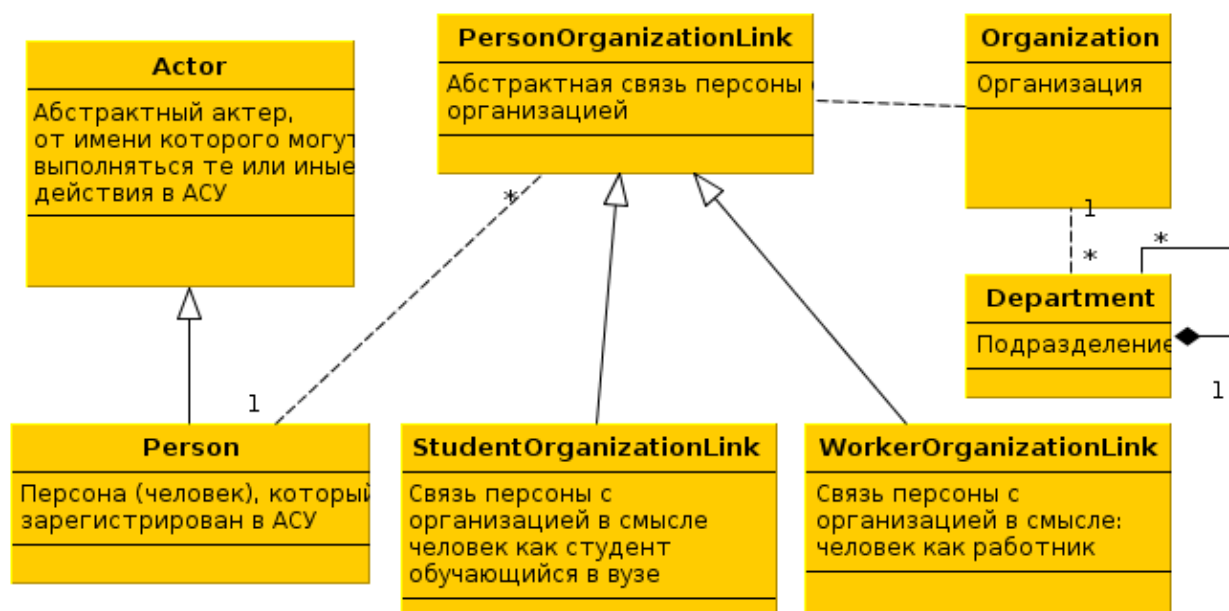


Рисунок 1.4 - Базова модель основних відносин з організацією

UML діаграма на рис. 1.4 вимагає наступних пояснень; Персона (Person) є окремим випадком більш загального поняття: Актор (Actor), який є суб'єктом

дій в інформаційній системі, між іншим варто відзначити - що в ролі зовнішнього суб'єкта може бути не тільки персона, а й деяка зовнішня програмна система, що взаємодіє з АІС. На цьому прикладі, також представлена базова модель ієрархічної структури підрозділів організації, з певним ставленням рекурсивної вкладеності (агрегації). Таку базову модель пропонує автор [55].

Для визначення місця проектованої системи розглянемо структурно-функціональну модель «Надання освітніх послуг»



Рисунок 1.5 - Контекстна діаграма IDEF0

Діаграма наочно ілюструє передумови і механізми надання освітньої послуги.

Передумови (вхідні дані і керуючі документи) досить докладно описані на схемі і пояснень не вимагають. Результати теж досить очевидні.

Механізми: сюди відноситься в першу чергу професорсько-викладацький склад (ПВС) як головний механізм забезпечення освітнього процесу. Природно варто окремо відокремлювати матеріальну базу (будівлі,

меблі, комп'ютери та техніка тощо). Крім викладачів діяльність забезпечує і ряд інших підрозділів: юристи, діловоди, секретарі, технічний персонал.

Завдання «забезпечення освітнього процесу» сформульована занадто широко і обсяг роботи не дозволяє розробити і описати повну модель. Тому обмежимося деякими обраними областями.

Для деталізації процесу використовується діаграма декомпозиції. Так діаграма декомпозиції надання освітніх послуг може виглядати наступним чином (рис. 1.6)

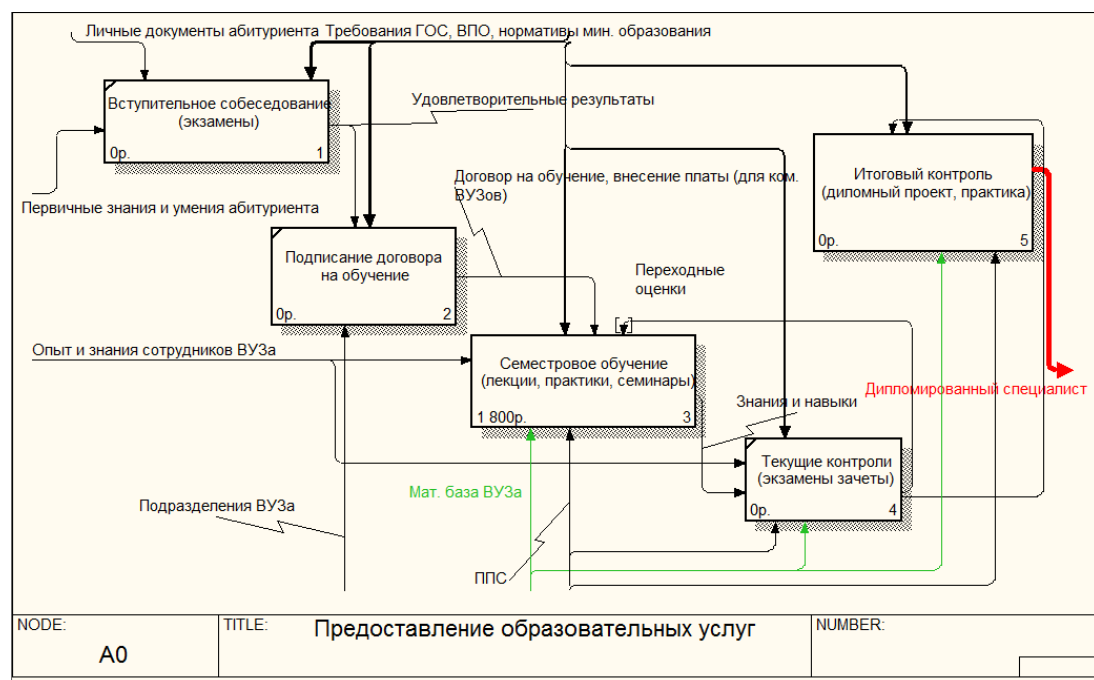


Рисунок 1.6 - Декомпозиція першого рівня

Уже на цій діаграмі декомпозиції видно процес «Поточні контролю». Саме цей процес і підлягає автоматизації за допомогою використання проектованої системи автоматизації тестування знань.

Навчання в семестрі (можливі і інші поділи: триместри, це не впливає кардинально на протікання процесів) можна в свою чергу деталізувати наступною діаграмою (рис. 1.7)



Рисунок 1.7 - Декомпозиція другого рівня

Система тестування з доповненням теоретичними матеріалами, для навчання за конкретною тематикою, може частково поєднати і процес «Виконання завдань» і процес «Контроль і оцінка знань». Частково тому, що не можна будувати завдання тільки на тестуванні. Завдання повинні бути комплексними і включати різні види робіт, серед яких тестування також доречно. Система має переваги в оцінюванні, так як оцінювання здійснюється програмою і є абсолютно об'єктивним, повністю, виключаючи суб'єктивний фактор і вплив відносин між «Студент» - «Викладач».

При створенні системи з наперед продуманим рівнем універсальності може використовуватися не тільки для тестування студентів. Так, наприклад, багато ВНЗ на своїй базі проводять різні курси перепідготовки, підвищення кваліфікації, курси від служб зайнятості та інших організацій. Розробляється система повинна припускати і таку переорієнтацію. Це істотно розширить спектр застосування системи. Як приклад можна розглядати тестування в напрямку «Охорона праці та безпека життєдіяльності»

1.2 Огляд літературних джерел за темою дослідження

На сучасному етапі склалася невідповідність між діючою системою управління навчальним процесом в навчальних закладах та новітніми вимогами щодо її вдосконалення. У спадок від радянської системи освіти Україна отримала традиційні засоби ведення документообігу, контролю, організації управлінської діяльності. І зараз у багатьох навчальних закладах документи, що відображають хід навчального процесу, який є основним виробничим процесом в навчальному закладі, формуються і зберігаються у вигляді паперових каталогів з різними списками, картами, відомостями, звітами і т.д. Такий спосіб організації управління негативно позначається на продуктивності праці в організаційних підрозділах закладу, створює несприятливі умови для ефективної роботи персоналу.

Така ж ситуація і з використанням ІТ технологій для навчання і наповнення освітнього процесу. У багатьох випадках НЗ володіють технічними засобами, але їх ефективне застосування обмежене відсутністю спеціалізованих орієнтованих на навчальний процес програм, відсутність налагоджених методик застосування ІТ. Часто використання ІТ в освітньому процесі зводиться до набору презентацій супроводжуючих виклад матеріалу.

Впровадження інформаційно-комунікаційних технологій в інформаційне забезпечення освітніх процесів здійснюється епізодично і недостатньо. Поступово формується ринок таких систем, де пропонуються як їх комерційні версії, створені незалежними виробниками, так і системи, розроблені безпосередньо в підрозділах освітньої сфери, де ці кошти і експлуатуються, адже науково - технічний потенціал освітян, особливо, якщо цей навчальний заклад технічного профілю, цілком дозволяє розробити діючу і всеосяжну систему автоматизації навчального процесом самостійно. Залежно від способу створення, подібні системи мають ряд характерних особливостей, як позитивних, так і негативних. Так, до недоліків інформаційних систем, створених незалежними розробниками, можна віднести їх надмірну

комерціалізацію, надлишкову універсальність, складність настройки, громіздкий і незрозумілий інтерфейс. Як позитивні ознаки можна відзначити хорошу документування та сервісну підтримку, широкий набір функціональних можливостей. Інформаційні системи індивідуальної розробки, як правило, менш стійкі, більше залежать від персонального супроводу своїми розробниками, гірше задокументовані. Однак системи, розроблені безпосередньо силами співробітників навчального закладу, точніше відповідають його специфічним умовам і вимогам, гнучко і оперативно реагують на зміни в законодавстві і загальній економічній ситуації, вони зручніше для використання персоналом певного навчального закладу. З цього випливає, що подібні системи потрібно обов'язково розробляти збірною командою і неодмінно залучати в процес розробки фахівців з освітньої сфери, дуже бажано, щоб ці фахівці були практикуючими вчителями, викладачами, менеджерами освітніх установ.

Потрібно відзначити, що ситуація позитивно змінюється в останні роки - наповнюється багаж програмного забезпечення для освітніх процесів, з'являються віртуальні лабораторії, спеціалізовані пошукові бібліотечні системи, інтегровані навчальні середовища.

Поява програмним забезпеченням дає змогу частково вирішити проблему, все більш широкого застосування набувають комп'ютеризовані інформаційні системи, що дозволяють автоматизувати і привести до ладу цю сферу діяльності, раціоналізувати інформаційні потоки в ній і розвантажити персонал навчальних закладів від одноманітної рутинної роботи.

У той же час ці програми вимагають постійного оновлення як за формою (орієнтація на ОС, розрядність, форма реалізації, інтерфейс) так і за змістом. Виходячи з цього - проблема розробки інформаційних систем і програмного забезпечення для автоматизації навчального процесу залишається постійно актуальною. Процес інформатизації є одним з перспективних напрямків розвитку освіти на сучасному етапі. Використання інформаційних технологій

в освіті дає якісно нові можливості і в ході протікання процесів і, як наслідок, позначається на кінцевому результаті - якості освіти. Державою передбачається створення умов для організації інформаційного забезпечення в освітньому закладі з використанням інформаційно-комунікаційних технологій: обладнання освітніх установ інформаційно-обчислювальною технікою, організація і проведення всеобучу з інформаційних технологій для управлінських і педагогічних кадрів.

На необхідність оптимізації інформаційно-аналітичної діяльності в управлінні школою вказують Н.К. Джамірзе багато дослідників (В. С. Лазарев, А. М. Моїсєєв, М.М. Поташник та ін.). Проблемі використання інформаційних технологій в освіті присвячені праці В.Ю. Бикова, І. Жалдака, С.А. Ракова, В.Н. Кухаренко, А.М. Гуржій, Ю.А. Жука та ін .. Дослідник А.Є. Кравчина розглядає проблему інформатизації організаційно - управлінської діяльності в ОУ (освітні установи). В.В. Дивак досліджує інформаційно-комунікаційні технології в професійній діяльності менеджера в сфері освіти. Дослідження вчених Н.В. Морзе і А.Г. Глазунової присвячені розробці моделі ефективного використання інформаційно-комунікаційних та дистанційних технологій навчання у вищому навчальному закладі. Н.А. Зінчук досліджує інформаційно-аналітичну компетентність, значення в професійній управлінській діяльності та передумови формування в вузі. У вітчизняній літературі досить повно розкрито питання про раціональної організації роботи з інформацією як одного з чинників оптимального управління. Питання оптимізації процесу обробки управлінської інформації розглядалися в роботах Л.І. Фишмана, Т.І. Шамової.

Багато авторів (наприклад Карпушина Т., Горбунова Л, Суботіна Е., Циганов О.) ставлять питання про необхідність комплексного підходу до реалізації інформаційного середовища НЗ, а також його постійного аудиту та модернізації. [2,3,32,34,50]

Інформаційне забезпечення освітніх процесів, побудоване на створенні інформаційних систем дає можливість знизити труднощі в обробці інформації та розширити сферу її використання. При цьому кожна інформаційна система постає як сукупність підсистем: інформаційної, організаційної, правової та програмної.

Інформаційна підсистема містить уніфіковані системи документації, схеми інформаційних потоків.

Організаційна підсистема регламентує взаємодію працівників з технічними засобами і між собою в процесі експлуатації інформаційної системи, визначає області управлінських завдань, що підлягають автоматизації; розподіляє функціональні обов'язків кожного при вирішенні цих завдань.

Правова підсистема включає в себе сукупність правових норм, що визначають створення і функціонування інформаційної системи: посадові інструкції, локальні нормативні акти, положення про порядок використання інформації.

Програмна підсистема складається з комплексу технічних засобів, що використовуються при функціонуванні інформаційної системи і сукупності всіх програм, в тому числі прикладних програм, як загального, так і спеціального призначення. Це дає можливість використовувати великі обсяги інформації, зіставляти показники, проводити швидкий пошук і обробку даних, а значить здійснювати вибір найкращого варіанту з можливих і приймати оптимальні рішення, в ході реалізації навчального процесу.

Будь-який навчальний заклад є системним об'єктом, і тому його інформатизація істотно впливає на всі сторони її життєдіяльності - від змісту освіти до фінансово-господарських питань, включаючи психологічний клімат в колективі, систему орієнтирів і точок зростання освітнього закладу. Таким чином, інформаційне середовище має пізнавальну цінність, надає емоційний

вплив, сприяє розвитку особистості учня (студента) і професійному становленню викладача.

Єдина інформаційна система НЗ вирішує ряд питань:

1. стимулює творчу активність, звільняє від фізіологічних обмежень, страху самовираження;
2. знімає навантаження - рутинні операції виконуються машиною, а вчитель оперативно залучає необхідні джерела текстової, графічної та аудіо-візуальної інформації;
3. розширюються можливості надання результатів навчальної діяльності. [54]

Єдиного інформаційного середовища освітньої установи повинна включати компоненти, що забезпечують інформатизацію основних видів діяльності НЗ:

- управління кадрами;
- управління ресурсами;
- забезпечення комунікації;
- управління контингентом учнів;
- наповнення навчального процесу;
- контроль і аналіз результатів навчання.

Загальні вимоги до реалізації інформаційного середовища освітньої установи включають:

- єдину базу даних;
- одноразовий введення даних з можливістю їх подальшого редагування;
- розмежування прав доступу до даних;
- використання одних і тих же даних в різних додатках і процесах;
- можливість обміну даними між різними прикладними програмами без виконання операцій експорту-імпорту.

Єдиного інформаційного середовища - це умова і засіб складних внутрішньо-системних взаємодій типу «людина - техніка», «людина -

людина», «людина - знакова система», «людина - художній образ». Єдиного інформаційного середовища розвивається як відкрита система, що самоорганізується відповідно до логіки і закономірностями власного розвитку і повинна породжувати нові форми діяльності в установі, змінювати і збагачувати всю систему освіти.

Розвиток єдиного інформаційного середовища пов'язано з постійним підвищенням рівня її організації та технічного оснащення. Єдність і цілісність структури інформаційного середовища визначаються єдністю педагогічних цілей, взаємозв'язком розв'язуваних педагогічних завдань і взаємодією учасників освітнього закладу.

При проектуванні єдиного інформаційного середовища освітньої установи необхідно наступне:

- забезпечувати на базі компетентнісного підходу формування інформаційної культури всіх учасників навчально-виховного процесу; досліджувати можливості впровадження інформаційних і комунікаційних технологій в практику викладання навчальних предметів, а також в діяльність виховної роботи освітньої установи;

- створити умови для практичного застосування комп'ютерної техніки учасниками освітнього процесу в навчальний та позанавчальний час, виходячи з потреб;

- забезпечити для учасників освітнього процесу відкритий доступ до інформаційних каналів локальної обчислювальної мережі, глобальної мережі Інтернет і ресурсів медіатек;

- організувати процес критичного осмислення накопичується емпіричного педагогічного досвіду, його узагальнення і аналізу, а також обмін досвідом з інформатизації освіти на різних рівнях;

- забезпечити безперервне розвиток технічної інфраструктури єдиного інформаційного середовища, роботу технічного персоналу, проведення

планових регламентних робіт, поточного ремонту та екстреного усунення несправностей, необхідне фінансування.

1.3 Тестування як засіб перевірки рівня знань

Тестування широко застосовується для оцінки рівня знань в навчальних закладах, при прийомі на роботу, для оцінки кваліфікації персоналу установ, тобто практично у всіх сферах діяльності людини, коли вимагається оцінка рівня знань чи навичок. Випробуваному пропонується ряд питань (тест), на які він повинен відповісти.

Зазвичай до кожного питання дається кілька варіантів відповіді, з яких треба вибрати правильну. Кожному варіанту відповіді відповідає деяка оцінка. Підсумовуванням оцінок за відповіді виходить загальний бал, на основі якого робиться висновок про рівень підготовленості випробуваного.

Основна відмінність тесту полягає в тому, що він завжди передбачає вимірювання – оцінку рівня знань. Оцінка, що виставляється за підсумками тестування, відрізняється більшою об'єктивністю і незалежністю від можливого суб'єктивізму того, хто проводить співбесіду або тестування усно. Головна відмінна риса тесту - об'єктивність, що гарантується вимірюванням, функція якого полягає в тому, щоб давати кількісну (бальну) оцінку.

Максимально підвищити надійність вимірювання і валідність використання тесту можна, якщо слідувати трьом основним етапам його створення:

- дати чітке і недвозначне теоретичне визначення умінь, які треба перевірити;
- точно встановити умову і операції, які повинні дотримуватися при проведенні тесту;
- кількісно визначити результати спостережень, з тим, щоб переконатися, що використовуються мірні шкали володіють всіма необхідними якостями.

Метод тестування в науковій літературі описаний давно історія розвитку тестів як науково обґрунтованого методу вимірювання почалася в кінці XIX в. і інтенсивно продовжується до наших днів. В Україні розробкою тестового методу вчені почали активно займатися тільки в дев'яностих роках минулого століття, за радянських часів технологія не вважалась досить доцільною для використання в навчальному процесі і практично не розвивалась.

Тестування сьогодні це окрема галузь, в якій постійно з'являються нові підходи, методи, засоби та інструменти. Але більшість досліджень в даному напрямку присвячено тестуванню знань з орієнтацією на учнів і студентів та досить мало робіт розглядають тестування як складову управління персоналом.

Під комп'ютерним тестом будемо розуміти реалізовану за допомогою персонального комп'ютера систему взаємопов'язаних предметним змістом завдань стандартної форми, що мають певну шкалу значень, виконання яких виявляє наявність знань, умінь і навичок і рівня підготовки фахівця. Сформованість умінь і навичок доцільно перевіряти, враховуючи такі взаємодоповнюючі критерії [46]:

1. правильність: результат будь-якої діяльності відповідає меті виконання;
2. осмисленість: готовність пояснити кожную з застосовуваних операцій;
3. усвідомленість: орієнтація на суттєві зв'язки і відносини, задані в умові при здійсненні діяльності;
4. раціональність: вибір такого способу виконання завдань, який швидше (оптимально) призведе до мети;
5. узагальнення: здатність студента виконувати необхідні дії в варіативних умовах;
6. абстрактність: можливість опису оперативного складу виконаної дії і послідовності виконання операцій, що входять в неї, відтворення необхідного правила;

При розробці системи слід враховувати наукові підходи, які сьогодні склалися в теорії тестування, - тести, орієнтовані на критерій (критеріально-орієнтовані) і тести, орієнтовані на певну норму (нормативно орієнтовані). Ці два підходи грають важливу роль в процесі створення тестів. Підготовлений тест необхідно експериментально апробувати і вивірити. Показників, що характеризують тест з тієї чи іншої точки зору, досить багато. Зазвичай виділяють: ступінь складності, валідність, надійність, простоту, однозначність.

Ступінь складності визначається відношенням

$$Q = W / N, \quad (1.1)$$

де W - кількість неправильних відповідей, N - загальна кількість людей, що беруть участь в тестуванні. Особливо складні і легкі питання вилучаються з тіста і перефразовуються, таким чином тест балансується.

Для вимірювання ступеня володіння інформацією на кожному рівні може використовувався коефіцієнт:

$$K = P1 / P2, \quad (1.2)$$

де P1 - кількість правильно виконаних завдань в процесі тестування, P2 - загальна кількість завдань у тесті ($0 < K < 1$). Коефіцієнт визначає необхідність проведення тренінгів та курсів підвищення кваліфікації для співробітників. Граничною величиною вважають значення 0,7.

Перевагою використання комп'ютерних тестів є їх інтерактивність, яка дозволяє студенту (співробітнику) здійснювати самоконтроль, що сприяє творчій активності фахівця.

Впровадження комп'ютерного тестування сприяє отриманню статистично достовірної картини індивідуального прогресу кожного студента, систематичному відстеженню якості і динаміки досягнень.

Систем тестування реалізовано велику кількість. Всі вони відрізняються ступенем функціональності, технологіями покладеними в основу реалізації,

вимогами до платформи і рядом інших критеріїв. Не можна сказати, що немає хороших систем, але розробка власного модуля має два яскравих переваги:

- можливість інтеграції в єдиний інформаційний простір і взаємозв'язок з іншими його елементами;
- можливість доопрацювання і розширення функціоналу специфічними проблемно-орієнтованими функціями. Наприклад - система тестування може бути доповнена аналітичним модулем: різними функціями збору статистики та аналіз результатів за період.

1.4 Характеристика програмного комплексу

В практичній частині необхідно реалізувати універсальну підсистему для автоматизації проведення тестування студентів (учнів).

Система не повинна мати безпосередню прив'язку до конкретного матеріалу, повинна передбачати можливість проектування різних тестів, з різними наборами питань та структурою.

Система повинна мати можливість працювати в локальній мережі, з авторизованим доступом для студентів, та окремим додатком для адміністрування системи.

Програмний комплекс повинен передбачати реалізацію двох складових:

Клієнт та Адміністратор (Адміністратор включає також і функціонал сервера БД)

Комплекс реалізується на основі взаємодії клієнта та серверної частини додатку. При умові працюючого сервера клієнти можуть проходити тестування в локальній мережі.

1) Сервер - пов'язаний з базою даних, і містить основні програмні засоби для рішення поставлених завдань, в цій частині реалізуються всі класи, які відповідають за обробку й передачу даних між іншими складовими системи.

2) Клієнт - програма призначена для організації діалогу з користувачем. Робота із сервером і БД організована через інтерфейс класів сервера, що за допомогою WCF передає необхідну інформацію на сервер.

Клієнт передбачає:

1 Авторизація входу в систему

2 Перегляд документів та матеріалів по тематиці (це можуть бути навчальні матеріали або ж тести для ознайомлення)

3 Пройти тестування

4 Переглянути свої результати

Адміністраторська частина передбачає:

1. Додавання користувачів системи (ті, що мають тестуватися, та ті які відповідають за процес тестування (умовно адміни) розділені на різні категорії)

2. Додавання матеріалів для ознайомлення

3. Створення тестів

Тести можуть включати питання трьох типів

ТП – ТВ

ПЗ – ТВ

ТП – ВЗ

ТП- текстове питання

ТВ – текстова відповідь

ЗП – питання-зображення

ВЗ – відповідь зображення (вибір картинки)

Редактор тесту має наступний вигляд:

Система тестирования - добавления тестов

№ теста: 1

Вид вопроса: В-То

Кол-во вопросов вида В-То: 0

Кол-во вопросов вида В-К: 0

Кол-во вопросов вида К-То: 0

Сохранить тест

Вопрос:

Ответ:

А: ☐

Б: ☐

В: ☐

Г: ☐

Правильный ответ: А

Балл за вопрос: 0

Добавить вопрос

Рисунок 1.8 – Опорный макет формы «Редагування тесту»

Для кожного тесту вказуються предмет і тема, а також визначається час на проходження тесту. Тести зберігаються в БД системи. Тести групуються за напрямками (дисциплінами) та характеризуються конкретною тематикою

4. Перегляд результатів тестування з можливістю фільтрації та пошуку по різним критеріям

2 АНАЛІЗ СИСТЕМИ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ

2.1 Коротка характеристика інструментарію

Основним інструментом розробки програми «Система обліку валютних операцій», обрана IDE Visual Studio. Середовище базується на використанні платформи і засобів .NET Framework [46], при відносній простоті характерна гнучкістю і масштабованістю розробляються в ній проєктів. Середовище має високий рівень документованості, спеціально для інформаційної підтримки розробників компанія виробник реалізувала потужну довідкову систему, яка постійно підтримується і оновлюється.

Microsoft Visual Studio - лінійка продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів.

Як мова розробки було обрано мову C #, розробником якої також є фірма Microsoft. Однією з причин такого вибору став той факт, що по даній мові досить легко можна знайти приклади і довідкову інформацію як в електронному (<http://msdn.microsoft.com>), так і в паперовому вигляді в зв'язку з великою популярністю даного мови.

Мова C # (C_Шарп)

Мова CШарп розроблявся як базова мова для платформи .NET. Платформа .NET характерна багатомовними підтримкою проєктів, (Система має подвійний проєкти - нововведення, що з'явилося саме з приходом .NET), це означає, що платформа і середовище розробки підтримує ще ряд популярних мов. У той же час CШарп є основною мовою, на неї орієнтована основна підтримка: розробка нових бібліотек, технічна підтримка, документування[33].

CШарп має деякі запозичення з мови Java, але їх схожість досить перебільшена. Більш правильним буде висловлювання - мова CШарп об'єднує в собі багато зручні конструкції і потужні інструменти багатьох мов, в першу чергу варто згадати C ++, Java, VisualBasic. У той же час багато моментів

суттєво спрощені для розробника - не потрібно використовувати явний деструктор, роль покажчиків істотно спрощена в порівнянні з C ++, контроль пам'яті багато в чому автоматизовано [35,66].

Пропонований перелік розробки різноманітних програмних елементів представлений на рис. 2.1.

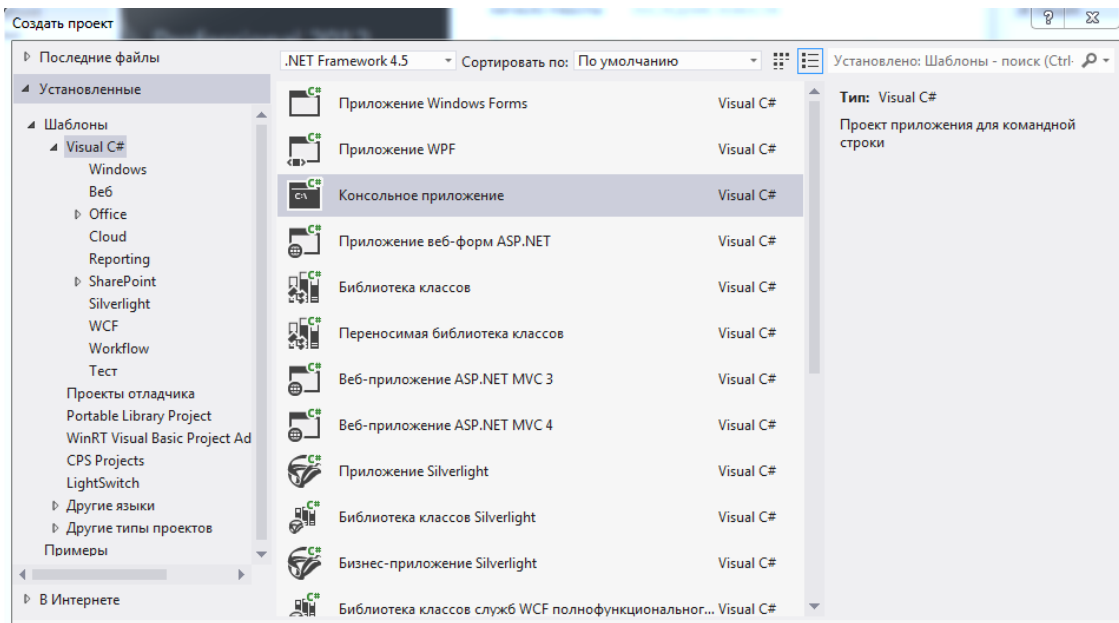


Рисунок 2.1 - Варіативність середовища з точки зору типу проектів і мови програмування

Зазначимо деякі переваги і зміни притаманні мові СШАРП в порівнянні з його найближчими «родинними» мовами:

1. Немає прямої необхідності використовувати покажчики, механізм покажчиків захований від розробника і використовується побічно.
2. Управління і контроль пам'яті (видалення невикористовуваних об'єктів, очищення пам'яті) автоматизовано за рахунок використання технології «Збірник сміття», команда delete в C # відсутня.
3. Реалізовано ряд формальних шаблонів і конструкторів для класичних конструкцій, делегатів і т.д ..
4. Перевантаження функцій (методів класу) підтримується і багато в чому спрощена в порівнянні з C ++.
5. Підтримується автоматизована технологія анотацій, що дозволяє уточнювати поведінку для членів класу.

Функціональність і можливості мови розширювалися з виходом кожної наступної версії .NET. Так у версії .NET 2.0 з'явилися наступні можливості:

1. Створення узагальнених типів і членів, що дозволяє істотно підвищити стабільність і безпеку коду в області роботи з типами.

2. Істотно спрощена модель "делегат-подія", реалізовані можливості: коваріантність, контрваріантність, можливість перетворення в групах методів.

3. Можливість перевизначення одного і того ж типу в декількох файлах коду

У наступній .NET 3.5 (вихід на ринок - кінець 2008 р) C # збагатився новими функціональними можливостями. [35]

1. Можливість створювати і працювати з типізованих запитами LINQ, запити орієнтовані на різні типи даних і істотно розширюють можливості їх обробки.

2. Анонімні типи, що визначають в першу чергу форму без обмежень по поведінці.

3. Використовуючи методи розширення з'явилася можливість розширювати типи.

4. Введено лямбда-операція ($\Rightarrow$) покликана спростити роботу з делегатами.

5. Ініціалізація властивість об'єкта під час створення.

З кожним новим оновленням .NET оновлюються і можливості мови, не стала винятком і остання версія. Варто також відзначити, що деякі не актуальні або застарілі, на думку розробників Microsoft, інструменти та функції перестають підтримуватися. Це не так критично для новостворюваних додатків (у більшості випадків існує новий більш зручний або потужний інструмент), але це досить критично для проектів написаних раніше, які підлягають підтримки та оновлення. Іноді відсутність підтримки застарілих

функцій обертається великою проблемою, яка потребує великих витрат на адаптацію проекту до нових умов [36].

Середовище розробки Visual Studio 2013

Visual Studio 2013 являє собою останню версію розроблену компанією Microsoft в лінійці продуктів Visual Studio. Як і попередні версії, дана версія середовища має графічний конструктор, підтримку повторного використання фрагментів коду, роботу із зовнішніми бібліотеками і DLL. Середовище має набір допоміжних утиліт орієнтованих на різні допоміжні завдання виникають в ході реалізації проекту: перегляд об'єктів та складових елементів проекту, інформацію про версію збірки, настройки збірки і ряд інших. Середовище включає інтегровану систему довідки. У Visual Studio 2013 пропонується безліч додаткових можливостей, найбільш важливі з яких перераховані нижче:

1. графічні редактори і конструктори XML;
2. підтримка розробки програм Windows, орієнтованих на мобільні пристрої;
3. підтримка розробки програм Microsoft Office;
4. підтримка розробки проектів Windows Workflow Foundation;
5. вбудована підтримка рефакторинга коду;
6. інструменти візуального конструювання класів.

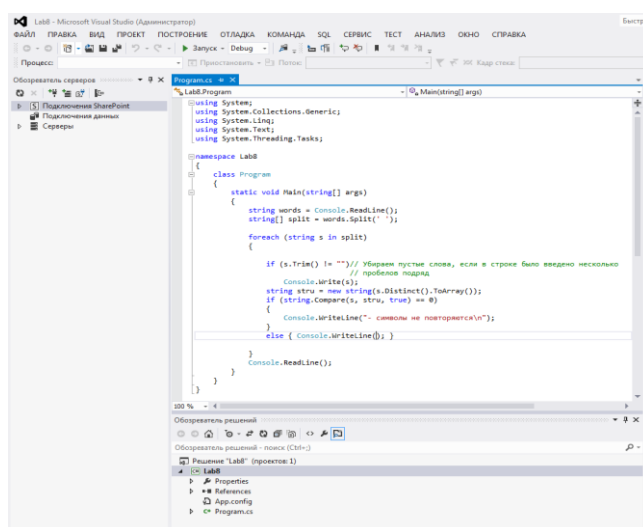


Рисунок 2.2 - Приклад реалізації проекту в середовищі VS

Останнім засобом в Visual Studio 2013, яким необхідно обов'язково вміти користуватися з самого початку, є повністю інтегрована довідкова система. Продукція, що поставляється з .NET Framework 4.5 SDK документація являє собою виключно хороший, дуже зрозумілий і насичений корисною інформацією джерело. При наявності з'єднання з Інтернетом переглядати документацію .NET Framework 4.5 SDK можна в онлайн-режимі за наступною адресою: <http://msdn.microsoft.com/library>.

2.2 Технічне та програмне забезпечення проекту

Технологія зберігання та обробки БД

Для реалізації зберігання даних в програмному комплексі використовується СУБД Microsoft Access. Система СУБД Access є універсальним засобом для створення і обслуговування баз даних, забезпечення доступу до даних і їх обробки.

Кожна база даних зберігається у вигляді файлу з розширенням * .mdb (*.accdb). Доступ до бази даних Microsoft Access можна організувати і з декількох комп'ютерів, об'єднаних в одну мережу. Хоча дана БД не відноситься до типу клієнт-серверних, обробка даних може бути організована на програмному рівні.

Обрана для проекту СУБД, підтримує реляційну модель даних, таблиці разом з атрибутами і зв'язками між таблицями перенести в середу СУБД з урахуванням вимог до відповідних об'єктів бази даних. Так, ідентифікатори (Імена) таблиць і полів задовольняють вимогам СУБД, типу даних, розміри полів, обмеження також приведені у відповідність з прийнятими в даній СУБД нормами.

Таким чином, послідовність робіт при створенні фізичної моделі на основі реляційної моделі даних є такою:

1. аналіз і вибір СУБД;

2. розробка структури (фізичної моделі) бази даних засобами обраної СУБД з урахуванням типів даних і обмежень цілісності атрибутів;
3. розробка схеми взаємозв'язків бази даних засобами обраної СУБД з урахуванням обмежень посилювальної цілісності;
4. розробка контрольного прикладу заповнення бази даних.

Кінцевим результатом роботи є представлення таблиць і зв'язків між ними, описані в моделі даних, в середовищі обраної СУБД.

Апаратно-технічне забезпечення

Технічне забезпечення - комплекс технічних засобів, призначених для роботи інформаційної системи, а також відповідна документація на ці засоби і технологічні процеси.

Комплекс технічних засобів становлять:

- комп'ютери;
- пристрої збору, накопичення, обробки, передачі та виведення інформації - жорсткі диски, пристрої зберігання даних, сканери, принтери, факсимільні апарати;
- пристрої передачі даних і ліній зв'язку - модеми;
- експлуатаційні матеріали - папір, CD (DVD) диски і т. П.

При виборі комп'ютера необхідно керуватися рядом характеристик. До таких характеристик відносяться надійність, вартісні витрати, продуктивність, простота використання та ін. Від значення зазначених параметрів залежить можливість роботи з необхідними програмними засобами, а отже, і успіх створення системи.

Для кожного з елементів даної схеми виділяється перелік критеріїв, найбільш важливих при здійсненні вибору технічного забезпечення.

Для робочих станцій найбільш часто використовуються наступні критерії:

- робоча частота процесора;
- швидкодія й обсяг оперативної пам'яті;

- обсяг жорсткого диска;
- розмір монітора;
- наявність можливості підключення до ЛВС;
- ергономічність клавіатури і миші.

Розроблена система має такі системні вимоги:

Таблиця 2.1 – Системні вимоги

Дисковий простір	100 мб вільного місця і більше (залежить від об'єму БД)
Операційна систем	Windows 7/8/8.1
.NET Framework	V 4.5

Для повноцінної роботи системи рекомендуються наступні вимоги до апаратної частини:

Таблиця 2.2 – Апаратні вимоги

Процесор	Двохядерний з частотою на менше 2GHz
ОЗП	2048 мб
Периферія	Клавіатура, миша
Принтер	Струйний/Лазерний

2.3 Сучасні технології та моделі тестування

На сьогоднішній день в світі існує велика кількість систем комп'ютерного тестування, що призводить до необхідності певної їх класифікації. В основу класифікації покладено ті або інші характерні ознаки, принципи побудови, схема взаємодії з користувачем, предметно-галузева спрямованість.

Можна виділити наступні типи системи комп'ютерного контролю знань:

- по можливості поповнення бази питань і внесення змін до програмної оболонки: відкриті і закриті;
- за схемою взаємодії з користувачем: гнучкі і формалізовані;

- по середовищем застосування: локальні (паритетні, "Клієнт-сервер"), глобальні;

- по предметно-галузевій спрямованості: профільні і універсальні.

Складові частини типової системи комп'ютерного тестування:

- підсистема тестування;
- підсистема створення, підготовки та редагування тестів;
- підсистема статистичних даних;
- підсистема управління користувачами
- база даних питань і пов'язаної з ними статистичної інформації;
- база даних користувачів.

Для того, щоб використання системи комп'ютерного тестування було доцільним, необхідним є дотримання кількох умов:

- комп'ютерне тестування має надавати результати не гірше, ніж його неавтоматизований аналог;
- результат комп'ютерного тестування повинен якісно наближуватись до результату іспиту;
- для того, щоб отримати додаткові переваги перед усною співбесідою, тестування повинно максимально абстрагуватися від людського фактору при перевірці робіт і при визначенні оцінки.

У системах комп'ютерного тестування повинно бути передбачено:

- точна і здатна до адаптації оцінка результатів тестування;
- можливість використання як для внутрішнього так і відкритого тестування;
- аналіз тестів для оцінки якості, придатності окремих питань і виявлення некоректно сформованих або просто неправильних питань, здійснюється шляхом відправлення спеціальних повідомлень адміністратору системи від користувача і апробація тестових наборів;
- легкість створення і модифікації тестів;

- відсутність жорсткої прив'язки до предмету, області знань;
- збір, зберігання і представлення в зручній формі різної статистичної інформації про процес тестування;
- можливість одночасного тестування великої кількості користувачів;
- відсутність жорсткої прив'язки до конкретного апаратного і програмного забезпечення;
- висока безпека, захищеність і стабільність;
- необмежена кількість тестів, тем, питань і варіантів відповідей на них;
- зручний і ефективний імпорт і експорт тестів;
- контроль часу тестування і збереження інформації по перерваного сеансу тестування;
- підтримка безпечного і універсального механізму управління користувачами з розподілом прав доступу;
- підтримка режиму модульного контролю, в процесі якого існує ряд питань і зафіксовано час, протягом якого на них необхідно відповісти.

Для повноти огляду розглянемо кілька on-line систем, а також систем для локальних мереж і локальних ПК.

Система тестування Brainbench є світовим лідером on-line сертифікації. пропонується величезний вибір онлайн тестів, які постійно оновлюються і змінюються (за станом на даний момент число тестів більше 900). У даній системі існують як платні (вартість здачі тесту - \$ 49.95), так і безкоштовні тести з різних дисциплін: інформаційні технології, мовні знання і ін .. Тести представлені англійською мовою. Здаючи тест, необхідно відповісти на питання, вибравши один або кілька правильних варіантів відповіді.

У тесті, як правило, 40 питань, на кожне питання 5-6 варіантів відповіді, час на обдумування від двох до п'яти хвилин. Перед здачею тесту слід уважно прочитати його опис. Як правило, до здачі основного тесту дається можливість пройти невеликий безкоштовний тест, що складається з 10-15 питань, для попередньої оцінки власних сил і складності тесту.

За даними Brainbench їх послугами вже скористалося близько 6000000 користувачів. При створенні цієї системи були використані сучасні технології і мови:

HTML, XML, SQL, PHP, Java, JavaScript.

В цілому, Brainbench є успішною та ефективною системою тестування знань, але все-таки присутні деякі недоліки при проходженні тестів:

- неможливість точно визначити особистість того, хто тестується, що дозволяє професіоналу проходити тести для інших користувачів;
- використовувати пошукові механізми Internet для відповіді на питання.

Ці недоліки не можна розглядати як істотні, так як більшість систем мають такі ж властивості і це є особливістю тестування знань в мережі Internet.

Система тестування Neuron є яскравим прикладом відкритої системи тестування знань, використовується тільки для публічних змагань і не містить інструментів ментів для адекватного контролю знань в контексті навчального процесу. Кожне питання характеризується предметом, до якого воно належить, кількістю балів за відповідь і рівнем складності.

Чим вище складність питання, тим більше балів можна отримати при правильній відповіді на нього. Показником успішності користувача є загальна сума балів, набрана їм під час тестування.

Основними недоліками системи є:

- практично повна непристосованість для використання в навчальному процесі;
- умовна і неефективна система нарахування балів за правильну відповідь;
- неефективна система поповнення бази питань, коли будь-який учасник може додати своє питання, що веде до виникнення великої кількості неправильних питань.

SunRav TestOfficePro

Розробник: SunRav Software

Сайт програми: <http://www.sunrav.ru/srtop/index.php>

Спосіб поширення: shareware

Ціна: 1970 грн. (Для всіх освітніх установ надається 20% знижка, тобто ціна для них 1575 грн.).

У пакет SunRav TestOfficePro входять програми для створення тестів, проведення тестування та обробки результатів тестування. З його допомогою можна організувати і провести тестування та іспити в освітніх установах (вузи, коледжі, школи), а підприємства та організації можуть здійснювати атестацію та сертифікацію своїх співробітників.

Пакет дозволяє створювати тести з предметів шкільної програми, вузівським дисциплін, тести для професійного тестування, психологічні тести тощо

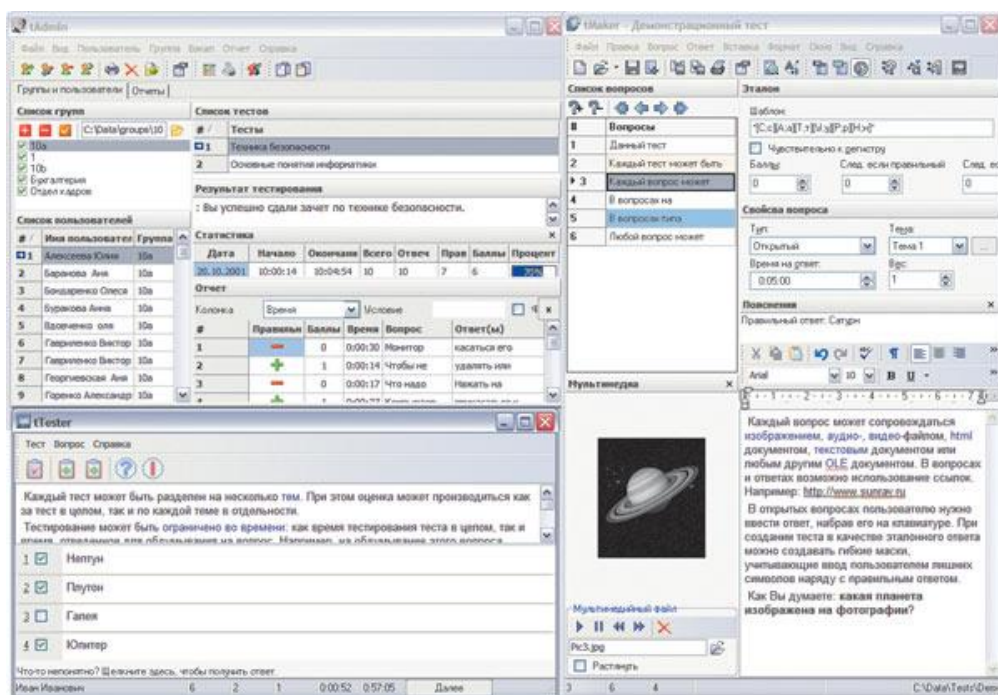


Рисунок 2.3 - Робоче вікно SunRav TestOfficePro

Всі тести і результати тестування шифруються методами стійкої криптографії, що виключає можливість підробки результатів. Крім того, на тест можна встановити паролі: на редагування захищає тест від перегляду його

структури, правильних відповідей і т.д .; на перегляд запобігає пробне тестування з метою з'ясування правильних відповідей.

Питання і варіанти відповіді можна формувати, використовуючи для цього вбудований текстовий редактор, близький за своїми функціями до MS WORD. В редакторі можна вставляти зображення, формули, схеми, таблиці, аудіо- та відеофайли, HTML-документи і будь-які OLE-документи.

У тестах можливе використання п'яти перерахованих вище типів питань.

Тест може бути розділений на декілька тем. При цьому можливо оцінювати знання тестованого як по кожній темі окремо, так і по тесту в цілому.



Рисунок 2.4 - Класична схема проходження процесу тестування

Питання в тесті можна перемішувати. Більш того, розробник тесту може визначити, скільки питань з кожної теми отримає користувач для тестування. Припустимо, кожна тема складається з 100 питань. Якщо вибрати випадковим чином тільки 10 питань, то тестовані отримають абсолютно різні набори питань з одного і того ж тесту. Варіанти відповідей також можна перемішати.

Порядок проходження питань може бути не тільки лінійним, але і залежати від відповідей користувача.

У пакет SunRav TestOfficePro входять наступні програми:

tMaker - програма, що дозволяє створювати і редагувати тести. Можливо імпортування тестів, створених в текстовому редакторі або в редакторі електронних таблиць;

tTester - програма для проведення тестування;

tAdmin - програма для віддаленого адміністрування користувачів і обробки результатів тестування. Дозволяє переглядати / друкувати результати тестування, а також створювати, друкувати, редагувати, експортувати звіти по тестуванню груп користувачів. Можливе створення матриці відповідей.

В даний час для даного пакета підготовлена велика кількість тестів. Зокрема, є освітні тести з інформатики, економіки, англійської мови, історії, суспільствознавства.

Конструктор тестів «Техносервіс плюс»

Розробник: ТОВ «Техносервіс плюс»

Сайт програми: <http://tsplus.narod.ru/test.html>

Спосіб поширення: замовлення по e-mail

Ціна: 1850 грн.

Програма призначена для створення тестів, іспитів, електронних підручників. Для розробки тесту або підручника користувачеві не обов'язково мати навички програмування програма адаптована і призначена насамперед для полегшення процесу перевірки знань в освітніх установах. Основною відмінністю продукту від аналогів є додаткова можливість створення модуля закріплення матеріалу, коли вже в процесі навчання користувач відповідає на питання по заданій темі.

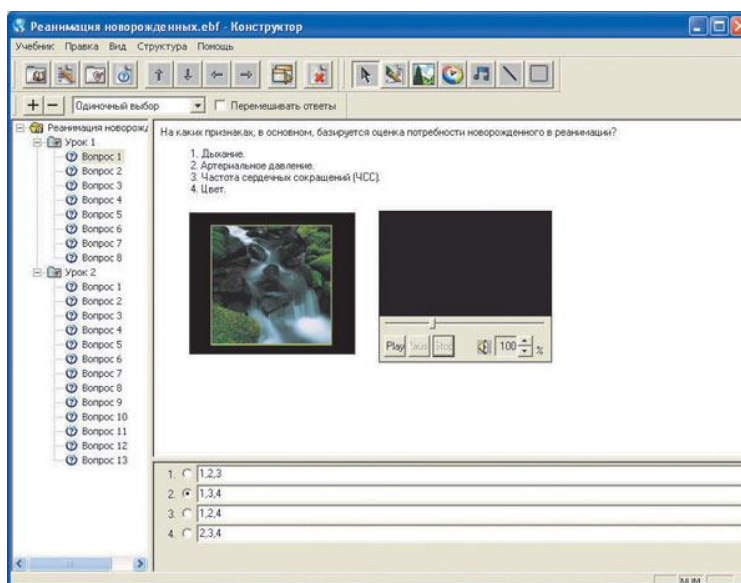


Рисунок 2.5 - Робоче вікно Конструктор тестів «Техносервіс плюс»

Програмний комплекс містить три модулі:

конструктор, що дозволяє створювати тести, підручники, а також модулі закріплення матеріалу. У конструкторі передбачена можливість використання текстового формату, формул, звукозапису, відеороликів і зображень;

- підручник - він створюється в конструкторі і може мати необмежену кількість розділів, параграфів і сторінок;

- тест, який також може бути прирівняний до іспиту, оскільки виключає можливість користуватися підручником під час перевірки знань, як при закріпленні матеріалу.

У цьому модулі створюються нові розділи і питання, а також вказуються відповіді на питання. При цьому у властивостях питання необхідно вказати кількість балів, що нараховуються за правильну відповідь на нього.

Програма дозволяє:

1. самостійно створювати тести для контролю знань учнів. Розробка тесту автоматизована і виконується швидко, передбачена можливість використання не тільки тексту і малюнків, а й звукозапису, відео, формул, графіків, складних графічних об'єктів та ін .;

2. з легкістю імпортувати в конструктор матеріали, за якими побудовані поточні електронні підручники;

3. проводити перевірку учнів з метою виявлення їх рівня знань під час вступу до установи (при переході в наступний клас (курс);

4. виробляти внутрішні планові / позапланові перевірки знань після проходження нового матеріалу для оцінки засвоєння матеріалу;

5. створювати електронні підручники, інструкції, методичні матеріали;

6. в режимі навчання реалізовувати опитування учня за обраними питань; в разі невірної відповіді програма вкаже на помилку і порекомендує ще раз прочитати відповідний розділ підручника.

У пакет програм «Конструктор тестів» входять:

1. модуль створення тестів і підручників;
2. модуль тестування;
3. модуль навчання;
4. модуль для перегляду підручників.

Обмежень за кількістю комп'ютерів, на які встановлюється програма, немає.

3 ОБГРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ СИСТЕМИ

3.1 Модель програмного комплексу

На сьогоднішній день існує багато технологій для реалізації систем з архітектурою клієнт-сервер, але серед цих технологій відрізняється Windows Communication Foundation (WCF). WCF - ім'я платформи наступного покоління, для мережеских систем з використанням розширення API .NET, WCF робить можливим побудову безпечних, надійних і транзакційних систем через спрощену уніфіковану програмну модель міжплатформного взаємодії. При використанні цієї системи в комплексі разом з базами даних, виникає велика кількість різноманітних можливостей для розробника програмного забезпечення.

Використовуючи WCF і мову C#, розроблена мережева система, основним завданням якої є організація проходження тестування студентів. Програма розділена на дві частини - клієнт і сервер.

1) Сервер - пов'язаний з базою даних, і містить основні програмні засоби для вирішення поставлених завдань, точніше саме в цій програмі реалізовані всі класи, які відповідають за обробку та передачу даних між іншими складовими системи.

2) Клієнт - програма призначена для організації діалогу з користувачем. Робота з сервером і БД організована через інтерфейс класів сервера, який за допомогою WCF передає необхідну інформацію на сервер.

Завдяки технологіям WCF, OleDb, мови C # і бази даних Access можна розробляти багатофункціональні системи автоматичного управління, орієнтовані на роботу в мережі, що значно спрощує як процес розробки програмного забезпечення, так і впровадження програмних комплексів в експлуатацію.

Безперечно, розробка сучасних програмних продуктів неможлива без попереднього моделювання. Накопичений досвід свідчить: чим більше і складніше проект, тим важливіше стає моделювання майбутньої системи. Не

варто сподіватися на успішність проекту, якщо не приділено достатньої уваги попередньому моделюванню системи.

Моделювання використовують не тільки при створенні великих систем. Адже чим більше і складніше система, тим більшого значення набуває моделювання при її розробці. Справа в тому, що моделювати складну систему необхідно в будь-якому випадку, оскільки в іншому випадку ми не зможемо її представити як єдине ціле. Моделі наочно демонструють бажану структуру і поведінку системи, відображають її архітектуру і допомагають уточнити деталі проекту з замовником для мінімізації майбутніх ризиків.

Для розроблюваної системи тестування, діаграма прецедентів приведена на рис. 3.1-3.2.

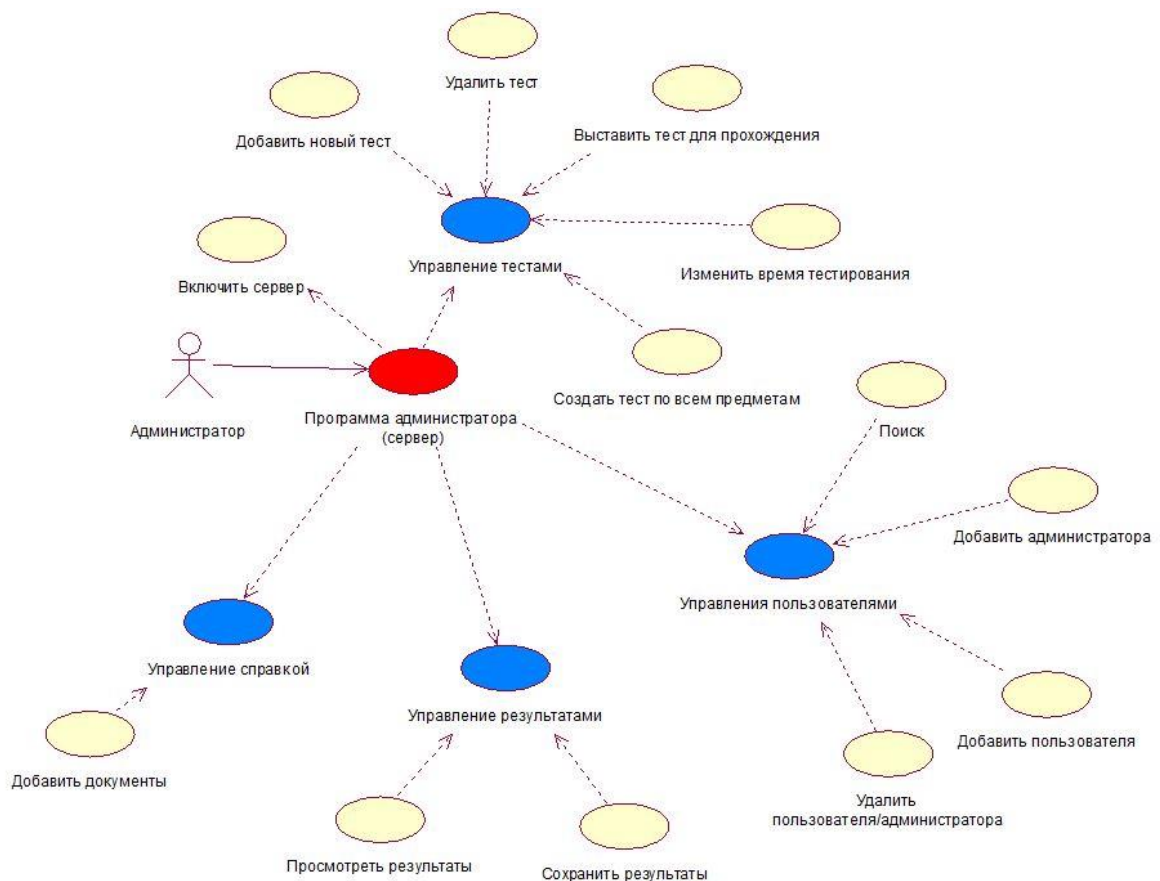


Рисунок 3.1 – Діаграма прецедентів для Адміністратора системи

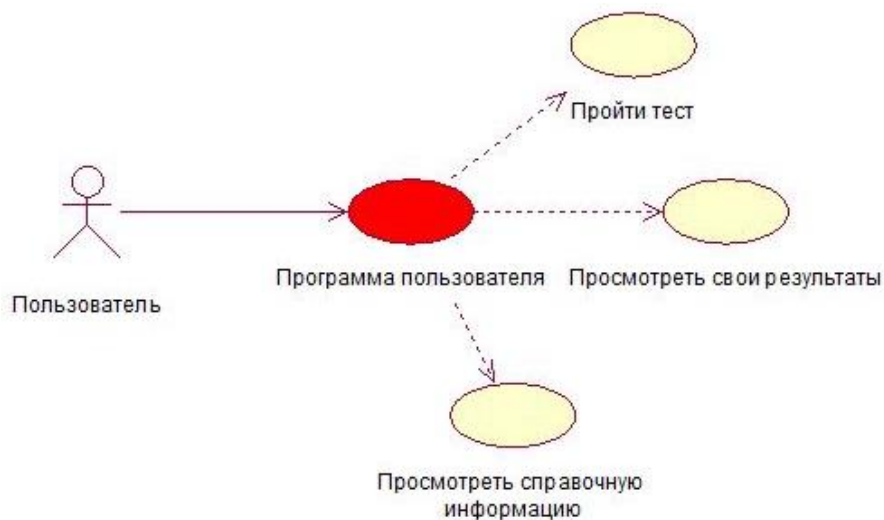


Рисунок 3.2 – Діаграма прецедентів для користувача

«Користувач» і «Адміністратор» є акторами моделі (об'єкти, які безпосередньо взаємодіють з системою).

Прецедентами ж є можливості модельованої системи (частина її функціональності), завдяки якій актор може отримати конкретний, вимірний і потрібний йому результат.

Так як розроблена програма є клієнт-серверним додатком, на рис. 3, показана модель взаємодії його основних залишають: клієнта, сервера і бази даних.

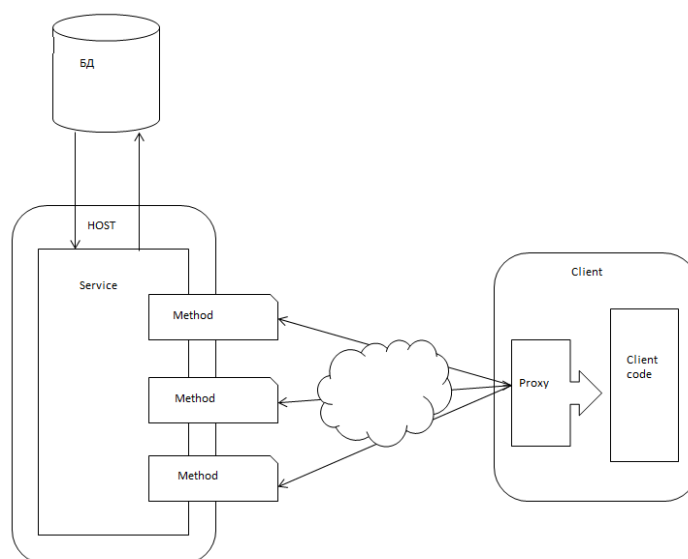


Рисунок 3.3 - Клієнт-серверна модель розробленої системи

Більш повне уявлення про програмний продукт, що розробляється в ДП, з точки зору взаємодії його компонентів між собою і з зовнішнім середовищем дає функціональна схема.

3.2 Функціональна схема проекту. Складові системи

Функціональна схема взаємодії компонентів програмного забезпечення з описом інформаційних потоків, складу даних в потоках із зазначенням використовуваних файлів і пристроїв показана на рис. 3.4

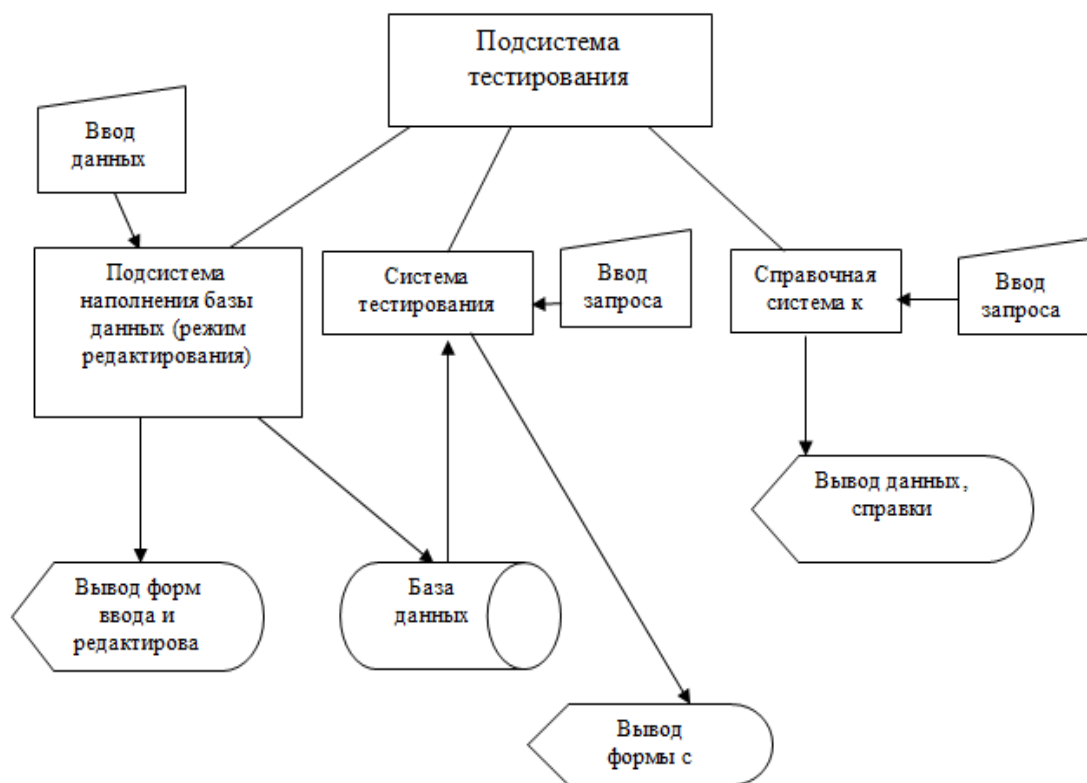


Рисунок 3.4 - Функціональна схема проекту

Автоматизована інформаційна система складається з двох основних частин. Програмної оболонки, яка дає користувачеві зручний інтерфейс, можливість формувати звіти, редагувати / додавати / видаляти дані та інше, а також з бази даних, яка зберігає всю інформацію, і приймаючи запити від програмної частини робить вибірку необхідних даних для тих чи інших завдань.

Програмна частина являє собою додаток написаний на мові С# з використанням .Net Framework 4.5.

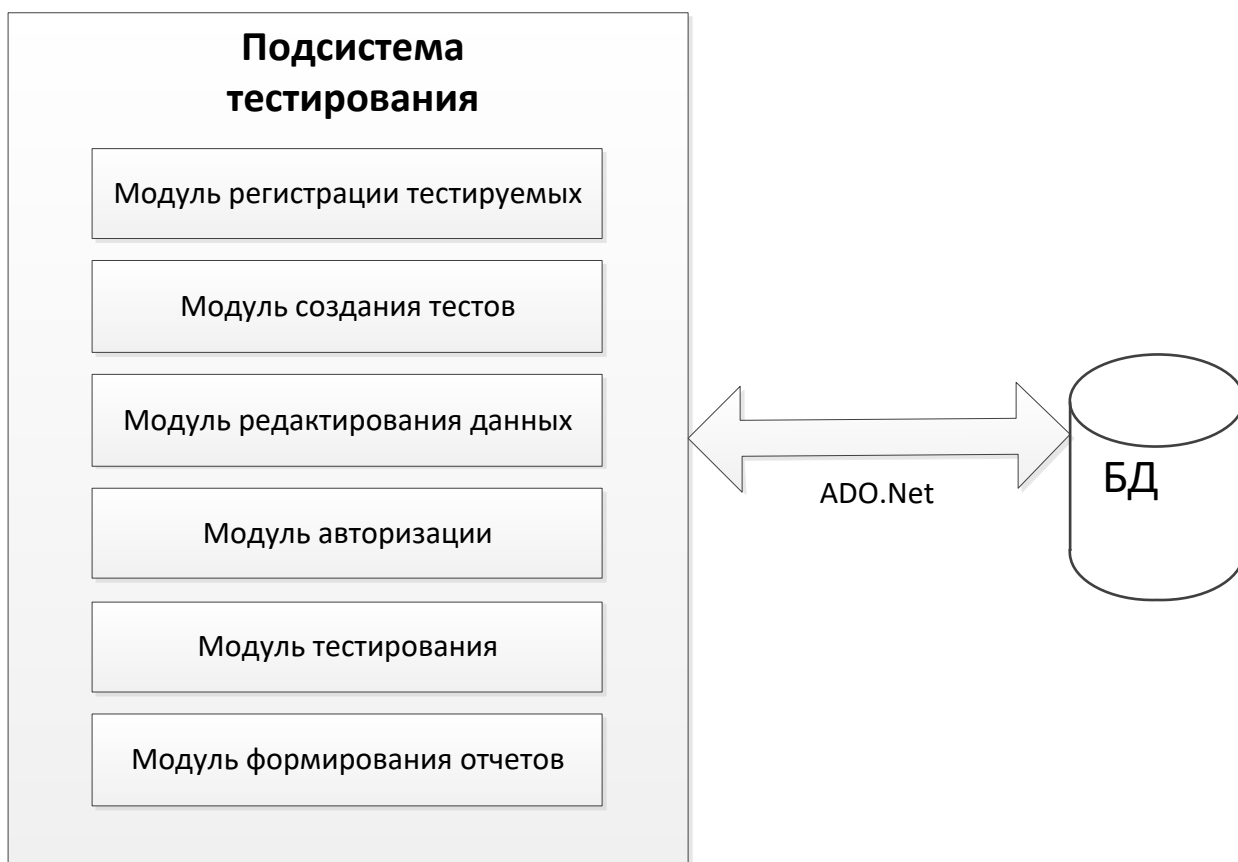


Рисунок 3.5 – Модульний склад проекту

Як ми можемо бачити з рисунка, представленого вище, програмна частина системи в свою чергу ділиться на модулі. Розглянемо їх докладніше:

- Модуль авторизації - призначено для авторизації користувача, і є частиною системи безпеки системи, так як виконує перевірку користувачів. Автоматично при реєстрації система визначає доступ до функціоналу, доступ до матеріалу, реквізити для збереження результатів.
- Для того щоб співробітник міг пройти тестування, він, попередньо має бути зареєстрований в ІС, за це відповідає модуль реєстрації;
- Модуль створення тестів – забезпечує можливість адміністратору системи створювати та редагувати тести в системі.
- Модуль тестування (разом з попереднім) реалізує основний функціонал системи.

- Підсистема редагування - призначена для редагування існуючих даних в базі даних.

- Підсистема генерації звітів - відповідає за генерацію звітів базуються на даних, які містяться в БД.

- Модуль взаємодії з базою даних - призначається для здійснення зв'язку бази даних з програмною частиною системи.

Друга частина системи, як уже згадувалося це база даних. Далі розглянемо модель бази даних і її безпосередню реалізацію.

3.3 Модель даних

Основними сутностями в системі є людина, що проходить тестування (учень, студент, співробітник) та сам тест.

Людина (користувач системи) в системі фігурує як основний суб'єкт, що взаємодіє з системою. Сутність «людина» визначається і уточнюється наступними даними:

- ПІБ;
- посада/категорія;
- логін/пароль для авторизації в системі

Тест являє собою набір питань певної структури, тест може бути вибіркоким набором n питань з великого набору, а може бути фіксованим переліком.

В результаті проходження тесту студент оцінюється, результати повинні зберігатися в БД.

Враховуючи вище сказане модель даних може бути представлено наступною схемою

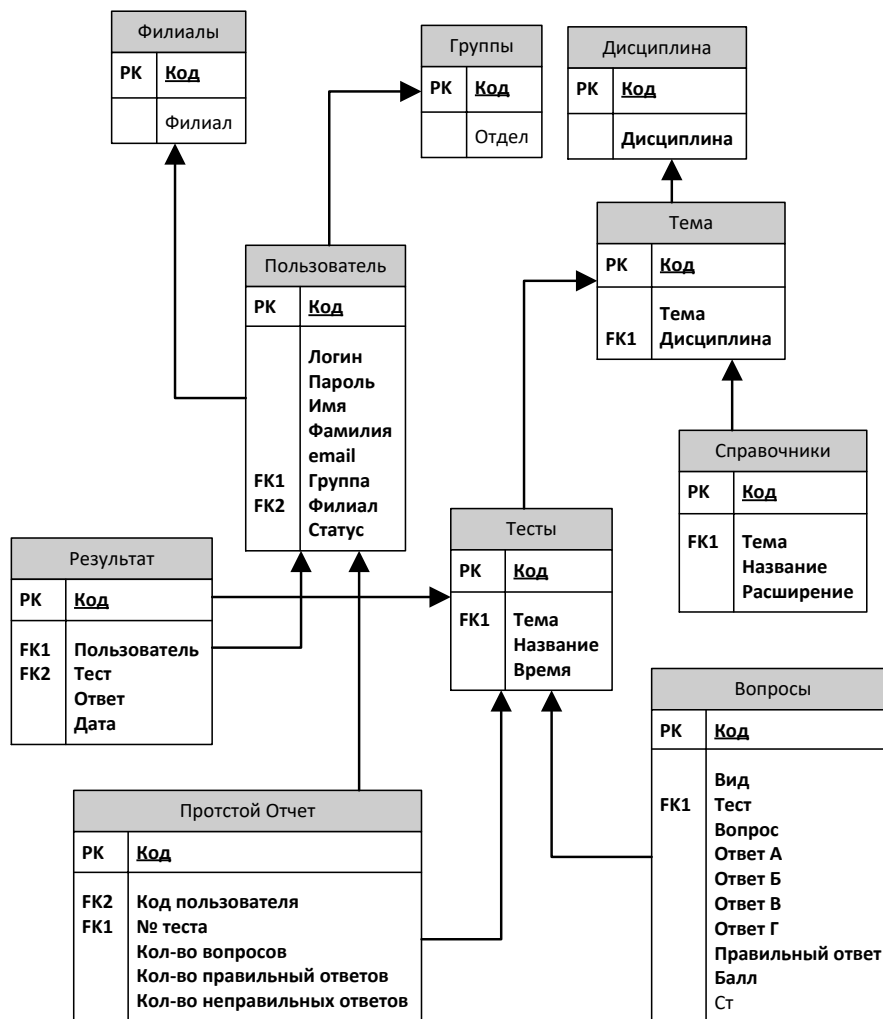


Рисунок 3.6 – Модель данных системы

3.4 Фізична схема бази даних

Для реалізації частини системи, що відповідає за збереження даних, використовується СУБД Microsoft Access. Система СУБД Access є універсальним засобом для створення й обслуговування баз даних, забезпечення доступу до даних і їхній обробки.

Для розроблювальної системи сутностями фізичної моделі даних є: «Питання», «Категорія користувача», «Дисципліна», «Користувачі», «Звіт», «Результат», «Довідники», «Тема», «Тести», «Філія».

Опишемо ці таблиці:

Таблиця 3.1- Сутність «Питання»

Сутність	Тип
Код	Лічильник
Вид	Текстовий
Тест	Числовий
Питання	Поле МЕМО
Відповідь А	Поле МЕМО
Відповідь Б	Поле МЕМО
Відповідь В	Поле МЕМО
Відповідь Г	Поле МЕМО
Правильна відповідь	Текстовий
Бал	Числовий

Сутність «Питання» зберігає основну інформацію по тесту: кожне окреме питання та варіанти відповідей. Як видно, всі питання є стандартизованими (мають єдиний формат).

Таблиця 3.2- Сутність «Категорія»

Код	Лічильник
Назва категорії	Текстовий

Сутність «Категорії» зберігає найменування категорій (студент, абітурієнт, студент_магістр і т.д.) для автоматизації вводу інформації та подальшого пошуку.

Таблиця 3.3 - Сутність «Дисципліна»

Код	Лічильник
Дисципліна	Текстовий

Сутність «Дисципліна» зберігає найменування дисциплін (напрямків), по яких може проходити тестування.

Таблиця 3.4- Сутність «Користувачі»

Код	Лічильник
Логін	Текстовий
Пароль	Текстовий
Ім'я	Текстовий
Прізвище	Текстовий
eMail	Текстовий
Категорія	Числовий
Філія	Числовий
Статус	Логічний

Сутність «Користувачі» містить основну інформацію пов'язану з персонами (наприклад - студентами), що проходили тестування. Без внесення цих даних, реєстрація й робота користувача із програмою буде неможливою.

Таблиця 3.5- Сутність «Простий_Звіт»

Код	Лічильник
Номер	Числовий
№ тесту	Числовий
Кількість питань	Текстовий
Кількість правильних відповідей	Текстовий
Кількість неправильних відповідей	Текстовий

Сутність « Простий_Звіт » містить первинну інформацію, що формується після завершення користувачем тестування. Ця інформація є результатом тестування й виводиться користувачеві безпосередньо після його проходження.

Таблиця 3.6- Сутність «Результат»

Код	Лічильник
Користувач	Числовий
Тест	Числовий
Відповідь	Текстовий
Дата	Текстовий

Сутність «Результат» містить інформацію про кожне конкретне пройдене питання користувача по тесту. Інформація заноситься після проходження всього блоку питань.

Таблиця 3.7 - Сутність «Довідники»

Код	Лічильник
Тема	Числовий
Назва	Текстовий
Розширення	Текстовий

Сутність «Довідники» містить інформацію про довідкову документацію доступну користувачам.

Таблиця 3.8- Сутність «Тема»

Код	Лічильник
Дисципліна	Числовий

Тема	Текстовий
------	-----------

Сутність «Тема» зберігає інформацію про можливі теми тестування. Кожна нова тема повинна бути пов'язана із уже внесеною в програму дисципліною.

Таблиця 3.9- Сутність «Тести»

Код	Лічильник
Тема	Числовий
Назва	Текстовий
Час	Числовий

Сутність «Тести» зберігає список доступних тестів для проходження.

Таблиця 3.10 - Сутність «Філії»

Код	Лічильник
Філія	Текстовий

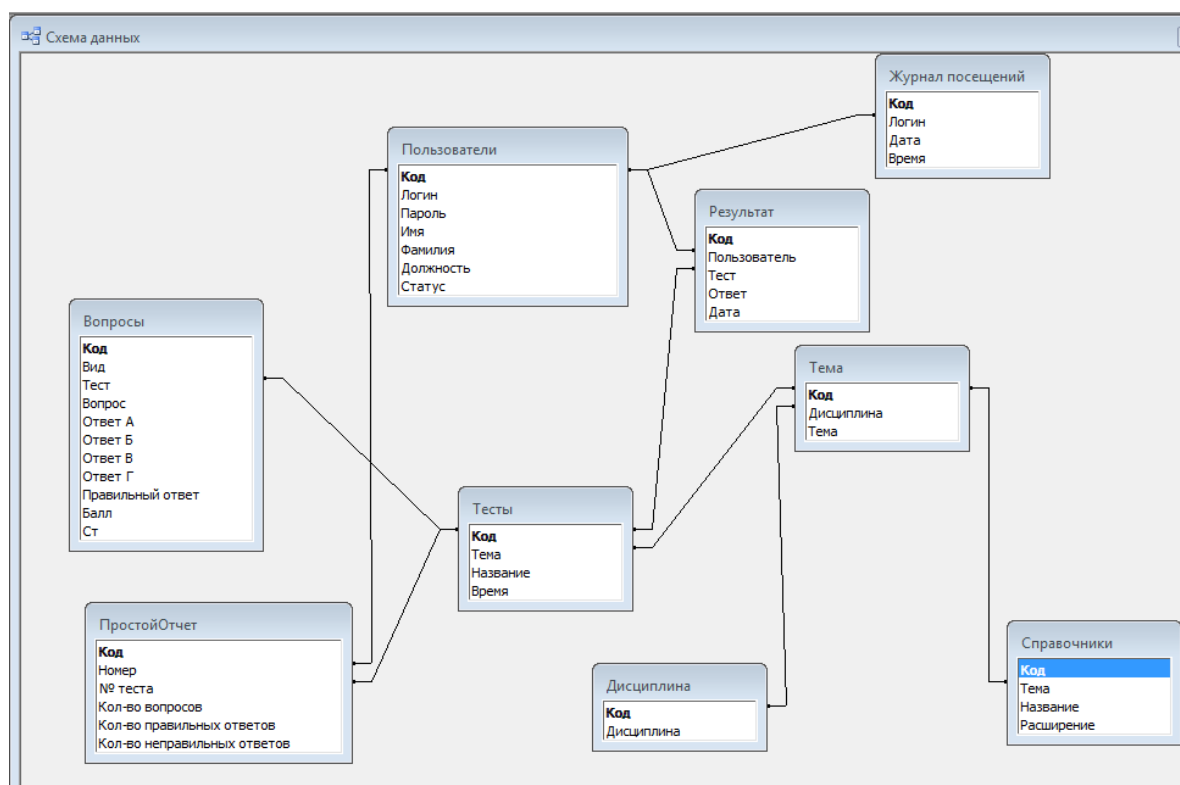


Рисунок 3.7 - Фізична схема бази даних

3.5 Програмна реалізація

3.5.1 Загальні положення (дерево функцій і сценарій діалогу)

Взаємодія із програмою відбувається через графічний інтерфейс, а саме діалогові вікна Windows Form. Технологія Windows Form розроблена компанією Microsoft і підтримує всі стандартні побудови користувацьких візуальних інтерфейсів, для платформи Windows.

Windows Forms дозволяє розробляти інтелектуальні клієнти. Інтелектуальний клієнт це додаток з багатим графічним інтерфейсом, простий в розгортанні й відновленні, здатний працювати при наявності або відсутності підключення до Інтернету й використовувати більш безпечний доступ до ресурсів на локальному комп'ютері в порівнянні із традиційними додатками Windows

У зв'язку з тим, що при рішенні завдання використовується технологія обробки інформації в режимі діалогу, взаємодія користувача із програмою можна представити у вигляді схеми діалогу (рис. 3.8).

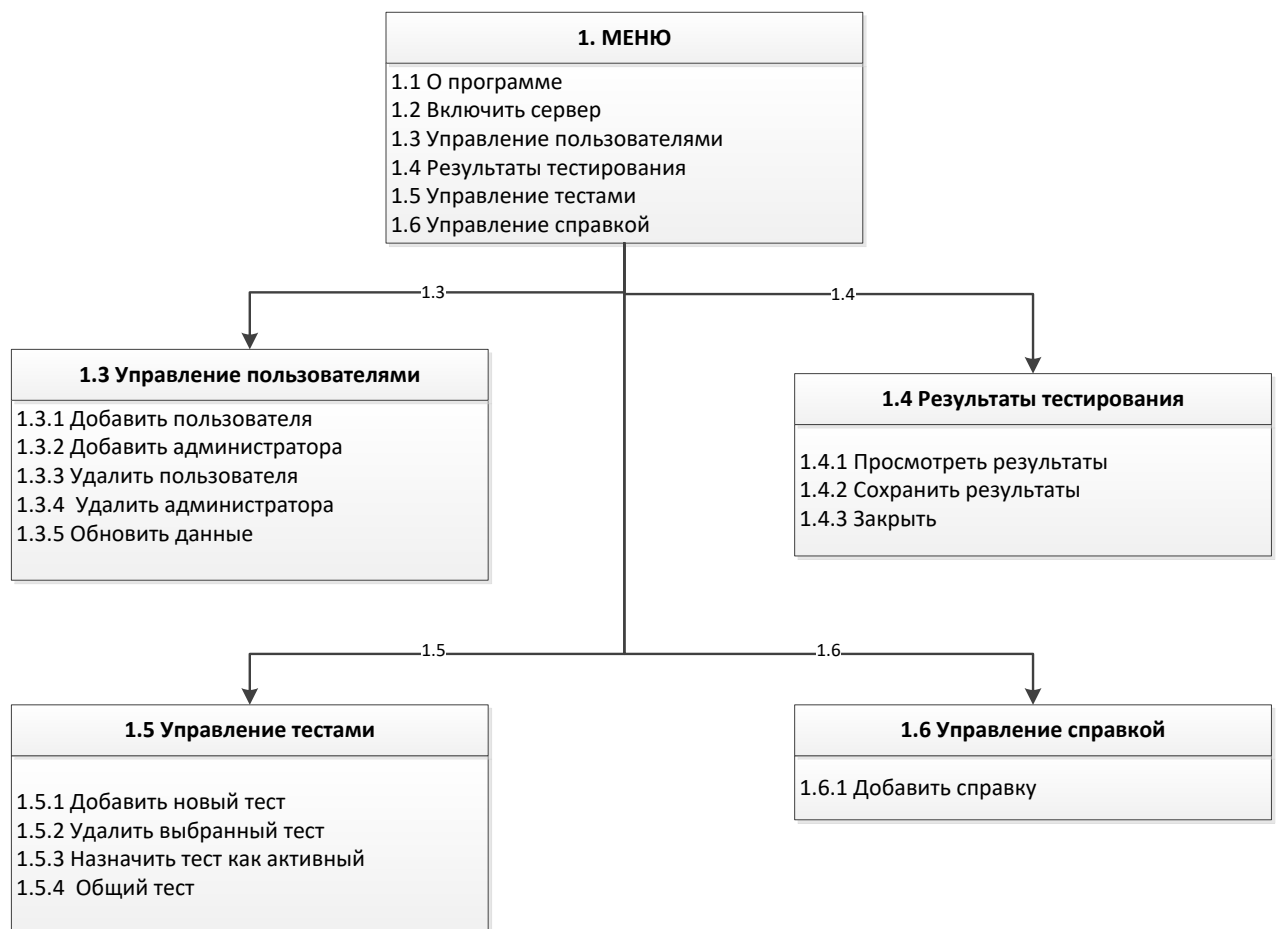


Рисунок 3.8 - Дерево діалогів



Рисунок 3.9 - Дерево функций

3.5.2 Опис програмних модулів

Технологія WCF дозволяє розмістити всі основні модулі програмного забезпечення в серверній його частині й надавати доступ до них через інтерфейси. Тому в даному пункті опишемо основні модулі, що призначено для реалізації повноти функціоналу програмного комплексу, що реалізується в дипломному проекті.

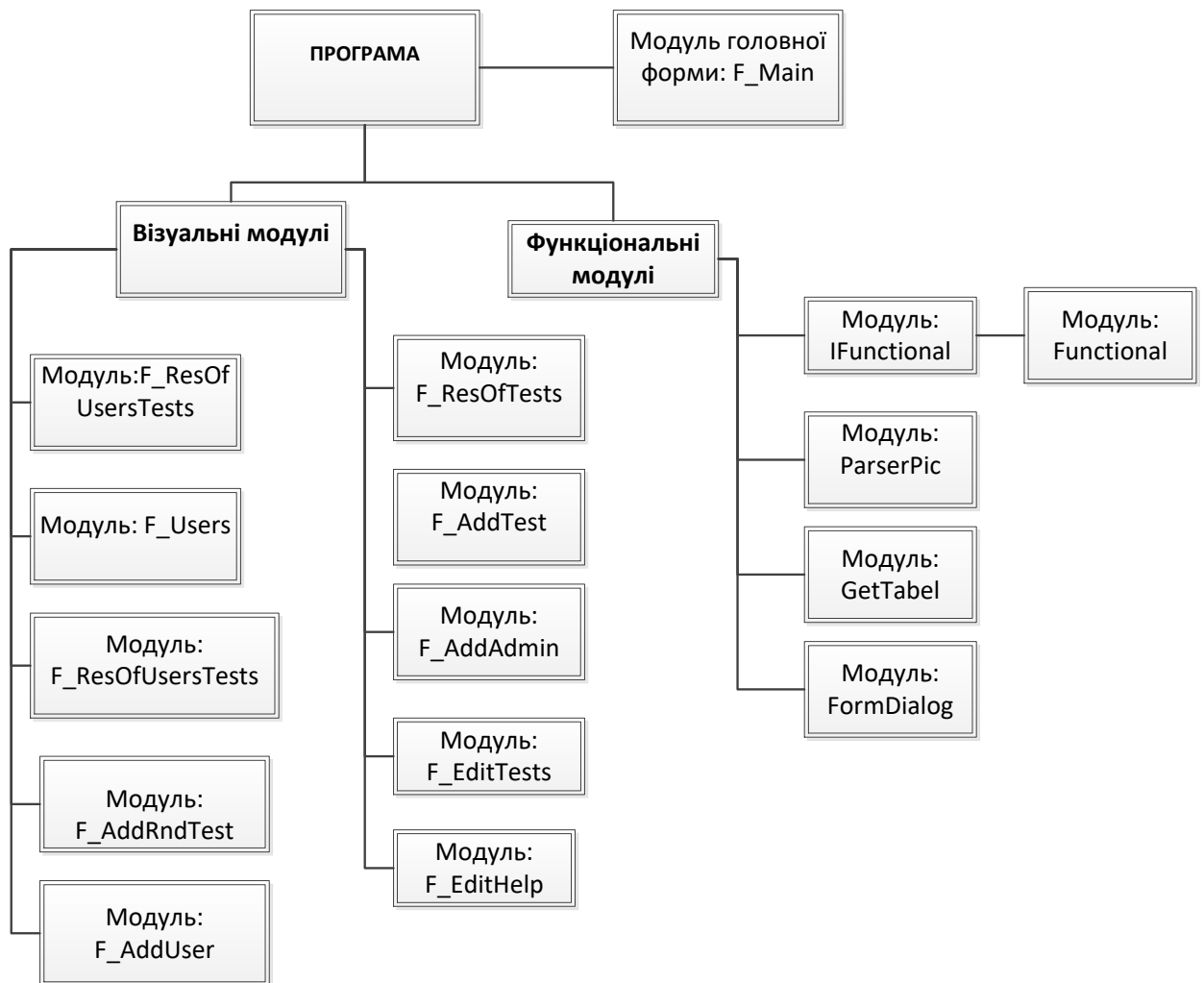


Рисунок 3.10 – Дерево модулів

Таблиця 3.11 - Опис модулів ПК

Назва	Опис
F_Main	Головне вікно програми
Функціональні модулі	
IFunctional	Інтерфейс методів для клієнтської частини програми
Functional	Набір методів по роботі із програмою
ParserPic	Клас відповідальний за парсинг зображень в XML форму для тестів
GetTabel	Клас , що має методи для зв'язку із БД
FormDialog	Зберігає глобальні змінні
Візуальні модулі	
F_ResOfUsersTests	Форма відображення детальної інформації з тесту користувача
F_Users	Форма виводу списку користувачів, надає функції їх редагуванню
F_ResOfTests	Форма, що відображає користувачів і їхні тести. Так само надаємо методи по пошуку й фільтрації даних
F_AddTest	Форма додавання нових тестів
F_AddAdmin	Форма додавання адміністраторів
F_EditTests	Форма редагування тестів
F_EditHelp	Форма додавання довідкової інформації
F_AddRndTest	Форма налаштування загального тесту
F_AddUser	Форма додавання користувачів

3.5.3 Реалізація доступу до бази даних

Для розробки системи автоматизованого тестування, використовувалося рішення OleDb, що являє собою драйвер для встановлення зв'язку програмного забезпечення з базою даних. Одна з основних переваг такого підходу є те, що користувальницький додаток спілкується з фізичною базою

даних через менеджер драйверів, фактично нічого не знаючи про тип використовуваної БД (спілкування менеджера драйверів і самих драйверів певною мірою стандартизовано, так що з погляду користувача всі джерела даних мають практично однакові властивості).

OLE DB - це відкрита специфікація, розроблена на основі специфікації ODBC. На відміну від технології ODBC, що була створена для доступу до реляційних баз даних, технологія OLEDB більш універсальна, вона розроблена для доступу як до реляційних, так і до нереляційних джерел даних.

Компоненти OLE DB складаються з таких компонентів:

- провайдери даних (data providers), які представляють свої дані;
- споживачі даних (data consumers), які використовують дані;
- сервісні компоненти (service components) - обробляють і транспортують дані.

OLEDB містить у собі зв'язок з ODBC, що дає можливість розроблювачам використовувати ODBC-драйвери реляційних БД, широко розповсюджені в цей час.



Рисунок 3.11 - Архітектура ODBC

3.6 Організація збору, передачі, обробки інформації

Розроблена система складається із клієнт-серверного додатка й бази даних. База даних забезпечує інформаційну частину системи, тобто, всі дані якими система оперує, зберігається безпосередньо там.

Клієнтський і серверний додатки написані мовою програмування C# з використанням графічного інтерфейсу Windows Form.

Серверна частина містить у собі основні механізми роботи системи. Містить у собі основні функції по обробці даних і надання функціональних інтерфейсів для клієнтської частини програми.

Взаємодія з користувачем, а точніше безпосередньо основну функціональну частину закладену в системі - проходження тестів, здійснюється клієнтським додатком, що за допомогою інтерфейсів, побудованих за технологією WCF, здійснює зв'язок із Сервером.

Нижче приведемо приклади реалізації деяких інтерфейсів і механізмів клієнтського й серверного додатків.

Фрагмент коду 1 - Приклад коду взаємодії системи з базою даних

```
static class GetTable
{
    static OleDbCommand OleDbComm = new OleDbCommand();
    static OleDbConnection OleDbConn = new OleDbConnection();
    static OleDbDataAdapter OleDbAdapter = new OleDbDataAdapter();
    // Устанавливает соединение с базой данных
    static public DataTable Get(string query)
    {
        OleDbConn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=DBDate.accdb;Persist Security Info=True";
        OleDbComm.Connection = OleDbConn;

        DataTable MainDataTable = new DataTable();
        OleDbAdapter.SelectCommand = new OleDbCommand(query, OleDbConn);
        OleDbAdapter.Fill(dt);
        try
        {
            OleDbConn.Open();
        }
        Catch {}
        OleDbConn.Close();
        return MainDataTable;
    }
}
```

Даний метод, за допомогою бібліотеки oledb здійснює доступ додатка до бази даних.

Тепер можна привести деякі основні методи й запити, які використовує програма в ході роботи.

Фрагмент коду 2 - Метод реєстрації користувачів

```
public Boolean Registration(String Pass, String Name, String Surname, String
eMail, Int32 Group, Int32 Branch)
{
    try
    {
        GetTabele.Get("INSERT INTO Пользователи (Логин, Пароль, Имя, Фамилия,
eMail, Группа, Филиал) VALUES ('" +
Surname + Name + "'", '" + Pass + "'", '" + Name + "'", '" + Surname + "'",
'" + eMail + "'", " + Group + " ", " + Branch + ")");
        return true;
    }
    catch { return false; }
}
```

Як видно з методу, за допомогою оператора INSERT, мови SQL, у таблицю «Користувачі» вносяться реєстраційні дані користувача: логин, пароль, ім'я, прізвище, eMail, категорія й філія.

Фрагмент коду 3 - Метод авторизації

```
{
    DataTable _dt = new DataTable();
    _dt = (GetTabele.Get("SELECT Код, Логин FROM Пользователи WHERE Логин='" +
Login + "' AND Пароль='" + Pass + "'"));
    DataTable dt = new DataTable("Пользователь");
    dt.Columns.AddRange(new DataColumn[2] { new DataColumn("Код"), new
DataColumn("Логин") });

    for (int i = 0; i < _dt.Rows.Count; i++)
        dt.Rows.Add(_dt.Rows[i][0], _dt.Rows[i][1]);
    return dt;
}
```

Даний метод за допомогою операторів SELECT і WHERE шукає в базі даних пари логін/пароль. Якщо збіг знайдений, вертаються дані про користувача, після чого можливий вхід у систему. На рис. 3.9 зображено даний процес у вигляді блок-схеми.

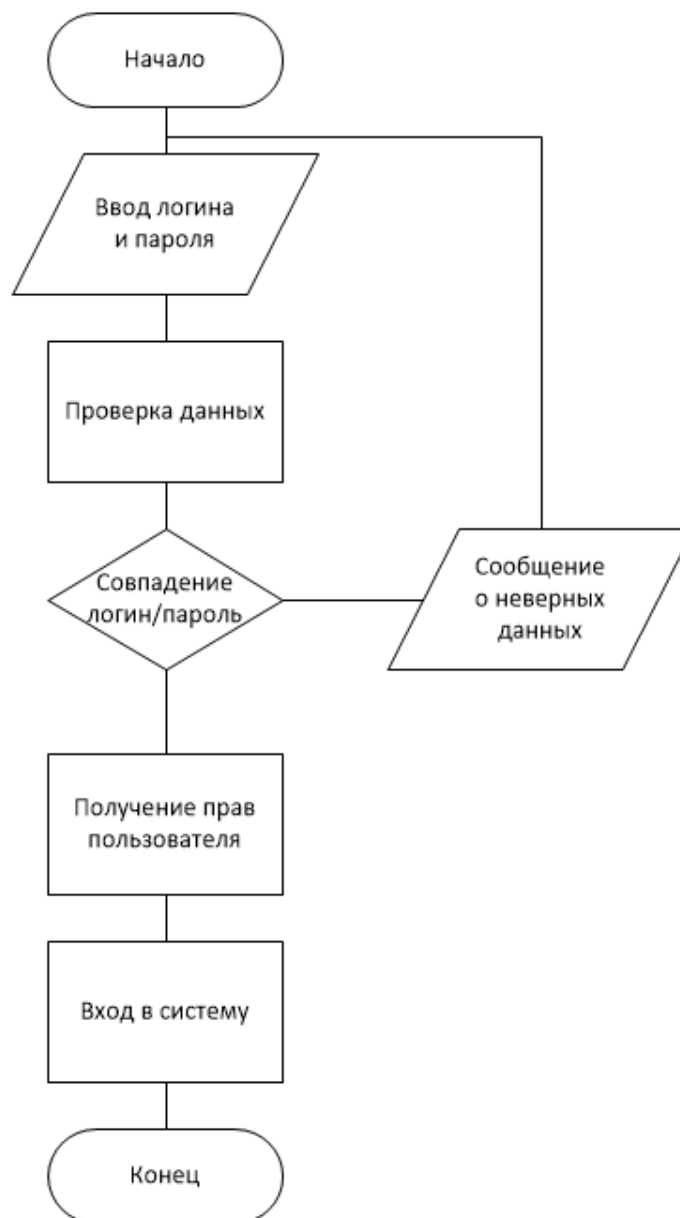


Рисунок 3.12 - Авторизація

Фрагмент коду 4 - Запит вибірки користувачів

```

"SELECT р.Код, р.Фамилия, р.Имя, р.Логин, р.eMail, g.Должность, b.Филиал FROM
    Пользователи р, Группы g, Филиалы b, р LEFT JOIN g ON р. Должность =g.Код, р
    LEFT JOIN b ON р.Филиал=b.Код WHERE р.Статус=false E
  
```

Представлений запит здійснює вибірку даних з таблиць: Користувачі, Посади, Філії й за допомогою оператора LEFT JOIN і включає їх в один результуючий потік.

Фрагмент коду 5 - Метод запуску сервера

```
public void StartServer()
{
    var thread2 = new Thread(() =>
    {
        if (_bool == false)
        {
            serviceHostClient = new
            ServiceHost(typeof(Functional),
            new Uri("http://localhost:7792/Testing"));
            serviceHostClient.AddServiceEndpoint(typeof(IFunctional), new
            BasicHttpBinding(), "");

            serviceHostClient.Open();
            _bool = true;
        }
        else
        {
            serviceHostClient.Close();
            _bool = false;
        }
    });
    thread2.Start();
}
```

Даний метод за допомогою служб WCF здійснює запуск прослуховування з'єднання в пошуку вхідних запитів із клієнтських додатків.

3.7 Проектування форм, інтерфейс

Інтерфейс розробленої системи реалізовано стандартними методами, що надаються середовищем Microsoft Visual Studio на базі Windows Forms.

При розробці інтерфейсу застосовувалися лише стандартні компоненти, вони описані в багатьох джерелах, зокрема в літературі [9, 31, 33, 36, 43]. З огляду на останнє, вважаємо детальний опис компонентів недоречним в роботі.

Приклад розробки однієї з форм приведено на рис. 3.13.

Загалом інтерфейс підтримує всі сучасні правила формування GUI на основі WinForms типу додатку.

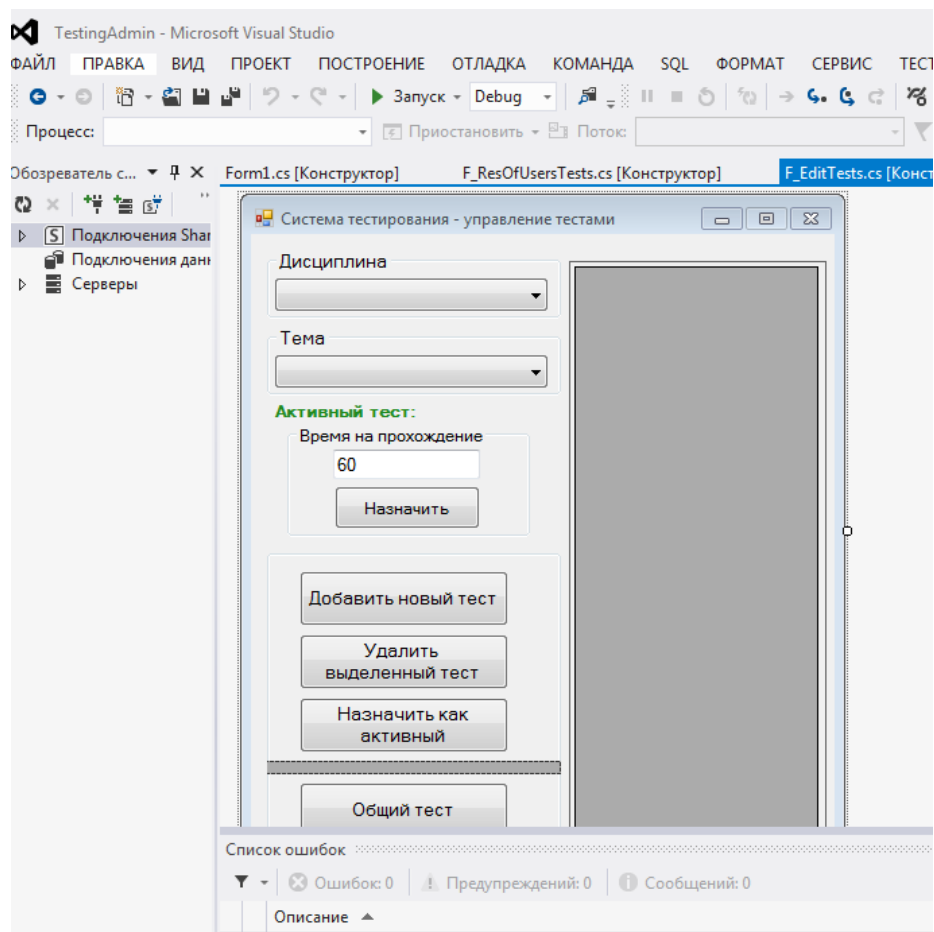


Рисунок 3.13 – Розробка інтерфейсу. Форма керування тестами

4 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

4.1 Планування і реалізація тестів по проекту

Тестування модуля, або автономне тестування (module testing, unit testing), - контроль окремого програмного модуля, зазвичай в ізольованому середовищі (т. Е. Ізольовано від всіх інших модулів). Тестування модуля іноді включає також математичне доказ.

Комплексне тестування (system testing) - контроль і / або випробування системи по відношенню до вихідних цілям. Комплексне тестування є процесом контролю, якщо воно виконується в моделюється середовищі, і процесом випробування, якщо виконується в середовищі реальної, життєвої.

Тестування прийнятності (acceptance testing) - перевірка відповідності програми вимогам користувача.

На сьогоднішній день автоматизовано більшість етапів розробки програмного забезпечення, в тому числі і етап тестування. Однак у вітчизняній практиці тестування програмних засобів відведена незаслужено маленька роль. Причинами можуть бути відсутність грошей на придбання дорогих CASE-засобів, що підтримують всі етапи розробки ПО, в тому числі тестування, або небажання тримати таку штатну одиницю, як фахівець з тестування. Зазвичай набувають засоби автоматизації проектування (створення ER-моделей, інформаційних моделей та ін.) І програмування (автоматичне створення БД на основі ER-моделі, створення інтерфейсу на основі ER-моделі, візуальне програмування).

Найбільш важливі види тестування ПЗ перераховані нижче.

1 Функціональне - тестування можливостей системи, її реакція на ті чи інші ситуації. Зазвичай результат тестування (реакція системи) порівнюється з постановкою завдання, при невідповідності фіксується помилка.

2 Регресійне - перевірка повноти реалізованих функцій системи в порівнянні з попередньою версією програмного продукту.

3 Навантажувальне - тестування роботи системи на піковому навантаженні, при цьому робиться висновок про продуктивність системи. Умовою для навантажувального тестування є виконання випробувань на одній і тій же конфігурації системи.

4 Контроль після виправлення (зворотний зв'язок). Цей вид тестування має на увазі під собою перевірку вже виправлених помилок.

5 Стресове тестування - перевірка реакції системи на поза штатні ситуації. Прикладом може служити перевірка системи на відновлення працездатності після відключення живлення на сервері бази даних.

6 Адаптаційне тестування - перевірка коректності перекладу програмного забезпечення на інший національну мову.

Розроблено спеціальні стандарти і вимоги, які регламентують поняття якості ПЗ. Стандарт ISO-8402 [20], присвячений опису систем і забезпечення якості програмного забезпечення, під якістю розуміє "сукупність характеристик програмного продукту, забезпечують його здатність задовольняти встановлені і передбачувані потреби клієнта". З іншого боку існує ряд стандартів [22,23,24], що визначають структуру ЖЦ і правила розробки ПО.

Але проблема полягає в тому, що дотримання стандартів не гарантує успіху програмного продукту, доказом цього можна привести ряд програм розроблених абсолютно поза будь-якими стандартами, часто навіть непрофесійними розробниками, які, при цьому, мають велику популярність і широке коло користувачів (прикладом можуть служити додатки з AndroidПлейМаркет). Основним фактором популярності і якості програмного продукту є передбачення вимог користувачів і вдале поєднання функціональності та простоти використання.

Одним зі складних етапів в життєвому циклі програмного забезпечення, як і раніше, залишається етап тестування.

Основною проблемою тестування є неможливість процесу з абсолютним результатом. Хоча в теорії тестування розглядається метод повного «абсолютного» тестування, але на практиці цей метод застосувати неможливо.

Ще в 1979 р. Майерс (Myers) [25] описав певний нескладний алгоритм, який містив тільки один цикл і кілька операторів розгалуження, для кодування даного алгоритму в більшості популярних мов програмування знадобилося б не більше 20 рядків коду. Але дана програма з досліджень автора має більш 1000000000000000 шляхів виконання.

Хоча процес тестування всіляко намагаються формалізувати, в той же час він має ряд евристичних ознак і вимагає комплексних зважених творчих підходів. Недарма публікації на цю тему часто розглядають питання психології, філософії, методів творчого пошуку і інші напрямки, на перший погляд здаються досить віддаленими від програмування і програмної інженерії. При проектуванні тестів програмного продукту перед фахівцем стоїть завдання вибору з великої кількості напрацьованих методів, в даній ситуації головне завдання тестувальника вибрати найбільш вдалий набір тестів, який може бути реалізований реального часу і при цьому буде мати максимальний ефект. Але навіть при найбільш вдалого вибору і правильної реалізації не існує ніяких гарантій, що всі помилки будуть знайдені [19].

Процес тестування у загальному вигляді може бути представлений таким чином

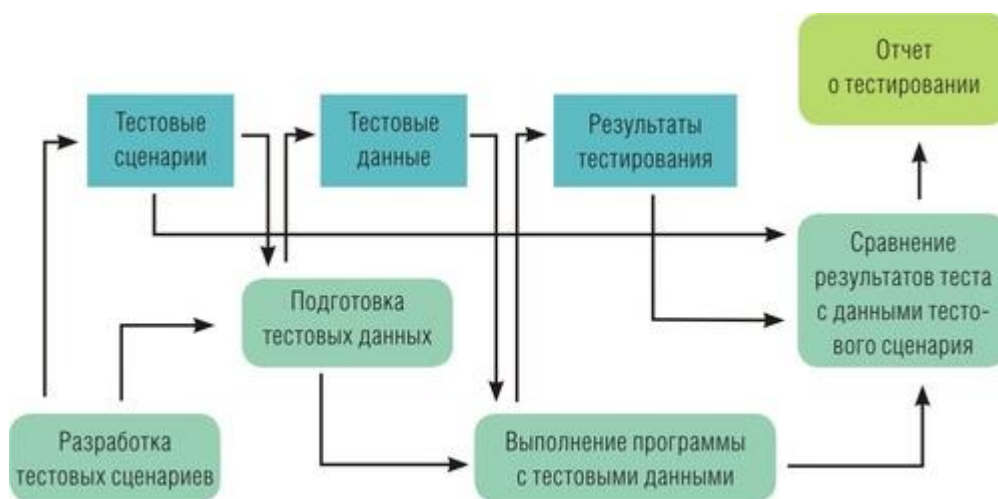


Рисунок 4.1 - Процес тестування

Для розроблюваної системи планується застосовувати ряд підходів.

План тестування

Спочатку - передпроектне тестування, це тестування на моделі. Для цього застосовується об'єктна модель (фрагменти наведені в роботі), модель даних та інші структурні та алгоритмічні моделі. Цей етап дозволяє попередньо протестувати систему ще до її створення. Визначити відповідність результату очікуванням користувачів, окреслити рамки системи, спрогнозувати можливі проблеми.

1 Тестування на моделі - розроблену модель (використовується функціональний і об'єктний підходи) пред'являють всім учасникам процесу: користувачеві, адміністратору. Кожен учасник пише звіт за результатами аналізу моделі. Обробка звітів допомагає розробнику оцінити правильність підходів, вибір стратегії і функціональний набір.

Аналіз цього анкетування є тестуванням глобальної структури системи і її відповідність очікуванням цільових користувачів. Після обробки даних отриманих від анкетування користувачів можуть вноситися корективи в структуру ІС, її функціональний набір, інтерфейсні зміни.

Для цього типу тесту застосовується спосіб опитування і інтерв'ювання. Досвід і практика тестерів і розробників показує, що на первинному етапі збору даних і перевірки адекватності побудованої моделі опитування є найбільш швидким, великим і практичним підходом. Другою перевагою цього способу є продукування артефактів (документи: анкети, інтерв'ю, опитувальні листи), які допомагають протоколювати і документувати процес.

2 Тестування БД

На основі спроектованої моделі даних (рис. 3.6) створюється фізична БД і розміщується на сервері. Тестування БД включає:

- 1) Оцінку структури БД і рівень нормалізації
- 2) Повноту представлення інформації за заявкою в системі

- 3) Можливість установки фізичної зв'язку між ІС і БД
- 4) Адекватність відпрацювання запитів.

Саме такий перелік пунктів і підхід до тестування обраний для БД ґрунтуючись на рекомендаціях фахівців (розробників БД). У загальній теорії тестування баз даних пропонується наступний опорний план:

- 1) Data mapping: відповідність проектної документації
- 2) ACID властивості транзакцій: атомарність, послідовність, ізоляція і міцність
- 3) Цілісність даних: останній стан даних БД однаково відображається для всіх складових компонентів системи (адміністратор, користувач, виконавець)
- 4) Точність реалізації бізнес-правил: реалізація бізнес-логіки на рівні БД (якщо такий підхід застосовувався).
- 5) Перевірка працездатності: реакція на транзакції, відпрацювання запитів в БД

З огляду на той факт, що в нашому випадку логіка в БД не використовується (бізнес-логіка реалізується на програмному рівні) пункт 4 вище представленого списку ми втрачаємо. Всі інші вимоги в нашому плані тестування враховані.

3 Функціональне тестування

Уже в ході розробки буде застосовуватися функціональне тестування, на завершальному етапі - комплексне і приймальним тестування.

Схема операційного тестування наведена на рис. 4.2

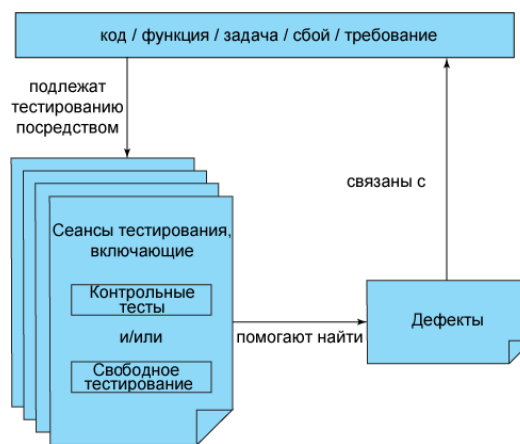


Рисунок 4.2 - Схема операційного тестування

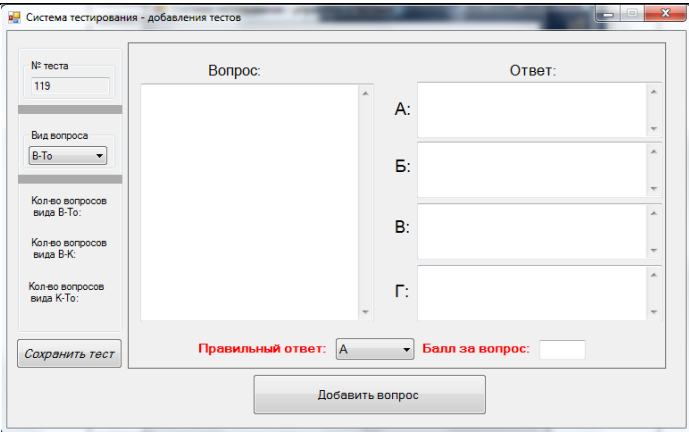
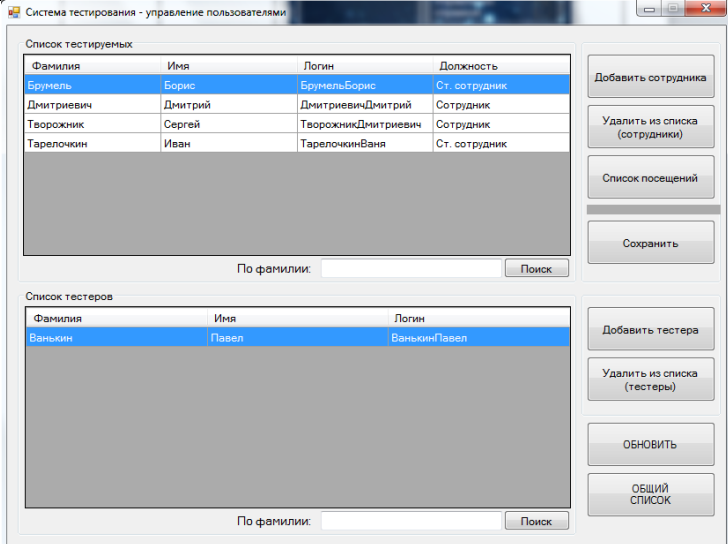
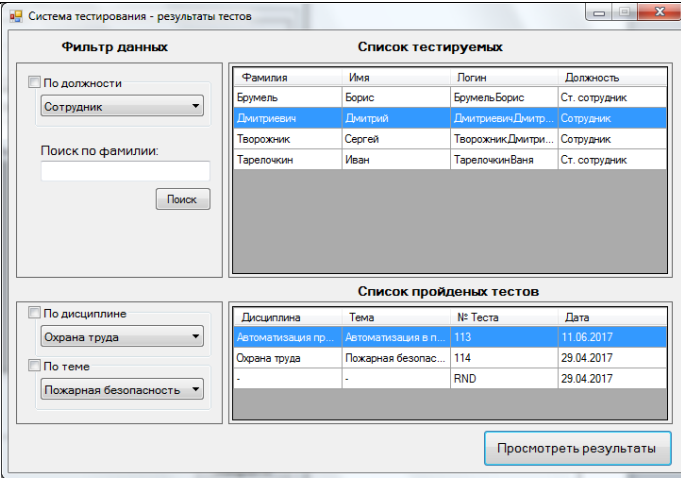
Динамічне операційне тестування передбачає запуск програмного коду програми. Якщо бути більш точним, динамічне тестування складається з [26]:

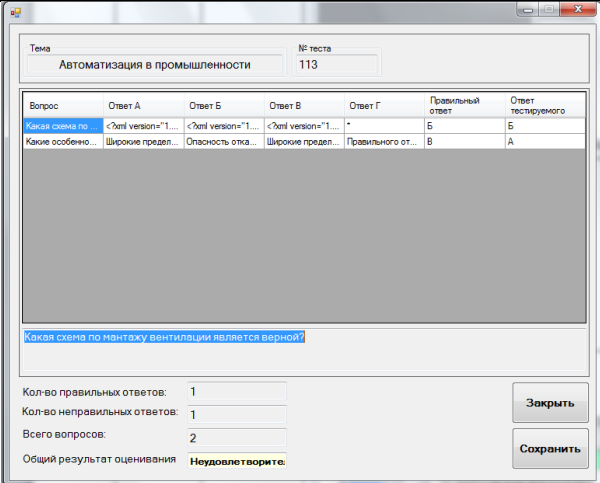
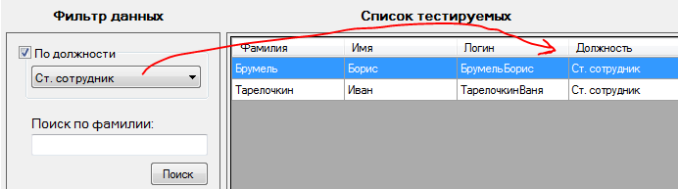
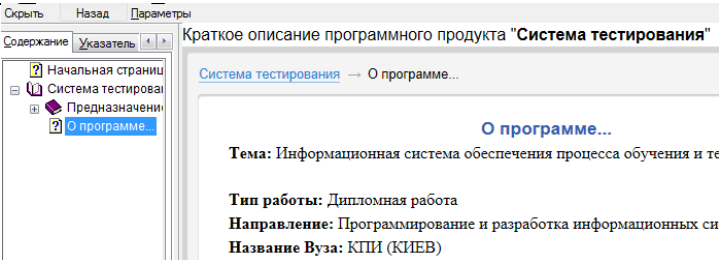
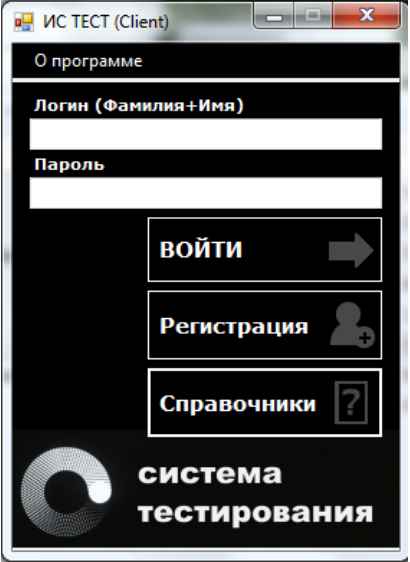
1. запуску системи або підсистеми;
2. виклику необхідних функціональних елементів або модулів;
3. порівняння через графічний інтерфейс користувача поведінки системи з очікуваним результатом поведінки.

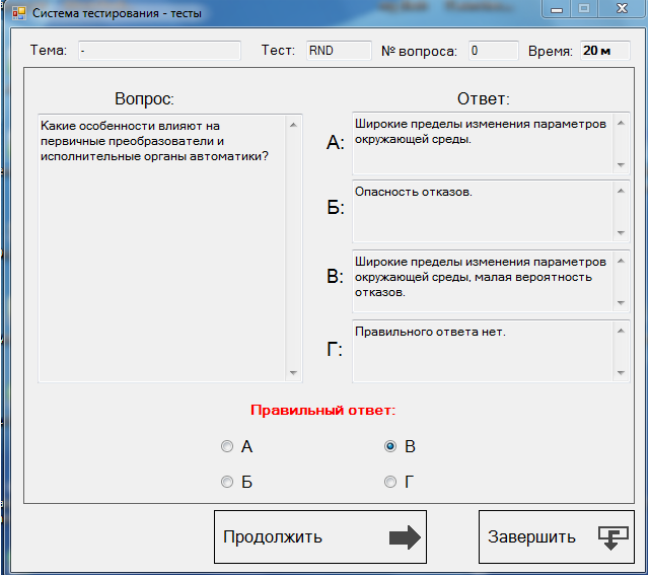
Процес операційного динамічного тестування може описуватися наступним чином. Отриманий результат в цій таблиці це намальовані екрани, які повинні відповідати реальним в процесі тестування.

Таблица 4.1 - Протокол функціонального тестування

Сценарій тесту	Очікуваний результат	Отриманий результат (Скріншот)
Запуск програми (частина адміністратора)	Завантажується головна форма програми	

Вибір режиму створення тесту	Завантажується редактор тесту	
Введено всі питання та обрано зберегти	Тест зберігається в системі	
Обрано режим перегляду результатів	Завантажується вікно з списком зареєстрованих користувачів	
Обрано конкретного користувача	Система надає данні про проходження тестів даним користувачем	

		
Фільтрація та пошук по списку користувачів	Система адекватно реагує на обрані критерії	
Виклик довідки	Завантажується довідникова система	
Запуск програми (частина користувача)	З'являється вікно авторизації	

Введено коректну інформацію	Завантажується головна форма програми (частина користувача)	
-----------------------------	---	--

Процес тестування повинен підтвердити або спростувати адекватну реакцію на дії користувача за допомогою вибору інтерфейсних елементів і введення даних. Виявлені в ході тестування помилки виправляються, і починається новий виток тестування. Ітерації повторюються до тих пір, поки результат тестування не буде оцінений як задовільний (підкреслимо - це не означає що помилок немає, це означає - що в більшості випадків система реагує прогнозовано і адекватно на запити і дії користувача).

4.2 Загальні характеристики розробленого комплексу

Технічні характеристики програми:

- інсталяція не потрібна, достатньо скопіювати програму на жорсткий диск;

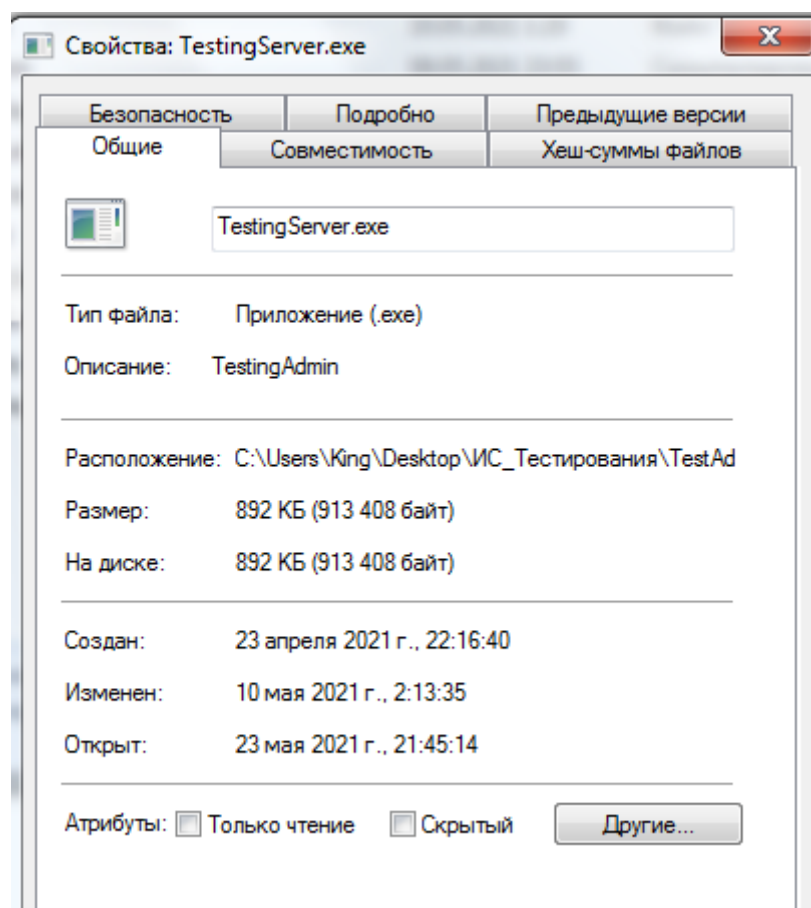
- розмір виконуваного файлу:

- 1) Client.exe - 182 Кбт

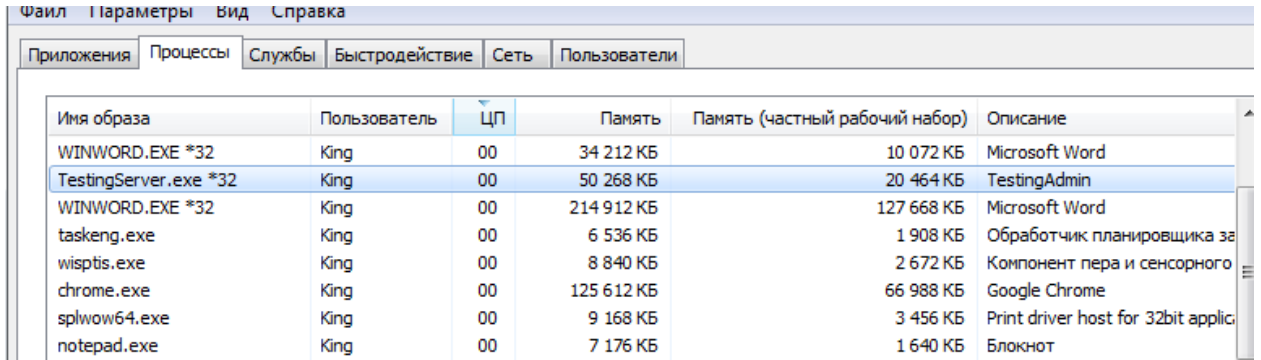
- 2) TestingServer.exe (admin) – 892 Кбт

- обсяг в оперативній пам'яті під час роботи 20 464 Кбт

Виконуваний файл



Процес



Файл Параметры Вид Справка					
Приложения Процессы Службы Быстродействие Сеть Пользователи					
Имя образа	Пользователь	ЦП	Память	Память (частный рабочий набор)	Описание
WINWORD.EXE *32	King	00	34 212 КБ	10 072 КБ	Microsoft Word
TestingServer.exe *32	King	00	50 268 КБ	20 464 КБ	TestingAdmin
WINWORD.EXE *32	King	00	214 912 КБ	127 668 КБ	Microsoft Word
taskeng.exe	King	00	6 536 КБ	1 908 КБ	Обработчик планировщика за
wisptis.exe	King	00	8 840 КБ	2 672 КБ	Компонент пера и сенсорного
chrome.exe	King	00	125 612 КБ	66 988 КБ	Google Chrome
splwow64.exe	King	00	9 168 КБ	3 456 КБ	Print driver host for 32bit applic
notepad.exe	King	00	7 176 КБ	1 640 КБ	Блокнот

Рисунок 4.3 - Розміри в пам'яті розробленої програми

Програма запускається на виконання або подвійним клацанням лівої кнопки миші по файлу TestingServer.exe в провіднику Windows, або натисканням правої кнопки «миші» по вказаного файлу і вибором пункту меню "Відкрити". Запускається частина програми для керування тестами та процесом тестування.

Для безпосереднього тестування необхідно завантажити програму-клієнт, та виконати попередню авторизацію.

Тестування програми показало адекватну реакцію на різноманітні ситуації. Неможливо гарантувати, що проведені тести покривають всі можливі ситуації. Варто відзначити, що з великою ймовірністю існують неперевірені ситуації і можливо некоректна реакція програми на ситуацію. Повне тестування провести дуже важко через велику варіативність можливих станів та ситуацій. Проведено тестування основних функцій і набір різних бізнес-ситуацій (неповне функціональне тестування)[26].

Навігація по інтерфейсним елементам працює адекватно, про це свідчать екранні форми наведені в протоколі тестування та приклади описані в наступному пункті.

4.3 Інструкція для користувача

Інструкція користувача й опис серверної частини представлено нижче:

Після запуску програми користувачеві відкривається доступ до його основної функціональної форми (Рис. 4.4).

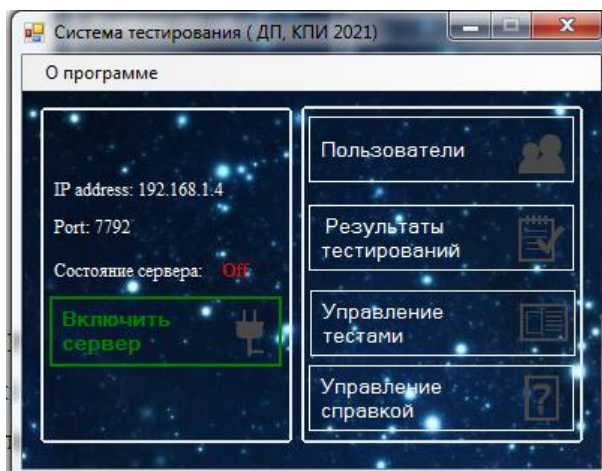


Рисунок 4.4 - Головна форма програми (адміністратор)

Далі опишемо основні функціональні частини головної форми: «Включить сервер», «Пользователи», «Результаты тестирований», «Управление тестами», «Управление справкой».

При натисканні на пункт меню «Включить сервер», будуть запущені служби, що забезпечують можливість клієнт-серверного з'єднання.

Пункт меню « Пользователи » здійснює перехід і відкриття форми керування користувачами й адміністраторами системи (Рис. 4.5).

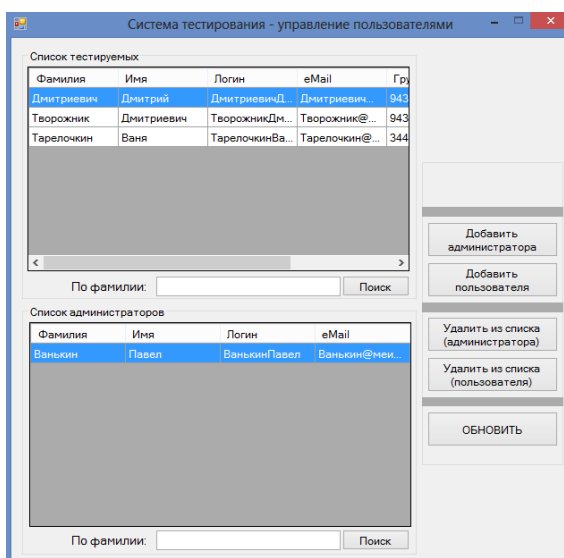


Рисунок 4.5 - Форма керування користувачами

При переході по пункту меню «Результаты тестирований» відкриється форма, що представляє собою список протестованих співробітників і

пройдених ними тестів. Так само в даній формі доступна опція фільтрації (сортування) даних по різних категоріях.

Рисунок 4.6 - Форма результатів тестування

Пункт меню «Управление тестами» дає можливість користувачеві вносити та вилучати тести в/з системи. Так само в даній формі можна встановлювати час проходження тестів і призначати активний тест.

Рисунок 4.7 - Форма керування тестами

З форми керування тестами можна виділити один найважливіший об'єкт функціонування - додавання нового тесту (Рис. 4.8).

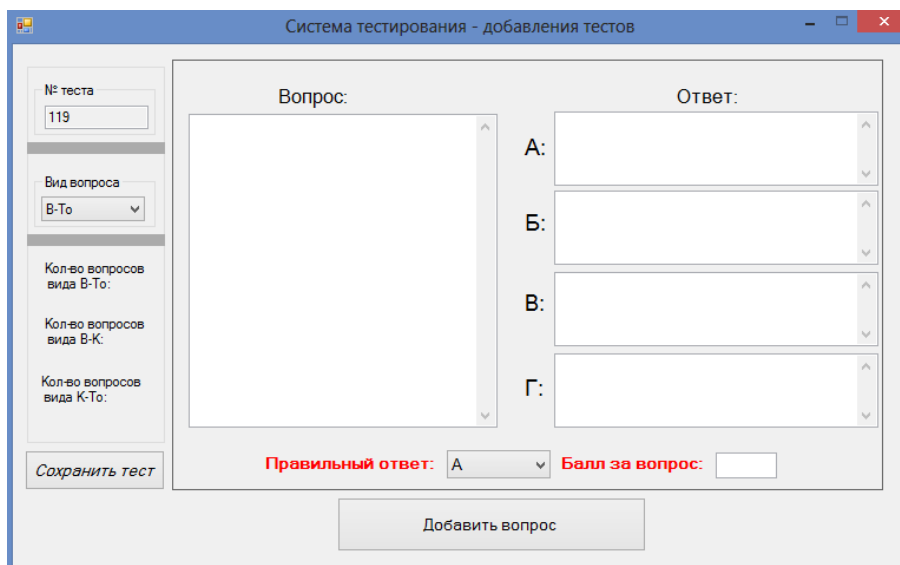


Рисунок 4.8 - Додавання нового тесту

Як видно з рисунка, питання тестів можуть складатися із трьох категорій (видів):

В-То: Питання - Текстова відповідь

В-К: Питання - Картинка (Малюнок)

К-То: Картинка (Малюнок) - Текстова відповідь.

Після додавання питань і вірної відповіді, необхідно додати оціночний бал за нього, після чого він внесеється в загальний список питань для тесту.

Остання функціональна частина головної форми, це «Управление справкой». Дана частина дозволяє вносити в систему різну довідкову документацію, що, надалі, можна буде переглядати в клієнтських додатках.

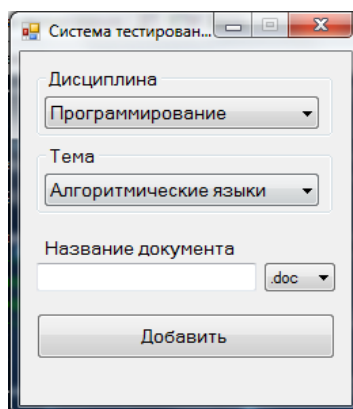


Рисунок 4.9 - Додавання довідкової документації

Після опису серверної частини, можна перейти до опису користувацького інтерфейсу клієнтської частини розробленого програмного забезпечення.

При відкритті клієнтського додаток з'явиться форма авторизації (Рис. 4.9) з можливістю реєстрації нового аккаунта в системі (Рис. 4.10).

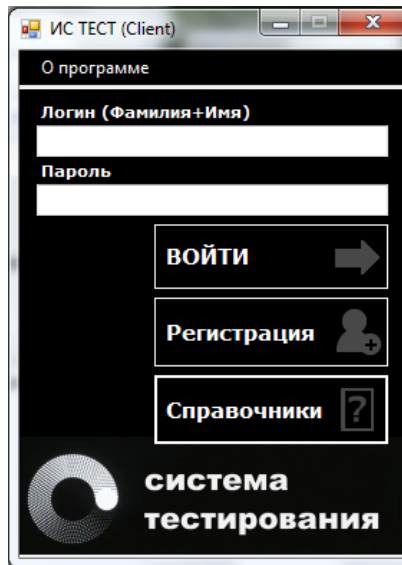


Рисунок 4.10 - Вікно авторизації

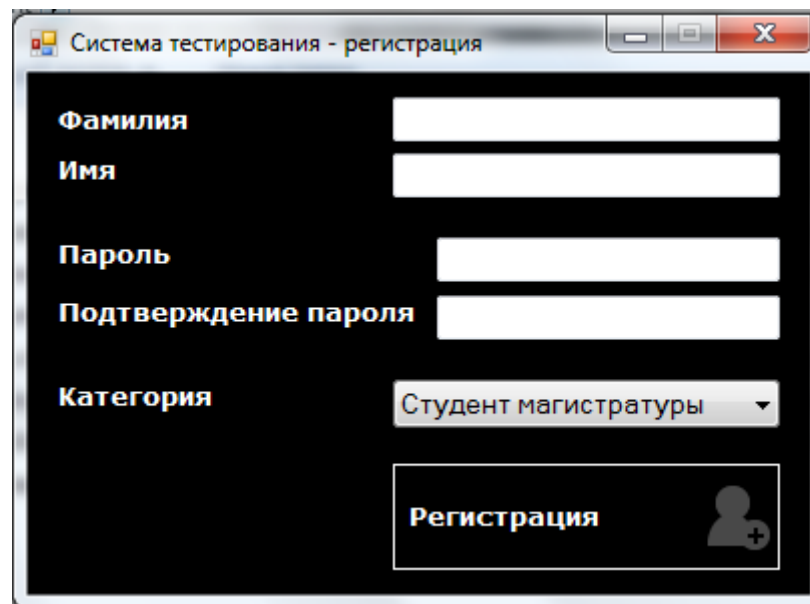


Рисунок 4.11 - Вікно реєстрації

Після успішної авторизації або реєстрації, користувач зможе почати проходження тестів. Логін користувача формується автоматично з його ім'я й прізвища.

Система тестирования - тесты

Тема: Алгоритмические языки Тест: 119 № вопроса: 0 Время: 60 м

Вопрос:

Языком программирования называется

Ответ:

А: система шифрования символов

Б: совокупность средств и правил перевода текста с формального языка на естественный

В: совокупность средств и правил представления алгоритма в виде, пригодном для выполнения

Г: язык, понятный программистам

Правильный ответ:

☐ А ☒ В

☐ Б ☐ Г

Продолжить ➡ Завершить ⬇

Рисунок 4.12 - Вікно проходження тестів

Після завершення процесу тестування результати автоматично фіксуються в системі та є доступними для обробки і аналізу адміністратором системи.

ВИСНОВКИ

Використання електронних посібників та автоматизованих систем тестування знань як сучасних інформаційних технологій в освіті дозволить внести суттєві зміни в процес навчання. Особливо актуальним є впровадження подібних засов навчання під час пандемії та вимушених кроків переходу на дистанційну форму освіти.

Аналіз питань використання сучасних засобів показує, що сьогодні системи тестування знань досить популярні і поширені. При цьому використовуються як локальні системи так і мережеві, в тому числі системи що працюють в мережі Інтернет. У той же час не всі навчальні заклади мають постійний стабільний доступ до мережі та ще й поширений на всі комп'ютерні класи (аудиторії). Тому до сьогодні залишаються актуальними і локальні програмні розробки які можуть використовуватися в локальній мережі НЗ або інтегруватися як модулі в загальну систему методичного забезпечення навчального процесу.

Системи тестування відносяться до досить складних програмних продуктів і вимагають для реалізації залучення ряду технологій та компонентів. Основою системи є база даних, але для вдалого та зручного використання необхідно продумати і ряд моментів пов'язаних з інтерфейсом, представленням даних, особливістю реалізації окремих елементів системи.

Розроблений в проекті програмний комплекс для тестування має ряд переваг та особливостей, в порівнянні з існуючими аналогами. До переваг варто віднести:

- гнучку системи типів та способу представлення інформації в питаннях;
- вбудований редактор тестів, що дає можливість проектувати авторські тести;

- наявність системи інформаційної підтримки для учнів/студентів з можливістю класифікації інформації та співвіднесення інформації до конкретних тем, що суттєво пришвидшує підготовку;

- можливість гнучкого керування тестами на стороні адміністратора;

- вбудована система аналізу результатів з можливістю імпорту результатів в зовнішні файли (зокрема талиці-Ексель та pdf-формат)

Робота включає 4 розділи основної частини.

В першому, аналітичному, розділі розглядається цільова організація (навчальний заклад та вищий навчальний заклад) як замовник продукту. Досліджуються питання моніторингу та оцінювання рівня знань як невід'ємної частини навчального процесу.

В технологічному розділі описується обраний для реалізації поставленої задачі інструментарій та основні технології, що планується використовувати ході реалізації.

Третій розділ присвячено безпосередньо програмній реалізації. Тут описано всі етапи проектування та розробки програмного додатку. Приділяється особлива увага моделюванню на всіх етапах реалізації, приводять конкретні програмні рішення, що використано в ході розробки продукту.

Останній, четвертий, розділ присвячено питанням тестування та відлагодження програмного додатку. В останньому пункті наводиться коротка інструкція користувача, демонструються екранні копії створених форм.

Зміст роботи відповідає темі та поставленим задачам.

Практична значимість дипломної роботи полягає в можливості застосування розробленої програми в реальних умовах для тестування студентів та учні. Практична складова роботи – програмний додаток повністю функціональний і може бути адаптованим під різні потреби.

Застосування розробленого додатку суттєво скоротить час викладача не лише на етапі проведення тестів, а й, головне, на етапі оцінювання і

формування статистики по результатам. Вся інформація про результати тестувань зберігається в базі даних і може бути піддана додатковій обробці та аналізу, за умови розширення функціонального набору програмного додатку відповідними модулями обробки результатів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України про освіту// Відомості Верховної Ради (ВВР), 2017, № 38-39, ст.380.
2. Карпушина Т. Инновационные подходы в обучении// педагогический клуб «Открытый урок» / "Технология" 20.05.2020
3. Горбунова Л. И., Субботина Е. А. Использование информационных технологий в процессе обучения // Молодой ученый. — 2015. — №4. — С. 544-547.
4. Вигерс Карл, Разработка требований к программному обеспечению, Пер, с англ. - М.: Издательско-торговый дом "Русская Редакция", 2008. -576с.: ил
5. Вендров А.М., CASE-технологии. Современные методы и средства проектирования информационных систем - М.: Финансы и статистика, 2007 г, 456 стр.
6. Гвоздева Т. В., Б. А. Баллод, Проектирование информационных систем, М, Издательство: Феникс, 2009 г., 512 стр.
7. Голицына О. Л., И. И. Попов, Н. В. Максимов, Т. Л. Партыка, Информационные технологии, М, Издательство Инфра-М, 2009 г., 608 стр.
8. Котляров В. П., Т. В. Коликова, Основы тестирования программного обеспечения, Издательства: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2011 г., 288 стр.
9. Пирогов В. Ю., Информационные системы и базы данных. М, Организация и проектирование, Издательство: БХВ-Петербург, 2009 г.528 стр.
10. Реляционные базы данных: практические приемы оптимальных решений. — СПб.: БХВ-Петербург, 2016 — 400с.:ил;
11. Емельянова Н. З., Партыка Т. Л., И. И. Попов, Проектирование информационных систем, М, Издательство: Форум, 2009 г., 432 стр.
12. Кузин А. В., С. В. Левонисова, Базы данных, М, Издательство: Академия, 2008 г., 320 стр.

13. Кузнецов С. Д., Основы баз данных, М, Издательства: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2007 г., 488 стр.
14. Симионов Ю.Ф., Боромотов В.В. Информационный менеджмент. — Ростов н. Д: Феникс, 2012, 250с., ил.;
15. Гвоздева Т. В., Б. А. Баллод, Проектирование информационных систем, М, Издательство: Феникс, 2009 г., 512 стр.
16. Илюшечкин В. М. , Основы использования и проектирования баз данных, М, Издательство Юрайт, 2010 г., 224 стр.
17. Пирогов В. Ю., Информационные системы и базы данных. М, Организация и проектирование, Издательство: БХВ-Петербург, 2009 г. 528 стр.
18. Мельников В. П., Клейменов С. А., Петраков А. М. Информационная безопасность и защита информации, М., Академия, 2011 г., 336 стр.
19. Сем Канер и др. Тестирование программного обеспечения. Фундаментальная концепция менеджмента бизнес – приложений: Пер. с англ./Сем Канер, Джек Фолк, Енг Кек Нгуен. –К.: Издательство «ДиаСофт», 2001. – 544 с.
20. ISO 8402:1994 Управление качеством и обеспечение качества.
21. Bertrand Meyer, Seven Principles of Software Testing. IEEE Computer, August 2008. IEEE Computer Society, 2008. All rights reserved. Reprinted with permission.
22. Стандарты типичного цикла разработки ПО (software development life cycle – SLDC), такие как IEEE,
23. Стандарт цикла разработки ISO – 12207
24. Стандарты CMMI.
25. Майерс. Г. Искусство тестирования программ: /Пер с англ. под ред. Б.А. Позина – М.: «Финансы и статистика», 1982. - 172 с.

26. Чипіга А. Ф., Информационная безопасность автоматизированных систем, М, Издательство: Гелиос АРВ, 2010 г., 336 стр.
27. Пирогов В. Ю., Особенности разработки автоматизированных систем на основе БД./ Организация и проектирование, Издательство: БХВ-Петербург, 2014 г. 528 стр.
28. Илюшечкин В. М. , Объектный подход в проектировании систем, М, Издательство Юрайт, 2016 г., 440 стр.
29. Грабер, М. Введение в SQL: Самое простое введение в SQL с полезными примерами и пробными объяснениями и др. / Пер. с англ. В. Ястребов – Издательство «Лори», 2015. – 375 с. – ISBN – 5-85582-010-6.
30. Коннолли, Т., Бегг, К., Страчан, А. Базы данных: Проектирование, Реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг, А. Страчан – М.: Вильямс, 2013. – 1436 с. – ISBN 5-8459-0527-3, 0-201-70857-4.
31. Культин, Н. Б. Microsoft Visual C# в задачах и примерах./ Н.Б. Культин. – СПб.: БХВ-Петербург, 2009. – 320 с.: ил. – ISBN 978-5-9775-0410-2.
32. Михеева, Е.В. Информационные технологии в профессиональной деятельности: учеб. Пособие 2 издание. – М.: ТК Велби, Изд-во Проспект, 2012. – 448 с. – ISBN 978-5-482-01569-8.
33. Нейгел, Кристиан, Ивѐн, Билл, Глинн, Джей, Скиннер, Морган, Уотсон, Карли. C# 2012 и платформа .NET 4.0 для профессионалов: Пер. с англ. / Кристиан Нейгел, Билл Ивѐн, Джей Глинн, Морган Скиннер, Карли Уотсон.– М.: Вильямс, 2013. – 1376 с. : ил. – ISBN 978-5-8459-1317-3 (рус.).
34. Балабанова Л.В., Сердак О.В. Отбор персонала через тестирование: учебное пособие: - М.: ИД «Профессионал», 2014. – 512 с.
35. Агуров П.В. C#. Сборник рецептов. 2017, 429 стр..
36. Бишоп Д. C# в кратком изложении. 2005, 467 стр..
37. Реляционные базы данных: практические приемы оптимальных решений. — СПб.: БХВ-Петербург, 2009 — 400с.:ил;

38. Баженова И. Ю. , Основы проектирования приложений баз данных, Издательства: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2008 г., , 328 стр.
39. Предметно-ориентированные экономические информационные системы, М, Издательство: Финансы и статистика, 2007 г., 224 стр.
40. ГОСТ ЕСПД 19.102-77 “Стадии разработки ”
41. Дейтел Х. С# в подлиннике. Наиболее полное руководство. 2006, 1057 стр..
42. Дубовцев А. Microsoft.NET. Наиболее полное руководство. 2004, 701 стр..
43. ИНТУИТ - Разработка Windows-приложений на основе Visual C# 2012.
44. Касимов, Ильяс. Информационные технологии в управлении образованием/ Ильяс Касимов//Педагогическая техника. - № 2, 2015г.
45. Квашнин Е.Г. Создание единой информационной образовательной среды образовательного учреждения/Егор Квашнин //Педагогическая техника. - № 4, 2017г.
46. Полат Е.С. Новые педагогические и информационные технологии в системе образования. – Київ: Освіта, 2016 г.
47. Справочник руководителя образовательного учреждения - № 10, октябрь 2010 г.
48. Черненко О.Н. Информационные технологии в учебном процессе: нормативное обеспечение, рекомендации из опыта работы. – Київ: Сучасний вчитель, 2018 г.
49. Менеджмент в управлении Вуза. Под редакцией Шамановой Т.И. – М., 2012 г.
50. Мониторинг в образовании. Майоров А.Н. – М.: Интеллект-Центр, 2005 г.

51. Гайдамакин Н.А. Автоматизированные информационные системы, базы и банки знаний. Вводный курс: Учебное пособие. – М.: Гелиос АРВ, 2016. – 368 с.
52. Горюнова М.А., Мелихова Л.Г., Мельников М.Г. Интернет-представительство образовательного учреждения. СПб.: ЛОИРО, 2013. – 60 с.
53. Д.А. Дмитриев ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ. Вектор науки ТГУ. 4(7). 2011 89
54. В.Е. Ланкин, Г.В. Горелова, В.Д. Сербин, Д.В. Арутюнова, А.В. Татарова, Г.Б. Баканов, Е.Л. Макарова Исследование и разработка организационных систем управления в высших учебных заведениях Монография. – Таганрог: Изд-во ТТИ ЮФУ, 2011. – 178 с.
55. А.В. Адаманский ИНФОРМАЦИОННАЯ МОДЕЛЬ УПРАВЛЕНИЯ ВЫСШИМ УЧЕБНЫМ ЗАВЕДЕНИЕМ, вестник Новосибирского государственного университета, серия Информационные технологии, т.8, 12.10.2015 [Электронный ресурс], - режим доступа: <http://old.softmotions.com/ru/index.php/news/30>.
56. Гайдамакин Н.А. Автоматизированные информационные системы, базы и банки знаний. Вводный курс: Учебное пособие. – М.: Гелиос АРВ, 2002. – 368 с.

ДОДАТКИ

Додаток А Текст програми (фрагменти)

```
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;

namespace Testing
{
    public partial class F_AddAdmin : Form
    {
        public F_AddAdmin()
        {
            InitializeComponent();
        }

        private void btn_Reg_Click_1(object sender, EventArgs e)
        {
            if (tb_Surname.Text.Length > 0 && tb_Name.Text.Length > 0 && tb_Pass.Text.Length
>0)
            {
                if (tb_Pass.Text == tb_ConPass.Text)
                {
                    try
                    {
                        GetTabele.Get("INSERT INTO Пользователи (Логин, Пароль, Имя, Фамилия,
Старус) VALUES ('" +
tb_Surname.Text + tb_Name.Text + "', '" + tb_Pass.Text + "', '" + tb_Name.Text + "', '" +
tb_Surname.Text + "', true)");
                        MessageBox.Show("Регистарция прошла успешно");
                        tb_Surname.Clear();
                        tb_Name.Clear();
                        tb_Pass.Clear();
                        tb_ConPass.Clear();
                    }
                    catch { MessageBox.Show("Ошибка регистрации", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error); }
                }
                else MessageBox.Show("Пароли не совпадают", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            else MessageBox.Show("Не все поля заполнены", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

```

private void tb_Surname_KeyPress(object sender, KeyPressEventArgs e)
{
    char l = e.KeyChar;
    if ((l < 'a' || l > 'я') && l != '\b' && (l < 'A' || l > 'Я'))
    {
        e.Handled = true;
    }
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    public partial class F_EditTests : Form
    {
        DataTable dtSubject = new DataTable();
        DataTable dtTopic = new DataTable();
        DataTable dtTests = new DataTable();

        Dictionary<String, int> dicTopic = new Dictionary<string, int>();

        F_AddTest F_AddTest = new F_AddTest();

        public F_EditTests()
        {
            InitializeComponent();
        }

        private void btn_AddTest_Click(object sender, EventArgs e)
        {
            FormDialog.IdOfTopic = dicTopic[cmb_Topic.Text]; //ID темы

            if (F_AddTest.Created == true) F_AddTest.Close();
            F_AddTest = new F_AddTest();
            F_AddTest.Show();
        }

        private void F_EditTests_Load(object sender, EventArgs e)

```

```

{
    dtSubject = GetTabele.Get("SELECT * FROM Дисциплина");
    dtTopic = GetTabele.Get("SELECT * FROM Тема");
    dtTests = GetTabele.Get("SELECT * FROM Тесты");

    cmb_Subject.DataSource = dtSubject;
    cmb_Subject.DisplayMember = "Дисциплина";
    GetActiveTest();
    dgv_Main.Columns[0].Visible = false;

    FormDialog.lb_Activetest = lb_ActiveTest;
}
String activeTest;
//Изменяет значение комбобокса с темами в зависимости от значения комбобокса
Дисциплин
void ChangeTopic()
{
    cmb_Topic.Items.Clear();
    dicTopic.Clear();
    for (int i = 0; i < dtTopic.Rows.Count; i++)
    {
        if (dtTopic.Rows[i][1].ToString() ==
dtSubject.Rows[cmb_Subject.SelectedIndex][0].ToString())
        {
            dicTopic.Add(dtTopic.Rows[i][2].ToString(), (int)dtTopic.Rows[i][0]);
            cmb_Topic.Items.Add(dtTopic.Rows[i][2].ToString());
        }
    }
    if (dicTopic.Count > 0) cmb_Topic.SelectedIndex = 0;
    dgv_Main.Columns[0].Visible = false;
}
DataTable ChangeDataGrid()
{
    DataRow dr;
    DataTable dt = new DataTable();
    dt.Columns.Add("Код");
    dt.Columns.Add("№ теста");
    dt.Columns.Add("Время прохождения");

    ;
    for (int i = 0; i < dtTests.Rows.Count; i++)
    {
        try
        {
            if (dicTopic[cmb_Topic.Text] == Convert.ToInt32(dtTests.Rows[i][1]))
            {
                dr = dt.NewRow();
                dr["Код"] = dtTests.Rows[i][0];
                dr["№ теста"] = dtTests.Rows[i][2];
                dr["Время прохождения"] = dtTests.Rows[i][3];
                dt.Rows.Add(dr);
            }
        }
    }
}

```

```

        }
    }
    catch { }
}
return dt;
}

private void cmb_Subject_SelectedIndexChanged(object sender, EventArgs e)
{
    ChangeTopic();
}

private void cmb_Topic_SelectedIndexChanged(object sender, EventArgs e)
{
    dgv_Main.DataSource = ChangeDataGrid();
}

private void btn_AddActiveTest_Click(object sender, EventArgs e)
{
    SetActiveTest();
    GetActiveTest();
}

void SetActiveTest()
{
    for (int i = 0; i < dtTests.Rows.Count; i++)
    {
        if (dtTests.Rows[i][2].ToString() == dgv_Main[1,
dgv_Main.CurrentRow.Index].Value.ToString())
            File.WriteAllText("Type.kay", dtTests.Rows[i][0].ToString(), Encoding.Default);
    }
}

void GetActiveTest()
{
    string strfilename = @"Type.kay";
    FileStream stream = new FileStream(strfilename, FileMode.Open);
    StreamReader reader = new StreamReader(stream);
    String FILE = reader.ReadToEnd();
    reader.Close();

    for (int i = 0; i < dtTests.Rows.Count; i++)
    {
        if (dtTests.Rows[i][0].ToString() == FILE)
            lb_ActiveTest.Text = "Активный тест: " + dtTests.Rows[i][2].ToString();
    }
}

F_AddRndTest F_AddRndTest = new F_AddRndTest();
private void btn_CreatureRNDTest_Click(object sender, EventArgs e)
{
    if (F_AddRndTest.Created == true) F_AddRndTest.Close();
    F_AddRndTest = new F_AddRndTest();
}

```

```

        F_AddRndTest.Show();
    }

    private void btn_DeleteTest_Click(object sender, EventArgs e)
    {
        DialogResult dialogRes = MessageBox.Show("Выполнить операцию удаления?", "",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question);
        if (dialogRes == System.Windows.Forms.DialogResult.OK)
        {
            GetTabele.Get("DELETE * FROM Тесты WHERE Код=" + dgv_Main[0,
            dgv_Main.CurrentRow.Index].Value.ToString());
            dtTests = GetTabele.Get("SELECT * FROM Тесты");
            ChangeDataGrid();
        }
    }

    private void tb_EditTime_Click(object sender, EventArgs e)
    {
        if (tb_Time.Text.Length > 0)
        {
            GetTabele.Get("UPDATE Тесты SET Время=" + tb_Time.Text);
            dtTests = GetTabele.Get("SELECT * FROM Тесты");
            dgv_Main.DataSource = ChangeDataGrid();
        }
        else MessageBox.Show("Значение поля времени не может быть пустым", "",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net;
using System.Net.Mail;
using System.ServiceModel;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    public partial class F_Main : Form
    {
        F_Users F_Users = new F_Users();
        F_ResOfTests F_ResOfTests = new F_ResOfTests();
    }
}

```

```

F_EditTests F_EditTests = new F_EditTests();
F_EditHelp F_EditHelp = new F_EditHelp();

//переменные для работы сервера
Boolean _bool = false; //Для проверки включен ли сервер или нет
Boolean offOn = false; //Проверка статуса кнопки вкл/выкл
ServiceHost serviceHostClient;
public void StartServer()
{
    var thread2 = new Thread(() =>
    {
        if (_bool == false)
        {
            serviceHostClient = new
            ServiceHost(typeof(Functional),
            new Uri("http://localhost:7792/Testing"));
            serviceHostClient.AddServiceEndpoint(typeof(IFunctional), new
BasicHttpBinding(), "");

            serviceHostClient.Open();
            _bool = true;
        }
        else
        {
            serviceHostClient.Close();
            _bool = false;
        }
    });
    thread2.Start();
}

public F_Main()
{
    InitializeComponent();
    btn_On.ForeColor = Color.Green; //начальный цвет кнопки вкл/выкл сервера
    lb_OnOff.ForeColor = Color.Red; //Начальный цвет лейбла статуса сервера
    ButtonsImages(btn_On, "Run.png");

    ButtonsImages(btn_EditUsers, "Users.png");
    ButtonsImages(btn_ResOfTests, "ResTest.png");
    ButtonsImages(btn_EditTests, "Tests.png");
    ButtonsImages(btn_EdutUserHelp, "_Help.png");
}
void ButtonsImages(Button button1, String image)
{
    button1.Image = Image.FromFile(@"images\" + image + "", false);
    //button1.BackgroundImageLayout = ImageLayout.Zoom;
    // Align the image and text on the button.
    button1.ImageAlign = ContentAlignment.MiddleLeft;
    button1.TextAlign = ContentAlignment.MiddleRight;
    // Give the button a flat appearance.

```

```

        button1.FlatStyle = FlatStyle.Flat;
    }
    private void btn_Test_Click(object sender, EventArgs e)
    {
        //SendMail.Send("smtp.yandex.ru", "Warickus@yandex.ru", "WarickusTest@i.ua",
"TEST", "TEST");
        // String str = GetTabele.Get("SELECT tp.Тема FROM Тесты test, Тема tp, test LEFT
JOIN tp ON test.Тема=tp.Код WHERE test.Код=1").Rows[0][0].ToString();
        //DataTable dtPreAnswer = new DataTable();
        // dtPreAnswer = GetTabele.Get("SELECT Ответ FROM Результат WHERE
Пользователь=1 AND Тест=18 AND Код=(SELECT MAX(Код) FROM Результат)");
        List<int> indexOfTests = new List<int>();
        indexOfTests.Add(1); indexOfTests.Add(2); indexOfTests.Add(3);
        var value = indexOfTests.Find(item => item == 5);

    }

    private void btn_On_Click(object sender, EventArgs e)
    {
        if (offOn == false)
        {
            lb_OnOff.Text = "On";
            lb_OnOff.ForeColor = Color.Green;
            btn_On.ForeColor = Color.Red;
            btn_On.Text = "ВЫКЛЮЧИТЬ" + Environment.NewLine + "сервер";
            offOn = true;
            ButtonsImages(btn_On, "Sleep.png");
        }
        else
        {
            lb_OnOff.Text = "Off";
            lb_OnOff.ForeColor = Color.Red;
            btn_On.ForeColor = Color.Green;
            btn_On.Text = "Включить" + Environment.NewLine + " сервер";
            offOn = false;
            //
            ButtonsImages(btn_On, "Run.png");
        }
        StartServer();
    }
    string myHost = System.Net.Dns.GetHostName();
    string myIP;
    private void F_Main_Load(object sender, EventArgs e)
    {
        FormDialog.rtb_ListOfConn = rtb_ListOfConn;
        myIP = System.Net.Dns.GetHostByName(myHost).AddressList[0].ToString();
        lb_IP.Text = "IP address: " + myIP;
    }
    private void btn_EditUsers_Click(object sender, EventArgs e)
    {
        if (F_Users.Created == true) F_Users.Close();
    }

```

```

        F_Users = new F_Users();
        F_Users.Show();
    }

    private void btn_ResOfTests_Click(object sender, EventArgs e)
    {
        if (F_ResOfTests.Created == true) F_ResOfTests.Close();
        F_ResOfTests = new F_ResOfTests();
        F_ResOfTests.Show();
    }

    private void btn_EditTests_Click(object sender, EventArgs e)
    {
        if (F_EditTests.Created == true) F_EditTests.Close();
        F_EditTests = new F_EditTests();
        F_EditTests.Show();
    }

    private void btn_EdutUserHelp_Click(object sender, EventArgs e)
    {
        if (F_EditHelp.Created == true) F_EditHelp.Close();
        F_EditHelp = new F_EditHelp();
        F_EditHelp.Show();
    }

    private void oПорпаммеToolStripMenuItem_Click(object sender, EventArgs e)
    {
        try
        {
            System.Windows.Forms.Help.ShowHelp(this, @"help.chm");
        }
        catch { MessageBox.Show("Файл справки не найден", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Error); }
    }

    private void btn_EditUsers_MouseEnter(object sender, EventArgs e)
    {
        btn_EditUsers.ForeColor = Color.Black;
    }

    private void btn_EditUsers_MouseLeave(object sender, EventArgs e)
    {
        btn_EditUsers.ForeColor = Color.White;
    }

    private void btn_ResOfTests_MouseEnter(object sender, EventArgs e)
    {
        btn_ResOfTests.ForeColor = Color.Black;
    }

    private void btn_ResOfTests_MouseLeave(object sender, EventArgs e)
    {

```

```

        btn_ResOfTests.ForeColor = Color.White;
    }

    private void btn_EditTests_MouseEnter(object sender, EventArgs e)
    {
        btn_EditTests.ForeColor = Color.Black;
    }

    private void btn_EditTests_MouseLeave(object sender, EventArgs e)
    {
        btn_EditTests.ForeColor = Color.White;
    }

    private void btn_EdutUserHelp_MouseEnter(object sender, EventArgs e)
    {
        btn_EdutUserHelp.ForeColor = Color.Black;
    }
    private void btn_EdutUserHelp_MouseLeave(object sender, EventArgs e)
    {
        btn_EdutUserHelp.ForeColor = Color.White;
    }
    private void F_Main_FormClosed(object sender, FormClosedEventArgs e)
    {
        FormDialog.F_Logo.Close();
    }
}
}
namespace Testing
{
    partial class F_Main
    {
        /// <summary>
        /// Требуется переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен быть удален;
        иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Код, автоматически созданный конструктором форм Windows

```

```

/// <summary>
/// Обязательный метод для поддержки конструктора - не изменяйте
/// содержимое данного метода при помощи редактора кода.
/// </summary>
private void InitializeComponent()
{
    System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(F_Main));
    this.groupBox1 = new System.Windows.Forms.GroupBox();
    this.btn_EditUsers = new System.Windows.Forms.Button();
    this.btn_EdutUserHelp = new System.Windows.Forms.Button();
    this.btn_ResOfTests = new System.Windows.Forms.Button();
    this.btn_EditTests = new System.Windows.Forms.Button();
    this.label3 = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.lb_IP = new System.Windows.Forms.Label();
    this.btn_On = new System.Windows.Forms.Button();
    this.groupBox3 = new System.Windows.Forms.GroupBox();
    this.lb_OnOff = new System.Windows.Forms.Label();
    this.rtb_ListOfConn = new System.Windows.Forms.TextBox();
    this.panel2 = new System.Windows.Forms.Panel();
    this.label1 = new System.Windows.Forms.Label();
    this.menuStrip1 = new System.Windows.Forms.MenuStrip();
    this.oIporpammeToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
    this.groupBox1.SuspendLayout();
    this.groupBox3.SuspendLayout();
    this.menuStrip1.SuspendLayout();
    this.SuspendLayout();
    //
    // groupBox1
    //
    this.groupBox1.BackColor = System.Drawing.Color.Transparent;
    this.groupBox1.Controls.Add(this.btn_EditUsers);
    this.groupBox1.Controls.Add(this.btn_EdutUserHelp);
    this.groupBox1.Controls.Add(this.btn_ResOfTests);
    this.groupBox1.Controls.Add(this.btn_EditTests);
    this.groupBox1.Location = new System.Drawing.Point(186, 28);
    this.groupBox1.Name = "groupBox1";
    this.groupBox1.Size = new System.Drawing.Size(188, 231);
    this.groupBox1.TabIndex = 0;
    this.groupBox1.TabStop = false;
    //
    // btn_EditUsers
    //
    this.btn_EditUsers.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
    this.btn_EditUsers.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
    this.btn_EditUsers.Location = new System.Drawing.Point(5, 13);
    this.btn_EditUsers.Name = "btn_EditUsers";

```

```

this.btn_EditUsers.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
this.btn_EditUsers.Size = new System.Drawing.Size(177, 44);
this.btn_EditUsers.TabIndex = 0;
this.btn_EditUsers.Text = "Пользователи";
this.btn_EditUsers.UseVisualStyleBackColor = true;
this.btn_EditUsers.Click += new System.EventHandler(this.btn_EditUsers_Click);
this.btn_EditUsers.MouseEnter += new
System.EventHandler(this.btn_EditUsers_MouseEnter);
this.btn_EditUsers.MouseLeave += new
System.EventHandler(this.btn_EditUsers_MouseLeave);
//
// btn_EdutUserHelp
//
this.btn_EdutUserHelp.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.btn_EdutUserHelp.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
this.btn_EdutUserHelp.Location = new System.Drawing.Point(5, 179);
this.btn_EdutUserHelp.Name = "btn_EdutUserHelp";
this.btn_EdutUserHelp.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
this.btn_EdutUserHelp.Size = new System.Drawing.Size(177, 44);
this.btn_EdutUserHelp.TabIndex = 0;
this.btn_EdutUserHelp.Text = "Управление\r\nисправкой";
this.btn_EdutUserHelp.UseVisualStyleBackColor = true;
this.btn_EdutUserHelp.Click += new
System.EventHandler(this.btn_EdutUserHelp_Click);
this.btn_EdutUserHelp.MouseEnter += new
System.EventHandler(this.btn_EdutUserHelp_MouseEnter);
this.btn_EdutUserHelp.MouseLeave += new
System.EventHandler(this.btn_EdutUserHelp_MouseLeave);
//
// btn_ResOfTests
//
this.btn_ResOfTests.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.btn_ResOfTests.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
this.btn_ResOfTests.Location = new System.Drawing.Point(5, 72);
this.btn_ResOfTests.Name = "btn_ResOfTests";
this.btn_ResOfTests.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
this.btn_ResOfTests.Size = new System.Drawing.Size(177, 44);
this.btn_ResOfTests.TabIndex = 0;
this.btn_ResOfTests.Text = "Результаты тестирований";
this.btn_ResOfTests.UseVisualStyleBackColor = true;
this.btn_ResOfTests.Click += new System.EventHandler(this.btn_ResOfTests_Click);
this.btn_ResOfTests.MouseEnter += new
System.EventHandler(this.btn_ResOfTests_MouseEnter);
this.btn_ResOfTests.MouseLeave += new
System.EventHandler(this.btn_ResOfTests_MouseLeave);
//
// btn_EditTests
//

```

```

        this.btn_EditTests.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.btn_EditTests.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
        this.btn_EditTests.Location = new System.Drawing.Point(6, 129);
        this.btn_EditTests.Name = "btn_EditTests";
        this.btn_EditTests.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
        this.btn_EditTests.Size = new System.Drawing.Size(177, 44);
        this.btn_EditTests.TabIndex = 0;
        this.btn_EditTests.Text = "Управление\r\nтестами";
        this.btn_EditTests.UseVisualStyleBackColor = true;
        this.btn_EditTests.Click += new System.EventHandler(this.btn_EditTests_Click);
        this.btn_EditTests.MouseEnter += new
System.EventHandler(this.btn_EditTests_MouseEnter);
        this.btn_EditTests.MouseLeave += new
System.EventHandler(this.btn_EditTests_MouseLeave);
        //
        // label3
        //
        this.label3.AutoSize = true;
        this.label3.Font = new System.Drawing.Font("Times New Roman", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label3.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
        this.label3.Location = new System.Drawing.Point(6, 108);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(106, 15);
        this.label3.TabIndex = 18;
        this.label3.Text = "Состояние сервера:";
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Font = new System.Drawing.Font("Times New Roman", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label2.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
        this.label2.Location = new System.Drawing.Point(6, 79);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(58, 15);
        this.label2.TabIndex = 17;
        this.label2.Text = "Port: 7792";
        //
        // lb_IP
        //
        this.lb_IP.AutoSize = true;
        this.lb_IP.Font = new System.Drawing.Font("Times New Roman", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.lb_IP.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
        this.lb_IP.Location = new System.Drawing.Point(6, 54);
        this.lb_IP.Name = "lb_IP";
        this.lb_IP.Size = new System.Drawing.Size(60, 15);
        this.lb_IP.TabIndex = 16;
        this.lb_IP.Text = "IP address:";

```

```

//
// btn_On
//
this.btn_On.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.btn_On.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
this.btn_On.Location = new System.Drawing.Point(6, 133);
this.btn_On.Name = "btn_On";
this.btn_On.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
this.btn_On.Size = new System.Drawing.Size(155, 47);
this.btn_On.TabIndex = 15;
this.btn_On.Text = "Включить\r\nсервер";
this.btn_On.UseVisualStyleBackColor = true;
this.btn_On.Click += new System.EventHandler(this.btn_On_Click);
//
// groupBox3
//
this.groupBox3.BackColor = System.Drawing.Color.Transparent;
this.groupBox3.Controls.Add(this.lb_OnOff);
this.groupBox3.Controls.Add(this.btn_On);
this.groupBox3.Controls.Add(this.lb_IP);
this.groupBox3.Controls.Add(this.label3);
this.groupBox3.Controls.Add(this.label2);
this.groupBox3.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.groupBox3.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
this.groupBox3.Location = new System.Drawing.Point(12, 28);
this.groupBox3.Name = "groupBox3";
this.groupBox3.Size = new System.Drawing.Size(168, 231);
this.groupBox3.TabIndex = 19;
this.groupBox3.TabStop = false;
//
// lb_OnOff
//
this.lb_OnOff.AutoSize = true;
this.lb_OnOff.Font = new System.Drawing.Font("Times New Roman", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.lb_OnOff.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(192)))),
((int)(((byte)(0)))), ((int)(((byte)(0))))));
this.lb_OnOff.Location = new System.Drawing.Point(118, 108);
this.lb_OnOff.Name = "lb_OnOff";
this.lb_OnOff.Size = new System.Drawing.Size(21, 15);
this.lb_OnOff.TabIndex = 19;
this.lb_OnOff.Text = "off";
//
// rtb_ListOfConn
//
this.rtb_ListOfConn.Location = new System.Drawing.Point(423, 69);
this.rtb_ListOfConn.Multiline = true;
this.rtb_ListOfConn.Name = "rtb_ListOfConn";
this.rtb_ListOfConn.ReadOnly = true;

```

```

this.rtb_ListOfConn.Size = new System.Drawing.Size(151, 185);
this.rtb_ListOfConn.TabIndex = 2;
//
// panel2
//
this.panel2.BackColor = System.Drawing.SystemColors.AppWorkspace;
this.panel2.Location = new System.Drawing.Point(434, 28);
this.panel2.Name = "panel2";
this.panel2.Size = new System.Drawing.Size(166, 10);
this.panel2.TabIndex = 1;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.label1.Location = new System.Drawing.Point(420, 50);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(154, 16);
this.label1.TabIndex = 18;
this.label1.Text = "Список соединений";
//
// menuStrip1
//
this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.oПорпammeToolStripMenuItem});
this.menuStrip1.Location = new System.Drawing.Point(0, 0);
this.menuStrip1.Name = "menuStrip1";
this.menuStrip1.Size = new System.Drawing.Size(386, 24);
this.menuStrip1.TabIndex = 20;
this.menuStrip1.Text = "menuStrip1";
//
// oПорпammeToolStripMenuItem
//
this.oПорпammeToolStripMenuItem.Name = "oПорпammeToolStripMenuItem";
this.oПорпammeToolStripMenuItem.Size = new System.Drawing.Size(94, 20);
this.oПорпammeToolStripMenuItem.Text = "О программе";
this.oПорпammeToolStripMenuItem.Click += new
System.EventHandler(this.oПорпammeToolStripMenuItem_Click);
//
// F_Main
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.AutoSizeMode = System.Windows.Forms.AutoSizeMode.GrowAndShrink;
this.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage")));
this.ClientSize = new System.Drawing.Size(386, 276);
this.Controls.Add(this.rtb_ListOfConn);
this.Controls.Add(this.panel2);
this.Controls.Add(this.groupBox3);

```

```

        this.Controls.Add(this.groupBox1);
        this.Controls.Add(this.menuStrip1);
        this.Controls.Add(this.label1);
        this.MainMenuStrip = this.menuStrip1;
        this.MaximizeBox = false;
        this.Name = "F_Main";
        this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
        this.Text = "Система тестирования (ДП 2017)";
        this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.F_Main_FormClosed);
        this.Load += new System.EventHandler(this.F_Main_Load);
        this.groupBox1.ResumeLayout(false);
        this.groupBox3.ResumeLayout(false);
        this.groupBox3.PerformLayout();
        this.menuStrip1.ResumeLayout(false);
        this.menuStrip1.PerformLayout();
        this.ResumeLayout(false);
        this.PerformLayout();
    }

    #endregion

    private System.Windows.Forms.GroupBox groupBox1;
    private System.Windows.Forms.Button btn_ResOfTests;
    private System.Windows.Forms.Button btn_EditUsers;
    private System.Windows.Forms.Button btn_EditTests;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.Label lb_IP;
    private System.Windows.Forms.Button btn_On;
    private System.Windows.Forms.GroupBox groupBox3;
    private System.Windows.Forms.Panel panel2;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Label lb_OnOff;
    private System.Windows.Forms.Button btn_EdutUserHelp;
    private System.Windows.Forms.TextBox rtb_ListOfConn;
    private System.Windows.Forms.MenuStrip menuStrip1;
    private System.Windows.Forms.ToolStripMenuItem oIporammeToolStripMenuItem;
    }
}
using System;
using System.Collections.Generic;
using System.Data.OleDb;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Windows.Forms;
namespace Testing
{
    static class GetTabele

```

```

{
    /// <summary>
    /// Выбирает/задает данные с/в БД
    /// </summary>
    /// <param name="quary"></param>
    /// <returns></returns>
    static public DataTable Get(string quary)
    {
        OleDbCommand OleDbComm = new OleDbCommand();
        OleDbConnection OleDbConn = new OleDbConnection();
        OleDbDataAdapter OleDbAdapter = new OleDbDataAdapter();

        OleDbConn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=TestsDB.mdb;Persist Security Info=True";
        //OleDbConn.ConnectionString = FormDialogMethods.ConnectionString;
        OleDbComm.Connection = OleDbConn;

        DataTable dt = new DataTable();

        OleDbAdapter.SelectCommand = new OleDbCommand(quary, OleDbConn);
        // try
        {
            OleDbAdapter.Fill(dt);
        }
        // catch (Exception ex) { MessageBox.Show(ex.ToString()); }
        try
        {
            OleDbConn.Open();
        }
        catch
        {
            OleDbConn.Close();
            return dt;
        }
        OleDbConn.Close();
        return dt;
    }
}

```

Додаток Б Акт впровадження

ЗАТВЕРДЖУЮ

Керівник підприємства


(підпис)

Р.С. Булік

АКТ ВПРОВАДЖЕННЯ

результатів бакалаврської роботи Мельник В.С. на тему
«Автоматизована система тестування», яка виконана в
Національному технічному університеті України
«Київський політехнічний інститут ім. Ігоря Сікорського»
 («КПІ ім. Ігоря Сікорського») "11" травня 2021 р.

Нами, представниками кафедри автоматизації проектування енергетичних процесів і систем «КПІ ім. Ігоря Сікорського» та ПП «BULIK Bis Sp. z o.o.», даний акт складено про те, що для використання в розробках спеціалізованого програмного забезпечення ПП «BULIK Bis Sp. z o.o.» прийняті результати бакалаврської роботи Мельник В.С., а саме програмний засіб - комп'ютерна автоматизована система тестування у вигляді запису на CD, а також документацію програмного супроводу.

Представник кафедри АПЕПС «КПІ ім. Ігоря Сікорського»

Керівник дипломної роботи

(прізвище, ім'я, по батькові)

(підпис)

Колумбет Вадим Петрович

Представник ПП «BULIK Bis Sp. z o.o.»

Керівник підприємства

Булік Руслан Сергійович

(прізвище, ім'я, по батькові)


(підпис)

BULIK Bis Sp. z o.o.
ul. Aleje Jerozolimskie, nr 85, lok. 28
02-001 Warszawa
KRS: 7010844996 REGON: 381067581