

Міністерство освіти і науки України
Департамент науки і освіти Харківської облдержадміністрації
Харківське територіальне відділення МАН України

Відділення: комп'ютерні науки
Секція: технології програмування

АВТОМАТИЗОВАНА СИСТЕМА ПРОФЕСІЙНОЇ ОРІЄНТАЦІЇ З ВИКОРИСТАННЯМ СУЧАСНИХ ТЕСТОВИХ ТЕХНОЛОГІЙ

Роботу виконав:

Громовий Данило Володимирович

учень I курсу

ДНЗ «Харківське вище професійне училище №6»

Науковий керівник:

Оруджова Дар'я Ейвазівна,

викладач інформатики

ДНЗ «Харківське вище професійне училище №6»

м.Харків - 2021

АВТОМАТИЗОВАНА СИСТЕМА ПРОФЕСІЙНОЇ ОРІЄНТАЦІЇ З ВИКОРИСТАННЯМ СУЧАСНИХ ТЕСТОВИХ ТЕХНОЛОГІЙ

Громовий Данило Володимирович

Харківське територіальне відділення МАН України

ДНЗ «Харківське вище професійне училище №6» , І курс,

група ОТН-1-15/16;м.Харків

Оруджова Дар'я Ейвазівна, викладач інформатики

ДНЗ «Харківське вище професійне училище №6»

На сьогодні професійна орієнтація спрямована на досягнення збалансованості між професійними інтересами та можливостями людини і потребами суспільства в конкретних видах трудової діяльності, вона покликана сприяти цілеспрямованому розвитку особистості, здібностей людини, зростанню її професіоналізму, працездатності.

Професійна орієнтація розглядається у двох вимірах: як система і як процес.

Система профорієнтації передбачає вироблення детермінованого структурою суспільних потреб на основі особистих нахилів та здібностей людини її вибіркового ставлення до професій, що визначає її місце у системі суспільного поділу праці.

Процес профорієнтації – це формування і реалізація професійних нахилів молоді під впливом комплексу суспільних умов у період її соціального становлення у школі, професійних навчальних закладах, трудових колективах.

За взаємообумовленості, взаємозв'язку та взаємодоповнюваності системи і процесу профорієнтації якісної визначеності їм надає цільова спрямованість чинників: система орієнтована переважно на розв'язання проблеми вибору, процес профорієнтації – на професійне становлення особистості. І першим кроком у цій справі є тестування.

Ключові слова: тестування, автоматизована система, об'єктно-орієнтована модель, функціональна модель, зовнішнє сховище, програмний продукт.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	6
РОЗДІЛ I ЗАГАЛЬНІ ПОЛОЖЕННЯ	8
1.1 Опис предметного середовища	8
1.1.1 Опис процесу діяльності.	9
1.1.2 Опис функціональної моделі.	11
1.2 Огляд наявних аналогів.....	12
РОЗДІЛ II ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	13
2.1 Аналіз предметної області	13
2.2 Проектування системи	14
2.2.1 Проектування бази даних.....	14
2.2.2 Побудова об'єктно-орієнтованої моделі.	16
2.3 Математичне та алгоритмічне забезпечення	20
РОЗДІЛ III ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	21
3.1 Засоби розробки	21
3.2 Вимоги до технічного та програмного забезпечення	21
3.3 Керівництво користувача.....	22
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	30
ДОДАТОК А	32

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

1. ЗВО – Заклад Вищої Освіти;
2. ЗНО – Зовнішнє Незалежне Оцінювання;
3. UML – Unified Modeling Language;
4. ОПГ – Опитувальник Професійної Готовності.
5. VCL – Visual Component Library;
6. IDE – Integrated Development Environment;
7. ЗМІ – Засоби масової інформації.

ВСТУП

Актуальність. Вибір професії – це важливий крок у доросле життя. В першу чергу – це відповідальність перед батьками, перед суспільством, а найголовніше перед собою. Ця відповідальність полягає в тому, щоб обрати саме свій шлях і ним йти. Робота і все, що з нею пов'язане, займає більшу половину життя людини. Знайти себе в світі професії – означає отримати можливість гідно заробляти, відчувати себе потрібним людям, максимально повно розкрити свої здібності. І тут найважливіше – не схибити.

Старшокласнику-випускнику дуже складно самотійно зорієнтуватися в такому складному життєвому виборі, якщо йому заздалегідь не допомогли визначитися з бажаннями, кращою сферою діяльності, батьки або школа. Незважаючи на те, що існує величезна кількість матеріалів з профорієнтації та професійного самовизначення в інтернеті і друкованих виданнях.

Важливо зрозуміти, вибір професії – це важливий етап в житті кожної людини.

Формування системи профорієнтаційної роботи дійсно важливо і значимо. Єдина система моніторингу освітніх потреб, формування механізму управління кадровим потенціалом дозволить не тільки оперативно реагувати на зміни, що відбуваються на ринку праці, а й прогнозувати ситуацію на ньому, і в підсумку готувати саме тих фахівців, які будуть затребувані через п'ять – шість років [1].

Об'єктом дослідження є автоматизована система професійної орієнтації з використанням сучасних тестових технологій.

Предмет дослідження – методи проектування та розробки комп'ютерного додатку.

Задачами дослідження є аналіз предметного середовища, побудова функціональної моделі, огляд аналогів, аналіз предметної області, опис проектування автоматизованої системи, вибір інструментарію та методів реалізації.

РОЗДІЛ I ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Тест на профорієнтацію являє собою комплекс питань, відповідаючи на які максимально чесно, людина проходить об'єктивне тестування емоційних і професійних уподобань, яке дозволяє визначитися з вибором майбутньої професії. Тест на профорієнтацію допомагає виявити інтереси і схильність людини до певної сфери діяльності.

Профорієнтаційні тести розширюють кругозір людини у світі професій і визначають його можливе місце в ньому.

Тестування дозволяє:

- визначити свої професійні нахили та здібності;
- виявити властивості характеру та особистісні якості;
- прояснити рівень розвитку своїх здібностей;
- вибрати галузь професійної діяльності.

Тести на профорієнтацію онлайн проливають світло на різні аспекти людської натури: одні досліджують бажання людини, інші тести на профорієнтацію перевіряють приватні здібності, треті – особистість людини.

Загальновідомо, що підлітки, старшокласники, випускники надають перевагу і більше довіряють інформації, отриманій через електронні джерела (інтернет), інколи незважаючи на ступінь її правдивості [2].

Згідно з проведеним соціологічним дослідженням навчальні заклади, організації, ЗМІ, які активно використовують у своїй роботі інтернет ресурси, старшокласники вважають більш сучасними і передовими, а отже, такими, що більше заслуговують на довіру. Під використанням інтернет

ресурсів розуміють не тільки розміщення новин на сайтах ЗМІ, а й створення і підтримку власних сайтів навчальних закладів, розміщення інформації в соціальних мережах (Facebook, Twitter, liveJornal, «Дневник»), на освітніх і молодіжних площадках (у тому числі форумах) [3].

У плануванні свого майбутнього важливо не безпам'ятно кидатися в сторону мрії дитинства (яка має всі шанси розсипатися при першій серйозній проблемі на шляху до неї), а краще придивитися до себе.

На щастя, розумні люди давно вже це помітили і ще на початку минулого століття почали розробляти програми профорієнтації, або, простіше кажучи, певні методи для визначення схильностей індивіда до певного роду діяльності. Під методами найчастіше розуміються різного роду тести, які на сьогоднішній день можна пройти безкоштовно і онлайн.

На превелику перевагу сучасного випускника школи, профорієнтаційні тести сьогодні вже не уявляють собою пачки листів, при заповненні яких респондент витрачає години на розрахунки результатів. На сьогодні те ж саме можна зробити тільки маючи комп'ютер, та початкові знання як з ним працювати.

Одна роботи але різними способами можна отримати результати. Тільки в другому варіанті перевагами є те, що витрачаються небагато часу, та є зацікавленість до використаних людиною технологій [4].

1.1.1 Опис процесу діяльності.

На основі аналізу предметного середовища була побудована діаграма діяльності для користувача, за допомогою UML діаграми (рис. 1.1) [5].

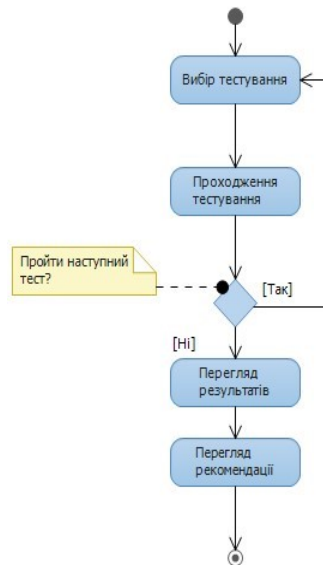


Рисунок 1.1 – Діаграма діяльності користувача

Аналізуючи побудовану діаграму діяльності можна визначити основні дії.

Користувач повинен обрати тест (з можливих запропонованих), пройти тести та переглянути результати (у разі проходження тесту). За бажанням користувач може переглянути рекомендації (необхідна інформація по предметам ЗНО за спеціальностями).

Розглянемо процес діяльності адміністратора за допомогою UML діаграми діяльності (рис. 1.2).

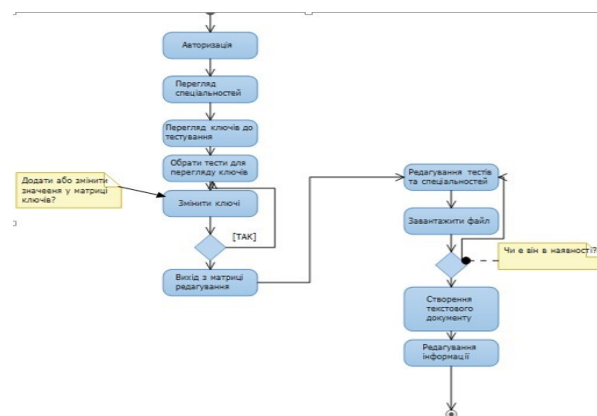


Рисунок 1.2 – Діаграма діяльності адміністратора

Аналізуючи побудовану діаграму діяльності можна виділити основні дії адміністратора:

- пройти авторизацію (ввести пароль);
- додавати тести (завантажити з зовнішнього сховища);
- додавати спеціальності (для результатів тестування); – підібрати ключі до визначення спеціальностей.

Для коректної роботи програми, адміністратор повинен завантажити усю необхідну інформацію.

1.1.2 Опис функціональної моделі.

За результатами проведення аналізу предметного середовища побудовано функціональну модель.

Функціональну модель можна показати у вигляді діаграми варіантів використання (рис. 1.3).

Позначимо дійових осіб (акторів) та дії, які кожен з них може виконувати у системі [6]:

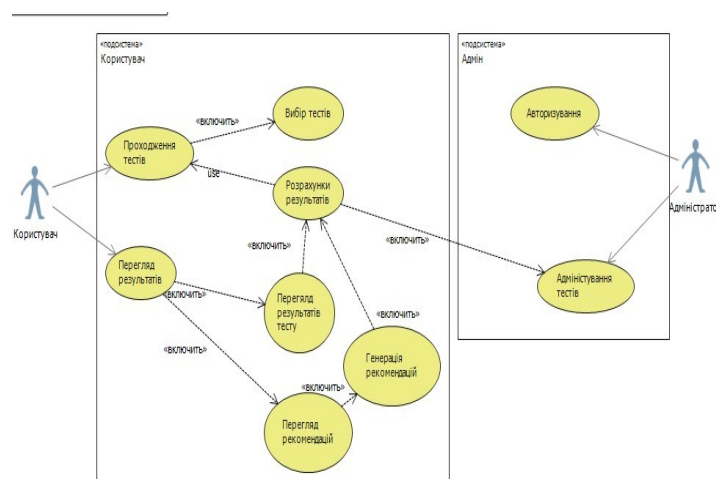


Рисунок 1.3 – Діаграма варіантів використання

Розглянемо дії з варіантів використання для кожного з акторів.

Користувач може виконувати наступні дії:

- проходження тестів – яка включає в себе вибір тесту;
- перегляд результатів (перегляд результатів тесту та перегляд рекомендацій);

Дія стосовно адміністратора:

- адміністрування тестів (додавання тестів, зміна інформації);
- авторизація (введення логіну).

1.2 Огляд наявних аналогів

Для реалізації побудованих діаграм діяльності та діаграм використання було виконано пошук наявних аналогів систем.

В мережі Інтернет представлений ряд сайтів, що пропонують свої послуги в сфері професійної орієнтації.

Сайт «Профорієнтація» [7], Сайт «ПрофГід» [8], Сайт «Тест на профорієнтацію» [9], Сайт «Карта інтересів [10].

В наш час багато ЗВО створюють свої автоматизовані системи для профорієнтаційної діяльності. Загальні автоматизовані системи, які використовуються ресурсами онлайн або центрами зайнятості, більш орієнтовані на загальну направленість схильності людини (тобто користувач, а саме школяр, може побачити перелік професій, які йому більш підходять за психологічними критеріями, встановленими психологами).

Вище зазначені програмні засоби мають багато специфічного та зайвого функціоналу, який не потрібний для вирішення поставленої задачі. Автоматизовані системи, що розробляються під конкретне ЗВО, адаптовані саме під його потреби. Була виявлено відсутність програмних продуктів, які могли б пропонувати вступнику глобальні спеціальності міста.

РОЗДІЛ II ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз предметної області

Система тестування – тип додатку, який дозволяє завдяки поставленим запитанням (автоматичним) обрати один з варіантів відповідей та отримати результати за схильностями людини.

Зазвичай користувачем даної системи являються абітурієнт, який сам формує свій життєвий вибір в плані вибору професії.

Вхідні дані надходять до системи від адміністратора та складаються з:

- питання – текстовий рядок до 255 символів;
- відповіді – текстовий рядок до 255 символів;
- сертифікати предметів ЗНО, які необхідно здати для вступу в ЗВО (3 з який бюджетної форми, та інші 3 контрактна форма) – текстовий рядок до 255 символів;
- напрямки спеціальностей, ЗВО на загальній базі (складаються з шифру спеціальностей, назви спеціальності та переліку сертифікатів предметів ЗНО);
- матриці з ключами (для кожної спеціальності зіставлена множина відповідей на питання тесту. Кожний елемент матриці є цілим числом) – ціле число.

Адміністратор завантажує спеціальності, предмети та матриці з ключами до системи у текстовому форматі.

На виході автоматизованої системи результатами будуть розподіл відсоткового співвідношення за пройденими тестами між спеціальностями та загальне усереднене значення між усіма спеціальностями враховуючи кількість пройдених користувачем тестів. Також на виході отримується

рекомендована інформація: перелік сертифікатів ЗНО, які необхідно отримати для вступу в ЗВО за запропонованими спеціальностями.

2.2 Проектування системи

2.2.1 Проектування бази даних.

На основі аналізу вхідних даних була спроектована структура даних [11] для автоматизованої системи, яка буде складатись з декількох зовнішніх сховищ:

- спеціальності;
- тест Соломін;
- тест Соломін 2;
- тест Холланд;
- матриця з ключами для всіх тестів.

Зовнішнє сховище буде представлено у вигляді тестового файлу.

Структура файлу «Спеціальності» містить в собі загальну інформацію про шифр спеціальності, спеціальності ЗВО та перелік предметів на які необхідно отримати сертифікати для вступу по окремим спеціальностям (подана в таблиці 2.1).

Таблиця 2.1 – Структура файлу «Спеціальності»

№	Ім'я поля в таблиці	Тип даних	Розмір поля
1	Шифр спеціальності	Цілочисловий	0 до 255
2	Спеціальності ЗВО	Текстовий	100
3	Сертифікати предметів	Текстовий	100

Структура файлу «Холланд» містить в собі загальну інформацію про тести. Містить питання та відповіді (подана в таблиці 2.2).

Таблиця 2.2 – Структура файлу «Тест Холланд»

№	Ім'я поля в таблиці	Тип даних	Розмір поля
1	Питання	Текстовий	100
2	Відповіді	Текстовий	100

Структура файлу «Тест Соломін» містить інформацію про тест Соломін та складається з питання та відповіді (подана в таблиці 2.3).

Таблиця 2.3 – Структура файлу «Тест Соломін»

№	Ім'я поля в таблиці	Тип даних	Розмір поля
1	Питання	Текстовий	100
2	Відповіді	Текстовий	100

Структура файлу «Ключі до матриці». Для кожної спеціальності зіставлена множина відповідей на тест який відповідний до спеціальності. Кожний елемент матриці є цілим числом (подана в таблиці 2.4).

Таблиця 2.4 – Структура файлу «Ключі до матриці»

№	Ім'я поля в таблиці	Тип даних	Розмір поля
1	Шифр спеціальності	Цілочисловий	0 до 255
2	Ключі	Цілочисловий	0 до 255

Файл «Ключі матриці», який складається з 3 різних текстових документів, відокремлених по окремим 3 тестам.

1. Тест Холланд складається з 43 запитань. Кожна відповідь на питання зіставляється елементу множини матриці.

Відповідь може приймати одне з значень -1 , 0 і 1 .

Кожне числове значення спів ставляться текстовому формату:

– (-1) (означає що професія зовсім не підходить);

- (0) (означає що людина можливо бачить себе в цій професії);
- 1 (найкраща професія).

2. Тест Соломін містить 34 запитання. Відповідь може приймати одне з значень 0, 1, 2,3.

Кожне числове значення спів ставляться текстовому формату;

- (0) зовсім ні;
- (1) можливо так;
- (2) вірно;
- (3) зовсім вірно.

3. Тест Соломін 2 складається з 35 запитань. Кожна відповідь на запитання приймається у числовому значенні 0, 1, 2, 3.

- 0 (зовсім ні);
- 1 (можливо так);
- 2 (вірно);
- 3 (зовсім вірно).

2.2.2 Побудова об'єктно-орієнтованої моделі.

На основі об'єктно-орієнтованого аналізу предметної області була спроектована модель, яка складається з 4 класів та 6 користувальницьких типів даних, що наведені на UML діаграмі [12] (рис. 2.1).

Сукупність опису об'єктів одного типу і методів роботи з ними називається класом [13].

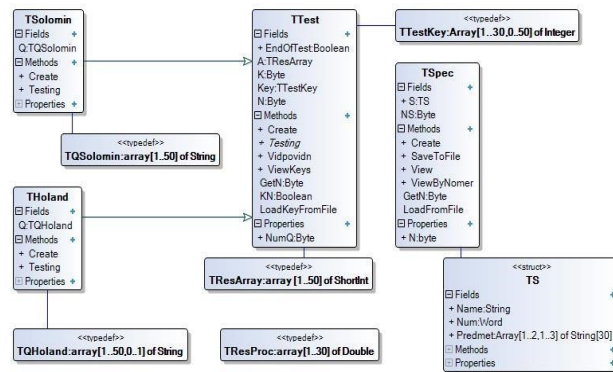


Рисунок 2.1 – UML діаграма класів

Побудована об'єктно-орієнтована модель включає в себе класи: «TTest», «THoland», «TSolomin», «TSpec».

Клас «TTest» призначений для оброблення інформації та реалізації розрахунку результатів. Клас складається з 4 полів (*A* – результати відповідей, *K* – номер поточного питання, *Key* – ключи до тесту, *N* – Кількість питань у тесті), методів та властивостей. Даний клас являється батьківським класом для «THoland», «TSolomin», «TSpec». Даний клас складається з наступних методів.

Метод *Create* містить *KFN* – файл з ключами. Містить інформацію про проходження теста.

Метод *LoadKeyFromFile* відповідає за файл, інформація якого містить питання та його проходження.

Метод *Testing* відповідає за проходження тестів. Містить інформацію про закінчення тестування та послідовне проходження тесту.

Метод *Vidpovidn* складається з полів 4 полів (*i*, *j*, *max*, *zn*), де *max* – кількість ключів, *zn* – кількість ключів, що співпали. Виконується процедура розрахунку усередненого значення до спеціальностей.

Метод *ViewKeys* містить головне поле *ks* – кількість спеціальностей.

Методи KN та GetN відповідають за отримання результатів за перевірка на досягнення кінця тесту та кількість питань у тесті.

Клас «THoland» містить інформацію по тесту Холланд. Складається з 1 поля, 2 методів. Методи класу «THoland» відповідають за створення та результат тестування на протязі проходження тесту. Даний клас є дочірнім класом класу «TTest» та містить наступні методи.

Метод Create відповідає за створення тесту. Містить 2 поля (F – текстовий файл, i – числовий).

Метод Testing відповідає за проходження тестів. Містить інформацію про закінчення тестування та послідовне проходження тесту.

Клас «TSolomin» містить інформацію по тесту Соломін. Складається з 1 поля, 2 методів. Методи класу «TSolomin» відповідають за створення та результат тестування на протязі проходження тесту. Даний клас є дочірнім класом до класу «TTest» та складається з наступних методів.

Метод Create відповідає за створення тесту. Містить 2 поля (F – текстовий файл, i – числовий).

Метод Testing відповідає за проходження тестів. Містить інформацію про закінчення тестування, послідовне проходження тесту та варіанти відповідей за запитання.

Клас «TSpec» відповідає за результати тестування, розподіл коефіцієнтів між усіма можливими спеціальностями. Складається з 1 поля (NS – Кількість спеціальностей), 6 методів та властивості. Спеціальності в даному класі зберігаються у вигляді масиву. Методи класу слугують основними функціями. Даний клас є дочірнім класом класу «TTest» та містить наступні методи.

Метод Create відповідає за функцію завантаження з файлу.

Метод SaveToFile призначений для збереження спеціальностей в файл. Складається з 2 полів (■ – текстовий файл, i – числовий).

Метод View призначений для перегляду інформація за предметами. Розподіл за двома формами: перші 3 предмети – бюджет, інші 3 – контракт.

Метод LoadFromFile призначений для завантаження інформації з файлу.

Метод ViewByNomer містить поле Nomer – номер. Призначений для виведення інформації щодо «Рекомендованої спеціальності» (вивід спеціальності та перелік сертифікатів предметів ЗНО які необхідні до вступу).

Користувальницький тип «TS» для розподілу спеціальності між двома критеріям (бюджет, контракт). Містить 3 поля (Ім'я, Номер та Предмет). Предмет складається з масивів для набору предметів, сертифікати ЗНО які необхідні для вступу.

Користувальницький тип даних «TResArray» призначений для відповідей користувача. Складається з масиву від 1 до 50.

Користувальницький тип «TResProc» призначений для відповідності користувача спеціальностям в відсотках.

Кожен з класів призначений для виконання конкретної відведеної задачі, а саме:

- реалізація взаємодії програми з файлу;
- реалізація алгоритму, що формує результати по тестуванню, згідно з вимогами користувача.

Реалізація механізму взаємодії з зовнішнім сховищем полягає в забезпеченні безпомилкового обміну даними між файлом і програмою, яка оброблює використовувані дані. Подібний підхід дозволяє виконати обробку різного роду винятків і різноманітних помилок, які, в теорії, могли б призвести до збою в роботі програми, яка виражена у формі буквального переривання роботи програми, або ж видачі хибного результату по завершенні обробки інформації.

Для розрахунків кінцевого результату була використана формула знаходження середнього значення, але адаптована під дану систему.

2.3 Математичне та алгоритмічне забезпечення

Для реалізації розрахунків результатів усередненого значення тестування була обрана формула знаходження середнього значення [14].

Середня арифметична – найпоширеніший вид середньої. Вона використовується, коли розрахунок здійснюється по незгрупованим статистичними даними, де потрібно здобути середній доданок.

Середня арифметична – це таке середнє значення ознаки, при отриманні якого зберігається незмінним загальний обсяг ознаки в сукупності.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

(2.1)

Система, яка розглядається, побудована на основі формули 2.1, але з змінними під результати:

$$\bar{S}_i = \frac{1}{N} \sum_{j=1}^N S_{ij}$$

(2.2)

де S_{ij} – відповідність результатів j -го тесту i -ої спеціальності;

S_i – загальний результат проходження тестів для i -ої спеціальності;

N – кількість тестів.

РОЗДІЛ III ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Для розробки програмної реалізації було обрано візуальне середовище Delphi [15].

Delphi – це Rapid Application Development (RAD) – середовище швидкої розробки додатків. Інакше кажучи, Delphi включає в себе не тільки всі засоби для роботи з мовою Object Pascal (тим більше, що починаючи з версії, 2005 у Delphi є підтримка інших мов програмування, зокрема, C #), а й ряд додаткових коштів, покликаних максимально прискорити і спростити створення програм. До таких засобів належать, перш за все, візуальний редактор форм, за допомогою якого за лічені хвилини і без зайвих зусиль можна створити повноцінно виглядають програму, а так само інші складові частини візуальної складової розробки програм.

В результаті час розробки програм істотно скорочується, оскільки відпадає необхідність прораховувати вручну розташування кожного елемента користувацького інтерфейсу [16].

На основі поставленої задачі було проведено огляд операційних середовищ та обрано саме Delphi 7.

3.2 Вимоги до технічного та програмного забезпечення

Данна програмна розробка представляє собою автономний «exe»файл.

Для того, щоб програма працювала без помилок та затримання необхідні:

- одноядерний процесор з тактовою частотою 1 ГГц та вище;

- 1 Гб оперативної пам'яті (RAM);
- 10 Мб вільного місця на жорсткому диску (HDD);
- монітор з роздільною здатністю від 1024x768;
- USB або PS/2 сумісна миша (або тачпад) та клавіатура;
- операційна система Microsoft Windows 7/8/8.1/10; – встановлений .NET Framework 4.

Програмна розробка дуже проста в освоєнні і розрахована не тільки на професіоналів, а й на пересічних користувачів, що працюють під Windows. Зручний інтуїтивно зрозумілий інтерфейс дозволяє працювати з програмою без попередньої підготовки.

Програма повинна видавати результати відразу після виконання відповідної дії користувачем.

3.3 Керівництво користувача

Для запуску програми на виконання слід запустити файл Tests.

Після запуску програми буде відображене головне вікно (рис. 3.1), яке поділено на частини: верхня панель, вкладки та місце для запитань та відповідей.

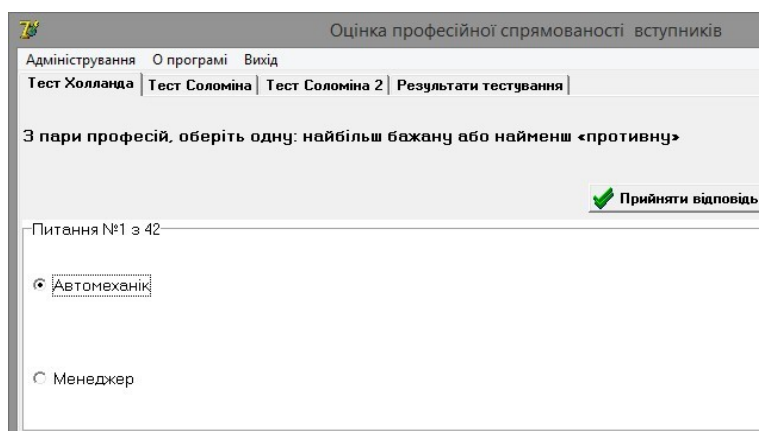


Рисунок 3.1 – Головне меню програми

Пункт «Адміністрування» призначений для роботи адміністратора та складається з підпунктів: авторизація, перегляд спеціальностей, додати спеціальність, ключи до тестів та редагування інформацій (рис. 3.2).

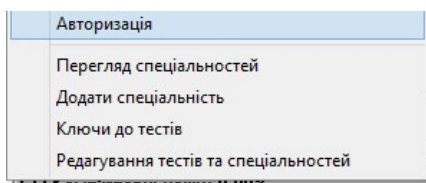


Рисунок 3.2 – Пункт «Адміністрування»

Адміністратор має пройти авторизацію – ввести пароль (рис. 3.3). У разі не вірно введених даних на екрані з’явиться попередження про помилку (рис. 3.4).

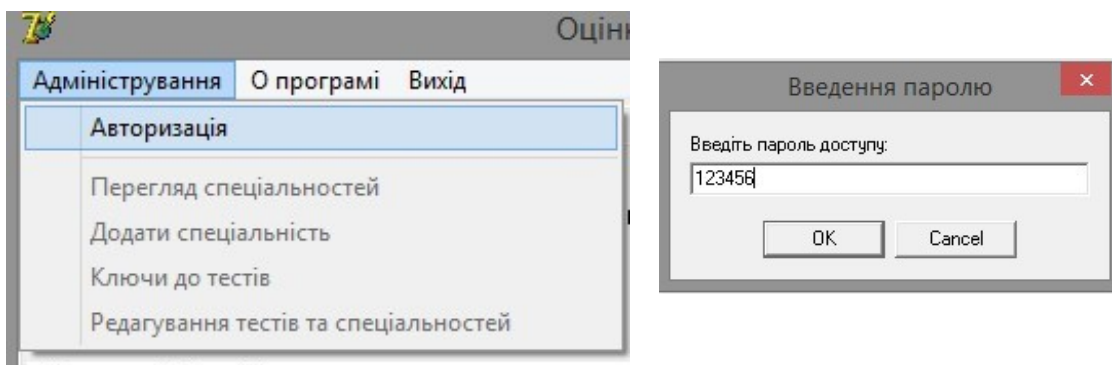


Рисунок 3.3 – Авторизація

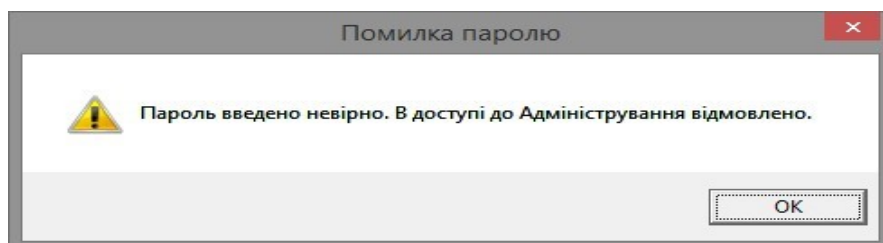


Рисунок 3.4 – Помилка у разі не вірного паролю

Інформація яка знаходиться у програмі заноситься від імені адміністратора.

Вносити інформацію треба таким чином (рис. 3.5)

- створюється текстовий документ;
- вноситься інформацію;
- зберігається результат;
- завантажується файл.

Таким чином виконується і зміна інформації. Обирається файл вже створений, вноситься зміна та зберігаються.

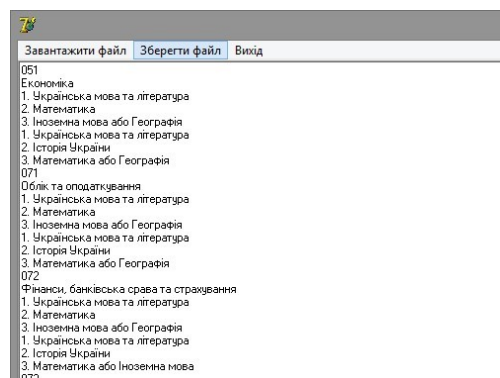
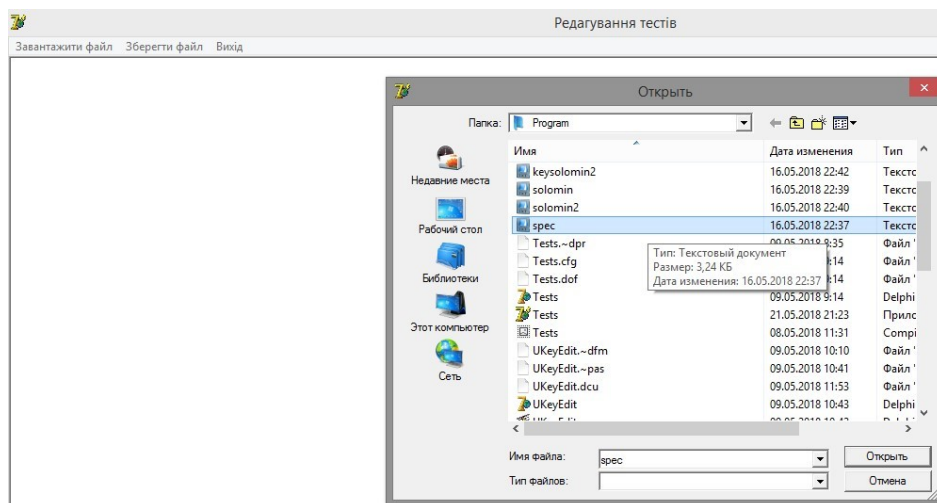


Рисунок 3.5 – Занесення інформації до програми

Користувач на головному екрані програми бачить вкладки з тестами, які йому запропоновані для проходження: Холланд, Соломін та Соломін 2.

Після він обирає ту вкладку, тест якого бажає пройти.

Тест Холланд містить 42 питання. Користувач бачить запитання та відповіді на нього. Із запропонованих попарно варіантів професій (спеціальностей) необхідно обрати ту, якої Ви віддаєте перевагу (рис. 3.6). Після вибору варіанту відповіді необхідно натиснути кнопку «Прийняти відповідь», тим самим відповідь заноситься до системи і так до самого останнього запитання.

Коли запитання закінчаться, програма заблокує кнопку «Прийняти відповідь» – тестування закінчене.

Рисунок 3.6– Тест Холланд

Тест Соломін містить 34 питання.

Користувач може оцінити за 4-х бальною шкалою ступінь свого бажання займатися кожним з 34 наведених у переліку видів діяльності.

Кожне питання складається з 4 варіантів відповідей, користувачу необхідно обрати один із запропонованих та натиснути кнопку «Прийняти відповідь».

Запитання складаються з схильностей людини (рис. 3.7).

Рисунок 3.7– Тест Соломін

Тест Соломін 2 містить 35 тестів.

Необхідно користувачу оцінити за 4-х бальною шкалою ступінь своїх здібностей до кожного з 35 заданих видів діяльності (рис. 3.8).

Рисунок 3.8– Тест Соломін 2

Після проходження тестів, користувач переходить на вкладку «Результати тестування», та бачить інформацію проходження тестів.

Результати поділяються на відсоткове розподілення спеціальностей по кожному тесту та загальні результати, враховуючи ті тести, які проходить користувач (рис. 3.9).

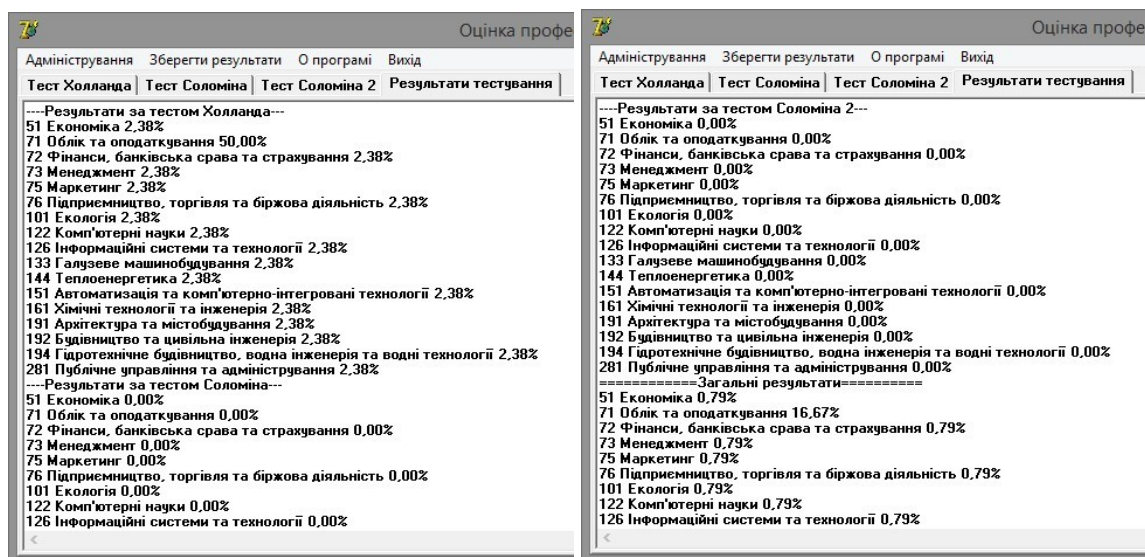


Рисунок 3.9 – Результати проходження тестів

За бажанням, користувач може переглянути рекомендовану йому спеціальність за пройденими тестами до вступу в ЗВО (рис. 3.10).

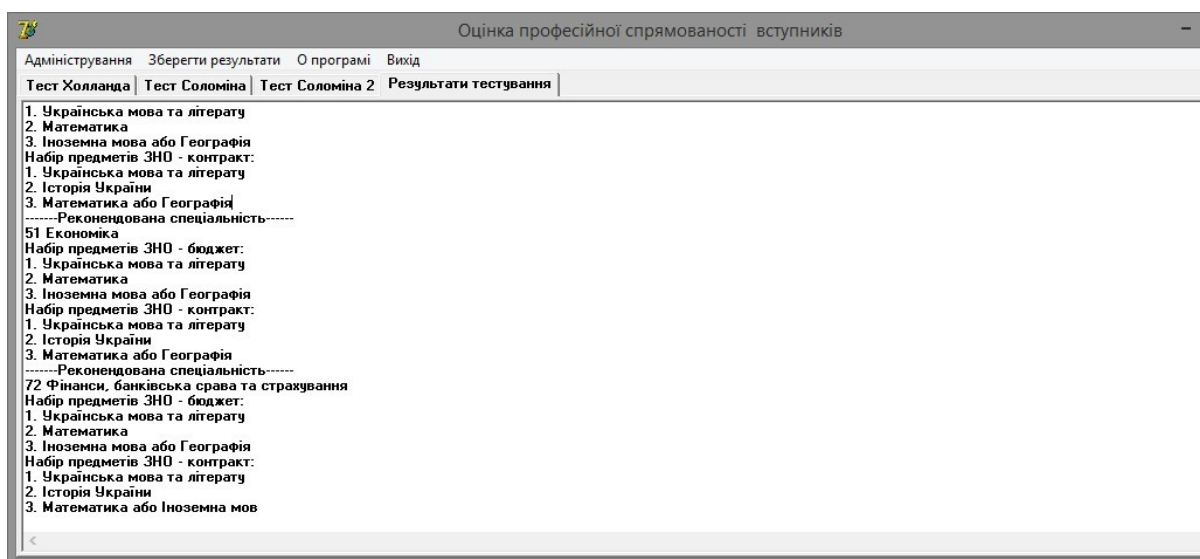


Рисунок 3.10 – Рекомендована інформація

Інформацію можна зберегти у текстовому форматі: натиснути на верхній панелі «Зберегти результати» та вказати адресу збереження.

Користувач отримає у текстовому вигляді результати проходження тестів (рис. 3.11).

```

----Результати за тестом Холланда---
51 Економіка 0,00%
71 Облік та оподаткування 50,00%
72 Фінанси, банківська справа та страхування 0,00%
73 Менеджмент 0,00%
75 Маркетинг 0,00%
76 Підприємництво, торгівля та біржова діяльність 0,00%
101 Екологія 0,00%
122 Комп'ютерні науки 0,00%
126 Інформаційні системи та технології 0,00%
133 Галузеве машинобудування 0,00%
144 Теплоенергетика 0,00%
151 Автоматизація та комп'ютерно-інтегровані технології 0,00%
161 Хімічні технології та інженерія 0,00%
191 Архітектура та містобудування 0,00%
192 Будівництво та цивільна інженерія 0,00%
194 Гідротехнічне будівництво, водна інженерія та водні технології 0,00%
281 Публічне управління та адміністрування 0,00%
----Результати за тестом Соломіна---
51 Економіка 0,00%
71 Облік та оподаткування 0,00%
72 Фінанси, банківська справа та страхування 0,00%
73 Менеджмент 0,00%
75 Маркетинг 0,00%
76 Підприємництво, торгівля та біржова діяльність 0,00%
101 Екологія 0,00%
122 Комп'ютерні науки 0,00%
126 Інформаційні системи та технології 0,00%
133 Галузеве машинобудування 0,00%
144 Теплоенергетика 0,00%
151 Автоматизація та комп'ютерно-інтегровані технології 0,00%
161 Хімічні технології та інженерія 0,00%
191 Архітектура та містобудування 0,00%
192 Будівництво та цивільна інженерія 0,00%
194 Гідротехнічне будівництво, водна інженерія та водні технології 0,00%
281 Публічне управління та адміністрування 0,00%

```

Рисунок 3.11 – Збереження результатів

При натисканні пункту «Вихід», програма буде закрита та усі данні очищуються.

ВИСНОВКИ

Результатами виконання роботи на тему «Автоматизована система професійної орієнтації з використанням сучасних тестових технологій» було наступне:

- проведено аналіз предметної області в результаті якої було визначено актуальність, мету та об'єкт дослідження.

- за результатами проведеного дослідження предметного середовища було визначено перелік задач, які потрібно вирішити для досягнення поставленої мети.

- під час аналізу предметної області було розглянуто існуючі програмні аналоги та виявлено, що ці програмні засоби мають багато специфічного та зайвого функціоналу, який не потрібний для вирішення поставленої задачі. Саме на відсутності гідного та безкоштовного програмного пакету формується задача створення власного засобу.

- було проведено детальний аналіз предметної області під час якого сформовані та описані вимоги до вхідних та вихідних даних. На підставі проведеного аналізу було побудовано об'єктно-орієнтовану модель та обрано математичні та алгоритмічні методи розв'язання поставленої задачі.

- за результатами об'єктно-орієнтованого програмування було обрано інструментарій для програмної реалізації автоматизованої системи. В якості середовища розробки обрано Delphi 7.

- розроблено керівництво користувача, вимоги до програмного забезпечення.

Таким чином, досліджені об'єкти характеризуються певними якостями, які дозволяють використовувати їх для застосування роботи у навчальних закладах міста.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дубровина И. В. Активные методы в работе школьного психолога: сборник научных трудов / И. В. Дубровина. // АПН СССР. – 1989. – 165 с.
2. Фопель К. М. На пороге взрослой жизни: Психологическая работа с подростковыми и юношескими проблемами. Планирование жизни. Решение проблем / К. М. Фопель. // Генезис. – 2008. – 179 с.
3. Пряжников Н. С. Профессиональное и личностное самоопределение / Николай Сергеевич Пряжников. // Институт практической психологии. – 1996. – С. 256.
4. Соломин И. Л. Ступени карьеры. Азбука профориентации / И. Л. Соломин, М. А. Бендюков. // Речь. – 2006. – 300 с.
5. Діаграми варіантів використання «UseCase» [Електронний ресурс] // Режим доступу: http://www.caseclub.ru/articles/use_case.html
6. Рамбо Д. UML 2.0. Объектно–ориентированное моделирование и разработка / Д. Рамбо, М. Блаха. // СПб: «Питер». – 2007. – 544 с.
7. Головний сайт «Профорієнтація» [Електронний ресурс] // Режим доступу: <https://smartia.me>
8. Головний сайт «ПрофГід» [Електронний ресурс] // Режим доступу: <https://sites.google.com/site/professiamoejmecty/raduga-professij>
9. Головний сайт «Тест на профорієнтацію» [Електронний ресурс] // Режим доступу: <http://mycareer.org.ua/test>
10. Головний сайт «Карта інтересів» [Електронний ресурс] // Режим доступу: <http://prof.osvita.org.ua/uk/determine/testing/6/index.html>
11. Архітектура БД [Електронний ресурс] // Режим доступу: <http://citforum.ru/ofis/backoffice/glava2.shtml>

12. Осипов Д. Базы данных и Delphi. Теория и практика / Дмитрий Осипов. // БХВ-Петербург. – 2011. – 856 с.
13. Осипов Д. Delphi. Профессиональное программирование / Дмитрий Осипов. // БХВ-Петербург. – 2006. – 350 с.
14. Дураков Б. К. Краткий курс высшей алгебры / Б. К. Дураков. // ФИЗМАТЛИТ. – 2006. – 232 с.
15. Чернилевский Г. Фундаментальные алгоритмы и структуры данных в Delphi / Георгий Чернилевский. // DiaSoft. – 2003. – 200 с.
16. Емельянов В. И. Основы программирования на Delphi / В. И. Емельянов, В. И. Воробьев, Т. П. Тюрина. // ООО «Юго-Восток, Лтд». – 2005. – 231 с.

ДОДАТОК А

Лістинг програми

```

unit UTest;

interface

Uses    SysUtils,ExtCtrls,Grids,StdCtrls;
Type
    TResArray = array [1..50] of ShortInt;  {Відповіді користувача}
    TResProc  =    array[1..30] of    Double;    {Відповідність
користувача спеціальностям в процентах}
    TS=Record                                {Тип для спеціальності}
        Num:Word;
        Name:String;
        Predmet:Array[1..2,1..3] of String[30]; {два набори предметів ЗНО}
    End;
    TTestKey=Array[1..30,0..50] of    Integer;           {Ключи    50
тестів        то    30 спеціальностей, 0-вий стовпець - номер спеціальності}
    TQHoland = array[1..50,0..1] of String;
    TQSolomin = array[1..50] of String;
    TTest=class    {Абстрактний базовий клас тесту}
private
    N:Byte; {Кількість питань у тесті}
    K:Byte; {Номер поточного питання}
    A:TResArray; {Результати відповідей}
    Key: TTestKey; {ключи до тесту}
    Function KN:Boolean; {перевірка на досягнення кінця тесту}

```


Procedure

LoadKeyFromFile(KFN:String);

Function GetN:Byte; public

EndOfTest:Boolean; {Ознака повного проходження тесту}

Property NumQ:Byte read GetN;

Procedure Testing; virtual; abstract;

Constructor Create(KFN:String);

Procedure ViewKeys(ks:Byte; SG:TStringGrid); {ks - кількість спеціальностей}

procedure Vidpovidn(ks:Byte; var RP:TResProc); {ks - кількість спеціальностей}

End;

THoland=class(TTest) {Тест Холланд}

private

Q:TQHoland;

public

Procedure Testing(var Rg:TRadioGroup); overload;

Constructor Create(TFN,KFN:String);

End;

TSolomin=class(TTest) {Тест СОЛОМІНА}

private

Q:TQSolomin;

public

Procedure Testing(var Rg:TRadioGroup); overload;

Constructor Create(TFN,KFN:String);

End;

```

TSpec=class
private
    NS:Byte; {Кількість спеціальностей}    procedure
LoadFromFile(FN:String); {Завантаження з файлу}
function GetN:Byte;    public

    S:Array[1..30] of TS; {Спеціальності
університету}    property N:byte read GetN;
Constructor Create(FN:String);
    Procedure SaveToFile(FN:String);
    Procedure View(SG:TStringGrid);
    Procedure ViewByNomer(Nomer:Byte; var M:TMemo);
End;

implementation
{TTest}
Constructor TTest.Create(KFN:String); {KFN - файл з ключами}
Var i:Byte;
Begin
inherited
Create;
N:=0;
For i:=1 to 50 do A[i]:=-1;
LoadKeyFromFile(KFN); k:=0;
{Початковий номер питання}
EndOfTest:=False;
End;

Function TTest.GetN:Byte;
Begin

```

Result:=N;

End;

Function TTest.KN:Boolean;

Begin

If K=N Then Result:=True

Else Result:=False;

EndOfTest:=Result;

End;

Procedure TTest.LoadKeyFromFile(KFN:String);

Var F:TextFile;

nn,i,j:Byte; {nn - Кількість питань}

Begin

AssignFile(F, KFN);

Reset(F);

Readln(f, nn);

i:=0;

While Not(EOF(F))

do Begin i:=i+1;

For j:=0 to nn do Read(F, Key[i,j]);

End;

CloseFile(F);

End;

Procedure TTest.ViewKeys(ks:Byte; SG:TStringGrid);

Var i,j:Byte;

Begin

For i:=1 to ks do For j:=0 to

N do

```

sg.Cells[j,i]:=IntToStr(Key[i,j])
; For j:=1 to n do
sg.Cells[j,0]:=IntToStr(j);
sg.ColCount:=N+1;
End;

```

```

Procedure TTest.Vidpovidn(ks:Byte; var RP:TResProc);
Var i,j,max,zn:Byte; {max - усього ключів, zn - співпало ключів}
Begin
  For i:=1 to ks do      {для кожної
спеціальності} Begin  max:=0;  zn:=0;  For
j:=1 to N do
    Begin
      If Key[i,j]<>-1 Then max:=max+1;
      If Key[i,j]=a[j] Then zn:=zn+1;
    End;
    RP[i]:=zn/max*100; {відсоток співпадань}
  End;
End;

```

```
{THoland}
```

```

Constructor
THoland.Create(TFN,KFN:String); Var
F:TextFile; i:Byte; Begin
  inherited Create(KFN);
  AssignFile(F,
TFN); Reset(F);
i:=0;

```

```
While not(EOF(F)) do
```

```
Begin
```

```
  i:=i+1;
```

```
  Readln(F, Q[i,0]);
```

```
  Readln(F, Q[i,1]);
```

```
End;
```

```
N:=i;
```

```
CloseFile(F);
```

```
End;
```

```
Procedure THoland.Testing(var Rg:TRadioGroup);
```

```
Begin
```

```
  If (K>0)and(rg.ItemIndex<>-1) Then A[K]:=rg.ItemIndex; {Зберігаємо  
Відповідь попереднього питання}
```

```
  If Not(KN) Then {якщо не кінець тестування}
```

```
  Begin
```

```
    K:=K+1; {Наступне питання}
```

```
  rg.ItemIndex:=-1;
```

```
  Rg.Caption:='Питання №'+IntToStr(K)+' з '+IntToStr(N);
```

```
  Rg.Items.Clear;
```

```
  Rg.Items.Add(Q[K,0]);
```

```
  Rg.Items.Add(Q[K,1]);
```

```
End;
```

```
End;
```

```
{TSolomin}
```

```
Constructor
```

```
TSolomin.Create(TFN,KFN:String); Var
```

```

F:TextFile; i:Byte; Begin inherited
Create(KFN);
  AssignFile(F,
TFN); Reset(F);
i:=0;
  While not(EOF(F))
do Begin i:=i+1;
  Readln(F, Q[i]);
  End;
N:=i;
  CloseFile(F);
End;

```

```

Procedure TSolomin.Testing(var Rg:TRadioGroup);
Begin
  If (K>0)and(rg.ItemIndex<>-1) Then A[K]:=rg.ItemIndex; {Зберігаємо
Відповідь попереднього питання}
  If Not(KN) Then {якщо не кінець тестування}
  Begin
    K:=K+1; {Наступне питання}
rg.ItemIndex:=-1;
    Rg.Caption:=Q[K]+'('+IntToStr(K)+'/'+IntToStr(N)+)';
    Rg.Items.Clear;
    Rg.Items.Add('Зовсім ні');
    Rg.Items.Add('Можливо так');
    Rg.Items.Add('Вірно');
    Rg.Items.Add('Зовсім вірно');
  End;
End;

```

```

                                {TSpec}
Procedure TSpec.LoadFromFile(FN:String);
Var F:TextFile;
Begin
    AssignFile(F, FN);
    Reset(F);
    NS:=0;
    While Not(EOF(F)) do
    Begin
        NS:=NS+1;
        ReadLn(F, S[NS].Num);
        ReadLn(F, S[NS].Name);
        ReadLn(F, S[NS].Predmet[1,1]);
        ReadLn(F, S[NS].Predmet[1,2]);
        ReadLn(F, S[NS].Predmet[1,3]);
        ReadLn(F, S[NS].Predmet[2,1]);
        ReadLn(F, S[NS].Predmet[2,2]);
        ReadLn(F, S[NS].Predmet[2,3]);
    End;
    CloseFile(F);
End;

Constructor
TSpec.Create(FN:String); Begin
inherited Create;
    LoadFromFile(FN);
End;

```

Procedure

TSpec.SaveToFile(FN:String); Var

F:TextFile; i:Byte;

Begin

AssignFile(F, FN);

Rewrite(F);

For i:=1 to NS do

Begin

Writeln(F, S[NS].Num);

Writeln(F, S[NS].Name);

Writeln(F, S[NS].Predmet[1,1]);

Writeln(F, S[NS].Predmet[1,2]);

Writeln(F, S[NS].Predmet[1,3]);

Writeln(F, S[NS].Predmet[2,1]);

Writeln(F, S[NS].Predmet[2,2]);

Writeln(F, S[NS].Predmet[2,3]);

End;

CloseFile(F);

End;

Procedure TSpec.View(SG:TStringGrid);

Var i:Byte;

Begin

SG.RowCount:=NS+1;

SG.Cells[0,0]:='Шифр';

SG.Cells[1,0]:='Назва';

SG.Cells[2,0]:='1-й предмет';

SG.Cells[3,0]:='2-й предмет';

SG.Cells[4,0]:='3-й предмет';


```

SG.Cells[5,0]:='1-й предмет';
SG.Cells[6,0]:='2-й предмет';
SG.Cells[7,0]:='3-й предмет';
For i:=1 to NS do
Begin
  SG.Cells[0,i]:=IntToStr(S[i].Num);
  SG.Cells[1,i]:=S[i].Name;
  SG.Cells[2,i]:=S[i].Predmet[1,1];
  SG.Cells[3,i]:=S[i].Predmet[1,2];
  SG.Cells[4,i]:=S[i].Predmet[1,3];
  SG.Cells[5,i]:=S[i].Predmet[2,1];
  SG.Cells[6,i]:=S[i].Predmet[2,2];
  SG.Cells[7,i]:=S[i].Predmet[2,3];
End;
End;

```

```

Procedure TSpec.ViewByNomer(Nomer:Byte; var M:TMemo);
Var i:Byte;
Begin
  M.Lines.Add('-----Рекомендована спеціальність-----');
  M.Lines.Add(IntToStr(S[Nomer].Num)+' '+S[Nomer].Name);
  M.Lines.Add('Набір предметів ЗНО - бюджет:');
  For i:=1 to 3 do
    M.Lines.Add(S[Nomer].Predmet[1,i]);
  M.Lines.Add('Набір предметів ЗНО - контракт:');
  For i:=1 to 3 do
    M.Lines.Add(S[Nomer].Predmet[2,i]);
End;
Function TSpec.GetN:byte;
Begin

```

Result:=N

S; End;

end.