КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка програмного забезпечення оцінки кандидатів та відбору фахівця на заміщення вакантної посади (на прикладі посади бухгалтера підприємства)»

Студентки 4 курсу, 9 групи,
спеціальності
122 «Комп'ютерні науки»

_____
підпис студента

Богданець Марія Вікторівна

Науковий керівник
кандидат технічних наук,
доцент

_____
підпис керівника

Демідов Павло Георгійович

Гарант освітньої програми
кандидат технічних наук, доцент

_____
підпис керівника

Демідов Павло Георгійович

Київ 2020
Київський національний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Затверджую
Зав. кафедри _____Пурський О.І.
«20» грудня 2019р.

Завдання
на випускний кваліфікаційний проект студентці

Богданець Марії Вікторівни
(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту

«Розробка програмного забезпечення оцінки кандидатів та відбору фахівця на заміщення вакантної посади (на прикладі посади бухгалтера підприємства)»

Затверджена наказом ректора від *«04» грудня 2019 р. № 4111*

2. Строк здачі студентом закінченого проекту *12 червня 2020 року*

3. Цільова установка та вихідні дані до проекту

Мета проекту: Розробити математичне, алгоритмічне та програмне забезпечення оцінки кандидатів та відбору фахівця на заміщення вакантної посади.

Об'єкт дослідження: процеси визначення можливого кандидата на заміщення вакантної посади.

Предмет дослідження: моделі, методи та інформаційні технології в системі оцінки кандидатів на заміщення вакантної посади

4. Перелік графічного матеріалу

__Вхідні_та_вихідні_дані_задачі_оцінки_та_відбору_кандидата_на_заміщення_вакантної_посади;_моделі_та_методи_вирішення_даної_задачі;_віконний інтерфейс_роботи_з_розробленою_комп'ютерною_програмою._____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Демідов П.Г.	15.12.2019 р.	15.12.2019 р.
2	Демідов П.Г.	15.12.2019 р.	15.12.2019 р.
3	Демідов П.Г.	15.12.2019 р.	15.12.2019 р.

6. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. Дослідження та аналіз підходів до оцінки кандидатів на заміщення вакантної посади

1.1. Проблеми оцінки та відбору фахівців на вакантні посади

1.2. Теорія нечітких множин та нечіткої логіки

1.3. Характеристика багатокритеріальних задач

РОЗДІЛ 2. Проектування програмного забезпечення вирішення задачі оцінки кандидатів та відбору фахівця на заміщення вакантної посади

2.1. Постановка задачі на розробку системи оцінки та відбору фахівців

2.2. Метод нечіткого логічного виведення в розв'язанні задачі відбору фахівців

2.3. Метод визначення вагових коефіцієнтів правил виведення

РОЗДІЛ 3. Інформаційна технологія відбору фахівців на заміщення вакантної посади

3.1. Алгоритмічна реалізація методу нечіткого логічного виведення

3.2. Алгоритмічна реалізація методу визначення вагових коефіцієнтів

3.3. Характеристика програмного забезпечення системи оцінки кандидатів на заміщення вакантної посади

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання проекту

№ Пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		За планом	фактично
1	2	3	4
1	*Вибір теми випускного кваліфікаційного проекту*	*01.10.2019*	*01.10.2019*
2	*Розробка та затвердження завдання на випускний кваліфікаційний проект*	*15.12.2019*	*15.12.2019*
3	*Вступ*	*03.02.2020*	*03.02.2020*
4	*РОЗДІЛ 1. Дослідження та аналіз підходів до оцінки кандидатів на заміщення вакантної посади*	*28.02.2020*	*28.02.2020*
5	*РОЗДІЛ 2. Проектування програмного забезпечення вирішення задачі оцінки кандидатів та відбору фахівця на заміщення вакантної посади*	*06.04.2020*	*06.04.2020*
6	*РОЗДІЛ 3. Інформаційна технологія відбору фахівців на заміщення вакантної посади*	*12.05.2020*	*12.05.2020*
7	*Висновки*	*15.05.2020*	*15.05.2020*
8	*Здача випускного кваліфікаційного проекту на кафедру науковому керівнику*	*20.05.2020*	*20.05.2020*
9	*Попередній захист випускного кваліфікаційного проекту*	*03.06.2020*	*03.06.2020*
11	*Виправлення зауважень, зовнішнє рецензування випускного кваліфікаційного проекту*	*09.06.2020*	*09.06.2020*
12	*Представлення готового зшитого випускного кваліфікаційного проекту на кафедру*	*12.06.2020*	*12.06.2020*
13	*Публічний захист випускного кваліфікаційного проекту*	*За розкладом роботи ЕК*	*За розкладом роботи ЕК*

8. Дата видачі завдання «15» грудня 2019 р.

9. Керівник випускного кваліфікаційного роботи проекту

Демідов П.Г.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Богданець М.В.

(прізвище, ініціали, підпис)

12. Відгук керівника випускного кваліфікаційного проекту

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Керівник випускного кваліфікаційного проекту

_____ 12.06.2020 р.

(підпис, дата)

13. Висновок про випускний кваліфікаційний проект

Випускний кваліфікаційний проект студента Богданець М. В._____

(прізвище, ініціали)

може бути допущений до захисту в екзаменаційній комісії.

Гарант освітньої програми_____Демідов П.Г.

(підпис, прізвище, ініціали)

Завідувач кафедри _____Пурський О.І.

(підпис, прізвище, ініціали)

«_____»_____2020 р.

Анотація

У випускній кваліфікаційній роботі здійснено комплексну розробку моделей та інформаційної технології оцінки та відбору фахівців на заміщення вакантної посади з метою підвищення ефективності діяльності підприємства. Теоретичного обґрунтовано основні положення формування і проведення оцінки та відбору кандидатів та запропоновано концепцію створення інформаційної системи оцінювання критеріїв кандидатів на заміщення вакантної посади. Розроблено метод автоматизованого розрахунку комплексної оцінки кандидатів. Створено програмне забезпечення відбору та оцінки кандидатів на заміщення вакантної посади.

Ключові слова: відбір кандидатів, оцінка кандидатів, метод нечітких множин та нечіткої логіки, інформаційна технологія.

Anotation

As a result of graduation thesis, a comprehensive development of models and information technology for evaluating and selecting specialists for filling a vacant position in order to increase the efficiency of the enterprise was conducted. The key elements of appointing applicants, their rating and selecting were theoretically justified. The concept of creating an information system for rating the criteria of applicants for a vacant position is proposed. The method for automated calculation of a comprehensive assessment of applicants was developed. The software of estimating and selecting applicants for filling a vacant position was created.

Key words: applicants selection, applicants rating, method of fuzzy sets and fuzzy logic, information technology.

Зміст

ВСТУП……. 8

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО ОЦІНКИ КАНДИДАТІВ НА ЗАМІЩЕННЯ ВАКАНТНОЇ ПОСАДИ

1.1. Проблеми оцінки та відбору фахівців на вакантні посади… 12

1.2. Теорія нечітких множин та нечіткої логіки……. 15

1.3. Характеристика багатокритеріальних задач…….. 19

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИРІШЕННЯ ЗАДАЧІ ОЦІНКИ КАНДИДАТІВ ТА ВІДБОРУ ФАХІВЦЯ НА ЗАМІЩЕННЯ ВАКАНТНОЇ ПОСАДИ

2.1. Постановка задачі на розробку системи оцінки та відбору фахівців……. 26

2.2. Метод нечіткого логічного виведення в розв’язанні задачі відбору фахівців…….. 29

2.3. Метод визначення вагових коефіцієнтів правил виведення.. 33

РОЗДІЛ 3. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВІДБОРУ ФАХІВЦЯ НА ЗАМІЩЕННЯ ВАКАНТНОЇ ПОСАДИ

3.1. Алгоритмічна реалізація методу нечіткого логічного виведення……. 38

3.2. Алгоритмічна реалізація методу визначення вагових коефіцієнтів…… 42

3.3. Характеристика програмного забезпечення системи оцінки кандидатів на заміщення вакантної посади……. 44

ВИСНОВКИ…….. 49

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ……. 50

ДОДАТКИ……. 54

ВСТУП

У процесі здійснення господарської чи комерційно-виробничої діяльності постійно виникає потреба у залученні робочої сили на підприємство. Процес залучення працівників можемо поділити на відбір па підбір. Відбір кадрів - це діяльність, у процесі якої здійснюється вибір зі списку можливих кандидатів одного, який більш повно за своїми якостями відповідає визначеним роботодавцем критеріям для заповнення того чи іншого вакантного робочого місця. Зазвичай, відбір проводиться поетапно, що залежить від вакантної посади, службових обов'язків та особистих якостей претендента [11].

Процес відбору досить трудомісткий і якщо у випадку з невеличкою компанією її засновник чи менеджер з кадрів котрий був в команді з її зародження може досить ефективно опиратися на власні судження і знання справи якою займається, то при рекрутингу людей до великих корпорацій підбір персоналу стає складнішим. Обов'язки підбору персоналу у різні відділи чи навіть офіси вже неможливо здійснити декількома людьми, які точно знають якого працівника шукати і якими рисами він повинен володіти. Ці обов'язки делегуються іншим людям, а ті делегують іншим і т.д. Це є нормальним процесом росту організації. Але це також і впливає на якість підбору кадрів. Будь то відсутність контролю за рекрутерами, некомпетентність чи банальні людські помилки. А підбір некомпетентного персоналу може привести до зменшення ефективності роботи підприємства, зменшення його прибутків або навіть до збитків. Таким чином постає необхідність розробки методів і засобів автоматизації підбору персоналу, з метою підвищення ефективності роботи підприємства, що і зумовило **актуальність** обраної теми дослідження.

Аналіз останніх досліджень і публікацій. Дослідження проблем підбору кадрів, а також формування ефективних систем управління підприємствами привертає усе більше уваги як науковців, так й фахівців. Їх реалізація вимагає від усіх учасників ринку та пов'язаних із ними суб'єктів,

постійного пошуку раціональних форм організації діяльності, що висуває на перший план підвищення ефективності управління проектами зі створення нових та забезпечення результативного функціонування існуючих підприємств.

Проблеми підвищення ефективності добору працівників, незважаючи на їх актуальність і практичну значимість, досліджується на недостатньому рівні. Активніше досліджено загальні проблеми управління людськими ресурсами.

Наявний цілий ряд робіт вчених В. А. Архипова, Л. О. Кості, Е. В. Маслова та інших, у яких за часів адміністративно-командної економіки досліджувалася сутність близького йому поняття „кадрове забезпечення підприємств", однак і за його розгляду єдина точка зору серед них була відсутня, що не дозволяло провести чітку межу між вищезазначеними поняттями та розробити методичні підходи щодо вдосконалення кадрового забезпечення.

За останнє десятиліття з'явилося чимало наукових праць, присвячених різним аспектам організації роботи з кадрами, авторами яких є: В. Г. Бикова, Н. Д. Лук'янченко, Г. М. Чорний, В. Н. Гончарова, С. И. Радомський, Л. С. Лісогора та інші [30].

Закордонний досвід із проблем керування кадрами описаний у роботах У. Бреддика, М. Вудкока, Дж. (мол.) Грейсона, К. О'делла, Г. Десслера, Д. Мерсера, Я. Мейтланда, Р. С. Сміта, Д. Френсіса, Дж. Ханта, Р. Дж. Еренберга та інших[43]. Концептуальні положення, що розкривають залежність зміни трудових процесів від застосування нових інформаційних технологій, розкриті в дослідженнях тільки закордонних авторів: Є. Дайсона, М. Кастельса, Дж. Милковича, М. Розенберга, Є. Тоффлера, П. Хейне, А. Шермана.[10]

Мета і завдання дослідження. Метою даного дослідження є розробка математичного, алгоритмічного та програмного забезпечення оцінки кандидатів та відбору фахівця на заміщення вакантної посади. Для досягнення поставленої мети необхідно було вирішити наступні **завдання**:

- провести комплексне аналітичне дослідження проблематики підбору персоналу на підприємство;
- дослідити методи відбору фахівців на заміщення вакантної посади;
- розробити математичну модель підбору персоналу;
- розробити метод автоматизованого визначення оцінки якостей персоналу;
- розробити алгоритм відбору персоналу;
- розробити автоматизовану систему підбору персоналу на заміщення вакантної посади бухгалтера підприємства.

Об'єкт дослідження: процеси визначення можливого кандидата на заміщення вакантної посади.

Предмет дослідження: моделі, методи та інформаційні технології в системі оцінки кандидатів на заміщення вакантної посади.

Методи дослідження: теоретичною основою дослідження є загальнонауковий аналітичний метод, теорія нечітких множин та нечіткої логіки, системний підхід і праці провідних вчених з розв'язання багатокритеріальних задач, з проблем дослідження підбору персоналу. Для практичного вирішення поставлених задач використовувались такі методи:

- загальнонауковий аналітичний метод;
- метод нечіткого логічного виведення в розв'язанні задачі відбору фахівців;
- метод визначення вагових коефіцієнтів правил виведення;
- метод алгоритмічного програмування, для створення автоматизованої системи підбору фахівця на заміщення вакантної посади бухгалтера.

Практичне значення. Така інформаційна технологія процесу підбору фахівця, може забезпечити підвищення продуктивності й ефективності, спростити виконання менеджером (керівником) своїх задач і скоротити кількість кроків у процесі добору і розподілу персоналу, а також може

дозволити знизити обсяг "паперової" роботи, у результаті чого значно скоротиться кількість помилок, підвищиться якість даних – їх цілісність і несуперечність.

Також при використанні такої системи скоротяться витрати – збільшуючи продуктивність і оптимізуючи розподіл завантаження робітників підприємства, можна значно скоротити операційні й адміністративні витрати.

РОЗДІЛ 1.
ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО ОЦІНКИ КАНДИДАТІВ НА ЗАМІЩЕННЯ ВАКАНТНОЇ ПОСАДИ

1.1. Проблеми оцінки та відбору фахівців на вакантні посади

Однією з найбільш важливих складових частин найму працівників є професійний відбір, який значною мірою визначає результативність управління персоналом. Тому в сучасних організаціях, як правило, відбору персоналу надається належна увага. Об'єктивне рішення про вибір, в залежності від обставин, може ґрунтуватися на створенні враження про кандидата, рівня його професійних навичок, досвіду попередньої роботи, особистих якостях[34, с.61-65].

Але цей процес досить трудомісткий і вимагає від підприємства витрат часу, коштів та людського капіталу. І на підприємствах, які не приділяють цьому належної уваги часто виникають проблеми, безпосередньо пов’язані з робою працівників. Наприклад, зниження ефективності роботи підприємства, зменшення його прибутків, плинність кадрів, конфлікти між керівниками та підлеглими, нездорова атмосфера в колективі.

Ці проблеми спричинені не лише браком часу чи коштів, існує рад інших причин, які перелічені в праці з управління персоналу Одегова Ю.Г. та Нікoнової Т.В.[36]:

- Стихійний відбір, в наслідок якого відбувається часта зміна працівників на одному робочому місці.
- Відсутність у компанії єдиних критеріїв відбору. Тобто в залежності від посади до кандидата застосовуються різні критерії оцінювання, що не дає змогу визначити відповідність фахівця загальним цілям і культурі підприємства, що і може спричинити внутрішні конфлікти.
- Відбір кандидата виключно за побажаннями керівника, що може суперечити внутрішній культурі організації.

- Невідповідальне ставлення до підбору фахівців, яке може проявлятись у формі занижених (готовність взяти будь-якого кандидата) або навпаки завищених вимог; у виборі кандидата з домінуванням “непрофільних” вимог.
- Невідпрацьована схема у прийнятті рішень щодо кандидата. Наприклад, рішення щодо одного кандидата може виносити один керівних, а щодо іншого - декілька керівників, причому у декілька етапів. Іншим прикладом незбалансованого процесу прийняття рішення є його надмірна тривалість (два і більше тижні).
- Недопрацьована система оцінювання, що проявляється у надмірних або навпаки у недостатніх вимогах до кандидата. Це може стосуватись як окремого підрозділу, так і підприємства в цілому.

Ті компанії, які все ж приділяють належну увагу підбору фахівців користуються методами оцінки, які можна розділити до двох груп: традиційні та нетрадиційні. До традиційних методів відносять тестування на профпридатність, біографічні тести та виявлення біографії, особистісні тести, інтерв’ю[16]. Нетрадиційними методами є використання поліграфу (детектор брехні)[12], методики психоаналізу [4] та тести, які ґрунтуються на аналізах сечі і крові, експертиза почерку, метод аналізу ранніх дитячих спогадів.

Але і ці методи оцінювання мають ряд недоліків. Що стосується традиційних методів, а саме тестувань, то фахівці виділяють такі недоліки:

- Труднощі у виконанні тестів через незвичну форму подання питань, що не зустрічається у повсякденному житті.
- Можливість підготуватись до тесту. Тести, які найчастіше використовуються підприємствами є в широкому доступі в Інтернеті та зустрічаються в спеціальній літературі.
- Використання неадаптованих в Україні закордонних тестів.
- Відсутність перевіреної інформації про зв’язок результатів тестів з результатами подальшої діяльності відібраних кандидатів[14].

Ще один популярний метод оцінювання - інтерв'ю є процесом міжособистісної взаємодії, а тому виникає ряд ризиків, які пов'язані саме з міжособистісним сприйняттям:

- помилка центральної тенденції виникає у випадку оцінювання більшості кандидатів середнім балом;
- ефект ореолу, який виникає якщо інтерв'юер орієнтується в оцінці кандидата на якусь певну рису (хорошу чи погану), яка є на його погляд головною;
- помилка контрасту, яка виникає, якщо середній кандидат отримує високу оцінку через те, що проходить інтерв'ю після декількох слабких претендентів і навпаки - низьку, якщо йде після сильних.

Нетрадиційні методи оцінки претендентів не можуть використовуватись як основні або єдині засоби оцінки тому що оцінюють переважно особистісні, а не професійні якості кандидата та є досить суб'єктивними.

Данюк В.М., Петюх В.М., Цимбалюк С.О. розрізняють методи в залежності від форми вираження кінцевого результату на три групи: описові, кількісні та комбіновані методи. При описових методах оцінка подається в текстовій формі, при кількісних і комбінованих використовуються шкали оцінок, які дозволяють кількісно виміряти різні рівні виконання робіт в організації[15].

Федулова Л.І. вважає, що методи оцінки персоналу необхідно поділяти на методи індивідуальної та методи групової оцінки [40].

Враховуючи вище наведені недоліки використання тих чи інших методів, можна сформувати загальні критерії до технологій оцінки претендентів на вакантні посади:

- Об'єктивність - незалежність від будь-якої приватної думки.
- Надійність - захищеність від впливу зовнішніх факторів(настрою, погоди, минулих успіхів чи невдач).

- Достовірність - оцінка реальних вмінь, знань та навичок претендента.
- Комплексність - відповідність кандидата не тільки вимогам посади, але й цілям та завданням підприємства.
- Доступність - цілі та методи, які використовуються при оцінці повинні бути знайомими та доступними не тільки вузькому колу спеціалістів, але й тим, кого оцінюють[27].

Саме для того, щоб уникнути помилок у прийнятті кандидата на посаду чи відмові йому, які спричинені людським фактором використовуються автоматизовані системи оцінки кандидатів. Процедура математичного моделювання заміняє трудомісткі тестування на розрахунки. При використанні таких методів порівняння варіантів відбувається значно швидше та точніше.

1.2. Теорія нечітких множин та нечіткої логіки

Для автоматизації процесу оцінювання кандидатів необхідний математичний апарат, який здатний взаємодіяти з людиною та переводити певні критерії в мову чітких і формальних математичних формул. Прикладом такого апарату є теорія нечітких множин, основи якої було закладено наприкінці 60-х років відомим американським математиком Лотфі Заде[3]. Заде створив апарат для опису процесів інтелектуальної діяльності, включаючи нечіткість та невизначеність виразів (наприклад, відповідний, більш відповідний, менш відповідний кандидат). І на відміну від класичного поняття множини, в якому функція може набувати лише значень 0 та 1, нечіткі множини за Заде можуть набувати будь-яких значень в інтервалі [0,1].

Методи нечітких множин особливо корисні за відсутності точної математичної моделі функціонування системи. Теорія нечітких множин дає можливість застосувати для прийняття рішень неточні та суб’єктивні експертні знання про предметну область без формалізації їх у вигляді традиційних математичних моделей. З використанням теорії нечітких множин

вирішуються питання узгодження суперечливих критеріїв прийняття рішень, створення логічних регуляторів систем. Нечіткі множини дають змогу застосовувати лінгвістичний опис складних процесів, установлювати нечіткі відношення між поняттями, прогнозувати поведінку системи, формувати множину альтернативних дій, виконувати формальний опис нечітких правил прийняття рішень[25].

Основними поняттями теорії нечітких множин є нечітка множина, лінгвістична змінна, терм, терм-множина.

Розглянемо поняття нечіткої множини. Нехай X – універсальна множина, x – елемент X, а G – деяка властивість. Нечіткою підмножиною A універсальної множини E називається множина впорядкованих пар: A = {μA(x) / x}, де μA(x) - функція належності, яка приймає значення в деякій заданій множині M (наприклад M = [0,1]). Функція належності μA(x) вказує ступінь істинності, з яким елемент x з X належить підмножині A. Нечітка підмножина A може бути також названою нечіткою множиною на універсальній множині X. Якщо множина M складається тільки з двох чисел 0 та 1, то нечітка підмножина A стає звичайною підмножиною універсальної множини X[24]. Функція належності може задаватись:

- самостійно дослідником на основі власного досвіду; - в просторовому випадку;
- із залученням експертів, обробкою їх оцінок - для більш відповідальних випадків[28].

Існує більше десятка типових форм кривих для задання функції належності. Найбільшою популярністю користуються трикутна, трапецієвидна та гаусова функції належності.

Найменування функції	Аналітичний вираз	Інтерпретація параметрів
Трикутна	$f_{\Delta}(x;a;b;c)=\begin{cases}0, & x\le a\\ \frac{x-a}{b-a}, & a\le x\le b\\ \frac{c-x}{c-b}, & b\le x\le c\\ 0, & c\le x\end{cases}$	a, c – носій нечіткої множини; b – координата максимуму
Трапецієвидна	$f_T(x;a;b;c;d)=\begin{cases}0, & x\le a\\ \frac{x-a}{b-a}, & a\le x\le b\\ 1, & b\le x\le c\\ \frac{d-x}{d-c}, & c\le x\le d\\ 0, & d\le x\end{cases}$	a, d – носій нечіткої множини; b, c – ядро нечіткої множини
Гаусова	$f_{\gamma}(x;\sigma;c)=e^{-\frac{(x-c)^2}{2\sigma^2}}$,	c – координата максимума; σ – коефіцієнт концентрації
Сигмоїдальна	$f_{sig}(x;a;b)=\frac{1}{1+e^{-a(x-b)}}$,	a – коефіцієнт крутизни; b – координата переходу через 0,5
Сінглтон	$f_{singl}(x;a)=\begin{cases}1, & x=a\\ 0, & x\ne a\end{cases}$,	a – нечітке число, що представлене нечіткою множиною

Рис.1 Основні функції належності

За допомогою нечітких множин формалізують терми лінгвістичних змінних нечітких моделей. Лінгвістична змінна характеризується набором (L, T (L), U, G, M), в якому L – назва змінної; T (L) – терм-множина змінної L, U – універсальна множина базових значень (область, в якій визначені значення лінгвістичної змінної); G – синтаксичне правило; M – семантичне правило [6]. Терм-множина T (L) являє собою сукупність термів – назв лінгвістичних значень змінної L. Кожному терму відповідає нечітка підмножина безлічі U, визначальне лінгвістичне значення терма. Іншими словами, зміст терма характеризується функцією належності μ: U → [0, 1], яка кожному елементу, де u належить U ставить у відповідність значення приналежності цього елементу терму. Синтаксичне правило G має зазвичай форму граматики, що породжує терми. Терм, що складається з одного або більше атомарних термів, називається складовим термом.

Семантичне правило M ставить у відповідність кожному атомарному

терму його зміст у вигляді нечіткої множини. Крім того, семантичне правило М пов'язує приналежності атомарних термів в складені лінгвістичні значення з приналежністю складеного значення[26].

Важливість використання логіко-лінгвістичних змінних обумовлена наступними причинами:

- не всі критерії оптимізації можуть задаватись кількісно;
- між рядом параметрів, які впливають на функціонування системи, не вдається встановити точних кількісних залежностей;
- процес автоматизації системи - багатокроковий, а зміст кожного кроку не завжди можна визначити однозначно;
- існуючі описи системи та її структури дуже громіздкі, їх важко, а часто і неможлива використовувати[24, с.78].

Для нечітких множин можна визначити такі ж логічні операції, які і для однозначної логіки, наприклад, NOT, OR, AND[6]. Також для нечітких множин характерні такі властивості, як: об’єднання відношень, операції включення, комутативність, асоціативність, ідемпотентність, дистрибутивність, рефлексивність, транзитивність, симетричність[17].

Основними характеристиками нечітких множин є наступні характеристики:

- Носій нечіткої множини A - чітка множина A1 таких точок в U, для яких величина $\mu A(x)$ додатна, тобто $A1 = \{ x \mid \mu A(x) > 0\}$.
- Висота нечіткої множини A - верхня межа її функції належності $\mu A(x)$.
- Нормальна нечітка множина A, якщо $\mu A(x) = 1$. В іншому випадку воно називається субнормальною нечіткою множиною [44].
- Ядро нечіткої множини - чітка підмножина універсальної множини U, елементи якої мають ступені належності рівні 1. Ядро субнормальної нечіткої множини пусте.
- Точка переходу нечіткої множини A - це такий елемент x є, для якого $\mu A(x) = 0,5$.

Головною операцією нечіткої логіки є процедура нечіткого висновку, за допомогою якої з нечітких умов отримують наближенні рішення. Процедура нечіткого висновку заснована на операції логічного слідування .Імплікація дозволяє формалізувати знання експерта за формою “якщо А, то В”, де А - передумова, а В - висновок. Стосовно нечіткого управління X - базова множина значень x регулярної змінної. Тоді X базова множина x, A - множина значень x, U - базова множина значень управління u, B - множина значень u.

В залежності від способу отримання логічних висновків з нечітких правил можуть бути різні регулятори. Наприклад, алгоритми Мамдані, Ларсена, Цукамото, Сугено, системи нечіткого логічного виведення із зваженою істинністю.

1.3. Характеристика багатокритеріальних задач

Вибір кандидата на посаду в підприємстві можна віднести до задачі багатокритеріального вибору, в якій у ролі альтернатив виступає саме скінченна кількість претендентів на вакантну посаду, і ці альтернативи якісно оцінюються. Особливістю таких задач, які виступають у ролі певної моделі є те, що вибір здійснюється з множини альтернатив, які не покращуються[37].

В економічні діяльності при моделюванні процесів прийняття рішень, які вимагають розгляду багатьох альтернатив, тобто є багатокритеріальними виникають дві ситуації:

1. Повна формалізація процесу пошуку найвигідніших для підприємства рішень, можлива лише для вивчених та кількісно сформульованих задач.

2. Для розв’язання якісно виражених задач повністю сформованих алгоритмів не існує, окрім примітивного та ненадійного алгоритму перебору (метод проб та помилок).

Також проблеми, що виникають в процесі управління соціально-економічними системами можна поділити на три типи.

До першого типу належать проблеми, які чітко виражені, тобто числами або символами і тому при їх вирішенні не виникає труднощів, оскільки вони проблеми такого типу добре формалізовані. Прикладами таких задач на підприємстві є задачі бухгалтерського обліку, складського обліку.

До другого типу проблем належать погано структуровані проблеми, для яких описані лише певні характеристики, ознаки та ресурси, але не встановлені зв'язки між ними, відсутні кількісні залежності. Розв'язання таких задач можливе лише у разі використання неформалізованих процедур, які базуються на неструктурованій та невизначеній інформації. До таких проблем на підприємстві відносяться проблеми, що пов'язані з плануванням, організаційним перетворенням, прийняттям стратегічних рішень. Більшість неструктурованих проблем розв'язується за допомогою евристичних методів, у яких не передбачена жодна упорядкована логічна процедура пошуку їх розв'язку, а сам метод цілком залежить від особистих характеристик людини (інформованості, кваліфікації, досвіду, інтуїції тощо). Особливістю евристичних методів є здатність здійснювати пошук рішення навіть тоді, коли не сформульовано задачу і невідомі способи її розв'язування[42, с.159].

До третього типу належать слабоструктуровані або так звані змішані проблеми, які містять як кількісні, так і якісні елементи. Для цього типу задач відсутні методи розв'язання, які базуються на перетворені даних. Тому часто рішення таких задач виноситься без достатньої кількості інформації. Наприклад, розподіл капіталовкладень, вибір проектів та наукових досліджень, складання плану виготовлення. Характерними особливостями слабоструктурованих задач є те, що рішення часто стосуються майбутнього; рішення є унікальним, адже кожна така проблема має певні відмінності, в порівнянні з попередньою; рішення пов'язані з невизначеністю в оцінках альтернатив, що об'єктивно зумовлена недостатньою кількістю інформації на момент прийняття рішення; оцінки альтернативних варіантів розв'язку слабоструктурованих задач мають якісний характер і в більшості випадків сформульовані вербально.

У загальному випадку прийняття рішення відбувається на основі дискретної множини альтернатив. І в результаті цього рішення буде отриману підмножину початкової множини. Звуження цієї множини можливо шляхом порівняння критеріїв між собою, та вибором найбільш привабливих. При цьому вважається, що множина, на якій потрібно цей вибір зробити є попередньо сформованою. В цій множині визначено цілі, заради яких здійснюється вибір та існують критерії оцінки та порівняння альтернатив.

Для вирішення багатокритеріальних задач математичним способом можемо використати наступний набір інформації: <T, X, R, A, F, G>, де T – постановка задачі вибору рішення; X – множина допустимих альтернатив; R – множина критеріїв вибору; A – множина шкал вимірювань за критеріями (порядкові, інтервальні, відношень, найменувань); F – відображення множини допустимих альтернатив на множину критеріальних оцінок (наслідки рішень); G – система переваг особи, що приймає рішення.

Математичні постановки багатокритеріальних задач прийняття рішень можуть суттєво відрізнятись одна від одної. Класифікація цих задач здійснюється за наступними ознаками[2;5]:

- за множиною альтернатив X задачі поділяються на злічені, скінченні та континуальні;
- за потужністю множини R (оцінка може здійснюватися за одним або декількома критеріями, які в свою чергу, можуть мати як кількісні, так і якісні характеристики) – задачі прийняття рішень бувають однокритеріальні (з скалярним критерієм) та багатокритеріальні (з векторним критерієм);
- за видом відображення множини F (детерміновані, ймовірнісні та невизначені) - розрізняють задачі прийняття рішень в умовах визначеності, коли наслідки прийнятого рішення точно відомі; в умовах ризику, якщо відомі ймовірності можливих наслідків прийнятих рішень; та в умовах невизначеності, коли жодних здогадів не існує;
- за типом систем переваг G (переваги однієї особи або переваги колективу) - задачі індивідуального та групового вибору;

- за режимом вибору рішення - одноразові задачі, або такі, що повторюються допускаючи навчання в процесі прийняття рішень;
- за відповідальністю прийняття рішень - односторонні чи багатосторонні. При багатосторонньому прийняті рішення ступінь його узгодженості може коливатись від повної узгодженості до її протилежності, тобто прийняттю рішення в конфліктних умовах.

В теорії для вирішення багатокритеріальних задач найкращою буде альтернатива, що матиме найменше значення за усіма n критеріями з множини X, де f(x), i = 1,2,..n. Рішення, яке відповідає такий альтернативі є найефективнішим. Проте на практиці розв'язання багатокритеріальних задач можливо лише в тому разі, якщо пошук здійснюється на множині узгоджених рішень. В іншому випадку, така задача не має єдиного розв'язку.

Найбільш поширеними методами розв'язання багатокритеріальних задач є[13, с.8]:

- метод скаляризації, суть якого полягає у зведенні багатокритеріальної задачі до однокритеріальної шляхом введення певного суперкритерію у вигляді скалярної функції векторного аргументу: $F(x) = G(f_1(x), f_2(x),...f_n(x))$;
- метод умовної мінімізації, суть якого полягає у використанні припущення про те, що окремі критерії, як правило, є нерівнозначними між собою (деякі менш важливі, ніж інші). Виділяють головний критерій, а решту розглядають як додаткові, неосновні. Тоді задача прийняття рішення полягатиме в знаходженні екстремуму основного критерію, за умови, що значення решти критеріїв буде знаходитись у певних інтервалах;
- метод послідовних поступок. Суть методу полягає у впорядкуванні частинних критеріїв у порядку убування їх важливості, виборі найважливішого критерію і визначенні найкращої за цим критерієм альтернативи. Далі визначають величину, на яку можливо зменшити отримане значення найважливішого критерію, щоб за його рахунок збільшити, наскільки можливо, значення наступного за важливістю критерію і т. д.;

• пошук множини Парето. За цим методом відмовляються від пошуку найкращої альтернативи і погоджуються з тим, що перевагу одній альтернативі перед іншою можна надавати лише у випадку, коли вона краща за усіма критеріями. У випадку, коли перевага за одним критерієм не збігається з перевагою за іншим, такі альтернативи вважають непорівнянними. У результаті попарного порівняння альтернатив, усі гірші за усіма критеріями альтернативи відхиляються, а ті, що залишились (не домінуючі), приймаються. Якщо усі максимально допустимі значення частинних критеріїв не відповідають єдиній альтернативі, то прийняті альтернативи утворюють множину Парето і на цьому процедура прийняття рішення закінчується. При необхідності вибору лише однієї альтернативи вводяться додаткові критерії, обмеження, залучають експертів;

• вибір на основі багатокритеріальної функції корисності використовується, якщо задані n альтернатив, m наборів зовнішніх умов та k критеріїв. Кожна альтернатива оцінюється за усіма критеріями для усіх наборів зовнішніх умов. Далі формують k матриць n x m. Кожен елемент матриці x являє собою числову оцінку i-ї альтернативи по критерію k при реалізації j-го набору зовнішніх умов. Для прийняття рішення вводиться багатокритеріальна функція корисності, що має вигляд числової функції, визначеної на множині можливих альтернатив і для якої виконується умова: якщо альтернатива x1 більш приваблива, ніж альтернатива x2, то $u(x1) \geq u(x2)$. Основна проблема методу полягає у складності побудови єдиної для усіх критеріїв функції корисності.

Але усі наведені вище методи мають певні обмеження при використанні у вирішенні реальних економічних проблем. Це пов’язано з тим, що значна кількість економічних параметрів є невизначеними, неточними, нечітко заданими, тому застосування при побудові формальних моделей прийняття рішень детермінованих методів призводить до значних помилок. У ситуаціях прийняття рішень, коли хоча б один із елементів задачі (альтернативи, критерії, обмеження, залежності) описується якісно (нечітко),

має місце задача прийняття рішення при нечіткій вхідній інформації. Перспективним напрямком розробки методів прийняття рішень у нечіткому середовищі є лінгвістичний підхід на базі теорії нечітких множин[1;39;8].

Задача прийняття рішення у нечіткому середовищі характеризується наступними поняттями:

- нечіткими критеріями;
- нечітким оператором максимізації(нечіткі інструкції типу ”бажано, щоб значення критерію x було якомога більшим”);
- нечітким критеріальним обмеженням(нечіткі інструкції типу “бажано, щоб значення критерію x було більше, менше або дорівнювало значенню α”);
- інформацією про приблизну важливість критеріїв(вагові коефіцієнти, низка пріоритетів);
- нечіткими обмеженнями на вектор незалежних змінних.

Прийняття рішення в нечітких умовах передбачає використання логічної схеми, важливою рисою якої є симетрія по відношенню до цілей та обмежень. Таким чином знімається різниця між цілями і обмеженнями, що дозволяє досить просто сформувати на їх основі рішення. Нехай X ={x} – задана множина альтернатив. Тоді нечітка ціль G буде ототожнюватися з фіксованою нечіткою множиною G в X. У традиційному підході задається критерій f(x), що використовується в процесі прийняття рішення для встановлення лінійної впорядкованості на множині альтернатив. Очевидно, що функція належності нечіткої цілі виконує ту ж задачу і може бути отримана з критерію f(x) за допомогою нормалізації, що зберігає встановлену лінійну впорядкованість. Схожим чином нечітке обмеження C в просторі X визначається як деяка нечітка множина в X. І ціль, і обмеження розглядаються як нечіткі множини в просторі альтернатив. Це дає можливість не робити між ними різниці при формуванні рішення.

Рішення – це вибір однієї чи декількох альтернатив із множини існуючих. Зважаючи на це, нечітке рішення визначають як нечітку множину в

просторі альтернатив, що отримана в результаті перетину заданих цілей та обмежень[21].

РОЗДІЛ 2.
ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИРІШЕННЯ ЗАДАЧІ ОЦІНКИ КАНДИДАТІВ ТА ВІДБОРУ ФАХІВЦЯ НА ЗАМІЩЕННЯ ВАКАНТНОЇ ПОСАДИ

2.1. Постановка задачі на розробку системи оцінки та відбору фахівців

Задача оцінки та відбору фахівця на вакантну посаду є багатокритеріальною задачею, адже містить в собі певну кількість альтернатив (претендентів на посаду), кожна з яких має певний пріоритет, який залежить від навиків та якостей якими володіють кандидати. Але характеристики цих кандидатів при відборі на посаду часто задаються не чітко, тобто не мають кількісного вираження, а лише якісне (наприклад, пунктуальність, відповідальність, стресостійкість і т. д.). В такому випадку має місце задача прийняття рішення при нечіткій вхідній інформації. Тому рішення таких задач здійснюється за допомогою теорії нечітких множин та нечіткої логіки.

Змістовна постановка задачі вибору кандидата на вакантну посаду. Елементи, що належать різним множинам назвемо призначенням, а сукупність n призначень, що охоплює усі суб’єкти та об’єкти - рішенням задачі. Пред’являючи вимоги до якості призначень, тобто до ступеня в відповідності характеристик елементів множини, допустимої при утворенні пар, формується область допустимих значень, яка також включатиме обов’язкові призначення або виключатиме неприпустимі. Формуючи призначення в області допустимих значень, посадова особа, яка приймає рішення прагне до одного з можливих рішень, за якого не можна поліпшити якість призначення для якої-небудь пари елементів, не погіршивши не при цьому якість призначення для інших пар. Назвемо такі рішення ефективними. Серед цих рішень посадова особа прагне відшукати таке, яке дає змогу отримати максимальну кількість якнайкращих можливих призначень. Варіант, який повністю задовольняє співвідношення між можливостями та потребами, назвемо оптимальним.

Враховуючи описані вище особливості , сформулюємо змістовну постановку задачі наступним чином.

Початковим етапом вирішення цієї задачі буде формування лінгвістичних термів типу “якщо, то” (наприклад, якщо кандидат має відповідну кваліфікацію, освіту та досвід бухгалтерського обліку, то він задовольняє вимогам). Наступник кроком буде визначення можливих значень лінгвістичних змінних X та Y, які будуть використовуватись для оцінки кандидатів, наприклад “ якщо X1 = відповідний, X2 = вища, X3 = достатній, то Y = задовільний ”. Далі потрібно визначити функцію належності та нечіткі множини оцінки кандидатів.

Таб.1.1

Нечіткі значення лінгвістичної шкали

Значення лінгвістичної змінної	Оцінка
Дуже низька	$\mu_{ДН}$={0,0,0.1,0.3}
Низька	$\mu_{Н}$={0.1,0.3,0.3,0.5}
Допустима	$\mu_{Д}$={0.3,0.5,0.5,0.7}
Висока	$\mu_{В}$={0.5,0.7,0.7,0.9}
Дуже висока	$\mu_{ДВ}$={0.7,0.9,1,1}

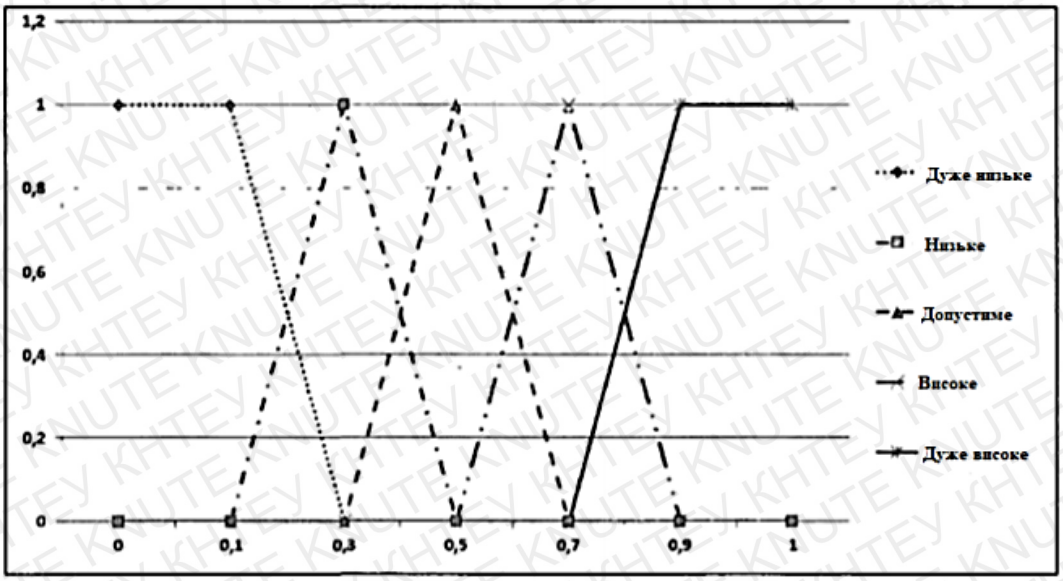

Рис.1.2. Функція належності п’ятибальної лінгвістичної нечіткої шкали

Як видно з малюнка, нечіткі числа «зачеплені» один за одного. Це відображає той факт, що немає різкого поділу між сусідніми оцінками, і перехід від однієї бальної оцінки до іншої відбувається поступово. В результаті оцінки якісної характеристики за допомогою бальної шкали фахівці можуть отримати його нечітку кількісну інтерпретацію.

Для визначення задовільності кожної альтернативи необхідно застосувати правило композиційного виведення в нечіткому середовищі. Потім порівняти нечіткі підмножини в одиничному інтервалі для порівняння найкращого рішення на основі точкових оцінок кожної альтернативи.

Точковими оцінками параметрів розподілу генеральної сукупності будуть такі оцінки, які визначаються одним числом. Наприклад, вибіркова середня x, вибіркова дисперсія D_b – точкові оцінки відповідних числових характеристик генеральної сукупності. Точкові оцінки параметрів розподілу є випадковими величинами, їх можна вважати первинними результатами обробки вибірки оскільки невідомо, з якою точністю кожна з них оцінює відповідну числову характеристику генеральної сукупності. Якщо об’єм вибірки досить великий, то точкові оцінки задовольняють практичні потреби точності[29]. При розв’язку нашої задачі, в якості найкращої вибираємо альтернативу з найбільшою точковою оцінкою.

Формалізація знань за допомогою правил дозволяє враховувати різну важливість критеріїв і самих правил. В задачі оцінки та вибору кандидата можливі різні підходи: м’який, жорсткий та раціональний. М’який підхід зазвичай має місце в умовах дефіциту часу і кваліфікованих кадрів, основну директиву цього методу можна сформулювати так “аби щось вмів робити”. Жорсткий підхід до вибору кандидата на вакантну посаду можливий у випадку великої кількості кваліфікованих кадрів та достатньої кількості часу, відведеного на вибір кандидата. Ціллю жорсткого відбору є пошук кандидата, який найбільш відповідає образу так званого “ідеального” кандидата. Підхід з

використанням правил, які мають однакову вагу можна вважати раціональним[2].

Отож, виходячи з вище сказаного для вирішення даної задачі необхідна:

- алгоритмічна реалізація методу нечіткого логічного виведення, який визначатиме найкращу альтернативу, тобто найбільш відповідного кандидата;
- алгоритмічна реалізація методу визначення вагових коефіцієнтів, який буде базуватись на вхідних даних у вигляді лінгвістичних змінних.

2.2. Метод нечіткого логічного виведення в розв'язанні задачі відбору фахівця

Існує достатньо методів розв'язання задач з нечіткими вхідними даними. Розглянемо їх більш детально.

Система нечіткого логічного виведення для одержання узагальнених оцінок об'єктів має n входів та один вихід – результат нечіткого логічного виведення. На вхід системи нечіткого логічного виведення подаються вхідні величини (впливові фактори) x_i, i =1,2,..k. На виході система видає узагальнену чітку оцінку, що є результатом системи нечіткого логічного виведення для заданих вхідних значень. Нечітке логічне виведення визначає відображення вектора вхідних даних у скалярне вихідне значення за допомогою нечітких правил. Система нечіткого логічного виведення складається з трьох основних компонентів: фазифікатора, механізму логічного виведення та дефазифікатора. Фазифікатор визначає ступінь належності вхідних значень, x_i, i =1,2,..k до нечітких множин входу Kx_i, $\{K^1_{xi}, K^2_{xi}, .., K^{mxi}_{xi}\}$, де m_{xi} – кількість лінгвістичних змінних з відповідної лінгвістичної шкали для i-го входу.

Необхідність у введенні процедури фазифікації зумовлена використанням у системі нечіткого логічного виведення лінгвістичних правил і здійснюється вона для визначення ступеня істинності кожної передумови

кожного правила. Основою механізму логічного виведення є нечітка база правил, яка містить лінгвістичні правила. Вони можуть бути задані експертним шляхом чи отримані із деяких інших міркувань. Зазначимо, що це один із найбільш трудомістких та ресурсовитратних етапів нечіткого логічного виведення. Механізм логічного виведення відображає вхідні нечіткі множини K^j_{xi}, i = 1,2,..k кожного правила ц вихідну множину K^j_y з набору вихідних лінгвістичних змінних $K_y = \{K^1_y, K^2_y,..,K^{my}_y\}$. Загалом у нечіткій базі j-те правило (j=1,2..,r) формулюється так:

якщо $x_1 \epsilon K^j_{x_1}$ і $x_2 \epsilon K^j_{x_2}$ і ... і $x_1 \epsilon K^j_{x_1} x_k \epsilon K^j_{x_k}$ то $y \epsilon K^j_y$ (2.1)

Вихідні нечіткі множини y_j, j = 2,1 ,..,r кожного правила об'єднуються в одну нечітку множину виведення y. Потім дефазифікатор відображає нечітку множину виведення ȳ у чітке число Ỹ, яке і буде результатом системи нечіткого логічного виведення для заданих вхідних значень x_i, i=1,2,..k. Часто вживаними методами дефазифікації є центроїдний метод, методи максимуму, метод центру максимумів, висотна дефазифікація[18].

Нечіткий алгоритм Мамдані [18]. У цьому алгоритмі процес нечіткого логічного виведення будується так. Б=аза знань складається з нечітких правил у вигляді. На етапі фазифікації визначаються ступені належності вхідних значень x_i , i = 1,2,…,k до нечітких множин входу, тобто визначаються ступені істинності $\mu^j_i(x_i)$ для кожної передумови кожного j-го правила. Потім для кожного правила на основі ступенів істинності передумов $\mu^j_i(x_i)$ розраховується ступінь його виконання a_j . Для цього застосовують композицію на основі оператора мінімуму:

$a_j = \min \{ \mu^j_1(x_1) , \mu^j_2(x_2),\ldots, \mu^j_k(x_k)\}$, j = 1,2,…r. (2.2)

Для кожного правила на основі ступеня виконання a_j , j = 1,2,…,r розраховується результат його виконання – вихідна нечітка множина з усіченою функцією належності μ^j (y). Визначають усічену функцію належності також за допомогою оператора мінімуму:

$\mu^j(y) = \min \{a_j, \mu^j(y)\}$, j = 1,2,…r. (2.3)

Механізм логічного виведення завершується тим, що вихідні нечіткі множини виконаних правил за допомогою оператора максимуму агрегуються в нечітку множину виведення $\bar{y}$, функція належності якої має вигляд:

$\mu_{\bar{y}} = \max \{\mu^1(y), \mu^2(y),.., \mu^r(y)\}$ (2.4)

Нечіткий алгоритм Ларсена [18]. За цим алгоритмом фазифікакція проводиться так само, як і за алгоритмом Мамдані. На відміну від використання оператора мінімуму, функцію належності μ^j (y) результату виконання кожного правила розраховують на основі оператора добутку:

$\mu^j(y) = a_j \times \mu^j(y)$, j = 1,2,…,r. (2.5)

Механізм логічного виведення завершується так само, як і за алгоритмом Мамдані. Для приведення до чіткості проводять процедуру дефазифікації нечіткої множини. Метод дефазифікації вибирають залежно від конкретної задачі. Придатним для багатьох класів задач та найпоширенішим є центроїдний метод, що полягає у знаходженні центру ваги (центроїду) нечіткої множини, який і обирається за кінцевий результат.

Нечіткий алгоритм Цукамото [18]. Цей алгоритм використовують у тому випадку, коли функції належності вхідних та вихідних величин є монотонними. Фазифікація входів проводиться так само, як і за алгоритмом Мамдані. Для кожного j-го правила на основі ступенів істинності передумов $\mu^j_i(x_i)$ розраховується ступінь його виконання a_j. Для цього застосовують композицію на основі оператора мінімуму. Механізм логічного виведення завершується тим, що розв’язуються рівняння виду:

$a_j = \mu^j(y)$, j =1,2,…,r (2.6)

визначаються чіткі значення для кожного правила. Чітке значення вихідної величини знаходиться як зважене середнє.

Нечіткий алгоритм Сугено [18]. У цьому алгоритмі нечіткого логічного виведення база правил складається з правил, у яких виходом є чітке значення, що визначається лінійною функцією:

Правило j : “Якщо $x_1 \in K^j_{x1}$ і $x_2 \in K^j_{x2}$ і…і $x_k \in K^j_{xk}$ то $\bar{y}^j$ дорівнює сумі $a^j_i x_i$, де a^j_i – константи..

Остаточний результат логічного виведення Ỹ розраховується як зважене середнє чітких результатів y^j виконаних правил, в якому ваговими коефіцієнтами є ступені виконання правил a_j.

Система нечіткого логічного виведення із зваженою істинністю [23]. База нечітких продукційних правил у системі нечіткого логічного виведення створюється або групою експертів, або її можна побудувати із деяких інших міркувань. Важливість всіх впливових факторів при складанні правил вважають однаковою. При застосуванні системи нечіткого логічного виведення коригують процес логічного виведення з урахуванням значень вагових коефіцієнтів усіх впливових факторів. Зазвичай вагові коефіцієнти є нормованими, і їх можна використовувати як коригувальні множники. Тобто, в системі нечіткого логічного виведення за таким методом ступінь істинності кожної передумови μ^j_i , що відповідає фактору x_i j-го правила, множиться на відповідний ваговий коефіцієнт p_i. Тоді ступінь виконання j-го правила a_j при застосуванні системи нечіткого логічного виведення визначатиметься так:

$$a_j = \min \{p_1\mu^j_1(x_1), p_2\mu^j_2(x_2),\dots p_i\mu^j_k(x_k)\}. \quad (2.6)$$

Але у такому випадку значення оцінки за менш важливим фактором, якому відповідає менше значення вагового коефіцієнта, значно впливатиме на значення ступеня виконання a j всього правила, тому що у процедурах композиції та імплікації відповідно до логічного “і” використовується оператор мінімуму. Отже, може виникати ситуація штучного заниження значення ступеня виконання певного правила a_j. Тому такий спосіб зважування ступенів істинності передумов правил для проблеми визначення рівня компетентності може бути некоректним. Для врахування значень вагових коефіцієнтів впливових факторів та забезпечення адекватної участі кожного фактора у формуванні значення ступенів виконання правил у результаті логічного виведення пропонується для визначення ступеня виконання a_j правила в процедурі композиції ступенів істинності передумов або замість оператора мінімуму використовувати спеціальний апарат зваженої агрегації значень ступенів істинності $\mu^j_i(x_i)$ передумов правила. При цьому

правило виконується, якщо значення ступенів істинності μ^j_i (x_i) кожної передумови правила більше за нуль. В окремих випадках доцільно застосовувати в процедурі композиції зваженої суми значень оцінок, тобто лінійної комбінації значень оцінок за всіма факторами та їх вагових коефіцієнтів. За такого підходу ступінь виконання j-го правила a_j при застосуванні системи нечіткого логічного виведення визначатиметься так:

a_j дорівнює сумі $p_i \times \mu^j_i (x_i)$, $\mu^j_i (x_i) > 0$.

Описана система нечіткого логічного виведення із зваженою істинністю об'єднує позитивні моменти ідей різних нечітких алгоритмів і відрізняється від зваженої системи нечіткого логічного виведення, в якій кожному правилу надається своє значення ваги.

2.3. Метод визначення вагових коефіцієнтів правил виведення

Відома значна кількість методів оцінювання вагових коефіцієнтів. В різних літературних джерелах вони можуть мати назву коефіцієнтів ваги, вагомості, впливу, значимості тощо. В найпростішому випадку ваговий коефіцієнт визначається як математичне сподівання частки вихідного ефекту, що вноситься окремим показником. Однак, коректніше було б шукати для вагових коефіцієнтів миттєві інтегральні значення, оскільки тоді максимально збільшувалася б вірогідність кінцевої оцінки, отриманої з їх використанням[9].

Існує два основні підходи визначення вагових коефіцієнтів : експертний та розрахунковий. Розглянемо детальніше кожний з них.

Перевагою експертних методів є простота застосування, недолік – можлива недостовірність даних, певна суб'єктивність, трудомісткість збору та обробки результатів. На сьогодні існує багато методів обробки результатів експертиз (методів експертного оцінювання та прогнозування): попарних зіставлень (порівнянь); надання переваг; рангів; нечітких експертних оцінок; аналізу ієрархій та їх удосконалені чи узагальнені варіанти. Крім того, існують методи організації проведення експертиз (інтерв'ю, анкетування, метод Делфі,

метод комісій та ін.) і методи визначення компетентності експертів та вагомості їх оцінок.

Для підтвердження узгодженості думок експертів необхідно застосувати коефіцієнт конкордації, який у подальшому був оцінений за критерієм Пірсона[20;38;41;22]. При оцінці суттєвості зазначених коефіцієнтів виходили з того, що чим ближче до одиниці наближається коефіцієнт конкордації, тим більш узгодженими є думки експертів. Одним із методів підтвердження істотності коефіцієнта конкордації є його оцінка за критерієм Пірсона x^2 . Якщо $x^2_{розп.} > x^2_{табл.}$, то коефіцієнт конкордації є істотним при ступенях свободи f = n–1 та заданому рівні значимості α = 0,01.

При визначенні вагових коефіцієнтів на основі попарних порівнянь[35] докладна числова інформація про досліджувані об'єкти не є обов'язковою, а порівняння можна здійснювати за якісними критеріями: "більше-менше", "краще-гірше", без уточнення, у скільки разів або на скільки більше чи краще. Метод попарних порівнянь було розроблено відомим американським математиком Т. Сааті.

Кожен показник, підгрупа або група показників співставляються з іншими (показниками, підгрупами та групами відповідно) на основі шкали відносної значущості. Експерти, виконуючи порівняння показників, заповнюють таблицю попарних порівнянь.

Оскільки значення показників невідомі заздалегідь, порівняння виконується на основі суб'єктивних суджень, які чисельно оцінюються за шкалою. Необхідно відмітити, що кожна клітинка таблиці містить не одне число (безпосередню оцінку експерта), а сукупність чисел, з урахуванням того, що порівняння показників проводиться опосередковано через порівняння їх з певним допоміжним фактором. Врахування цих додаткових порівнянь дозволяє підвищити надійність отриманих результатів, або зменшити кількість необхідних експертів Результатом порівняння є певне числове значення: показник, який має перевагу, оцінюється цілим числом (a_{ij}), протилежний – оберненим до нього ($1/a_{ij}$). Наприклад, якщо показник 1 за

рішенням експерта має “помірну перевагу” над показником 2, то перший оцінюється числом $a_{12} = 2$, а другий – $a_{21}=1/2$. Отримані значення записуються у вигляді матриці попарних порівнянь (табл. 4). Таблиця заповнюється по рядках (при порівнянні показників 1 з 2, оцінка 2 ставиться на перетині рядка 1 і стовпця 2, а оцінка 1/2 – на перетині рядка 2 і стовпця 1). На діагоналі заповненої таблиці завжди будуть стояти одиниці – порівняння показника самого з собою буде давати рівноцінність.

Методом ранжування[32] передбачає упорядкування елементів за мірою збільшення або зменшення пріоритетності за деякою ознакою. На основі отриманих кількісних оцінок об’єктів обирають найбільш пріоритетний. Упорядкування проводять у часі чи просторі або за конкретною ознакою для випадків, коли значення ознаки неможливо виміряти з причин практичного або теоретичного характеру.

Визначення вагових коефіцієнтів методом адаптивної оцінки[33] використовується у випадках, коли значущість показників змінюється з часом і відповідно виникає необхідність зменшення чи збільшення їх вагових коефіцієнтів:

- протягом тривалого часу показник стабільний і його значення практично не змінюється;
- величина вагового коефіцієнта може бути пропорційною темпу його зміни;
- при аналізі індикаторів економічної безпеки ряд показників може сягнути порогового значення;
- доцільно зменшити значення його вагового коефіцієнта, збільшивши при цьому ваги показників, які не досягли необхідного порогу; – показник може втратити свій економічний сенс з різних економічних, політичних і соціальних причин.

Визначення вагових коефіцієнтів методом парної кореляції[33] є розрахунковим методом, який проводиться з метою виявлення основних тенденцій зв’язку між показниками економіки. Якщо x_i (t) – значення і-го

показника в момент часу t, то коефіцієнт кореляції між значеннями показників (додатного чи від'ємного) знаходиться за формулою:

$$r_{ij}(t) = \langle \frac{\langle (x_i(t) - \langle x_i \rangle)(x_j(t) - \langle x_j \rangle) \rangle}{\sqrt{\langle (x_i(t) - \langle y_j \rangle)^2 \rangle \langle (x_j(t) - \langle x_j \rangle)^2 \rangle}} \rangle \quad (3.1)$$

Коефіцієнт парної кореляції між показниками, близький до одиниці, свідчить про тісний зв'язок між ними. У цьому випадку один з показників можна виключити з розгляду, істотно зменшити його вагу або об'єднати декілька показників в один, додавши їх ваги.

Метод вагових коефіцієнтів, який найчастіше використовують в умовах нечіткої логіки передбачає, що дослідник володіє певними апріорними даними, наприклад, про вагомість окремих показників властивостей. Але вагові коефіцієнти, що ймовірно можуть бути приписані показникам, характеризуються лише нечисловою, неточною і можливо неповною інформацією (nnn-інформацією) на зразок того, що показник властивостей Q_i є важливішим для узагальненого показника якості, ніж показник Q_j , або вказані межі варіювання значень вагових коефіцієнтів $0 \leq a \leq w \leq b \leq 1$. Застосовуючи для прикладу метод, який можна було б назвати рендомізацією невизначеності і який базується на застосуванні концепції Т. Баєса, сукупність i = 1, m вагових коефіцієнтів інтерпретується як вектор в m - вимірному евклідовому просторі.

Суть методу полягає в моделюванні невизначеності вибору конкретного вектору вагових коефіцієнтів $w = (w_1, w_2, .., w_m)$ з множини W(I) всіх допустимих вагових коефіцієнтів за допомогою рендомізованого вектору вагових коефіцієнтів, заданого за допомогою деякого розподілу на множині W(I) . Підставлення рендомізованого вектору вагових коефіцієнтів замість детермінованого дає рендомізований узагальнений комплексний показник, який можна інтерпретувати як "числове зображення nnn-інформації" і який поєднує інформацію про окремі значення показників властивостей разом з нечисловою, неточною і нечіткою інформацію про значимість цих окремих

показників властивостей. Тим самим задача порівняння багатокритеріальних оцінок якості двох складних об’єктів зведена до виявлення якого-небудь стохастичного домінування між двома випадковими величинами. Метод потребує хоча і нечіткої, та все ж апріорної інформації, яку можна отримати знову ж таки експертним методом. В поєднанні з методом регресивних залежностей обидва підходи можна трактувати як шлях об’єктивізації процесу ранжування вагових коефіцієнтів[7].

РОЗДІЛ 3.
ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВІДБОРУ ФАХІВЦІВ НА ЗАМІЩЕННЯ ВАКАНТНОЇ ПОСАДИ

3.1. Алгоритмічна реалізація методу нечіткого логічного виведення

Для вирішення задачі відбору та оцінки кандидатів на посаду кандидата будуть використані певні критерії їх оцінки. Розділимо ці критерії на декілька груп : функціональні, когнітивні та надпредметні (метапредметні).

Когнітивні критерії включають в себе освіту та рівень знань в широкому розумінні, здобуті в процесі навчання або особистісного пізнавального досвіду. Функціональні критерії включають в себе практичні навички, здобутий досвід. До надпредметних критеріїв відносяться знання та навички, які прямо не стосуються даної професії, в нашому випадку посади бухгалтера. Наприклад, юридична грамотність, знання іноземних мов, знання мов програмування[31].

Для нашої задачі відбору визначимо п’ять критеріїв : кваліфікація, освіта, досвід роботи на подібній посаді, вміння працювати з сучасним програмним забезпеченням та юридична грамотність. Даним критеріям присвоїмо значення X_1, X_2, X_3, X_4, X_5, а Y буде означати ступінь відповідності. Для формування правил необхідно визначити можливі значення лінгвістичних змінних X_i та Y, які будуть використовуватись для оцінки кандидатів:

d_1: ”Якщо X_1 = належна і X_2 = вища і X_3 = достатнє, то Y = задовільний”;

d_2: ”Якщо X_1 = належна і X_2 = вища і X_3 = достатнє і X_4 = здатний, то Y = більш ніж задовільний”;

d_3: ”Якщо X_1 = належна і X_2 = вища і X_3 = достатнє і X_4 = здатний і X_5 = володіє, то Y = бездоганний”;

d_4: ”Якщо X_1 = належна і X_2 = вища і X_3 = достатнє і X_4 = володіє, то Y = дуже задовільний ”;

d_5: ”Якщо X_1 = належна і X_2 = не вища і X_3 = достатнє і X_5 = володіє, то Y = задовільний ”;

d_6: ”Якщо X_1 = не має і X_3 = недостатнє, то Y = незадовільний”.

Змінна Y задана на множині J = {0; 0,1; 0,2;…;1}. Значення змінної Y задаються відповідно до наступних функцій належності:

S = задовільний, що визначено як $\mu_S(x) = x$, x є J;

MS = більш ніж задовільний - $\mu_{MS}(x) = sqrt(x)$, x є J;

P = бездоганний - як $\mu_P(x) = \{1$, якщо x =1; 0, якщо x < 1}, x є J;

VS = дуже задовільний - як $\mu_{VS}(x) = x^2$, x є J;

US = незадовільний - як $\mu_{US}(x) = 1 - x$, x є J.

Вибір здійснюється з-поміж п’яти кандидатів на множині U = $\{u_1, u_2, u_3, u_4, u_5\}$. В даній задачі оцінки кандидатів задані наступними нечіткими множинами:

належна (кваліфікація) A = $\{0{,}8/u_1, 0{,}61u_2, 0{,}5/u_3, 0{,}1/u_4, 0{,}3/u_5\}$;

вища освіта B = $\{0{,}5/u_1, 1/u_2, 0/u_3, 0{,}5/u_4, 1/u_5\}$;

достатній досвід C = $\{0{,}6/u_1, 0{,}9/и_2, 1/u_3, 0{,}7/u_4, 1/u_5\}$;

здатний (працювати з програмним забезпеченням) D = $\{1/u_1, 0{,}3/и_2, 1/u_3, 0/u_4, 0/u_5\}$;

володіє (юридичними знаннями) E = $\{0/u_1, 0{,}5/u_2, 1/u_3, 0{,}8/u_4, 1/u_5\}$.

З врахуванням уведених означень правила $d_{1...}d_6$ приймають вигляд:

d_1: “Якщо X= A і B, і C, то Y =S”;

d_2: "Якщо X= A і B, і C, і D, то Y = MS";

d_3: “Якщо X= A і B, і C, і D, і E, то Y = P”;

d_4: “Якщо X = A і B, і C, і E, то Y = VS”;

d_5: “Якщо X = A, і не B, і C, и E, то Y = S”;

d_6: “Якщо X = не A і не C, то Y = US”.

Визначимо функції належності μ_m для лівих частин приведених правил:

для d_1: $\mu_{M1}(u) = \min(\mu_A(u), \mu_B(u), \mu_C(u))$;

$M_1 = \{0{,}5/u_1,\ 0{,}6/u_2, 0/u_3, 0{,}1/u_4, 0{,}3/u_5,\}$;

для d_2: $\mu_{M2}(u) = \min(\mu_A(u), \mu_B(u), \mu_C(u), \mu_D(u))$;

$M_2 = \{0{,}5/u_1,\ 0{,}3/u_2, 0/u_3, 0/u_4, 0/u_5,\};$

для d_3: $\mu_{M3}\ (u) = \min(\mu_A(u), \mu_B(u), \mu_C(u), \mu_D(u)\mu_E(u));$

$M_3 = \{0/u_1,\ 0{,}3/u_2, 0/u_3, 0/u_4, 0/u_5,\};$

для d_4: $\mu_{M4}\ (u) = \min(\mu_A(u), \mu_B(u), \mu_C(u), \mu_D(u)\mu_E(u));$

$M_4 = \{0/u_1,\ 0{,}5/u_2, 0{,}5/u_3, 0{,}1/u_4, 0{,}3/u_5,\};$

для d_5: $\mu_{M5}\ (u) = \min(\mu_A(u), 1 - \mu_B(u), \mu_C(u), \mu_E(u));$

$M_5 = \{0/u_1,\ 0/u_2, 0{,}5/u_3, 0{,}1/u_4, 0/u_5,\};$

для d_6: $\mu_{M6}\ (u) = \min(1 - \mu_A(u), 1 - \mu_C(u));$

$M_6 = \{0{,}2/u_1,\ 0{,}1/u_2, 0/u_3, 0{,}3/u_4, 0/u_5,\};$

Тоді правила можна записати у вигляді:

d_1: Якщо X = M_1, то Y = S;

d_2: Якщо X = M_2, то Y = MS;

d_3: Якщо X = M_3, то Y = P;

d_4: Якщо X = M_4, то Y = VS;

d_5: Якщо X = M_5, то Y = S;

d_6: Якщо X = M_6, то Y = US.

Використовуючи, для перетворення правило виду “Якщо X = M, то Y = Q” імплікацію Лукасевича[19] μ_d (u, j) = min(1, 1 - μ_M /(u) + $\mu_y(j)$), для кожної пари (u, j) є U x J отримуємо наступні нечіткі відношення на U x J:

$$D_1 = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & 0{,}1 & 0{,}2 & 0{,}3 & 0{,}4 & 0{,}5 & 0{,}6 & 0{,}7 & 0{,}8 & 0{,}9 & 1 \end{array} \\ \left\| \begin{array}{ccccccccccc} 0{,}5 & 0{,}6 & 0{,}7 & 0{,}8 & 0{,}9 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}4 & 0{,}5 & 0{,}6 & 0{,}7 & 0{,}8 & 0{,}9 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}9 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}7 & 0{,}8 & 0{,}9 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} \right\| \end{array}$$

$$D_2 = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & 0{,}1 & 0{,}2 & 0{,}3 & 0{,}4 & 0{,}5 & 0{,}6 & 0{,}7 & 0{,}8 & 0{,}9 & 1 \end{array} \\ \left\| \begin{array}{ccccccccccc} 0{,}5 & 0{,}82 & 0{,}95 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} \right\| \end{array}$$

$$D_3 = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & 0{,}1 & 0{,}2 & 0{,}3 & 0{,}4 & 0{,}5 & 0{,}6 & 0{,}7 & 0{,}8 & 0{,}9 & 1 \end{array} \\ \left\| \begin{array}{ccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}7 & 0{,}7 & 0{,}7 & 0{,}7 & 0{,}7 & 0{,}7 & 0{,}7 & 0{,}7 & 0{,}7 & 0{,}7 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} \right\| \end{array}$$

$$D_4 = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & 0{,}1 & 0{,}2 & 0{,}3 & 0{,}4 & 0{,}5 & 0{,}6 & 0{,}7 & 0{,}8 & 0{,}9 & 1 \end{array} \\ \left\| \begin{array}{ccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}5 & 0{,}51 & 0{,}54 & 0{,}59 & 0{,}66 & 0{,}75 & 0{,}86 & 0{,}99 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}9 & 0{,}91 & 0{,}94 & 0{,}99 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}7 & 0{,}71 & 0{,}74 & 0{,}79 & 0{,}86 & 0{,}95 & 1 & 1 & 1 & 1 & 1 \end{array} \right\| \end{array}$$

$$D_5 = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & 0{,}1 & 0{,}2 & 0{,}3 & 0{,}4 & 0{,}5 & 0{,}6 & 0{,}7 & 0{,}8 & 0{,}9 & 1 \end{array} \\ \left\| \begin{array}{ccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}5 & 0{,}6 & 0{,}7 & 0{,}8 & 0{,}9 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}9 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} \right\| \end{array}$$

$$D_6 = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & 0{,}1 & 0{,}2 & 0{,}3 & 0{,}4 & 0{,}5 & 0{,}6 & 0{,}7 & 0{,}8 & 0{,}9 & 1 \end{array} \\ \left\| \begin{array}{ccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0{,}9 & 0{,}8 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0{,}9 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0{,}9 & 0{,}8 & 0{,}7 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} \right\| \end{array}$$

В результаті перетину відношень $D_1,...D_6$ отримуємо загальне функціональне рішення:

$$D = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & 0{,}1 & 0{,}2 & 0{,}3 & 0{,}4 & 0{,}5 & 0{,}6 & 0{,}7 & 0{,}8 & 0{,}9 & 1 \end{array} \\ \left\| \begin{array}{ccccccccccc} 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 1 & 1 & 1 & 1 & 0{,}9 & 0{,}8 \\ 0{,}4 & 0{,}5 & 0{,}54 & 0{,}59 & 0{,}66 & 0{,}7 & 0{,}7 & 0{,}7 & 0{,}7 & 0{,}7 & 0{,}9 \\ 0.51 & 0.6 & 0.7 & 0.8 & 0.9 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0{,}9 & 0{,}91 & 0{,}94 & 0{,}99 & 1 & 1 & 1 & 1 & 0{,}9 & 0{,}8 & 0{,}7 \\ 0{,}7 & 0{,}71 & 0{,}74 & 0{,}79 & 0{,}86 & 0{,}95 & 1 & 1 & 1 & 1 & 1 \end{array} \right\| \end{array}$$

Для розрахування задовільності кожної із альтернатив застосуємо правило композиційного виводу в нечіткому середовищі:

$E_k = G_k \circ D$, де E_k — степінь задовільності альтернативи k; Gk - відображення альтернативи k у вигляді нечіткої підмножини на U, D – загальне функціональне рішення. Тоді:

$$\mu_{E_k}(i) = \max_{u \epsilon U} \left(\min\left(\mu_{G_k}(u), \mu_D(u)\right)\right) \quad (1.1)$$

Крім цього, у випадку $\mu_G(u) = 0$; $u \neq u_k$, $\mu_G(u) = 1$; $u = u_k$. Звідси $\mu_E(i) = \mu_K(u_k, i)$. Іншими словами, E_k є k-тий рядок в матриці D.

3.2. Алгоритмічна реалізація методу визначення вагових коефіцієнтів

Для визначення вагових коефіцієнтів використаємо метод точкових оцінок. Застосуємо вище описану процедуру для порівняння нечітких підмножин в одиничному інтервалі для отримання найкращого рішення на основі точкових оцінок.

Для першої альтернативи:

E_1 = {0,5/0; 0,6/0,1; 0,7/0,2; 0,8/0,3; 0,9/0,4; 1/0,5; 1/0,6; 1/0,7; 1/0,8; 0,9/0,9; 0,8/1}.

Вирахуємо множини E_{ja} і потужність такої множини M(Ea) за формулою:

$$M(E_{ja}) \sum_{i=1}^{n} \frac{x_i}{n}: (2.1)$$

для $0 < a < 0{,}5; da = 0{,}5$

$E_{1a} = \{0; 0{,}1; 0{,}2; 0{,}3; 0{,}4; 0{,}5; 0{,}6; 0{,}7; 0{,}8; 0{,}9; 1\}\ M(E_{1a}) = 0{,}5$

для $0{,}5 < a < 0{,}6; da = 0{,}1$

$E_{1a} = \{0{,}1; 0{,}2; 0{,}3; 0{,}4; 0{,}5; 0{,}6; 0{,}7; 0{,}8; 0{,}9; 1\}\ M(E_{1a}) = 0{,}55$

для $0{,}6 < a < 0{,}7; da = 0{,}1$

$E_{1a} = \{0{,}2; 0{,}3; 0{,}4; 0{,}5; 0{,}6; 0{,}7; 0{,}8; 0{,}9; 1\}\ M(E_{1a}) = 0{,}6$

для $0{,}7 < a < 0{,}8; da = 0{,}1$

$E_{1a} = \{0{,}3; 0{,}4; 0{,}5; 0{,}6; 0{,}7; 0{,}8; 0{,}9; 1\}\ M(E_{1a}) = 0{,}65$

для $0{,}8 < a < 0{,}9; da = 0{,}1$

$E_{1a} = \{0{,}4; 0{,}5; 0{,}6; 0{,}7; 0{,}8; 0{,}9\}\ M(E_{1a}) = 0{,}65$

для $0{,}9 < a < 1; da = 0{,}1$

$E_{1a} = \{0{,}5; 0{,}6; 0{,}7; 0{,}8\}\ M(E_{1a}) = 0{,}65$

Розрахуємо точкову оцінку для E_1:

$$F(E_1) = \frac{1}{\alpha_{max}} \int_0^{\alpha_{max}} M(E_{1a})da = \frac{1}{1} \int_0^1 M(E_{1a})da = 0{,}5 * 0{,}5 +$$

$$+0{,}55 * 0{,}1 + 0{,}6 * 0{,}1 + 0{,}65 * 0{,}1 + 0{,}65 * 0{,}1 + 0{,}65 * 0{,}1 = 0{,}560$$

Аналогічним чином розрахуємо точкові оцінки для інших альтернатив: для другої альтернативи $F(E_2) = 0{,}656$; для третьої - $F(E_3) = 0{,}575$; для четвертої - $F(E_4) = 0{,}483$; для п'ятої - $F(E_5) = 0{,}562$. В якості найкращої обираємо альтернативу, яка має найбільшу точкову оцінку.

В даному випадку це альтернатива u_2, відповідно вона і буд найкращою. Друге місце займе альтернатива u_3; третье - u_5 ; четверте - u_1, а найгіршої з усіх альтернатив є u_4.

Описаний вище алгоритм оцінки та відбору кандидата, який буде використовуватись в програмному забезпеченні зобразимо у вигляді блок-схеми:

Рис.2.1 Схема відбору кандидатів

3.3. Характеристика програмного забезпечення системи оцінки кандидатів на заміщення вакантної посади

Кожна розробка програмного продукту починається з архітектури а саме: архітектура бази даних та архітектура програмного засобу. Архітектура бази даних включає в себе вибір бази даних та саму структуру даних. Для реалізації програмного забезпечення системи оцінки кандидатів було обрано систему управління базами даних, яка входить до складу пакету офісних програм Microsoft Office – Microsoft Access. Основними компонентами даної програми є конструктори таблиць, екранних форм, запитів SQL (мова структурованих запитів) та конструктор звітів, що виводиться на друк.

Програма Access сумісна з іншими джерелами даних, що дозволяє будувати повноцінні клієнт-серверні додатки. Отже, в середовищі програми Access були створені наступні таблиці, які виконують роль бази даних для даного програмного продукту:

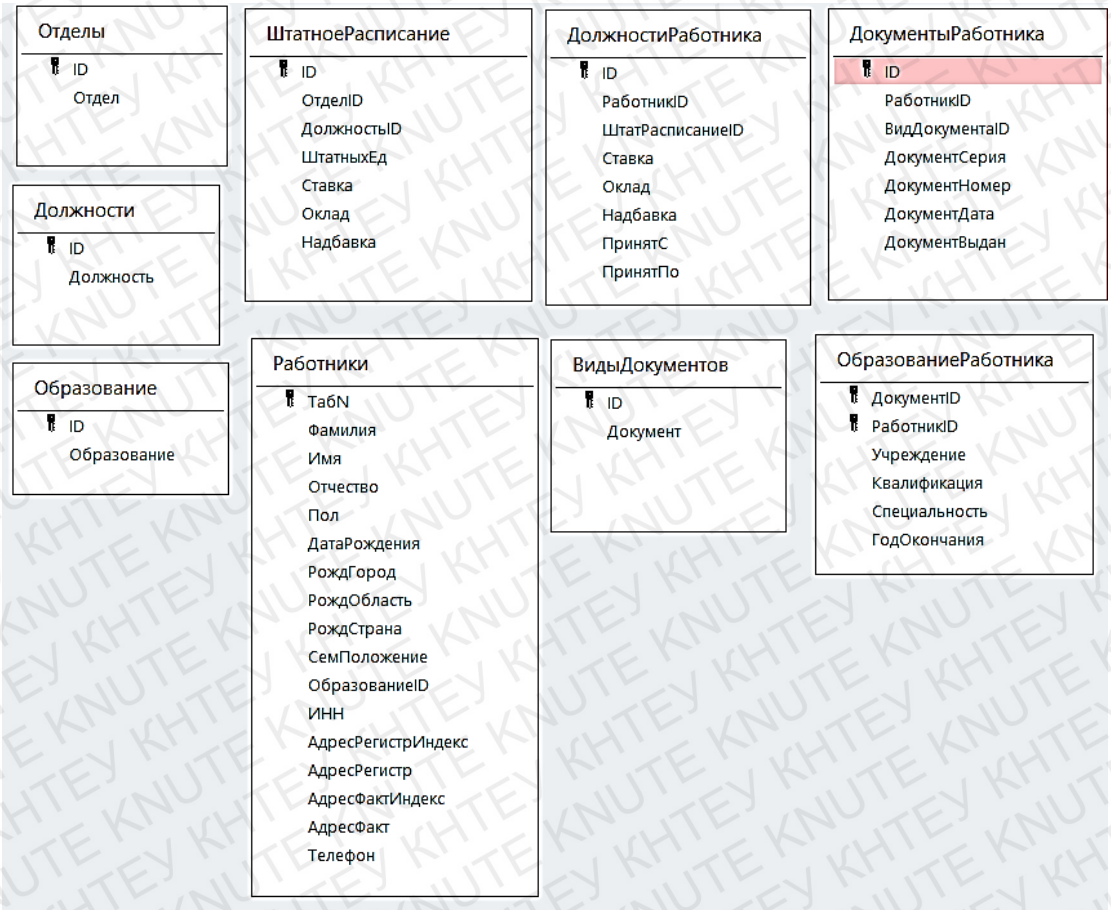

Рис.3.1 Таблиці бази даних

Таблиця “Відділи” містить в собі інформацію про відділи, які існують на підприємств. В таблиці “Посада Працівника” записані дані про посаду кожного працівника підприємства. В таблиці “Працівники” розміщені особисті дані кожного працівника. В таблиці “Освіта Працівника” знаходиться інформація про рівень освіти та кваліфікацію.

Аналогічно архітектурі бази даних архітектура програмного засобу складається х двох частин: вибір середовища та мови програмування; вибір типу програмування. Для розробки даного програмного забезпечення використовувалось мова програмування Delphi, яка є імперативною та структурованою мовою програмування і часто використовується саме для розробки прикладного програмного забезпечення. В основі даної мови лежить об’єктно-орієнтований тип програмування. Розробка програми здійснювалась в однойменному середовищі Delphi.

Програмне забезпечення розраховане на роботу зі звичайним користувачем. Тобто це може бути працівник відділу кадрів або керівник, який зможе легко самостійно розрахувати відповідність кандидата або кандидатів на посаду бухгалтера. Після проведених розрахунків користувач може додати обраного кандидата до списку працівників, заповнивши картку з особистими даними, які будуть внесені до загальної бази даних.

Прообразом майбутнього вікна програми є форма. Для кожного вікна програміст створює форму, наповнюючи її певними компонентами з палітри. Вміст кожної форми описується в окремому модулі.

Модуль - це самостійна частина програми. Модуль створюється кожного разу, як створюється нова форма. При компіляції програми Delphi було створено файли з різним розширеннями для кожного модуля. Тому в програмі є файли з розширеннями pas (файл містить копію тексту з вікна кода програми), dfm (у файлі зберігаються описання змісту форми), dcu (файл містить результат перетворення в машинний код текстів з обох файлів), dpr

(файл головного модуля проекту), dof (файл, що описує сам проект), res (файл ресурсів).

Щодо технічних вимог до користувачі програми, то на клієнтській машині повинна бути встановлена операційна система Windows. Інших обмежень не має.

На початку роботи користувач повинен пройти процес авторизації, який полягає у введені паролю та логіну(рис.3.2).

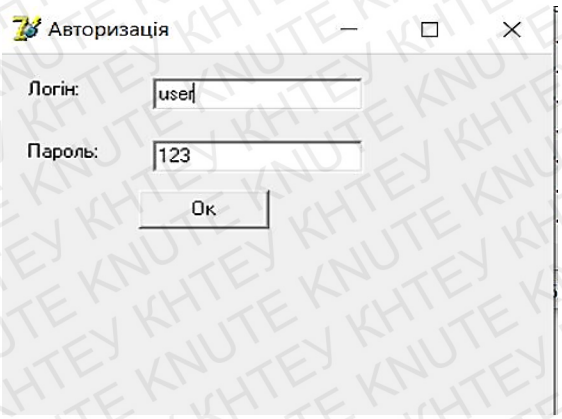

Рис.3.2 Вікно авторизації

У разі не правильно введених даних на екран користувача буде виведено повідомлення про помилку.

Для розрахунку оцінки кандидатів на вакантну посаду в головному вікні необхідно обрати вкладку “Оцінка кандидатів” після чого з’явиться наступне віко(рис.3.3)

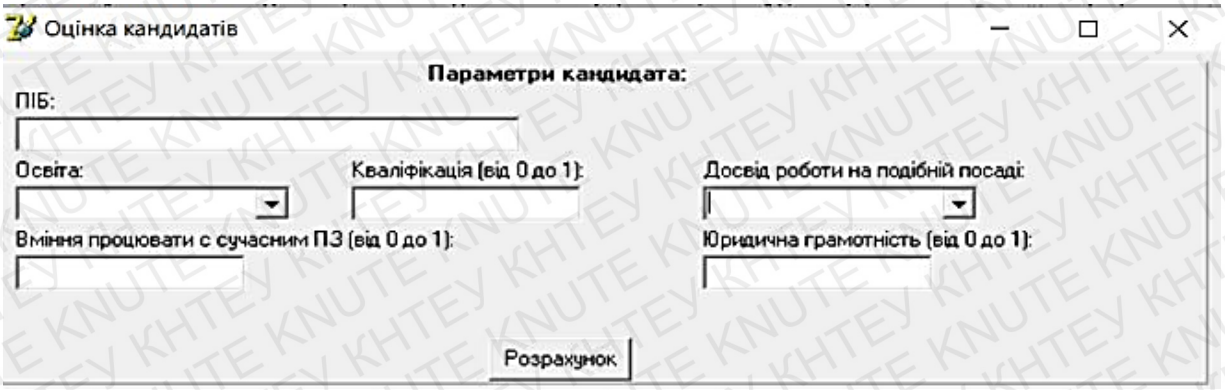

Рис.3.3 Вікно вводу параметрів кандидата на посаду

Наступним кроком для оцінки кандидата є введення його особистих даних у відповідні рядки вікна. Далі буде проведено розрахунок персональної оцінки кандидата. Для порівняння кандидатів та вибору найбільш

відповідного необхідно здійснити аналіз шляхом натиснення відповідної клавіші у даному вікні.(рис.3.4)

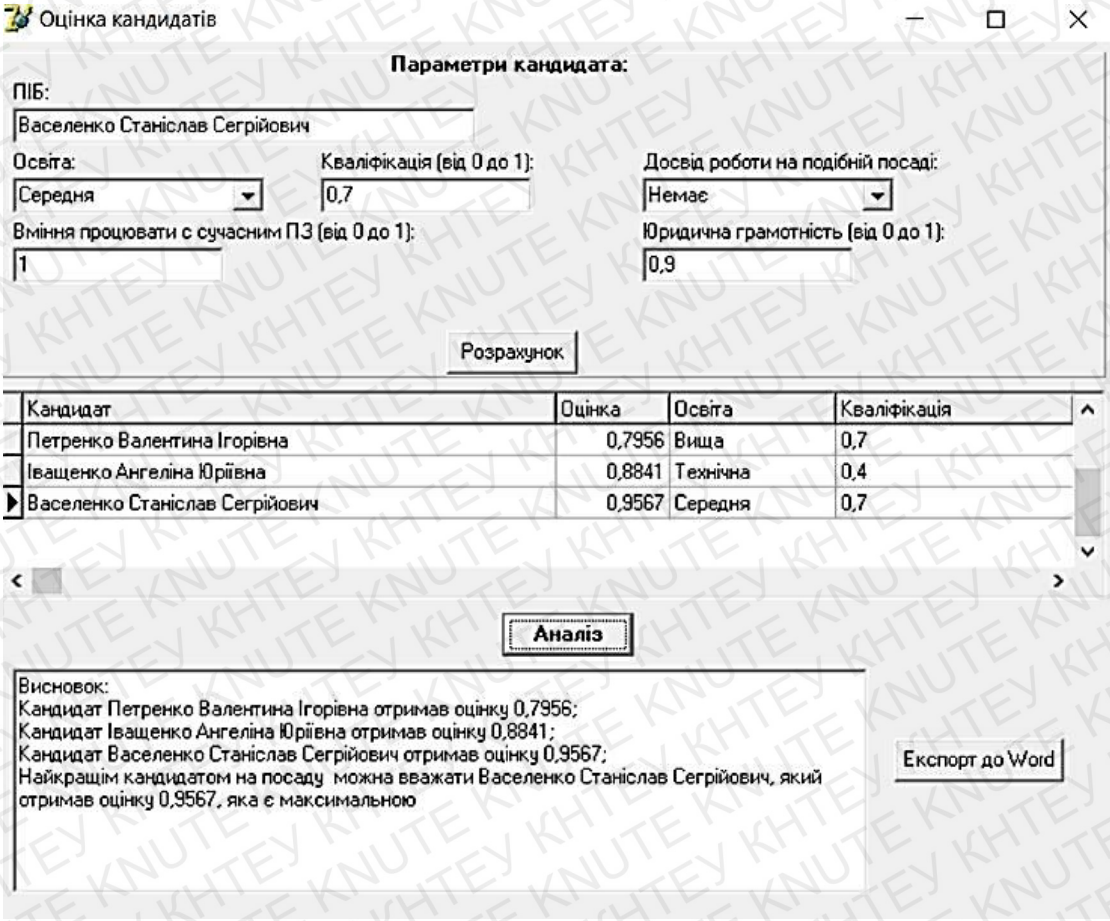

Рис.3.4. Вікно вводу параметрів кандидата на посаду

Отриманий результат користувач має змогу експортувати до Word для надання звіту. Після підписання договору з обраним кандидатом його дані будуть внесені до особистої картки співробітника та збережені у базі даних(рис.3.5).

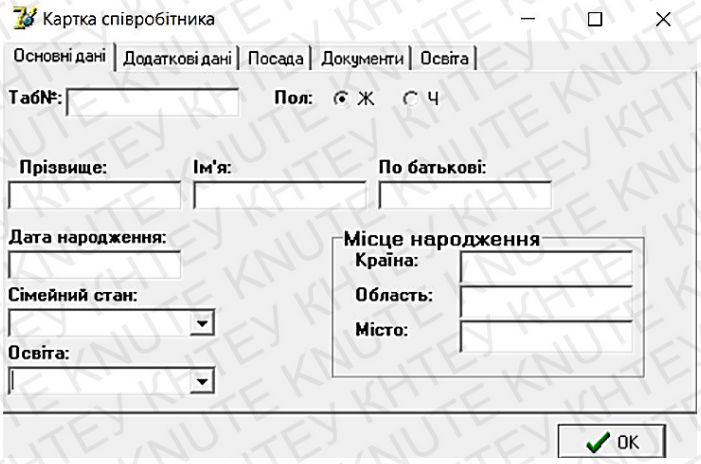

Рис.3.5 Картка співробітника

Таким чином, розроблена і реалізована за допомогою сучасних програмних засобів автоматизована система відбору та оцінки кандидата проста у використанні, має інтуїтивно зрозумілий інтерфейс, забезпечує інтерактивний діалог користувача системи в процесі її використання, не потребує спеціалізованої підготовки користувачів та значних витрат на її впровадження в систему.

ВИСНОВКИ

У випускній кваліфікаційній роботі представлені результати теоретичних та прикладних досліджень , що полягають у розробці інформаційної технології оцінки та відбору кандидатів на вакантну посаду. Результати прикладних досліджень стали основою для створення автоматизованої системи оцінювання критеріїв претендентів на вакантну посаду. В результаті проведеного дослідження було зроблено такі висновки:

- від забезпечення підприємства кваліфікованими кадрами, раціонального їх використання залежать організаційно-технічний рівень виробництва, імідж підприємства, його фінансова стабільність, усі показники діяльності підприємства і важливим крок у процесі забезпечення чи фірми організаційної системи відповідними її профілю людськими ресурсами є відбір кандидатів, що являє собою вибір із усієї підібраної групи індивідуальних працівників для наступного найму.
- при оцінці та відборі кандидатів на вакантні посади доцільно використовувати сучасні інформаційні технології, що дозволить приймати адекватні та найбільш вигідні для підприємства рішення.
- досліджено метод визначення вагових коефіцієнтів та методи нелогічного виводу , що стали основою інформаційної технології відбору та оцінки кандидатів на вакантну посаду.
- здійснено програмну реалізацію методів вагових коефіцієнтів та нелогічного виводу у вигляді автоматизованої системи оцінки та відбору кандидатів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алтунин А. Е., Семухин М. В. Модели и алгоритмы принятия решений в нечетких условиях: Монография. – Тюмень: Издательство Тюменского государственного университета, 2000. 352 с.

2. Андрейчиков А. В., Андрейчикова О. В. Анализ, синтез, планирование решений в экомике , Моска "Финансы и Статистика"-2000, с. 100, с.368.

3. Беллман Р., Заде Л. (1976). Принятие решений в расплывчатых условиях - в книге Вопросы анализа и принятия решений (ru). Москва: Мир. с. 172–215.

4. Березін Ф. Б. Методика багатостороннього дослідження особистості. / Ф. Б. Березін, М. П. Мірошников, Е. Д. Соколова .-- М .: Фолиум, 2004. с. 79.

5. Блюмин С. Л., Шуйкова И. А. Модели и методы принятия решений в условиях неопределенности. – Липецк: ЛЭГИ, 2001.138 с.

6. Блюмин С. Л., Шуйкова И. А., Сараев П. В, Черпаков И. В. Нечеткая логика: алгебраические основы и приложения: Монография – Липецк: ЛЭГИ, 2002. 113 с.

7. Бойко Т. Г. Огляд методів визначення вагових коефіцієнтів показників властивостей продукції, ISSN 1993-9981 Методи та прилади контролю якості, № 24, 2010, с.87.

8. Бочарников В. П. Fuzzy-технология: Математические основы. Практика моделирования в экономике. – СПб: "Наука", РАН, 2001. 328 с.

9. Варжапетян А. Г. Имитационное моделирование на GPSS/H: монография / А. Г. Варжапетян. – М.: Вузовская книга, 2004. – 340 с.

10. Василенко О. В. Проблема підбору управлінських кадрів та використання інноваційних технологій управління колективом у ресторанному господарстві при підготовці фахівців ресторанної справи [Електронний ресурс]: ЦДПУ, Київ, 2014. – Режим доступу: https://www.cuspu.edu.ua/ю. - Загл. з екрану (10.05.2020).

11. Войтко С. В., Мельниченко А. А. Управління трудовими ресурсами/ Войтко С. В - ТОВ НВП ‘Інтерсервіс’ Київ, 2016 – 35 с.

12. Гайнуллова Т. Використання поліграфа (детектора брехні) при роботі з персоналом / Т. Гайнуллова // Управління персоналом, 2001.. -№3. с.6.

13. Гафт М. Г. Принятие решений при многих критериях. – М.: Знание, 1979. 64 с.

14. Дзьюба Наталія Методи відбору кандидатів [Електронний ресурс]. - Режим доступу: https://www.trn.ua/articles/11242/ - Загл. з екрану (10.05.2020).

15. Дмитренко та ін. – К. : МАУП, 2004. – 368 с. 3. Данюк В. М. Менеджмент персоналу : навч. посіб. / В.М. Данюк, В. М. Петюх, С. О. Цимбалюк та ін. ; за заг. ред. В. М. Данюка, В. М. Петюха. – К. : КНЕУ, 2004. 398 с.

16. Дугіна О. Метод центру оцінки (Assessment-Center). Місце оцінки персоналу в кадровій роботі / О. Дугіна // Кадровий вісник, 2004. -№ 2 (14) . - с. 24.

17. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений: Пер. с англ. — М.: Мир, 1976. — 165 с.

18. Зайченко Ю. П. Нечёткие модели и методы в интеллектуальных системах: учеб. пособие / Ю. П. Зайченко. – К.: Изд. дом “Слово”, 2008. – 354с.

19.Ивин А.А. Модальные теории Я. Лукасевича, 2001, с.19.

20. Kendall M.G. Rank Correlation Methods / M. G. Kendall. – H.Y.: Hafner Publ. Go.,1995. – 196 p. (DOI: 10.1111/j.2044-8317. 1956. tb00172.x).

21. Ковальчук В. М. Особливості розв’язання багатокритеріальних задач прийняття рішень у нечіткому середовищі. - Наукові записки. Серія “Економіка”. Випуск 14, 2010. 449 - 455 с.

22. Ковтун Н. В. Теорія статистики: курс лекцій, практикум / Н. В. Ковтун. – К.: ІМЕКС. – 2007. – 276 с.

23. Коршевнюк Л. О. Система нечіткого логічного виводу із зваженою істинністю / Л. О. Коршевнюк, М. Ю. Мінін // Единое информационное

пространство ’2004: Сб. докл. II-й Междунар. науч.-практ. конф. – Днепропетровск: ИПК ИнКомЦентра УГХТУ, 2004. – с. 114–117.

24. Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь, 1982. 432 с.

25. Кравець П. Системи прийняття рішень з нечіткою логікою / П. Кравець, Р. Киркало // Національний університет “Львівська політехніка”. – 2009. с. 115–123.

26. Круглов В. В. Нечёткая логика и искусственные нейронные сети. Учеб. пособие / В. В. Круглов, М. И.

27.Кузьмін О. Є., Громовик Б. П. Менеджмент у фармації. - НОВА КНИГА, 2005. 335 с.

28. Ладанюк А. П. Основи системного аналізу, Вінниця - 2004, с.76-78.

29. Ліхоузова Т. А. Теорія імовірностей та математична статистика. Курс лекцій.: КПІ ім. Ігоря Сікорського, 2018. – 182с.

30. Лук’янченко Н. Д. Керування трудом на промислових підприємствах. / Н. Д. Лук’янченко – Донецьк: Донбас, 2000. – 196 с.

31. Ляшенко О. І. Компетентність як об’єкт оцінювання навчальних досягнень учнів / О. І. Ляшенко // Збірник наукових праць Кам’янецьПодільського національного університету ім. Івана Огієнка. Серія: Педагогічна. – 2014. – Вип. 20. – С. 36–39. 12.

32. Медиковский М. О., Шуневич О. Б. Дослідження ефективності методів визначення вагових коефіцієнтів. Вісник Хмельницького національного університету №5, 2011. — С. 120-123.

33. Митяков Е. С. К вопросу о выборе весов при нахождении интегральных показателей экономической динамики / Е. С. Митяков, Д. А. Корнилов // Труды НГТУ. – Н. Новгород: Изд-во Нижегород. гос. техн. ун-т., 2011. №3(90). С. 289–300.

34. Михаляк Т. Эффективность использования различных методов при отборе персонала / Т. Михаляк // Менеджер по персоналу. – 2010. – № 11. – с. 48–64.

35. Новосад В. П., Селіверстов Р. Г. – К. Методологія експертного оцінювання: конспект лекцій – К.: Вид-во НАДУ, 2008. – 48 с.

36. Одегов Ю. Г., Ніконова Т. В. Управління персоналом. Практикум: конкретні ситуації. – М.: Видавництво «Іспит», 2003. – 192 с.

37. Семенова Н. В., Колєчкіна Л. М., Нагірна А. М. Багатокритеріальні задачі лексикографічної оптимізації на нечіткій множині альтернатив: УДК 519.8, 2010. 42 с.

38. Сеньо П. С. Теорія ймовірностей та математична статистика: підруч. / П. С. Сеньо. – [2-ге вид., перероб. і доп.]. – К.: Знання, 2007. – 556 с.

39. Сявавко М. С., Рибицька О. М. Математичне моделювання за умов невизначеності. – Львів: Українські технології, 2000. 320 с.

40. Федулова І.Л., Сокирник І.В., Стадник В.В., Йохна М.А., Новикова О.С., Рясних Є.Г. Менеджмент організацій : підручник – К. : Либідь, 2004. 448 с.

41. Черняк О. І. Теорія ймовірностей та математична статистика. Збірник задач: навч. посіб. / О. І. Черняк, О. М. Обушна, А. В. Ставицький. – [2-ге вид.]. – К.: Знання, КЦЦЦ, 2002. – 199 с.

42. Шарапов О. Д., Дербенцев В. Д., Семьонов Д. Є. Економічна кібернетика: Навч. посібник. – К.: КНЕУ, 2004. 231 с.

43. Шахмалов Ф. И. Американский менеджмент. Теория и практика / Ф. И. Шахмалов. – М. : Наука, 2002. – 272 с.

44. Под ред. Р.Р. Ягера. М. Нечеткие множества и теория возможностей. Последние достижения.: Радио и связь, 1986. 408 с.

ДОДАТКИ

Програмний код реалізації оцінки та відбору кандидатів.

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Unit2, Unit3;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Button1: TButton;
    Label1: TLabel;
    Label2: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var Form2:Tform2;
Form3:TForm3;
begin
if (Edit1.Text='user') and (Edit2.Text='123') then
```

```
begin
Form2:=TForm2.Create(Self);
Form2.ShowModal;
Close;
end
else
begin
Form3:=TForm3.Create(Self);
Form3.ShowModal;
end;
end;
end.
nit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ExtCtrls, DB, ADODB, ComCtrls, Buttons, DBCtrls, Grids,unit4,
  DBGrids, unit5, unit6, unit7, StdCtrls, Unit8, Unit12, Unit10, Unit11, Unit13,
Unit15;
type
  TForm2 = class(TForm)
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
```

```
N11: TMenuItem;
N12: TMenuItem;
N4: TMenuItem;
TreeView1: TTreeView;
ADOTable1: TADOTable;
Panel2: TPanel;
SpeedButton1: TSpeedButton;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
SpeedButton4: TSpeedButton;
SpeedButton5: TSpeedButton;
SpeedButton6: TSpeedButton;
SpeedButton7: TSpeedButton;
DataSource1: TDataSource;
DBGrid1: TDBGrid;
ADOQuery1: TADOQuery;
DBNavigator2: TDBNavigator;
SpeedButton8: TSpeedButton;
N8: TMenuItem;
ADOTable2: TADOTable;
procedure N5Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure TreeView1Change(Sender: TObject; Node: TTreeNode);
procedure N6Click(Sender: TObject);
procedure N7Click(Sender: TObject);
procedure N9Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure N11Click(Sender: TObject);
procedure N12Click(Sender: TObject);
```

```
    procedure N4Click(Sender: TObject);
    procedure N8Click(Sender: TObject);
    procedure N13Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form2: TForm2;
  i:TTreeNode;
  t:string;
  q:integer;
implementation
uses unit1;
{$R *.dfm}
procedure TForm2.N5Click(Sender: TObject);
begin
Form1.Close;
Form2.Close;
end;
procedure TForm2.TreeView1Change(Sender: TObject; Node: TTreeNode);
begin
q:=node.AbsoluteIndex;
i:= Node;
t:=TreeView1.Items.Item[q].Text;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select Работники.ТабN,Работники.Фамилия,
Работники.Имя, Работники.Отчество,
Должности.Должность,Образование.Образование, Работники.Телефон');
```

```
ADOQuery1.SQL.Add('from Отделы, Работники,
Должности,Образование,ДолжностиРаботника, ШтатноеРасписание');
ADOQuery1.SQL.Add('where Работники.ОбразованиеID =Образование.ID
and');
ADOQuery1.SQL.Add('Работники.ТабN = ДолжностиРаботника.РаботникID
and');
ADOQuery1.SQL.Add(' ДолжностиРаботника.ШтатРасписаниеID =
ШтатноеРасписание.ID and');
ADOQuery1.SQL.Add(' ШтатноеРасписание.ДолжностьID = Должности.ID
and' );
ADOQuery1.SQL.Add(' ШтатноеРасписание.ОтделID = Отделы.ID and');
ADOQuery1.SQL.Add(' Отделы.Отдел = '"+t+'"');
ADOQuery1.Active:=True;
end;
procedure TForm2.FormCreate(Sender: TObject);
begin
While not ADOTable1.Eof do
begin
TreeView1.Items.Add(i,ADOTable1.FieldByName('Отдел').AsString);
ADOTable1.Next;
end;
end;
procedure TForm2.N6Click(Sender: TObject);
var Form4:Tform4;
begin
Form4:=TForm4.Create(Self);
Form4.ShowModal;
end;
procedure TForm2.N7Click(Sender: TObject);
var Form5:Tform5;
```

```
begin
Form5:=TForm5.Create(Self);
Form5.ShowModal;
end;
procedure TForm2.N9Click(Sender: TObject);
var Form7:Tform7;
begin
Form7:=TForm7.Create(Self);
Form7.ShowModal;
end;
procedure TForm2.N10Click(Sender: TObject);
var Form6:Tform6;
begin
Form6:=TForm6.Create(Self);
Form6.ShowModal;
end;
procedure TForm2.SpeedButton6Click(Sender: TObject);
var Form8:Tform8;
begin
Form8:=TForm8.Create(Self);
Form8.ShowModal;
end;
procedure TForm2.N11Click(Sender: TObject);
var Form11:Tform11;
begin
Form11:=TForm11.Create(Self);
Form11.ShowModal;
end;
procedure TForm2.N12Click(Sender: TObject);
var Form10:Tform10;
```

```
begin
Form10:=TForm10.Create(Self);
Form10.ShowModal;
end;
procedure TForm2.N4Click(Sender: TObject);
var Form12:Tform12;
begin
Form12:=TForm12.Create(Self);
Form12.ShowModal;
end;
procedure TForm2.N8Click(Sender: TObject);
var Form13:Tform13;
begin
Form13:=TForm13.Create(Self);
Form13.ShowModal;
end;
procedure TForm2.N13Click(Sender: TObject);
var Form15:Tform15;
begin
Form15:=TForm15.Create(Self);
Form15.ShowModal;
end;
end.
unit Unit3;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm3 = class(TForm)
```

```
    Label1: TLabel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form3: TForm3;
implementation
{$R *.dfm}
end.
unit Unit4;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, DBCtrls, Grids, DBGrids, DB, ADODB;
type
  TForm4 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form4: TForm4;
implementation
```

```
{$R *.dfm}
end.
unit Unit5;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, Grids, DBGrids, ExtCtrls, DBCtrls;
type
  TForm5 = class(TForm)
    DataSource1: TDataSource;
    ADOTable1: TADOTable;
    DBGrid2: TDBGrid;
    DBNavigator2: TDBNavigator;
    ADOTable2: TADOTable;
    DataSource2: TDataSource;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form5: TForm5;
implementation
{$R *.dfm}
end.
unit Unit6;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```

```
  Dialogs, DB, ADODB, ExtCtrls, DBCtrls, Grids, DBGrids, StdCtrls, ComCtrls,
ComObj;
type
  TForm6 = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    Button1: TButton;
    Button2: TButton;
    ADOTable2: TADOTable;
    TreeView1: TTreeView;
    ADOQuery1: TADOQuery;
    procedure FormCreate(Sender: TObject);
    procedure TreeView1Change(Sender: TObject; Node: TTreeNode);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form6: TForm6;
   i:TTreeNode;
  t:string;
  q:integer;
  ExcelApplication:Variant;
implementation
uses unit2;
```

```
{$R *.dfm}
procedure TForm6.FormCreate(Sender: TObject);
begin
While not ADOTable1.Eof do
begin
TreeView1.Items.Add(i,ADOTable1.FieldByName('Отдел').AsString);
ADOTable1.Next;
end;
end;
procedure TForm6.TreeView1Change(Sender: TObject; Node: TTreeNode);
begin
q:=node.AbsoluteIndex;
i:= Node;
t:=TreeView1.Items.Item[q].Text;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select    ШтатноеРасписание.ID,Должности.Должность,
ШтатноеРасписание.ШтатныхЕд,
ШтатноеРасписание.Ставка,ШтатноеРасписание.Оклад,
ШтатноеРасписание.Надбавка, Отделы.Отдел');
ADOQuery1.SQL.Add('from ШтатноеРасписание, Отделы, Должности ');
ADOQuery1.SQL.Add('      Where      ШтатноеРасписание.ДолжностьID      =
Должности.ID and' );
ADOQuery1.SQL.Add('  ШтатноеРасписание.ОтделID = Отделы.ID and');
ADOQuery1.SQL.Add(' Отделы.Отдел = '"+t+'"');
ADOQuery1.Active:=True;
end;
procedure TForm6.Button1Click(Sender: TObject);
begin
close;
end;
```

```
procedure TForm6.Button2Click(Sender: TObject);
var Path, FileName : String;
Excel: Variant;
i:integer;
begin
Path := ExtractFilePath(ParamStr(0));
Excel:=CreateOleObject('Excel.Application');
Excel.Application.WorkBooks.Add( ""+Path + 'Книга1.xlsx');
Excel.Visible:=True;
Excel.DisplayAlerts:=False;
Excel.WorkSheets.Item['Лист1'].Cells[2, 1]:='ID';
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
Excel.WorkSheets.Item['Лист1'].Cells[2, 2]:='Должность';
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
Excel.WorkSheets.Item['Лист1'].Cells[2, 3]:='Штатных Ед';
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
Excel.WorkSheets.Item['Лист1'].Cells[2, 4]:='Оклад';
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
Excel.WorkSheets.Item['Лист1'].Cells[2, 5]:='Надбавка';
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
Excel.WorkSheets.Item['Лист1'].Cells[2, 6]:='Отдел';
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
ADOQuery1.First;
i:=3;
 while not ADOQuery1.Eof do
      begin
Excel.WorkSheets.Item['Лист1'].Cells[i,
1]:=ADOQuery1.FieldByName('ID').AsInteger;
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
```

```
Excel.WorkSheets.Item['Лист1'].Cells[i,
2]:=ADOQuery1.FieldByName('Должность').AsString;
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
Excel.WorkSheets.Item['Лист1'].Cells[i,
3]:=ADOQuery1.FieldByName('ШтатныхЕд').AsString;
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
Excel.WorkSheets.Item['Лист1'].Cells[i,
4]:=ADOQuery1.FieldByName('Оклад').AsString;
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
Excel.WorkSheets.Item['Лист1'].Cells[i,
5]:=ADOQuery1.FieldByName('Надбавка').AsString;
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
Excel.WorkSheets.Item['Лист1'].Cells[i,
6]:=ADOQuery1.FieldByName('Отдел').AsString;
Excel.ActiveWorkbook.SaveAs('Книга 1.xlsx');
i:=i+1;
   ADOQuery1.Next;
   end;
end
end.
nit Unit7;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, ExtCtrls, DBCtrls, Grids, DBGrids;
type
  TForm7 = class(TForm)
   DBGrid1: TDBGrid;
   DBNavigator1: TDBNavigator;
   DataSource1: TDataSource;
```

```
    ADOTable1: TADOTable;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form7: TForm7;
implementation
{$R *.dfm}
end.
nit Unit8;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ADODB, StdCtrls, DB, ExtCtrls, DBCtrls, Grids, DBGrids
type
  TForm8 = class(TForm)
    ComboBox1: TComboBox;
    Label1: TLabel;
    ADOQuery1: TADOQuery;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    DataSource1: TDataSource;
    Button1: TButton;
    ADOTable1: TADOTable;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
```

```
  public
    { Public declarations }
  end;
var
  Form8: TForm8;
implementation
{$R *.dfm}
procedure TForm8.FormCreate(Sender: TObject);
begin
ADOTable1.First;
While not ADOTable1.Eof do
begin
Combobox1.Items.Add(ADOTable1.FieldByName('Должность').AsString);
ADOTable1.Next;
end;
end;
procedure TForm8.Button1Click(Sender: TObject);
var t:string;
begin
t:=Combobox1.Text;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select Работники.ТабN,Работники.Фамилия,
Работники.Имя, Работники.Отчество,
Должности.Должность,Образование.Образование, Работники.Телефон');
ADOQuery1.SQL.Add('from Отделы, Работники,
Должности,Образование,ДолжностиРаботника, ШтатноеРасписание');
ADOQuery1.SQL.Add('where Работники.ОбразованиеID =Образование.ID
and');
ADOQuery1.SQL.Add('Работники.ТабN = ДолжностиРаботника.РаботникID
and');
```

```
ADOQuery1.SQL.Add('ДолжностиРаботника.ШтатРасписаниеID = ШтатноеРасписание.ID and');
ADOQuery1.SQL.Add(' ШтатноеРасписание.ДолжностьID = Должности.ID and' );
ADOQuery1.SQL.Add(' ШтатноеРасписание.ОтделID = Отделы.ID and');
ADOQuery1.SQL.Add(' Должности.Должность = '''+t+'''');
ADOQuery1.Active:=True;
end;
end.
unit Unit9;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, DBCtrls, DB, ADODB, Grids, DBGrids, ComCtrls,ComObj;
type
  TForm9 = class(TForm)
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    DBNavigator1: TDBNavigator;
    ADOTable1: TADOTable;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form9: TForm9;
   ExcelApplication:Variant;
implementation
```

```
{$R *.dfm
unit Unit10;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls, ComCtrls, Grids, Buttons, DBGrids,
  ExtCtrls, DBCtrls;
type
  TForm10 = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    TabSheet3: TTabSheet;
    TabSheet4: TTabSheet;
    TabSheet5: TTabSheet;
    Edit1: TEdit;
    Label1: TLabel;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Label6: TLabel;
    ComboBox1: TComboBox;
    Label7: TLabel;
    ComboBox2: TComboBox;
```

```
GroupBox1: TGroupBox;
Label8: TLabel;
Label9: TLabel;
Label11: TLabel;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
ADOTable1: TADOTable;
Label10: TLabel;
Edit8: TEdit;
GroupBox2: TGroupBox;
Label12: TLabel;
Label14: TLabel;
Edit9: TEdit;
Edit10: TEdit;
GroupBox3: TGroupBox;
Label13: TLabel;
Label15: TLabel;
Edit11: TEdit;
Edit12: TEdit;
Label16: TLabel;
Edit13: TEdit;
Label17: TLabel;
ComboBox3: TComboBox;
ADOTable2: TADOTable;
Label18: TLabel;
Edit14: TEdit;
Label19: TLabel;
Edit15: TEdit;
Label20: TLabel;
```

Edit16: TEdit;
Label21: TLabel;
Edit17: TEdit;
Label22: TLabel;
Edit18: TEdit;
Label23: TLabel;
Edit19: TEdit;
Label24: TLabel;
Label25: TLabel;
Label26: TLabel;
Edit20: TEdit;
Label27: TLabel;
Label28: TLabel;
Edit21: TEdit;
Label29: TLabel;
Edit22: TEdit;
Label30: TLabel;
Edit23: TEdit;
ADOTable3: TADOTable;
Label31: TLabel;
ComboBox4: TComboBox;
Label32: TLabel;
Edit24: TEdit;
Label33: TLabel;
Edit25: TEdit;
Label34: TLabel;
Edit26: TEdit;
Label35: TLabel;
Edit27: TEdit;
Button1: TButton;

```
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    ADOTable4: TADOTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    ADOTable5: TADOTable;
    Label36: TLabel;
    Edit28: TEdit;
    ADOTable6: TADOTable;
    Label37: TLabel;
    Edit29: TEdit;
    ADOTable7: TADOTable;
    ADOTable8: TADOTable;
    DBNavigator1: TDBNavigator;
    ADOTable9: TADOTable;
    procedure FormCreate(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form10: TForm10;
implementation
{$R *.dfm}
procedure TForm10.FormCreate(Sender: TObject);
begin
```

```
ADOTable4.First;
While not ADOTable4.Eof do
begin
ADOTable4.Delete;
end;
ADOTable9.First;
While not ADOTable9.Eof do
begin
Combobox1.Items.Add(ADOTable9.FieldByName('СемейноеПоложение').AsStr
ing);
ADOTable9.Next;
end;
ADOTable1.First;
While not ADOTable1.Eof do
begin
Combobox2.Items.Add(ADOTable1.FieldByName('Образование').AsString);
ADOTable1.Next;
end;
ADOTable3.First;
While not ADOTable3.Eof do
begin
Combobox4.Items.Add(ADOTable3.FieldByName('Документ').AsString);
ADOTable3.Next;
end;
ADOTable2.First;
While not ADOTable2.Eof do
begin
Combobox3.Items.Add(inttostr(ADOTable2.FieldByName('ID').AsInteger));
ADOTable2.Next;
end;
```

```
end;
procedure TForm10.BitBtn1Click(Sender: TObject);
begin
ADOTable2.First;
While not ADOTable2.Eof do
begin
if(Combobox3.Text=(inttostr(ADOTable2.FieldByName('ID').AsInteger)))then
begin
Edit16.Text:=floattostr(ADOTable2.FieldByName('Ставка').AsFloat);
Edit17.Text:=inttostr(ADOTable2.FieldByName('Оклад').AsInteger);
Edit18.Text:=inttostr(ADOTable2.FieldByName('Надбавка').AsInteger);
Edit19.Text:=floattostr(ADOTable2.FieldByName('Ставка').AsFloat*ADOTable2
.FieldByName('Оклад').AsInteger+ADOTable2.FieldByName('Надбавка').AsInte
ger);
end;
ADOTable2.Next;
end;
end;
procedure TForm10.Button1Click(Sender: TObject);
begin
ADOTable4.Insert  ;
ADOTable4.FieldByName('Документ').AsString:=ComboBox4.Text ;
ADOTable4.FieldByName('Серия').AsString:=Edit24.Text;
ADOTable4.FieldByName('Номер').AsInteger:= strtoint(Edit25.Text);
ADOTable4.FieldByName('Дата').AsString:=Edit26.Text;
ADOTable4.FieldByName('Кем выдан').AsString:= Edit27.Text;
ADOTable4.Post ;
end;
procedure TForm10.BitBtn2Click(Sender: TObject);
var q,q1,q2:integer;
```

```
a:boolean;
begin
ADOTable9.First;
While not ADOTable9.Eof do
begin
if ADOTable9.FieldByName('СемейноеПоложение').AsString
=ComboBox1.Text then
q2:= ADOTable9.FieldByName('ID').Asinteger;
ADOTable9.Next;
end;
ADOTable1.First;
While not ADOTable1.Eof do
begin
if ADOTable1.FieldByName('Образование').AsString =ComboBox2.Text then
q:= ADOTable1.FieldByName('ID').Asinteger;
ADOTable1.Next;
end;
if  RadioButton1.Checked=True then
a:=False;
if  RadioButton2.Checked=True then
a:=True;
ADOTable5.Insert;
ADOTable5.FieldByName('ТабN').AsInteger:=strtoint(Edit1.Text) ;
ADOTable5.FieldByName('Фамилия').AsString:=Edit2.Text;
ADOTable5.FieldByName('Имя').AsString:= Edit3.Text;
ADOTable5.FieldByName('Отчество').AsString:=Edit4.Text;
ADOTable5.FieldByName('Пол').AsBoolean:= a;
ADOTable5.FieldByName('ДатаРождения').AsString:=Edit28.Text ;
ADOTable5.FieldByName('РождГород').AsString:=Edit7.Text;
ADOTable5.FieldByName('РождОбласть').AsString:= Edit6.Text;
```

```
ADOTable5.FieldByName('РождСтрана').AsString:=Edit5.Text;
ADOTable5.FieldByName('СемПоложение').AsInteger:=q2;
ADOTable5.FieldByName('ОбразованиеID').AsInteger:=q;
ADOTable5.FieldByName('ИНН').AsString:=Edit8.Text;
ADOTable5.FieldByName('АдресРегистрИндекс').AsString:=Edit9.Text;
ADOTable5.FieldByName('АдресРегистр').AsString:=Edit10.Text;
ADOTable5.FieldByName('АдресФактИндекс').AsString:=Edit11.Text;
ADOTable5.FieldByName('АдресФакт').AsString:=Edit12.Text;
ADOTable5.FieldByName('Телефон').AsString:=Edit13.Text;
ADOTable5.Post ;
ADOTable6.Insert;
ADOTable6.FieldByName('РаботникID').AsInteger:=strtoint(Edit1.Text) ;
ADOTable6.FieldByName('ДокументID').AsString:=Edit29.Text;
ADOTable6.FieldByName('Учреждение').AsString:= Edit21.Text;
ADOTable6.FieldByName('Квалификация').AsString:=Edit22.Text;
ADOTable6.FieldByName('Специальность').AsString:= Edit23.Text;
ADOTable6.FieldByName('ГодОкончания').AsString:=Edit20.Text ;
ADOTable6.Post ;
ADOTable7.Insert  ;
ADOTable7.FieldByName('РаботникID').AsInteger:=strtoint(Edit1.Text);
ADOTable7.FieldByName('ШтатРасписаниеID').AsInteger:=strtoint(ComboBox
3.Text);
ADOTable7.FieldByName('Ставка').AsFloat:=strtofloat(Edit16.Text);
ADOTable7.FieldByName('Оклад').AsFloat:= strtoint(Edit17.Text);
ADOTable7.FieldByName('Надбавка').AsFloat:=strtoint(Edit18.Text);
ADOTable7.FieldByName('ПринятС').AsString:=Edit14.Text;
ADOTable7.FieldByName('ПринятПо').AsString:= Edit15.Text;
ADOTable7.Post ;
ADOTable4.First;
While not ADOTable4.Eof do
```

```
begin
ADOTable3.First;
While not ADOTable3.Eof do
begin
if
ADOTable3.FieldByName('Документ').AsString=ADOTable4.FieldByName('Док
умент').AsString then
begin
q1:= ADOTable3.FieldByName('ID').AsInteger;
ADOTable8.Insert  ;
ADOTable8.FieldByName('РаботникID').AsInteger:=strtoint(Edit1.Text);
ADOTable8.FieldByName('ВидДокументаID').AsInteger:=q1;
ADOTable8.FieldByName('ДокументСерия').AsString:=ADOTable4.FieldByNa
me('Серия').AsString;
ADOTable8.FieldByName('ДокументНомер').AsString:=ADOTable4.FieldByNa
me('Номер').AsString;
ADOTable8.FieldByName('ДокументДата').AsString:=ADOTable4.FieldByNam
e('Дата').AsString;;
ADOTable8.FieldByName('ДокументВыдан').AsString:=ADOTable4.FieldByNa
me('Кем выдан').AsString;
ADOTable8.Post ;
end;
ADOTable3.Next;
end;
ADOTable4.Next;
end;
ADOTable8.Insert  ;
ADOTable8.FieldByName('РаботникID').AsInteger:=strtoint(Edit1.Text);
ADOTable8.FieldByName('ВидДокументаID').AsInteger:=strtoint(ComboBox3.
Text);
```

```
ADOTable8.FieldByName('ДокументСерия').AsInteger:=strtoint(Edit16.Text);
ADOTable8.FieldByName('ДокументНомер').AsFloat:= strtoint(Edit17.Text);
ADOTable8.FieldByName('ДокументДата').AsFloat:=strtoint(Edit18.Text);
ADOTable8.FieldByName('ДокументВыдан').AsString:=Edit14.Text;
ADOTable8.Post ;
Close;
end
unit Unit11;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, DBCtrls, DB, ADODB, Grids, DBGrids, StdCtrls, Mask,
Unit16;
type
  TForm11 = class(TForm)
    DBGrid1: TDBGrid;
    ADOTable3: TADOTable;
    DataSource1: TDataSource;
    DBNavigator1: TDBNavigator;
    Label1: TLabel;
    Label2: TLabel;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit5: TDBEdit;
    DBEdit6: TDBEdit;
    DBEdit7: TDBEdit;
    Label3: TLabel;
    Label4: TLabel;
```

```
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Button2: TButton;
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form11: TForm11;
implementation
{$R *.dfm}
procedure TForm11.Button2Click(Sender: TObject);
var Form16:Tform16;
begin
Form16:=TForm16.Create(Self);
Form16.ShowModal;
end;
end.
unit Unit12;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Grids, Spin, DB, ADODB, DBGrids, Mask,
  DBCtrls, Menus, ComObj, Math, jpeg;   // стандартный набор модулей
type
```

```
  TForm12 = class(TForm)
    Memo1: TMemo;
    Button3: TButton;
    Panel1: TPanel;
    Label14: TLabel;
    Label15: TLabel;
    Label17: TLabel;
    Label1: TLabel;
    Label9: TLabel;
    Label18: TLabel;
    ComboBox2: TComboBox;
    ComboBox4: TComboBox;
    Button1: TButton;
    Button2: TButton;
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Label2: TLabel;
    Label5: TLabel;
    ADOTable1: TADOTable;
    Image1: TImage
    procedure N1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
```

```
    { Private declarations }
  public
    { Public declarations }
  end;
Var
  Form12: TForm12;
implementation
uses Unit11, Unit1, Unit2;
{$R *.dfm}
procedure TForm12.N1Click(Sender: TObject);
begin
Form1.Close;
Form2.Close;
Form12.Close;
end;
procedure TForm12.Button1Click(Sender: TObject);
Var   min,res,sum,w:real;
q:array[1..5] of real ;
m:array[1..6] of real ;
e:array[0..10] of real ;
d:array[1..6,0..10] of real ;
i,j,k:integer;
st:string;
begin
Image1.Visible:=False;
DBGrid1.Visible:=true;
Button2.Visible:=true;
if ComboBox4.Text='Вища' then
q[1]:=0.9;
if ComboBox4.Text='Технічна' then
```

```
q[1]:=0.5;
if ComboBox4.Text='Середнє' then
q[1]:=0.25;
if ComboBox4.Text='Немає' then
q[1]:=0.01;
if ComboBox2.Text='від 1 року до 5' then
q[2]:=0.25;
if ComboBox2.Text='від 5 до 10' then
q[2]:=0.5;
if ComboBox2.Text='більше 10' then
q[2]:=0.9;
if ComboBox2.Text='Немає' then
q[2]:=0.01;
q[3]:=strtofloat(Edit1.Text);
q[4]:=strtofloat(Edit2.Text);
q[5]:=strtofloat(Edit3.Text);
min:=q[1];
for i:=2 to 3 do
if q[i]<min then min:=q[i];
m[1]:=min;
if q[5]<min then m[4]:=q[5] else m[4]:=min;
if q[4]<min then m[2]:=q[4] else m[2]:=min;
for i:=4 to 5 do
if q[i]<min then min:=q[i];
m[3]:=min;
if q[3]<(1-q[1]) then min:=q[3] else min:=1-q[1];
if q[2]<min then min:=q[2];
if q[5]<min then min:=q[5];
m[5]:=min;
if (1-q[3])<(1-q[2]) then m[6]:=1-q[3] else min:=1-q[2];
```

```
for i:=0 to 10 do
if (1-m[1]+(i/10))<1 then d[1,i]:=(1-m[1]+(i/10)) else  d[1,i]:=1;
for i:=0 to 10 do
if (1-m[2]+sqrt(i/10))<1 then d[2,i]:=(1-m[2]+sqrt(i/10)) else  d[2,i]:=1;
for i:=0 to 9 do
if (1-m[3])<1 then d[3,i]:=(1-m[3]) else  d[3,i]:=1;
if (1-m[3]+1)<1 then d[3,10]:=(1-m[3]+1) else  d[3,10]:=1;
for i:=0 to 10 do
if (1-m[4]+sqr(i/10))<1 then d[4,i]:=(1-m[4]+sqr(i/10)) else  d[4,i]:=1;
for i:=0 to 10 do
if (1-m[5]+(i/10))<1 then d[5,i]:=(1-m[5]+(i/10)) else  d[5,i]:=1;
for i:=0 to 10 do
if (1-m[6]+(1-(i/10)))<1 then d[6,i]:=(1-m[6]+(1-(i/10))) else  d[6,i]:=1;
for j:=0 to 10 do
begin
min:=d[1,j];
for i:=1 to 6 do
if d[i,j]<min then min:=d[i,j] ;
e[j]:=min;
end;
w:=0;
res:=0;
While w<>1 do
begin
min:=1;
for j:=0 to 10 do
if (e[j]<min) and (e[j]>w) then min:=e[j];
k:=1;
sum:=0;
for j:=0 to 10 do
```

```
if e[j]>=min  then
begin
sum:=sum+e[j];
k:=k+1;
end;
res:=res+(sum/(k-1)*(min-w));
w:=min;
end;
ADOTable1.Insert;
ADOTable1.FieldByName('Кандидат').AsString:=Edit4.Text ;
ADOTable1.FieldByName('Оцінка').AsString:=floattostr(RoundTo(res,-4));
ADOTable1.FieldByName('Освіта').AsString:= ComboBox4.Text;
ADOTable1.FieldByName('Кваліфікація').AsString:=Edit1.Text;
ADOTable1.FieldByName('Вміння працювати з ПЗ').AsString:=Edit2.Text;
ADOTable1.FieldByName('Опит').AsString:=ComboBox2.Text ;
ADOTable1.FieldByName('Юридична грамотність').AsString:=Edit3.Text ;
ADOTable1.Post ;
end;
procedure TForm12.FormCreate(Sender: TObject);
begin
ADOTable1.First;
while not ADOTable1.EOF do
begin
ADOTable1.Delete;
ADOTable1.Next;
end;
ADOTable1.First;
while not ADOTable1.EOF do
begin
ADOTable1.Delete;
```

```
ADOTable1.Next;
end;
end;
procedure TForm12.Button2Click(Sender: TObject);
var pib:string ;
max:real;
begin
Button3.Visible:=true;
Memo1.Visible:=true;
Memo1.Lines.Add('Висновок:');
ADOTable1.First;
While not ADOTable1.Eof do
begin
Memo1.Lines.Add('Кандидат '+ADOTable1.FieldByName('Кандидат').AsString+
' отримав оцінку '+ADOTable1.FieldByName('Оцінка').AsString+';');
ADOTable1.Next;
end;
max:=0;
ADOTable1.First;
While not ADOTable1.Eof do
begin
if (ADOTable1.FieldByName('Оцінка').AsFloat>max) then
begin
max:= ADOTable1.FieldByName('Оцінка').AsFloat;
pib:=ADOTable1.FieldByName('Кандидат').AsString
end;
ADOTable1.Next;
end;
Memo1.Lines.Add('Найкращім кандидатом на посаду  можна вважати '+pib+ ',
який отримав оцінку '+floattostr(max)+', яка є максимальною');
```

```
end;
procedure TForm12.Button3Click(Sender: TObject);
var WordApplication:variant;
begin
WordApplication:=CreateOLEObject('Word.Application');
WordApplication.Visible:=True;
WordApplication.DisplayAlerts:=True;
Memo1.SelectAll;
Memo1.CopyToClipboard;
WordApplication.Documents.Add;
WordApplication.Documents.Item(1).Range.Paste;
end;
end.
unit Unit13;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ADODB, Grids, DBGrids, DB, DBCtrls, Unit14;
type
  TForm13 = class(TForm)
    ADOTable1: TADOTable;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    DBComboBox1: TDBComboBox;
    DBComboBox2: TDBComboBox;
    DBComboBox3: TDBComboBox;
    DBComboBox4: TDBComboBox;
```

```
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    DBGrid1: TDBGrid;
    ADOQuery1: TADOQuery;
    Button1: TButton;
    ADOTable2: TADOTable;
    ADOTable3: TADOTable;
    ADOTable4: TADOTable;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form13: TForm13;
implementation
{$R *.dfm}
procedure TForm13.Button1Click(Sender: TObject);
var Form14:TForm14;
begin
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select *');
ADOQuery1.SQL.Add('from Работники ');
ADOQuery1.SQL.Add(' Where Фамилия = '"+DBComboBox2.Text+'" and' );
ADOQuery1.SQL.Add('  Имя= '"+DBComboBox3.Text+'" and');
ADOQuery1.SQL.Add('  Отчество = '"+DBComboBox4.Text+'" and');
ADOQuery1.SQL.Add('  ТабN = '+DBComboBox1.Text+");
ADOQuery1.Active:=True;
Form14:=TForm14.Create(Self);
```

```
Form14.ShowModal;
end;
end.
unit Unit14;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, DBCtrls, Grids, DBGrids, StdCtrls, Buttons, ComCtrls,
  ADODB, DB;
type
  TForm14 = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label36: TLabel;
    Edit1: TEdit;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    GroupBox1: TGroupBox;
```

```
Label8: TLabel;
Label9: TLabel;
Label11: TLabel;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Edit28: TEdit;
TabSheet2: TTabSheet;
Label10: TLabel;
Label16: TLabel;
Edit8: TEdit;
GroupBox2: TGroupBox;
Label12: TLabel;
Label14: TLabel;
Edit9: TEdit;
Edit10: TEdit;
GroupBox3: TGroupBox;
Label13: TLabel;
Label15: TLabel;
Edit11: TEdit;
Edit12: TEdit;
Edit13: TEdit;
TabSheet3: TTabSheet;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
Label20: TLabel;
Label21: TLabel;
Label22: TLabel;
ComboBox3: TComboBox;
```

```
    Edit14: TEdit;
    Edit15: TEdit;
    Edit16: TEdit;
    Edit17: TEdit;
    Edit18: TEdit;
    ADOTable3: TADOTable;
    ADOTable4: TADOTable;
    ADOTable2: TADOTable;
    ADOQuery1: TADOQuery;
    ADOTable1: TADOTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form14: TForm14;
implementation
{$R *.dfm}
procedure TForm14.FormCreate(Sender: TObject);
begin
ADOTable2.First;
While not ADOTable2.Eof do
begin
if ADOTable2.FieldByName('ID').AsInteger
=ADOQuery1.FieldByName('ОбразованиеID').AsInteger then
ComboBox1.Text:=ADOTable2.FieldByName('Образование').AsString;
ADOTable2.Next;
end;
```

```
ADOQuery1.First;
if  ADOQuery1.FieldByName('Пол').AsBoolean=False then
RadioButton1.Checked:=True;
if  ADOQuery1.FieldByName('Пол').AsBoolean=True then
RadioButton2.Checked:=False;
Edit1.Text:=inttostr(ADOQuery1.FieldByName('ТабN').AsInteger);
Edit2.Text:=ADOQuery1.FieldByName('Фамилия').AsString;
Edit3.Text:=ADOQuery1.FieldByName('Имя').AsString;
Edit4.Text:=ADOQuery1.FieldByName('Отчество').AsString;
Edit28.Text:=ADOQuery1.FieldByName('ДатаРождения').AsString;
Edit7.Text:=ADOQuery1.FieldByName('РождГород').AsString;
Edit6.Text:=ADOQuery1.FieldByName('РождОбласть').AsString;
Edit5.Text:=ADOQuery1.FieldByName('РождСтрана').AsString;
ComboBox1.Text:=ADOQuery1.FieldByName('СемПоложение').AsString;
Edit8.Text:=ADOQuery1.FieldByName('ИНН').AsString;
Edit9.Text:=ADOQuery1.FieldByName('АдресРегистрИндекс').AsString;
Edit10.Text:=ADOQuery1.FieldByName('АдресРегистр').AsString;
Edit11.Text:=ADOQuery1.FieldByName('АдресФактИндекс').AsString;
Edit12.Text:=ADOQuery1.FieldByName('АдресФакт').AsString;
Edit13.Text:=ADOQuery1.FieldByName('Телефон').AsString;
ADOTable4.First;
While not ADOTable4.Eof do
begin
if                                ADOTable4.FieldByName('РаботникID').AsInteger
=ADOQuery1.FieldByName('ТабN').AsInteger then
begin
//ComboBox3.Text:=inttostr(ADOTable4.FieldByName('ШтатРасписаниеID').As
Integer);
Edit16.Text:=floattostr(ADOTable4.FieldByName('Ставка').AsFloat);
Edit17.Text:=floattostr(ADOTable4.FieldByName('Оклад').AsFloat);
```

```
Edit18.Text:=floattostr(ADOTable4.FieldByName('Надбавка').AsFloat);
Edit14.Text:=ADOTable4.FieldByName('ПринятС').AsString;
Edit15.Text:=ADOTable4.FieldByName('ПринятПо').AsString;
end;
ADOTable4.Next;
end;
end;
end.
nit Unit15;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComObj, DB, ADODB, ExtCtrls, DBCtrls, Grids, DBGrids;
type
  TForm15 = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    DataSource1: TDataSource;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form15: TForm15;
implementation
uses Unit12;
{$R *.dfm
nit Unit16;
interface
```

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ADODB, DBCtrls, Menus, ExtCtrls;
type
  TForm16 = class(TForm)
    Panel1: TPanel;
    Label14: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    Label17: TLabel;
    Label1: TLabel;
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    ComboBox3: TComboBox;
    ComboBox4: TComboBox;
    ComboBox5: TComboBox;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    Panel2: TPanel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
```

```
  ADOTable1: TADOTable;
  ADOTable2: TADOTable;
  DataSource2: TDataSource;
  Edit8: TEdit;
  DataSource1: TDataSource;
  ComboBox6: TComboBox;
  Label9: TLabel;
  Label10: TLabel;
  ComboBox7: TComboBox;
  Label11: TLabel;
  ComboBox8: TComboBox;
  Label12: TLabel;
  ComboBox9: TComboBox;
  Label13: TLabel;
  ComboBox10: TComboBox;
  Label18: TLabel;
  Label19: TLabel;
  Button1: TButton;
  Edit5: TEdit;
  Button2: TButton;
  ADOTable3: TADOTable;
  ComboBox11: TComboBox;
  ComboBox12: TComboBox;
  procedure N1Click(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
```

```
    { Public declarations }
  end;
var
  Form16: TForm16;
implementation
uses Unit1, Unit2;
{$R *.dfm}
procedure TForm16.N1Click(Sender: TObject);
begin
Form1.Close;
Form2.Close;
Form16.Close;
end;
procedure TForm16.Button1Click(Sender: TObject);
Var q:integer;
begin
edit5.Visible:=true;
q:=0;
if ComboBox4.Text='Высшее образование' then
q:=3;
if ComboBox4.Text='Техническое' then
q:=2;
if ComboBox4.Text='Среднее общее образование' then
q:=1;
if ComboBox2.Text='от 1 года до 5' then
q:=q+1;
if ComboBox2.Text='от 5 до 10' then
q:=q+2;
if ComboBox2.Text='более 10' then
q:=q+3;
```

```
if ComboBox5.Text='Да' then
q:=q+3;
 if ComboBox5.Text='Нет' then
q:=q+1;
  if ComboBox1.Text='Да' then
q:=q+3;
 if ComboBox1.Text='Нет' then
q:=q+1;
  if ComboBox3.Text='Есть' then
q:=q+3;
 if ComboBox3.Text='Нет' then
q:=q+1;
if ComboBox6.Text='от 55 до 70' then
q:=q+1;
if ComboBox6.Text='от 71 до 85' then
q:=q+2;
if ComboBox6.Text='от 86' then
q:=q+3;
if ComboBox7.Text='Хорошо понимает свои обязанности' then
q:=q+3;
if ComboBox7.Text='Многие вопросы необходимо объяснять' then
q:=q+2;
if ComboBox7.Text='Плохо понимает свою работу' then
q:=q+1;
if ComboBox8.Text='Сохраняет выдержку' then
q:=q+3;
if ComboBox8.Text='Старается сдерживать себя в сложных ситуациях' then
q:=q+2;
if ComboBox8.Text='Нервное поведение' then
q:=q+1;
```

```
if ComboBox9.Text='Внимательно слушает, задает уточняющие вопросы' then
q:=q+3;
if ComboBox9.Text='Внимательно слушает' then
q:=q+2;
if ComboBox9.Text='Не слушает, перебивает' then
q:=q+1;
if ComboBox10.Text='Хорошо излагает свои мысли' then
q:=q+3;
if ComboBox10.Text='Возникают сложности в логическом обосновании своих
мыслей' then
q:=q+2;
if ComboBox10.Text='С трудом излагает свои мысли ' then
q:=q+1;
edit5.Text:=floattostr(q/10);
end;
procedure TForm16.Button2Click(Sender: TObject);
begin
ADOTable3.Insert;
ADOTable3.FieldByName('ID').AsInteger:=strtoint(Edit1.Text);
ADOTable3.FieldByName('Отдел').AsString:=ComboBox11.Text;
ADOTable3.FieldByName('Должность').AsString:= ComboBox12.Text;
ADOTable3.FieldByName('Количество').AsInteger:=strtoint(Edit2.Text);
ADOTable3.FieldByName('Ставка').AsInteger:=strtoint(Edit3.Text);
ADOTable3.FieldByName('Оклад').AsInteger:=strtoint(Edit4.Text);
ADOTable3.FieldByName('Надбавка').AsInteger:=strtoint(Edit8.Text);
ADOTable3.FieldByName('Рейтинг').AsFloat:=strtofloat(Edit5.Text);
ADOTable3.Post ;
end;
procedure TForm16.FormCreate(Sender: TObject);
begin
```

```
ADOTable1.First;
While not ADOTable1.Eof do
begin
Combobox12.Items.Add(ADOTable1.FieldByName('Должность').AsString);
ADOTable1.Next;
end;
ADOTable2.First;
While not ADOTable2.Eof do
begin
Combobox11.Items.Add(ADOTable2.FieldByName('Отдел').AsString);
ADOTable2.Next;
end;
end;
end.
```