

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

СЕМЕНЮК АНДРІЙ МИХАЙЛОВИЧ

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій
канд. техн. наук, доцент
_____ О. В. Зелінська
« ____ » _____ 20__ р.

**РОЗРОБКА ІНФОРМАЦІЙНО-ЕКСПЕРТНОЇ
МЕДИЧНОЇ СИСТЕМИ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:

Т. В. Січко, доцент кафедри
інформаційних технологій,
к. т. н., доцент

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

Вінниця - 2025

АНОТАЦІЯ

Семенюк А.М **Розробка інформаційно-експертної медичної системи**
Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця 2024.

У кваліфікаційній (бакалаврській) роботі розглянуто поняття, структуру та алгоритм роботи медичної інформаційної системи. Засовуючи такі технології як Delphi, SQL, Firebird, MS Excel, та застосуванням принципів нечіткої логіки було створено власну ІС.

Ключові слова: медична інформаційна система, база даних, експертна система, реляційна модель, електронна карта пацієнта, SQL, GUI, автоматизація, діагностика, електронна система охорони здоров'я, медичний заклад, інтерфейс користувача, графічний інтерфейс, статистична інформація, підтримка прийняття рішень, Delphi, InterBase, Firebird, звіт.

60 ст. 21 рис., 2 табл., 4 дод., 20 джерел.

ABSTRACT

Semenyuk A.M. Development of an Information-Expert Medical System. Major 122 "Computer Science", educational program "Computer Science". Vasyl Stus Donetsk National University, Vinnytsia, 2024.

The bachelor's qualification thesis explores the concept, structure, and operation algorithm of a medical information system. Using technologies such as Delphi, SQL, Firebird, MS Excel, and applying principles of fuzzy logic, a custom information system was developed.

Keywords: medical information system, database, expert system, relational model, electronic patient record, SQL, GUI, automation, diagnostics, electronic healthcare system, medical institution, user interface, graphical interface, statistical information, decision support, Delphi, InterBase, Firebird, report.

60 pages, 21 figures, 2 table, 4 appendices, 20 references.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	4
ВСТУП	6
РОЗДІЛ 1	
АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ	13
1.1 Аналіз предметної області	13
1.2 Аналіз функцій медичної інформаційної системи	17
1.3 Постановка задачі розробки МІС	20
РОЗДІЛ 2	
ПРОЕКТУВАННЯ МЕДИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	23
2.1 Структура інформаційної системи	25
2.2 Проектування бази даних інформаційної системи	28
2.3 Проектування експертного блоку	35
РОЗДІЛ 3	
ОПИС ПРОГРАМИ	43
3.1 Початок роботи, введення даних	43
3.2 Графічний інтерфейс	46
3.3 Попередня експертна оцінка стану пацієнта	49
3.4 Формування звітів, статистичної інформації, фільтри даних	53
3.5 Керівництво користувача	54
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58
ДОДАТКИ	61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АСУ	– автоматизована система управління;
БД	– база даних;
ВНМУ	– Вінницький національний медичний університет;
ДР	– діабетична ретинопатія;
ЕСОЗ	– електронна система охорони здоров'я;
ЕК	– експертна комісія;
ЕКГ	– електрокардіограма;
ЕКП	– електронна карта пацієнта;
ЕОМ	– електронно-обчислювальна машина;
ЗОЗ	– заклад охорони здоров'я;
ІКТ	– інформаційно-комунікаційні технології;
ІТ	– інформаційні технології;
ІС	– інформаційна система;
КТ	– комп'ютерна томографія;
ЛМІ	– людино-машинний інтерфейс;
ЛПЗ	– лікарсько-профілактичний заклад;
МАІ	– метод аналізу ієрархії;
МІС	– медичні інформаційні системи;
МКП	– медична карта пацієнта;
МОЗ України	– Міністерство охорони здоров'я України;
НСЗУ	– Національна служба здоров'я України;
ОПР	– особа прийняття рішень;
ПЗ	– програмне забезпечення;
ПК	– персональний комп'ютер;
ПП	– пакет програм;
ПрО	– предметна область;
СКД	– система компоновки даних;

СППР	– система підтримки прийняття рішень;
СУБД	– система управління базами даних;
ЦБТ	– центральна база даних;
SQL	– structured query language.



ВСТУП

У сучасному світі інформаційні системи використовуються для автоматизації діяльності процесів різноманітних організацій, підприємств або установ. Це дозволяє спростити та пришвидшити певні робочі процеси.

В цьому плані взаємодія користувача та обчислювального середовища є виконання інтерактивними обчислювальними системами задач на потребу людини. І суттєвим аспектом цієї взаємодії є забезпечення задоволення користувачів. А тому, нагальним є створення інформаційно-обчислювальних систем не лише "широкого" профілю, а й пристосованих під специфіку окремих підрозділів.

Медична галузь не є винятком. Завдання які можливо виконати в медицині та галузі охорони здоров'я з використанням комп'ютерних технологій, багатогранні. Їх можна класифікувати за такими параметрами [1]:

- ціль;
- зміст;
- напрямками використання;
- рівень використання.

Задачі таких систем можна розподілити [1]:

- надання інформаційної допомоги медичному персоналу;
- проведення моніторингу стану здоров'я населення;
- інформаційне забезпечення невідкладної допомоги у екстрених випадках;
- інформаційне забезпечення наукової та дослідної роботи.

Нині існує великий вибір програмних засобів для комп'ютеризованого ведення медичних карточок пацієнта, організації роботи медичного закладу, взаємодії з пацієнтом. Однак, окремих програм, які враховують не достатню "комп'ютерну" підготовку персоналу, і головною функцією яких є автоматизація формування медичної документації на рівні окремих відділень медичного закладу, створення звітних статистичних даних, та попередніх експертних оцінок стану особи мала кількість.

Актуальність роботи обумовлена:

- відсутністю найбільш зручного програмного рішення задач, що відповідають специфіці та критеріям окремих відділень та малих амбулаторій;
- необхідністю скорочення часу на виконання дій необхідних для реєстрації пацієнтів, формуванню заключень та статистичних форм;
- необхідністю зменшення кількості помилок при введення особових даних;
- стандартизації типових відповідей на анкетні запитання для уникнення дублювання ідентичних;
- впровадженням інформаційної підтримки роботи медичного персоналу;
- зменшення затрат робочого часу на виконання рутинних операцій.

Мета кваліфікаційної роботи – розробка медичної інформаційної системи обробки та аналізу даних, яка дозволяє автоматизувати реєстрацію осіб в медичному закладі експертного профілю, призначення та контроль діагностичних досліджень, формування вихідних заключень, та статистичної звітності з метою покращення показників охорони здоров'я населення, поліпшення медичних досліджень. Також проводиться оцінка впливу новітніх інформаційних технологій на надання експертно-медичних послуг формування клінічних рішень і рекомендацій пацієнтам.

Об'єкт дослідження – організація раціонального процесу взаємодії окремих підрозділів, медичного персоналу з пацієнтом, та закладу в цілому, людино-машинна взаємодія в широкому значенні.

Предметом дослідження є – інформаційні системи, алгоритми, методи та підходи які направлені на автоматизацію формування медичних справ, експертних висновків, статистичних аналізів.

Дана розробка зможе надати можливість покращити обслуговування пацієнтів шляхом зменшення часу який витрачається на формування первинної, поточної та заключної медичної документації, дозволить уникнути помилок при введенні типових даних, надати можливість використання АРМ персоналом лише

з базовою комп'ютерною підготовкою, формуванню інформаційних баз для медико-статистичних досліджень, виконанню постійного й оперативного аналізу роботи відділення.

Апробація результатів дослідження. Результати кваліфікаційної (бакалаврської) роботи апробовано:

– III Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «ПРИКЛАДНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ», яка відбулась 22 квітня 2022 р. на базі кафедри інформаційних технологій Донецького національного університету імені Василя Стуса. Тези на тему: **"РОЗРОБКА АРМ БАЗ ДАНИХ МЕДИЧНОГО РЕЄСТРАТОРА"** були опубліковані у збірнику наукових праць– Вінниця, ДонНУ імені Василя Стуса, 2022, 138 с., С.89÷92, <https://jait.donnu.edu.ua/article/view/12271>, doi: 2708-5821

– Міжнародна науково-практична конференція «Молодіжна наука заради миру та розвитку», яка відбулась 9–11 листопада 2022 року, Чернівці. Тези на тему: **"SOFTWARE COMPLEX OF THE EXPERT DEPARTMENT"** були опубліковані збірнику наукових праць Чернівецького національного університету ім. Ю. Федьковича, 2022. 656 с., С.574÷577, ISBN 978-966-423-755-7

– I МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ «Прикладні аспекти сучасних міждисциплінарних досліджень», яка відбулась 18 листопада 2022 року на базі кафедри інформаційних технологій Донецького національного університету імені Василя Стуса. Тези на тему: **"ООП ДЛЯ КОМП'ЮТЕРНОГО ОБЛІКУ ТА АНАЛІЗУ"** були опубліковані в електронному збірнику наукових праць, який можна переглянути за посиланням <https://jpasmd.donnu.edu.ua/article/view/12961>

– III Всеукраїнська науково-практична конференція «КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ОБРОБКИ ДАНИХ», яка відбулась 8 грудня 2022 р., на базі кафедри інформаційних технологій Донецького національного університету імені Василя Стуса. Тези на тему: **"ОБРОБКА ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ РЕЛЯЦІЙНИХ БАЗ ДАНИХ"** були опубліковані в

електронному збірнику наукових праць, який можна переглянути за посиланням <https://jktod.donnu.edu.ua/article/view/13107>

– ЛІІ Науково-технічна конференція підрозділів ВНТУ, «ВСС ВНТУ ФІТ та КІ», яка відбулась 21-23 червня 2023 р., м. Вінниця на базі факультету ІТ та КІ ВНТУ. Тези на тему: **"СТВОРЕННЯ РЕЄСТРАЦІЙНОЇ СИСТЕМИ МСЕК"** були опубліковані в збірнику наукових праць НТКП ВНТУ–2023 Вінниця, 21–23 червня 2023 р. С.744–746, електронний збірник наукових праць <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2023>

– ІІ Міжнародна науково-практична конференція «МОЛОДІЖНА НАУКА ЗАРАДИ МИРУ ТА РОЗВИТКУ», яка відбулась 08-10 листопада 2023 р., м. Чернівці на базі . Тези на тему: **"СТАТИСТИЧНІ ДОСЛІДЖЕННЯ РЕАБІЛІТАЦІЙНИХ ЗАХОДІВ"** були опубліковані у збірнику наукових праць Чернівецького національного університету ім. Ю. Федьковича, 2023. 430 с., С.291÷292 , ISBN 978-966-423-811-0

– ІІ Міжнародна науково-практична конференція «ПРИКЛАДНІ АСПЕКТИ СУЧАСНИХ МІЖДИСЦИПЛІНАРНИХ ДОСЛІДЖЕНЬ», яка відбулась 24 листопада 2023 р на базі кафедри інформаційних технологій Донецького національного університету імені Василя Стуса. Тези на тему: **"МОДЕЛЮВАННЯ СИСТЕМ ПРИЙНЯТТЯ РІШЕНЬ МЕДИЧНОЇ ЕКСПЕРТИЗИ"** та **"ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ У КЛІЄНТ-СЕРВЕРНИХ ДОДАТКАХ НА ПРИКЛАДІ МЕДИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ"** були опубліковані в електронному збірнику наукових праць, який можна переглянути за посиланням <https://jpasmd.donnu.edu.ua/article/download/14758>, <https://jpasmd.donnu.edu.ua/article/view/14809>

– ІV Всеукраїнська науково-практична конференція «КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ОБРОБКИ ДАНИХ», яка відбулась 8 грудня 2023 р., на базі кафедри інформаційних технологій Донецького національного університету імені Василя Стуса. Тези на тему: **"ОПТИМІЗАЦІЯ ВИРОБНИЦТВА РЕАБІЛІТАЦІЙНИХ ЗАСОБІВ З ВИКОРИСТАННЯМ ІМІТАЦІЙНОГО**

МОДЕЛЮВАННЯ" були опубліковані в електронному збірнику наукових праць, який можна переглянути за посиланням <https://jktod.donnu.edu.ua/article/view/16222>

– V Міжнародна науково-практична конференція молодих вчених та студентів присвячена 125-й річниці КПІ ім. Ігоря Сікорського «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ І ПЕРЕДОВІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ», яка відбулась 19–21 грудня 2023 р., м. Київ, на базі . Тези на тему: **"ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СТАТИСТИЧНИХ ДАНИХ ПРОПОЗИЦІЙ РЕАБІЛІТАЦІЙНИХ ПОСЛУГАХ"** були опубліковані у збірнику наукових праць м. Київ, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», ФІОТ, 399 с., С.296÷298, ISSN 2306-7233 (Online), ISSN 2306-6962 (Print)

– “StudMath-IT International Students Conference”, яка відбулась 20÷21 листопада 2024 року на базі “Aurel Vlaicu” University of Arad Romania. Тези на тему: **«STATISTICAL RESEARCH OF REHABILITATION MEASURES PERSONS WITH DISABILITIES»** були опубліковані в електронному збірнику наукових праць, який можна переглянути за посиланням https://studmathit-uav.ro/wp-content/uploads/2025/01/StudMathIT2024-Book_Of_Abstracts_FinalVersion_.pdf.

– V ВСЕУКРАЇНСЬКА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ «КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ОБРОБКИ ДАНИХ», яка відбулась 6 грудня 2024 р. на базі кафедри інформаційних технологій Донецького національного університету імені Василя Стуса м. Вінниця. Тези на тему: **"СИСТЕМА ЕКСПЕРТНОЇ ОЦІНКИ НА БАЗІ FUZZY LOGIC"**

– III Міжнародна науково-практична конференція "МОЛОДІЖНА НАУКА ЗАРАДИ МИРУ ТА РОЗВИТКУ", яка відбулась 12-14 грудня 2024 р., м. Чернівці на базі. Тези на тему: **"ВИКОРИСТАННЯ МОДЕЛІ ТАКАГІ-СУГЕНО В МЕДИЧНІЙ СТАТИСТИЦІ"** були опубліковані у збірнику наукових праць Чернівецького національного університету ім. Ю. Федьковича, 2024. 346 с., с.85÷86 , ISBN 978-966-423-938-4

– В. І. Шевчук, Р. Я. Перепелична, Н. М. Беляєва, Л. О. Сторожук, І. В. Куриленко, Л. Г. Семененко, М. В. Семенюк, А. М. Семенюк, **«ОСНОВНІ ПОКАЗНИКИ МЕДИКО-СОЦІАЛЬНОЇ РЕАБІЛІТАЦІЇ ОСІБ З ІНВАЛІДНІСТЮ В УКРАЇНІ ЗА 2021 РІК»**, Аналітико-інформаційний довідник, – Вінниця: ФОП Данилюк В. Г., 2022. – 120 с., ISBN 978-617-8028-11-4

– В. І. Шевчук, Р. Я. Перепелична, Л. О. Сторожук, І. В. Куриленко, Е. М. Дрозденко, Л. Г. Семененко, М. В. Семенюк, А. М. Семенюк, **«ОСНОВНІ ПОКАЗНИКИ МЕДИКО-СОЦІАЛЬНОЇ РЕАБІЛІТАЦІЇ ОСІБ З ІНВАЛІДНІСТЮ В УКРАЇНІ ЗА 2022 РІК»**, Аналітико-інформаційний довідник, – Вінниця: ФОП Рогальська І. О., 2023. – 112 с., ISBN 978-617-8028-19-0

– В. І. Шевчук, Р. Я. Перепелична, Л. О. Сторожук, І. В. Куриленко, Е. М. Дрозденко, Л. Г. Семененко, М. В. Семенюк, А. М. Семенюк, **«ОСНОВНІ ПОКАЗНИКИ МЕДИКО-СОЦІАЛЬНОЇ РЕАБІЛІТАЦІЇ ОСІБ З ІНВАЛІДНІСТЮ В УКРАЇНІ ЗА 2023 РІК»**, Аналітико-інформаційний довідник, – Вінниця: ФОП Данилюк В. Г., 2024. – 114 с., ISBN 978-617-8028-19-0

Мета роботи і полягає у поширенні використання МІС на "низових" рівнях – у відділеннях, а саме у відділеннях які входять до складу Науково-дослідного інституту реабілітації осіб з інвалідністю Вінницького національного медичного університету ім. М. І. Пирогова (з 2025 року Університетська лікарня ВНМУ ім. М. І. Пирогова).

Для початкової стадії впровадження було відібране експертне відділення. Його функції полягають:

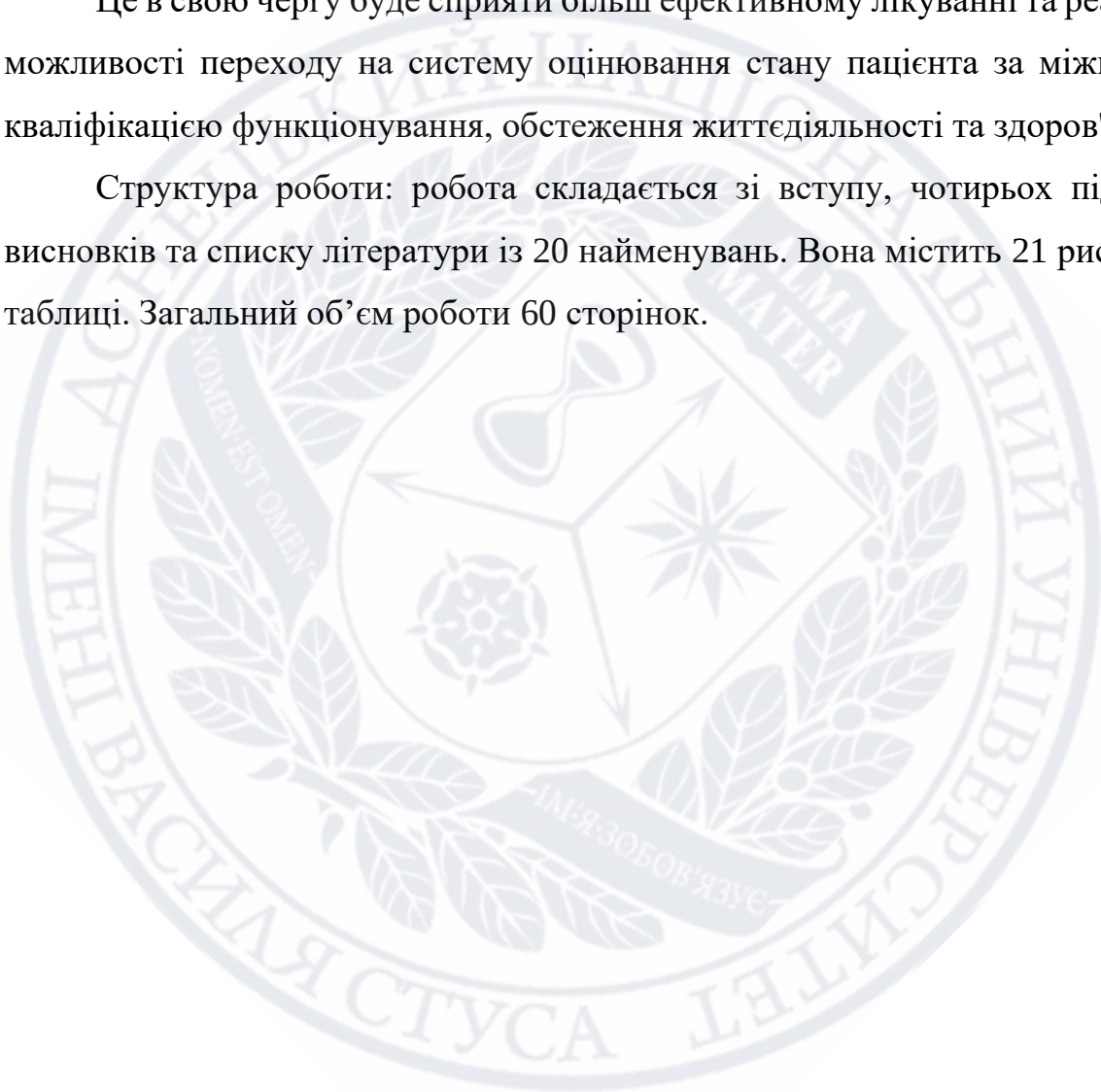
- визначення ступеню обмеження життєдіяльності пацієнтів;
- визначення потреб реабілітації;
- визначення реабілітаційного потенціалу;
- визначення групи інвалідності, часу і причини отримання;
- визначення ступеню втрати професійної працездатності (у відсотках)

працівників, які одержали ушкодження здоров'я, пов'язане з виконанням ними трудових обов'язків.

Даний підрозділ охоплює в своїй роботі широкий контингент осіб і координує свою діяльність з іншими, більш вузькоспеціалізованими відділеннями. Тому напрацювання отримані від впровадження МІС зможуть бути легко поширені для використання в цілому у закладі.

Це в свою чергу буде сприяти більш ефективному лікуванню та реабілітації, можливості переходу на систему оцінювання стану пацієнта за міжнародною кваліфікацією функціонування, обстеження життєдіяльності та здоров'я.

Структура роботи: робота складається зі вступу, чотирьох підрозділів, висновків та списку літератури із 20 найменувань. Вона містить 21 рисунок та 2 таблиці. Загальний об'єм роботи 60 сторінок.



РОЗДІЛ 1

АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

В діяльності держави не може бути неважливих галузей, але деякі з них, як от медична галузь, повинна стати невід'ємною частиною "цифрової" України, тому що інформаційно-комунікаційні технології є необхідним засобом процвітання держави, та її руху до складу Європейської спільноти.

Її розвиток і стан безпосередньо впливає на всі аспекти піднесення країни, а тому поліпшення організаційно-управлінського та лікувального процесів є першочерговим для утримання галузі на передовому рівні. Державі необхідно подбати щоб лікарі були забезпечені зручним і швидким інструментом створення, збереження та передачі медичних даних. Для вирішення цих задач розробляються медичні інформаційні системи (МІС). Подібні системи є невід'ємною складовою взаємодії між сучасними "цифровим" пацієнтом і "цифровим" лікарем [2].

На сьогоднішній день, інформація це не тільки соціальний аспект, а й велика база даних якій необхідне впорядкування та систематизація. Даний процес створення інформаційного набору потребує формування індивідуальних "порцій" даних з яких отримуємо контейнер відомостей стосовно заданої теми або предмету дослідження.

Спосіб ведення документації в електронному вигляді на сьогоднішній день не підлягає сумніву. Найбільш зручний засіб зберігання інформації у вигляді електронних баз даних, або електронних таблиць. Надалі це дозволить систематизувати її засобами комп'ютерної обробки з використанням різноманітних критеріїв. Дану тезу підтверджує наявність великої кількості вітчизняних та закордонних публікацій.

Аналіз впровадження МІС [3] і порівняння використання різних інформаційних баз закладів первинної медичної допомоги доводить, що останнім часом задачам розробки різноманітних МІС надавали багато уваги.

Використання систем обробки інформації у галузі охорони здоров'я зросло в декілька разів, тому що подібні новації сприяють наданню якісних медичних послуг, оптимізують робочий процес медичних працівників, а це веде до покращення ефективності роботи персоналу. Дана модель дозволяє зберігати відомості у вигляді електронних таблиць, здійснювати пошук інформації в одній чи декількох таблицях на основі пошукового ключа, та її систематизацію. Цей процес не потребує зміни структури таблиць БД, і може бути виконаний повторно як з даним набором запитів та даних, так і з видозміненим.

Для керівника закладу дана розробка надає ряд організаційних переваг:

- підвищення ефективності процесу;
- зменшення можливих людських помилок;
- оптимізація використання ресурсів.

За структурною організацією і функціональним призначенням МІС [4] класифікуються;

- професійні МІС, потужні системи для медичних установ;
- індивідуальні МІС, системи персонального призначення (медичні калькулятори), використовуються для експрес-аналізу і встановлення індивідуальних показників стану пацієнта;
- відбіркові МІС, застосовуються для скринінгових досліджень з використанням комп'ютерних систем і технологій;
- комбіновані МІС.

На даний момент в медичній галузі України широко використовуються декілька професійних МІС [5, 6]:

- «МедІнфоСервіс»;
- «Helsi»;
- «Health24»;
- «Doctor Eleks».

Ці системи дозволяють вести обмін даними у центральній компоненті "eHealth", створеній в Національній службі здоров'я України (НСЗУ).

Вони дозволяють накопичувати великий об'єм інформації, доступ до неї з

використанням як внутрішньої мережі закладу так і ВЕБ-технологій і мають свої переваги та недоліки [6, 7]. Розглянемо найбільш поширені.

Система **Helsi** виконана у форматі веб-застосунку. МІС складається з трьох складових:

- *helsi.me* – модуль призначений для пацієнтів;
- *helsi.pro* – модуль призначений для персоналу медичної установи та лікарів (можливість прийому пацієнтів в форматі eHealth, з подальшим встановленням діагнозів відповідно до ICPC2 або МКХ-11);
- *reform.helsi.me* – модуль для використання адміністрацією мед установи.

Серед головних недоліками системи Helsi виділяють:

- модуль *helsi.pro* має низьку швидкість роботи;
- не інтуїтивний та складний інтерфейс для користувачів без досвіду роботи з МІС;
- "великовагова", що ускладнює використання ФОПами і малими закладами;
- всі три модулі мають різний інтерфейс, що викликає труднощі у лікарів при переході з *reform.helsi.me* на *helsi.pro*.

Doctor Eleks – також широко розповсюджена медична інформаційна система. Це окремий програмний комплекс з власним інтерфейсом.

Серед головних недоліками системи Doctor Eleks виділяють:

- неможливість мультиплатформного використання АРМ лікаря, основний модуль системи розроблено під операційну систему Windows;
- незвичний дизайн застосунку, для рядового (з базовою комп'ютерною підготовкою) користувача не зрозумілий;
- висока вартість встановлення системи – без урахування додаткових витрат що включають ПК, сервер та дворічну технічну підтримку складає 400 доларів США за одне робоче місце,.

Всі медичні інформаційні системи реалізуються у форматі веб-застосунку або десктопному. Аналіз переваг та недоліків найбільш широко

використовуваних МІС, а також існуючих вимог Національної служби здоров'я України (НСЗУ) дозволяє зробити наступні висновки:

- погано розвинута модульна архітектура не дозволяє налаштовувати МІС до вимог конкретного ЛПЗ;
- "важкість" систем та складність комунікацій з різними розробниками унеможливлюють оперативність змін до нових вимог НСЗУ і eHealth;
- більшість систем потребують високої комп'ютерна підготовка персоналу;
- професійні МІС практично недоступні для використання лікарями і пацієнтами за межами ЗОЗ.

В той же час, багато медичних установ не потребують таких складних проектів, які, до речі вимагають досить потужні апаратні ресурси та фінансові затрати. Для цих закладів є актуальним обробка даних і ведення документації у відповідності з профілем установи, з її специфікою.

Наприклад для невеликих фельдшерських пунктів, окремих відділень, спеціалізованих підрозділів (наприклад ВЛК чи ЕК), що мають специфічні задачі, використання стандартизованих МІС потребує налаштування цих систем під потрібні функції. Це в свою чергу веде до необхідності зміни програмного забезпечення, а отже додаткових затрат та часу роботи виробників ПЗ. В деяких випадках технічні умови різних підрозділів можуть мати не сумісні запити, що ускладнює, а іноді і унеможлиблює розробку універсальних програм.

Звичайно альтернативою можливе використання різних офісних пакетів як тих що поширюються на умовах вільного доступу, online-сервісів, або на платній основі. Але для них також потрібно розробляти шаблони для заповнення форм, формування звітів, аналітики, статистичних даних. А ця задача вимагає знань більших ніж базова комп'ютерна підготовка на рівні користувача ПЕОМ. Звичайно поступово рівень комп'ютерних знань персоналу підвищується, але на даний час з врахуванням специфіки галузі він недостатній для самостійної, на "місцях" автоматизації реєстраційних і аналітичних функцій.

Тому актуальною є необхідність розробки спеціалізованих програмних

засобів із врахуванням вимог під особливі запити та обмежене фінансування підрозділів і установ.

Комбіновані МІС надають можливість об'єднати переваги систем різних типів для задач персоніфікованої медицини.

Написання такого програмного забезпечення дозволить:

- 1) врахувати "некомп'ютерну" освіту персоналу;
- 2) максимально використовувати заповнення форм з вибором інформації з готових варіантів за допомогою маніпулятора "мишка" або стрілок навігації на клавіатурі;
- 3) зменшити помилки вводу і позбутися випадків дублювання одних і тих самих даних пов'язаних з використанням великих і малих літер, різного скорочення, літер українського та англійського алфавітів з однаковим відображенням на екрані (наприклад – "а", "о", "і"; особливо "с" яка розміщена на тій самій клавіші);
- 4) формувати довідкову інформацію під конкретні технічні вимоги.

Тобто, незважаючи на широку присутність різних програм, як універсальних так і галузевих; за функціональністю та вартістю розробки, інформаційні системи під особливі задачі, з можливістю адаптації і супроводу, є актуальними і необхідними.

1.2 Аналіз функцій медичної інформаційної системи

З урахуванням правил функціонування експертного відділення (рис. 1.1) в МІС повинні бути враховані ряд вимог і побажань до вигляду інтерфейсу програми, її функціональної структури, структури і виду баз даних де зберігається інформація. Уніфіковане введення даних дозволить з максимальною ефективністю використовувати інформацію, та позбутися суттєвих недоліків рукописних форм.

Основні серед них такі:

- слабка структурованість медичної карти;
- зберігання у картці лише кінцевого варіанту аналізів без накопичення

проміжних результатів;

- зберігання результатів досліджень у вигляді виписок, тобто висновків лікаря-діагноста, а самі оригінали даних при цьому знаходяться в архіві діагностичного відділення, оперативний доступ до якого неможливий;
- неохайний вигляд рукописних записів;
- висока ймовірність втратити чи пошкодити історії хвороби як пацієнтом, так і лікувальним закладом.

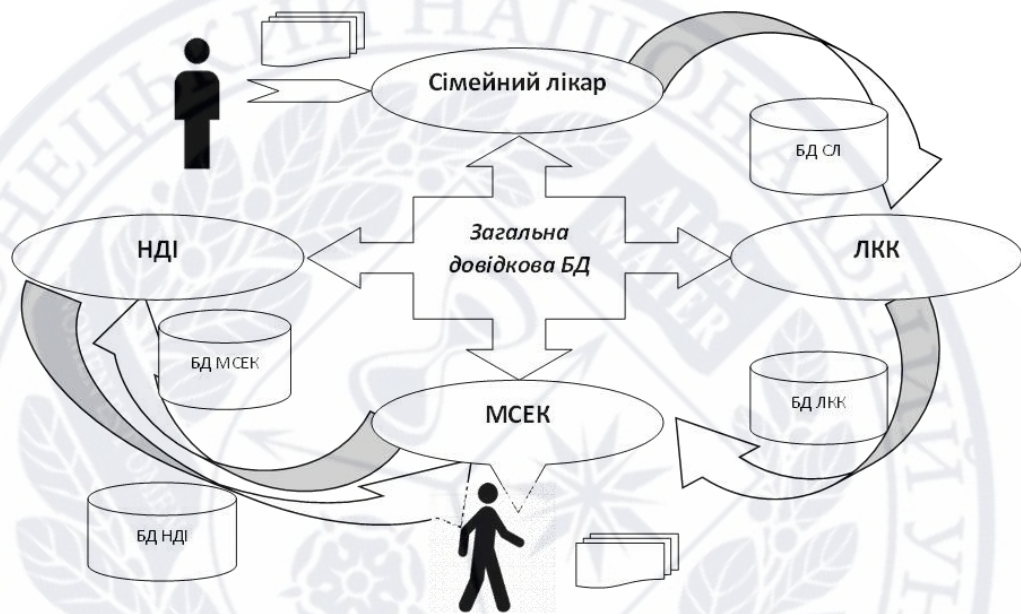


Рисунок 1.1 – Модель взаємодії МІС [4]

Збереження і доступ до інформації повинен бути максимально спрощеним з врахуванням знань персоналу.

Зберігання записів в базі даних забезпечується з можливістю її переносу і встановленню на інший комп'ютер, та використання для доступу до збереженої інформації сторонніх пакетів обробки БД.

Бази даних повинні бути реляційного типу, та використовувати для визначення і обробки даних структуровану мову запитів типу SQL (Structured Query Language). Цей вибір дозволяє звести до мінімуму ряд ризиків пов'язаних із застосуванням БД, так як мова SQL одна з найбільш поширених мов запитів і досить гнучких у використанні.

Робота програми забезпечується на ПК "офісного класу", без необхідності

встановлення додаткового стороннього платного ПЗ.

Формування звітної документації повинно відбуватися відповідно до вимог і правил медичної галузі та затверджених форм звітності.

Персональна інформація пацієнтів захищена згідно із законодавством України про використання і доступ подібних даних.

Як вже зазначалось, електронний варіант медичної картки реалізується у вигляді реляційних баз даних. Інформаційна система містить в собі декілька пов'язаних таблиць. Записи у цих таблицях є чітко структурованими рядками даних, а їх кількість довільна. Подібний спосіб збереження інформації має безліч переваг, основними з яких є [9]:

- швидкий і зручний доступ в автоматичному режимі до всієї бази в цілому і до окремих записів;
- доступ до необхідної діагностичної інформації за певними ознаками або критеріями за допомогою запитів;
- можливість оперативного обміну необхідною медичною інформацією на різних рівнях – від кабінетів і відділень до установи;
- можливість створення вибірки даних за певною ознакою або набору ознак (фільтрація);
- впорядкування отриманих вибірок за різними стовпцями таблиці, за такими, як наприклад: дата, прізвище, діагноз;
- розрахунок статистичних параметрів;

МІС реалізується з використанням графічного інтерфейсу який повинен мати зручні засоби введення та відображення даних. Для введення текстової і цифрової інформації можуть використовуватися наперед підготовлені шаблони і варіанти значень.

В системі повинно бути передбачене автоматичне і "ручне" архівування даних і створення резервної копії БД для захисту інформації від випадкової втрати внаслідок аварійної роботи обладнання або його пошкодження.

Загалом структуру системи узагальнено можливо розділити на дві компоненти:

- введення даних;
- статистичний аналіз даних.

1.3 Постановка задачі розробки МІС

Мета проекту – створення інформаційної системи, що дозволить вдосконалити роботу експертного відділення реабілітації осіб з інвалідністю Університетської лікарні ВНМУ ім. М. І. Пирогова.

Проект планує створення програмного забезпечення що дозволить автоматизувати процедуру реєстрації пацієнтів, оформлення медичної документації, формуванню аналітично-статистичної інформації.

Проектування МІС медичної установи складається з декількох етапів (рис 1.2) [10].

На початку необхідно провести анкетування вимог користувачів (лікарів, медичних сестер, адміністраторів тощо) пов'язаних з видом інтерфейсу, функціональністю та можливостями ПЗ. Наступним кроком є їх детальний аналіз для визначення порядку та пріоритетності функцій взаємодії користувачів з МІС. На етапі проектування оформлюється загальна структура (структура бази даних, задачі клієнтської частини, вигляд інтерфейсу користувача). Розробка та реалізація системи передбачає створення окремих модулів системи, їх інтеграція в загальний проект, тестування окремих модулів та МІС в цілому. На етапі впровадження системи для використання в ЛПЗ проводяться фінальні тести програми та бази даних, налаштування параметрів та підготовка робочого місця до експлуатації в реальних умовах, інструктаж та навчання користувачів.



Рисунок 1.2 – Основні етапи впровадження МІС [10]

Для забезпечення функціонування в медичному закладі інформаційної системи необхідно виконати ряд умов:

- ведення реєстраційної (паспортної) документації;
- ведення медичної документації;
- статистичний аналіз щодо результатів огляду;
- формування звітних форм відповідно до вимог МОЗ України;
- формування заключної медичної документації затвердженого виду.

Використовуваний тип бази даних повинен бути сумісним з базами даних використовуваних в закладі програмних засобів і підтримувати розширення стандартної підмножини мови SQL.

Графічний інтерфейс повинен забезпечувати введення даних до картки пацієнта за допомогою шаблонів, які надають можливість використовувати стандартний набір термінів та виразів (використання довідників). Редагування шаблонів виконується або внутрішніми засобами програми, або за допомогою зовнішніх текстових редакторів.

В програмі передбачається наявність система пошуку даних.

Формування вихідних даних виконується у вигляді файлів стандартних

текстових типів для можливості їх використання незалежними офісними програмами

Отже, у розділі були всебічно розглянуті автоматизація та інформатизація процесів медичного закладу, що дозволяє збільшити ефективність та швидкість роботи, підвищити рівень безпеки пацієнтів та забезпечити швидке та якісне обслуговування.

Було проаналізовано декілька медичних інформаційних систем та баз даних, які використовуються у медичній галузі. Дослідження МІС включало детальне вивчення існуючих розробок. Вони мають свої переваги та недоліки, і, у зв'язку з цим, вирішено розробити спеціалізований додаток та базу даних для автоматизації процесів з метою покращення ефективності роботи закладу.

Цей розділ підтверджує актуальність і зростаюче значення інформаційних технологій. МІС – це один із видів інформаційних та телекомунікаційних систем, що є частиною загальної системи ЕСОЗ. Вона допомагає автоматизувати роботу лікарні. Подібні системи використовуються як для зберігання інформації про електронну медичну карту пацієнта (ЕКП), так і забезпечують звітно-організаційну роботу підрозділу.

ЕКП полегшує вирішення багатьох завдань:

- документування (збір, нагромадження, надійне зберігання) інформації;
- пошук (фільтрацію за певним критерієм) необхідної інформації;
- відстеження часової динаміки певних параметрів або процесів;
- дослідження ефективності роботи медичного закладу та ін.

Поставлена задача роботи та описано основні компоненти розроблюваної системи з використанням сучасних технологій та програмних засобів, що дозволить створити зручний та функціональний продукт з можливістю подальшого розширення та удосконалення.

РОЗДІЛ 2

ПРОЕКТУВАННЯ МЕДИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Технології проектування МІС припускають розробку системи в декілька стадій (рис. 2.1) [1].

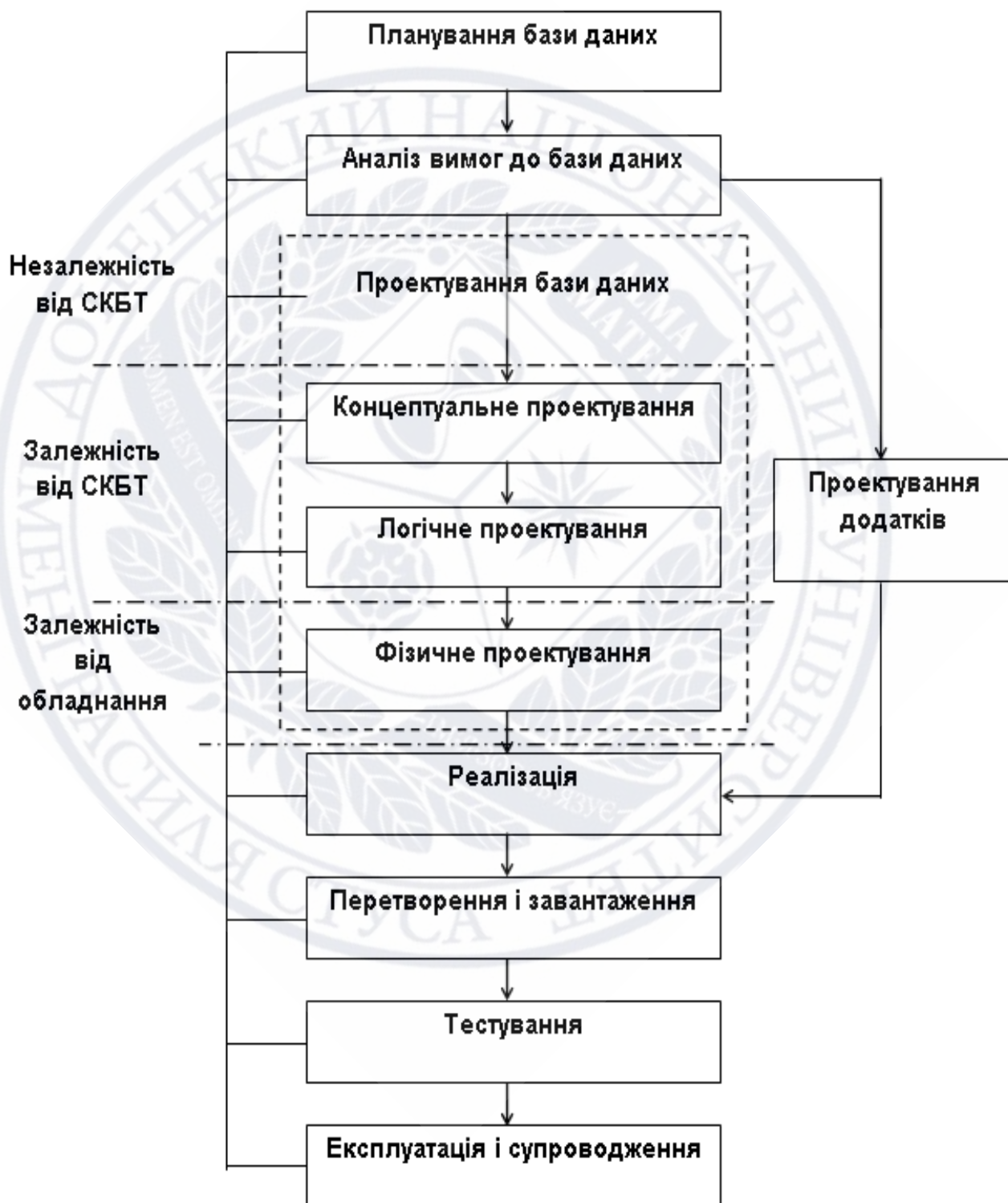


Рисунок 2.1 – Етапи створення МІС

1. Аналіз і планування задач МІС. Дана фаза є передпроектною. Вона охоплює дослідження та аналіз об'єкта плануємої МІС, визначення вимог до інформаційної системи, вимог до технічного завдання (ТЗ) на розробку програми та БД, а також формування техніко-економічного обґрунтування (ТЕО) проекту. У ТЗ включаються призначення МІС, вимоги до неї, та її підсистем, і обмеження на засоби проектування та експлуатації. В ТЕО приводяться економічні розрахунки, які підтверджують доцільність та необхідність розроблення МІС.

2. Проектування МІС – це процес логічного проектування. Оформлення технічного проекту повинно відповідати вимогам які були прописані раніше, складу затверджених автоматизованих функцій, а також складу допоміжних підсистем, структурі бази зберігання даних.

3. Реалізація – це процес фізичного проектування. Він включає в себе етапи розробки основної і допоміжних програм, наповнення БД інформацією, написання експлуатаційних інструкцій для персоналу, підготовка і оформлення робочого проекту. Даний етап ґрунтується на вимогах технічного проекту МІС.

4. Впровадження – це етап тестування системи та попередня експлуатація системи. Він включає комплексне налагодження підсистем, поетапне впровадження інформаційної системи по підрозділам установи, виконання приймально-здавальних робіт, і подальша передача програмного комплексу в експлуатацію, а також навчання персоналу [16, 17].

5. Експлуатація – це заключний етап який полягає в супроводі та модернізації системи. Він включає збір статистичних даних про функціонування МІС, анкетування персоналу, виправлення "багів", недопрацювань, модернізація системи відповідно до умов та параметрів які змінилися за час використання МІС.

На сьогоднішній день в Україні впроваджена електронна система охорони здоров'я (ЕСОЗ). Це інформаційно-телекомунікаційна система яка використовується з метою автоматизації управління електронними медичними даними, а також для ведення обліку медичних послуг установами охорони здоров'я які співпрацюють з Національною службою здоров'я України (НСЗУ).

Архітектурою ЕСОЗ є гібридна модель, яка представлена у вигляді двокомпонентної системи. В ній використовуються декілька медичних інформаційних систем (МІС) розроблених приватними структурами, та єдина центральна база даних (ЦБД), яка є державною власністю [2, 4, 5].

Держава встановлює стандарти використання і правила розробки систем для ЕСОЗ (рис. 2.2), та забезпечує охорону, зберігання та цілісність даних, а МІС територіального та державного рівня гарантують забезпечення своєчасного якісного надання своїх сервісів та послуг користувачам систем.



Рисунок 2.2 – Ключові ролі в ЕСОЗ [8]

Автоматизація процесів формування та передачі даних дозволяє налагодити зв'язок з пацієнтами, ЛПЗ та ЦБТ, що надає можливість закладам виконати більшу кількість звернень, зменшити час їх обробки та упорядкувати роботу підрозділів, спростивши процедуру формування списку пацієнтів на прийом.

2.1 Структура інформаційної системи

Інструментом для оптимізації роботи установ стали різні медичні програми обліку, які дозволили налагоджувати бізнес-процеси в організаціях найбільш зручним для них способом.

Медичні інформаційні системи класифікують з використанням різних ознак [8].

За критеріями автоматизації процесів збору й обробки даних МІС діляться на автоматизовані й автоматичні. В перших частина операцій збирання та обробки даних виконується людиною. Другі припускають відторгнення персоналу з процесів маніпулювання інформацією.

За критеріями типу інформаційної бази системи поділяються на, такі що оперують даними, та такі які оперують знаннями. Останні відносяться до експертних систем. Їх використання істотно ґрунтується на знаннях які отримані від людини-експерта, а результат роботи близький до результатів діяльності аналітиків-експертів.

За критеріями виду розв'язуваних задач інформаційні системи діляться на три групи:

- автоматизовані системи управління (АСУ) – керуючі;
- системи автоматизованого пошуку (САП) та вимірювальні системи (ВС) – інформаційно-довідкові системи;
- системи діагностики, прогнозування, моніторингу – інформаційно-логічні системи.

МІС також класифікують відповідно багаторівневій структурі галузі охорони здоров'я. Це ієрархічний принцип за яким існує чотири рівня розподілу:

- рівень лікарів різного профілю – базовий (клінічний);
- рівень поліклініки, стаціонару, диспансеру, станції швидкої допомоги – медичний заклад;
- рівень профільних, спеціалізованих медичних служб, а також регіональні органи керування – територіальний;
- рівень державних закладів та органів управління – державний.

Кожен рівень МІС в свою чергу класифікується за функціональним принципом, у відповідності задачам та цілям які вирішуються системою.

Розроблювана інформаційна система експертного відділення відноситься до автоматизованих інформаційно-довідкових систем базового класу що

оперують даними та знаннями.

МІС базового рівня репрезентовані медико-технологічними ІС, тобто комплексами підтримки технологічних інформаційних процесів на клінічному рівні. Такі системи використовуються для інформаційного забезпечення прийняття рішень пов'язаних з професійною діяльністю лікарів різного профілю. Основна мета системи – комп'ютерна інформаційна підтримка роботи лікаря, лаборанта, реєстратора, ін. Ці МІС дозволяють покращити якість роботи первинної ланки – профілактичної, лікувально-діагностичної, а також формування аналітично-статичних звітів та форм, що особливо актуально в умовах дефіциту кваліфікованих спеціалістів і часу при масовому обслуговуванні.

Медичні інформаційно-довідкові системи розроблюваного класу призначені для пошуку і формуванні зрізів медичних даних відповідно до запиту користувача. Масиви даних таких систем вміщують довідкову медичну інформацію різного плану.

Структура МІС яка відповідає заявленим вимогам представлена на рис. 2.3.

Медичний аспект подібних систем полягає:

- у належній підготовці знань та даних медичного характеру – це стосується стандартизації та єдності термінології, формалізація понять;
- у створенні відповідного інтерфейсу до загальної структури інформаційної БД;
- у побудові математичної моделі медико-біологічних, фізіологічних і патологічних процесів, та ін.

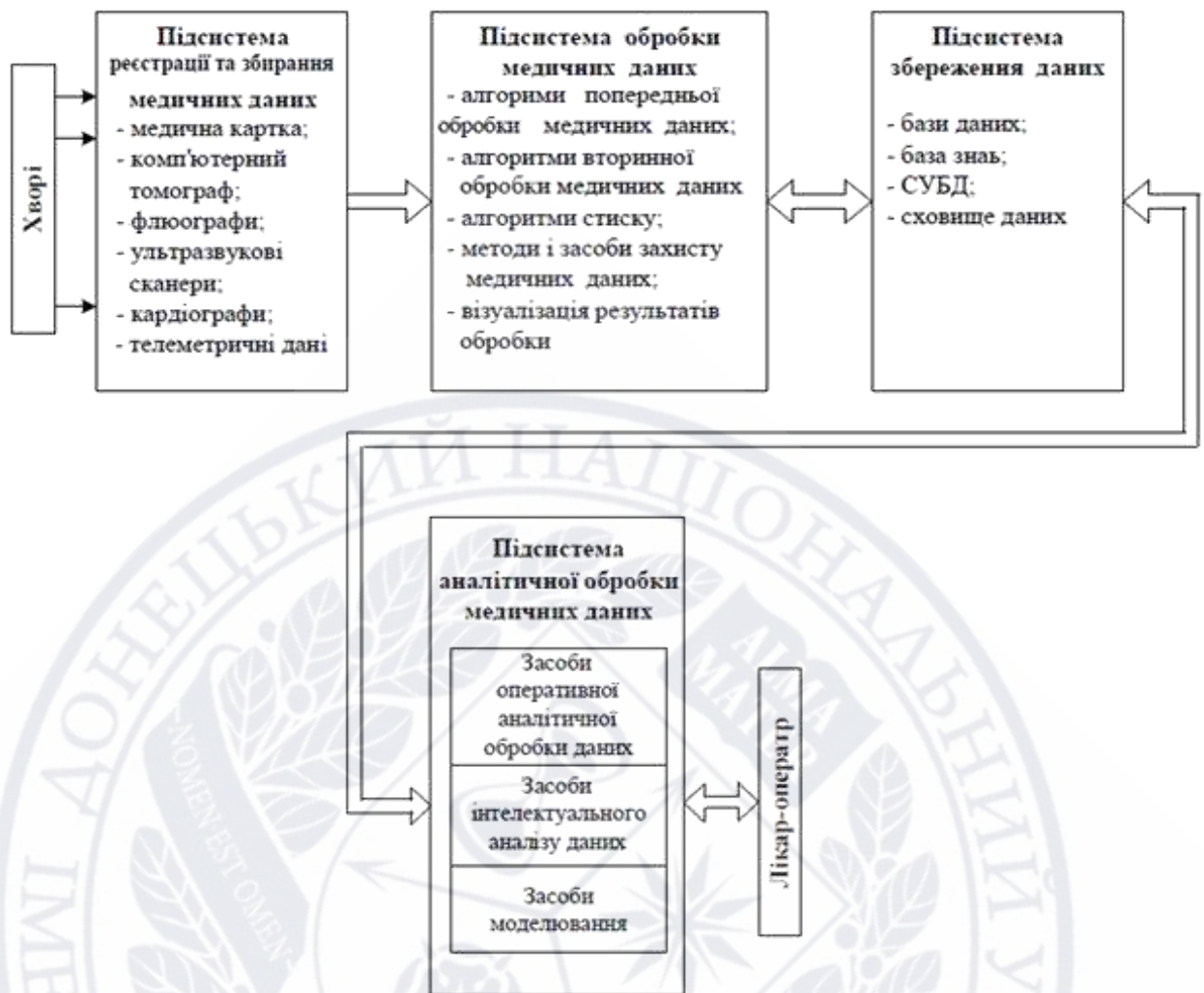


Рисунок 2.3 – Загальна архітектура МІС [9]

2.2 Проектування бази даних інформаційної системи

Розробка теоретичних моделей будови бази даних і знань що дозволять вирішити відповідні медичні задачі, а також відповідна програмно-апаратна реалізація такої ІБД, що ґрунтується на розроблених моделях представляють собою технічний аспект задачі [10, 11, 12].

Технологічний аспект полягає у приведенні у відповідність побудованої технічної системи до технологічної моделі лікувально-діагностичного процесу (метод вбудовування ІС у лікувально-діагностичний процес).

Психолого-педагогічний аспект полягає у належній підготовці персоналу медичного закладу.

Описані критерії МІС накладають вимоги на проектування таких систем –

кваліфікування складових елементів системи, а саме – програмні модулі, архітектура системи, компоненти та інтерфейси, вхідна і вихідна інформація. Розглянемо основний критерій, задля якого і виконується проектування МІС – база даних (рис. 2.4).



Рисунок 2.4 – Структура бази даних МІС

Проектування бази даних [11, 12] зводимо до розбиття всього потоку інформації, яку опрацьовують та аналізують співробітники, на блоки (таблиці), що містять опис кожного з аспектів даних. Це дозволить визначити яка інформація є основною, а яка допоміжною і її достатньо занести в таблиці-довідники. В залежності від розподілу інформації та від взаємозв'язків даних з різних таблиць створюємо модель БД [11, 12], використовуючи наступні умови: “забезпечення цілісності даних” “каскадне оновлення”, “каскадне видалення”, а також враховуючи співвідношення залежних полів з різних таблиць: “один до одного”, “один до багатьох”.

Для досягнення поставленої мети в програмі розроблено наступні модулі:

- перший – відповідає за інтерфейс з користувачем;
- другий – виконання адміністративних функцій з БД;
- третій – інформаційно-експертний, формування даних та аналіз;
- четвертий – формування звітності роботи підрозділу та вибірки і

фільтрації даних.

Згідно цієї структури інтерфейс користувача розділяємо на наступні блоки:

- блок введення та редагування бази даних (БВР);
- блок формування вихідних форм (БВФ);
- блок формування статистичних звітів (БСЗ);
- блок службових команд (БСК).

БВР реалізовано у вигляді графічної оболонки (GUI) і використовується для введення паспортних і особистих даних, медичної інформації – електронної медичної карточки особи.

БВФ та БСЗ функціонально однотипні і використовують методи фільтрації і вибірки для формування затверджених звітних форм, або необхідної аналітичної інформації. Взаємодії з користувачем також відбувається за допомогою GUI.

За допомогою блоку службових команд відбувається налаштування програми та взаємодія з базою даних в цілому: копіювання, об'єднання декількох БД чи розділення на частин, формування нової структури.

В табл. 2.1 наведені для порівняння базові параметри найбільш часто використовуваних систем керування базами даних (СКБД) [13, 17].

Для реалізації проектуємої інформаційно системи з множини існуючих систем управління базами даних – *FoxPro*, *Oracle*, *Paradox*, *PostgreSQL*, *Karat*, *Microsoft Access* та ін. – вибрана система *InterBase*, яка побудована на версійній крос-платформній [13, 14] архітектурі зберігання/читання/запису даних.

Цей підхід має ряд переваг над блокуючими СКБД:

- для відновлення баз даних *Interbase* після системних порушень роботи не потрібно підтримувати журнал транзакцій (*log transaction*);
- клієнти які зчитують дані ніколи не блокують клієнтів які виконують запис.

Таблиця 2.1 Основні характеристики СКБД

Характеристика	dBASE	MS SQL Server	Oracle	InterBase	MS Access
некомерційна необмежена версія	-	-	+	+	-
адміністрування	складне	просте	складне	просте	просте
швидкодія	низька	висока	дуже висока	висока	середня
підтримка технології клієнт-сервер	-	+	+/-	+	-
відновлення баз даних			+/-	+	-
виправлення помилок	-	+	+	+	-
підтримка багатопроцесорної архітектури	-	+	+/-	+	-
обмеження розмірів полів та записів	+	-	-/+	-	-
обмеження розмірів бази даних	+	-	-/+	-	+
версійна архітектура	-	-	+	+	-
резервне копіювання на "льоту"	-	+	+	+	-
використання мов програмування	обмежене	специф.	обмежене	без обмежень	специф.
сфера використання	особиста, невеликі, середні, великі підприємства	середні, великі підприємства	великі підприємства	особиста, невеликі, середні, великі підприємства	особиста, невеликі підприємства
можливість додавати стовпці, змінювати назву, типи даних для view без їх знищення	+	+	+	+	-

InterBase – реляційна система управління базами даних клієнт-серверної архітектури, розробляється компанією *Embarcadero*, з'явилася в середині 1980-х років. *InterBase* відрізняється від інших СКБД малим розміром, нескладними правилами адміністрування, та архітектурою типу multi-generational.

Вибір цієї СКБД ґрунтується на таких перевагах:

- 1) повнофункціональна реляційна система управління базами даних, яку можливо вбудовувати в розроблювані програмні продукти;
- 2) відрізняється потужністю, низькою вартістю, високою продуктивністю;
- 3) не вимагає адміністрування, просто встановлюється, розгортається і масштабується;

- 4) пропонує вбудовані можливості захисту даних;
- 5) забезпечують збереження даних навіть у разі збою сервера;
- 6) підтримує всі основні мови програмування, в тому числі Java, C, C++, .NET, Delphi, Object Pascal, PHP і Ruby;
- 7) інтегрована в такі засоби розробки додатків, як RAD Studio, Delphi;
- 8) дозволяє розгортати додатки, які безпосередньо звертаються до DLL-бібліотекам ядра InterBase, при цьому не запускається окремий процес сервера бази даних, а ядро бази даних функціонує в області процесу додатків, споживаючи вкрай мало ресурсів;
- 9) можливість підключення до бази даних InterBase через ADO.NET або ODBC;
- 10) мультиплатформеність – доступна для Android, iOS, Windows і OS X;
- 11) пакетне виконання SQL запитів дозволяє зменшити мережевий трафік і збільшити продуктивність;
- 12) код InterBase став основою вільно розповсюджуваної системи управління базами даних Firebird, яка зберігає всі її переваги.

Ще одним з критеріїв вибору цієї БД є необхідність подальшого інтегрування з існуючими пакетами програмного забезпечення які використовуються в медичних установах – "МедСтат" та "Кадри", в яких використовується даний тип баз даних.

Обрана СКБД, розрахована на управління заздалегідь визначеною структурою інформації й вирішення цілком певного й обмеженого кола завдань, що дозволяє майбутнім користувачам не відволікатися на вивчення питань пов'язаних з роботою баз даних і засобами керування ними.

В процесі створення баз даних вибір моделі інформаційної бази в першу чергу залежить від об'єкту використання, а також від етапу впровадження. Модель даних завжди повинна бути орієнтована на структуру закладу. Якщо структура БД не буде якісно прорахована, то це може вплинути на ефективність використання інформації.

В БД кількість інформації в одному записі може набувати великих об'ємів

і при аналізі не завжди вся потрібна, тому розділимо її на три таблиці пов'язаних зв'язками між сутностями (рис. 2.5).

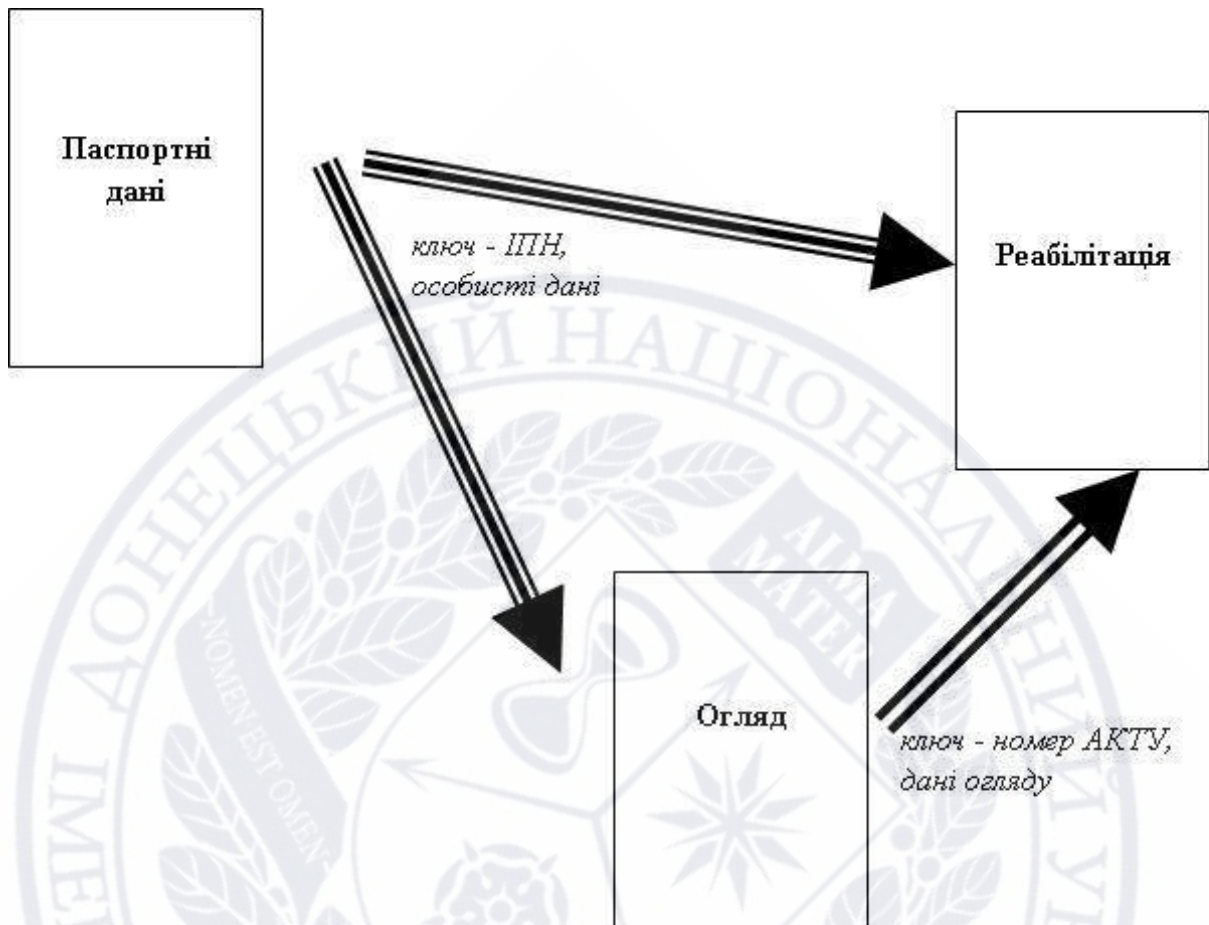


Рисунок 2.5 – Модель бази даних

Ця модель відзеркалює інформаційно-логічний рівень узагальнення, що пов'язаний з формалізацією об'єктів предметної області, взаємозв'язків між собою, та їх властивостями.

В першій таблиці будуть зберігатись паспортні дані, адреса, професія, освіта, та ін. В якості первинного ключового поля використаємо індивідуальний податковий номер особи (ІПН, для осіб в яких він відсутній будемо генерувати особисту цифрову послідовність) як унікальний елемент, що залишається для громадян незмінним.

В другу таблицю записуються результати медичного огляду особи, висновки роботи комісії експертного відділення, параметри огляду (первинний/повторний, мета, місце проведення, тощо). В якості первинного ключа використовується поле номер запису (статистичного талону, акту).

Оскільки звернення до комісії особою може бути неодноразовим, то для її ідентифікації використовують первинний ключ 1-ї таблиці. При подальших зверненнях за ним проводиться пошук необхідної інформації.

В третій таблиці накопичуються дані реабілітаційних рекомендацій згідно медичних показників. В цій таблиці використовуються два ключових поля: ПІН та номер Акту. Ключі не обов'язково є самостійними і в даному випадку вони будуть запозиченими.

Серед полів таблиць бази даних існують поля в яких дані для різних записів однакові (можуть повторюватись): імена, назви міст, вулиць, параметри огляду, медичних закладів, тощо. Тому, для полегшення роботи, зведення до мінімуму помилок вводу і різного написання даних, при заповненні бази даних будемо використовувати довідники – додаткові, неосновні таблиці. Крім того, це приведе до зменшення об'єму вихідного файлу за рахунок зберігання кодів елементів (індексів), а не самих параметрів, і до деякої кодифікації інформації.

Інформація про структуру БД представляється з використанням графічних діаграм (рис. 2.6). Вона побудована за моделлю "сутність - зв'язок" (ER-модель, ERM). Серед переваг подібної моделі наступні критерії:

- досить проста;
- однозначне визначення понять;
- використання природної мови;
- легкість для розуміння.

На користь даного рішення вплинули такі можливості СКБД:

- кожен об'єкт БД в ERD діаграмах це іменований об'єкт, з великою частотою відображається в таблицях СКБД;
- інформація представлена у вигляді переліку атрибутів що фіксують окремий сегмент інформації (заголовок меню), об'єкта інформації (безпосередньо меню);
- асоціація між суб'єктами керування.

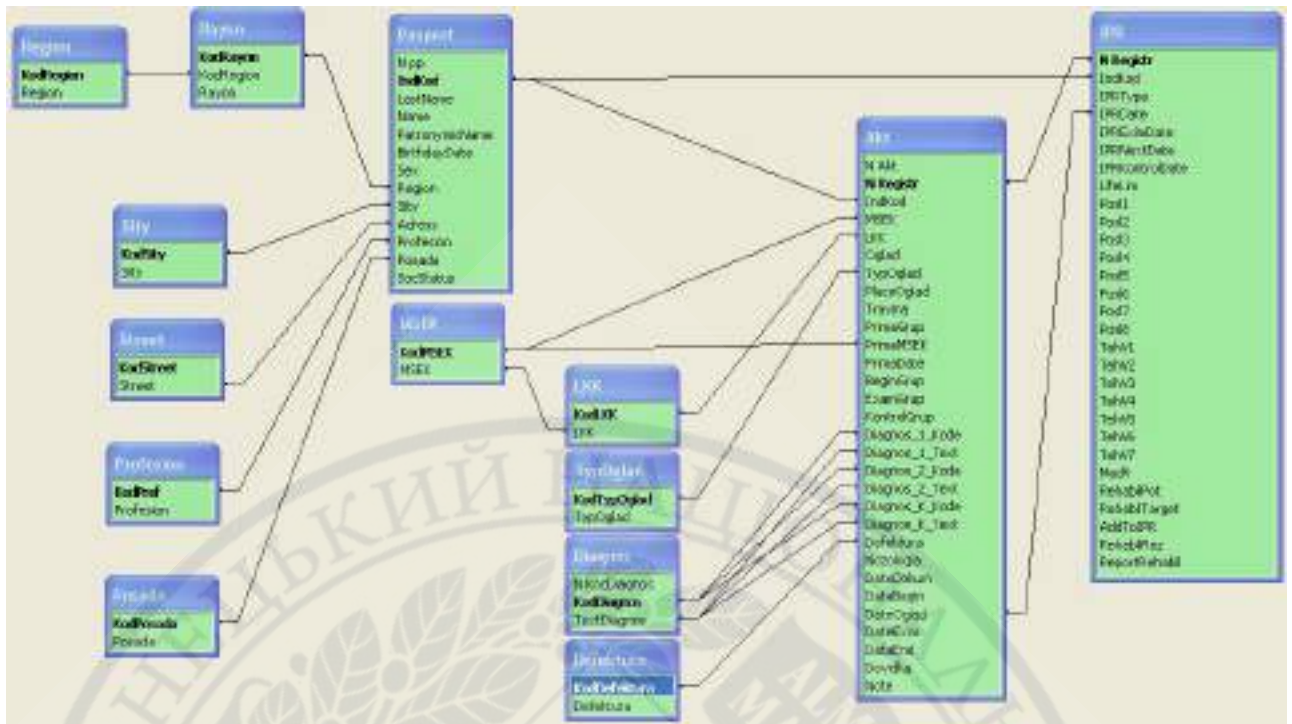


Рисунок 2.6 – Основні таблиці бази даних

В процесі виконання ПЗ можливе створення тимчасових таблиць, поля яких частково є полями основних таблиць, а частково вираховуваними..

2.3 Проектування експертного блоку

В медицині одне з головних питань – це необхідність обробки великих обсягів даних для прийняття рішень, що впливає на якість та оперативність діагностування, зменшення медичних помилок. Системи підтримки рішень використовують різноманітні технології для полегшення прийняття медичних рішень, зокрема – алгоритми машинного навчання, статистичні методи, нечітку логіку (fuzzy logic) та інші сучасні підходи.

Дані для моделювання складних та нечітких процесів діагностування, класифікації симптомів та клінічних досліджень не завжди чітко визначені, та можуть бути характерними для кількох різних захворювань. Методи fuzzy logic надають інструмент на основі нечітких або суперечливих даних для визначення ймовірності можливих діагнозів, та вибрати серед них найбільш вірогідний [15].

Вхідні (початкові) медичні дані бувають двох видів:

- Структуровані дані: лабораторних аналізи, клінічні вимірювання,

діагнози, параметри пацієнтів.

- Неструктуровані дані: такі як медичні зображення, текстові нотатки лікарів, результати тестів та результати медичних обстежень.

У медичних системах структуровані дані зазвичай використовуються для різних завдань: від прогнозування ризику захворювань до оцінки ефективності лікування. Неструктуровані для уточнення діагнозів, рекомендацій лікування.

Вихідні дані – діагноз: ймовірне захворювання, яке може бути поставлене на основі вхідних даних. Вони визначається (2.1) через класифікацію на основі моделі нечіткої логіки.

$$D = \sum_{i=1}^n w_i * f_i(x) + b, \quad (2.1)$$

де,

D – показник ймовірності конкретного діагнозу;

x – множина вхідних даних;

w_i – вага i -го параметра;

f_i – функція перетворення значення параметра, що нормалізує його;

b – базова ймовірність діагнозу без впливу даних (константа).

Система нечіткого логічного висновку (НЛВ) [17] приймати вхідні дані про пацієнта (вік, стать, симптоми, результати аналізів) та моделює підпорядкування $Y = f(X)$ з використанням нечіткої логіки, змінна Y – вектор вихідних змінних, змінна X – вектор вхідних змінних (рис. 2.7).



Рисунок 2.7 – Структурна схема системи НЛВ [17]

Функції елементів системи наступні:

- фазіфікатор використовується для перетворення фіксованого вектору вхідних змінних X з використанням функцій належностей у вектор нечітких множин;
- в базі знань зберігається інформацію про функцію $Y = f(X)$ у вигляді нечітких правил “якщо - то”;
- у блоці нечіткого логічного висновку передбачається вихідних вектор нечітких значень змінних, він відповідний до вхідного вектору нечітких значень змінних;
- у дефаззіфікаторі відбувається перетворення вектору нечітких множин у звичайний скалярний вектор Y .

Розробка програми враховувала обмеження щодо ресурсів і обробки даних; неоднорідність медичної інформації та потреба в скороченні часу обробки. Медичну діагностику осіб стосовно яких проводиться експертиза в першу чергу починають з фізіологічних змін організму, що впливають на біофізіологічними показниками.

Нечітка логіка діагностування є методом цифрового процесу що імітує прийняття рішення людиною, з використанням багатозначної логіки. Експертний модуль МІС аналізує показники стану людини та прогнозує ступінь ризику наявності у пацієнтів захворювання з використанням алгоритмів нечіткої логіки, що є імітацією консилиуму лікарів.

Базою для формалізації семантичних знань відносно предметної області є спрямований граф (рис. 2.8) з позначеними вершинами і дугами. Такий граф називають семантична мережа. Її вершини його є конкретні форми об'єкт-поняття, об'єкт-характеристика, об'єкт-подія, дуги – це зв'язки між об'єктами у вигляді двомісних асоціацій (семантичні міжоб'єктні відношення)

Fuzzy logic заснована на правилах використання лінгвістичних змінних – термів (наприклад, «низький», «середній», «високий») – для прийняття рішень. Кожна лінгвістична змінна має відповідну фізичну величину, що визначає якісні ознаки станів досліджуваної системи (пацієнта) за набором термінів (станів та показників). Кожен з термінів належить до діапазону значень змінної. Ступінь

приналежності фізичної величини до терму рівне "1", всі інші значення приймаються у відповідності до вибраної функції належності. Тому потрібно розрахувати засоби перетворення нечітких значень до певних конкретних величин, тобто заміну лінгвістичних змінних числовими аналогами.

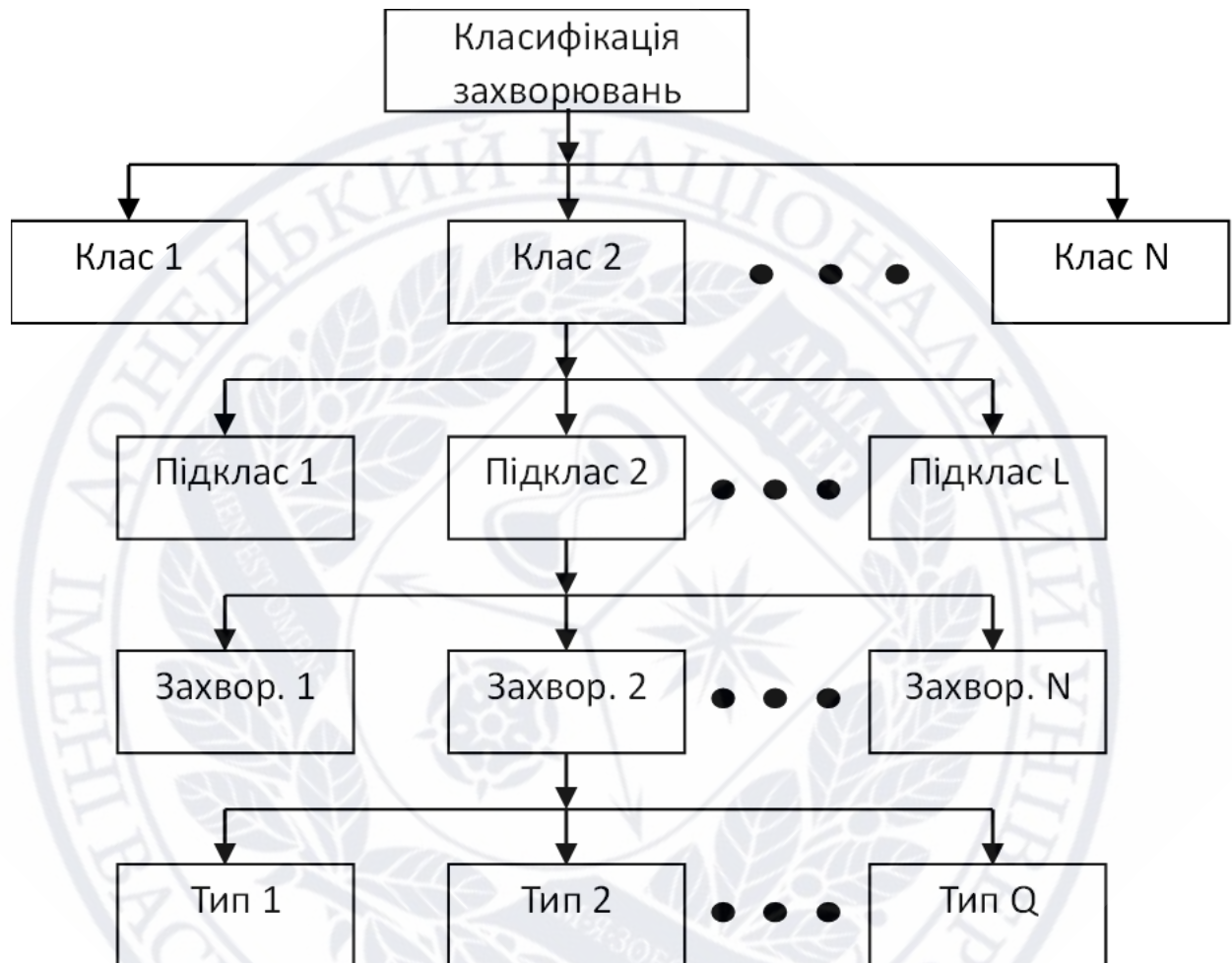


Рисунок 2.8 – Семантична модель класифікації захворювань

Набір правил нечіткої логіки для діагностування пацієнта розроблено на основі базових фізіологічних показників [18]:

- температура тіла (T);
- тиск крові (BP);
- пульс (HR);
- насичення крові киснем (SO₂);
- частота дихання (RR);

- рівень у крові цукру (BS), та ін.

Кожному набору показників належності (рис. 2.9) призначаємо семантичні терміни, а діагностичні висновки формуються у відповідності з виміряними значеннями для кожної групи ризику (див. табл. 2.2):

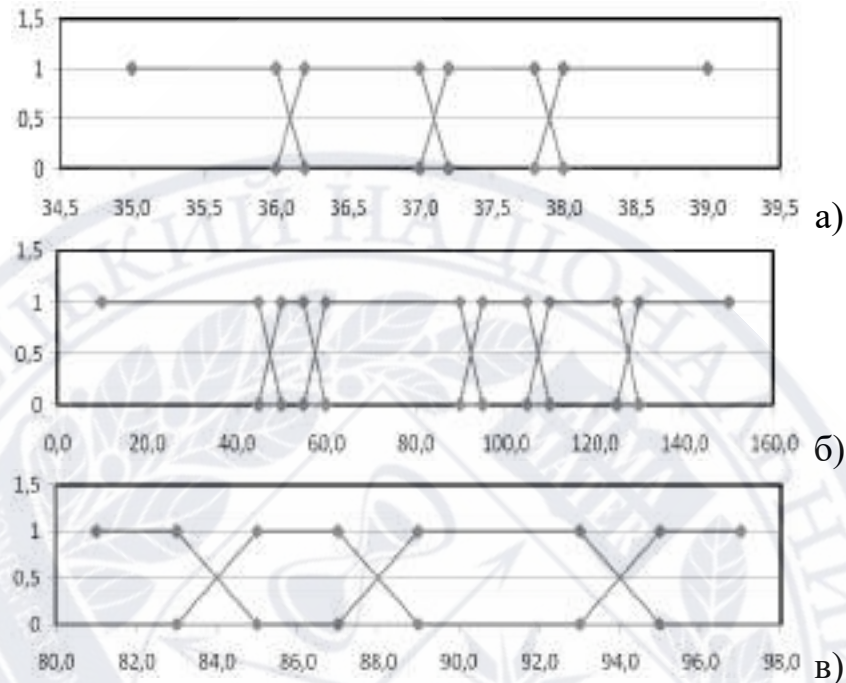


Рисунок 2.9 – Приклад семантичних термінів: а) $\mu(T)$; б) $\mu(HR)$; в) $\mu(SO_2)$

Правила для датчиків що визначають показники вихідної функції μ , визначаються за такими правилами (наприклад для температури (Temp) і пульсу (Pulse)),):

```
IF Temp is N0Temp=0 AND Pulse is N0Puls=0 THEN Risk is N0N0=0;
***** <<<<< ***** >>>>> *****
IF Temp is L2Temp=2 AND Pulse is H3Puls=3 THEN Risk is L2H3=5;
IF Temp is H2Temp=2 AND Pulse is H3Puls=3 THEN Risk is H2H3=5.
```

Таблиця 2.2 Ризик прояву захворювання відповідно до показників

Група	LOW3	LOW2	LOW1	NORM	HIGH1	HIGH2	HIGH3
Оцінка	3	2	1	0	1	2	3
		T<36.2 HR<50	45<HR<60	36<T<37.2 55<HR<100	37<T<38 95<HR<110	T>37.5 105<HR<130	HR>125
	SPO2<85 SBR<75	83<SPO2<90 70<SBR<85	87<SPO2<95 80<SBR<100	SPO2>93 95<SBR<190		SBR>185	
	BS<66	RR<15 63<BS<72		10<RR<30 70<BS<110	RR<25	106<BS<150	BS>140

Вихідна змінна (Y) показує ступінь ризику, який розділяється на групи.

Розглянемо вихідну функцію під час вимірюванні температури тіла (Temp) та частоти пульсу (Puls). Результат наведено на графіку, де рівень ризику вихідної змінної поділено на шість груп нечітких множин (рис 2.10):

- $NR0 = \text{FuzzySet}(0.0, 0.0, 0.0, 0.5);$
- $LR1 = \text{FuzzySet}(0.5, 1.0, 1.0, 1.5);$
- $LR2 = \text{FuzzySet}(1.5, 2.0, 2.0, 2.5);$
- $LR3 = \text{FuzzySet}(2.5, 3.0, 3.0, 3.5);$
- $HR1 = \text{FuzzySet}(3.5, 4.0, 4.0, 4.5);$
- $HR2 = \text{FuzzySet}(4.5, 5.0, 5.0, 5.0).$



Рисунок 2.10 Вихідна змінна (Y) для температури (Temp) і пульсу (Pulse)

Наприклад, низький кров'яний тиск $L2Puls=2$ і низька температура тіла $L2Temp=2$ вказує на сильну кровотечу з великою втратою крові, а доданий показник низького насичення крові киснем $L3SO2=3$ свідчить про пошкоджені легені.

Ваги значимості неструктурованих даних дозволяють уточнити не тільки діагноз, а і ступінь втрати працездатності – визначити групу інвалідності (дані носять рекомендаційний характер, встановлення – прерогатива експертної комісії).

Неструктурованих даних, розподілених за нозологіями, класами захворювань та травм [19], наприклад:

- для діабетичної ретинопатії (ДР):
 - поява “мушок” у полі зору,
 - розмитість і спотворення зображення,
 - труднощі з фокусуванням погляду,

- діабетична нефропатія,
- вагітність,
- фаза гормональної перебудови,
- підвищення проникливості судинної стінки,
- утворення мікротромбів і оклюзія капілярів;
- для травматичної ампутації:
 - кукси обох нижніх кінцівок на рівні нижньої третини стегон і вище,
 - кукси обох верхніх кінцівок на рівні плеча,
 - кукси двох нижніх кінцівок на рівні гомілки,
 - кукси верхніх кінцівок на рівні передпліччя,
 - коротка кукса стегна у разі неможливості протезування,
 - кукса стопи на рівні суглоба,
 - відсутність усіх фаланг чотирьох пальців кисті,
 - відсутність перших пальців обох кистей, та ін.;
- для ішемічної хвороби:
 - нейроциркуляторна дистонія легкого ступеня,
 - нейроциркуляторна дистонія середнього ступеня,
 - нейроциркуляторна дистонія тяжкого ступеня,
 - стенокардія першого функціонального класу,
 - стенокардія другого функціонального класу,
 - стенокардія третього функціонального класу.

Деякі з них уточнюють основний чи супутній діагноз та групу інвалідності (для діабетичної ретинопатії); деякі з додатковими компонентами системи моніторингу ЕКГ, КТ, холтер дозволять визначити діагноз (для ішемічної хвороби); а деякі мають ваги які не потребують структурованих даних для експертної оцінки (ампутації).

База даних під час формуванні медичного звіту об'єднує різні типи даних, враховує критичні показники особи та активізує процедури, пов'язані з наданням експертного висновку пацієнту.

Отримані в процесі експлуатації знання використовуються для подальшого навчання МІС, уточнення бази знань, корегування вагових коефіцієнтів.

Отже, проведений аналіз МІС дозволяє зробити висновок про необхідність інформаційних систем не тільки територіального та державного рівня, а також і локального – пристосованого під задачі, функції, можливості лікувальних закладів.

Бажано при проектуванні та розробці таких програм притримуватись концепції наявних програмних засобів, та враховувати базовий рівень комп'ютерної підготовки персоналу. Використовувані типи баз даних повинні бути сумісними з тими що використовуються в закладі, що дозволять адміністраторам МІС формувати зрізи даних які потребує заклад.

Проектуєма система призначена для початкової діагностики та експертної оцінки стану пацієнта з використанням методу нечіткої логіки що дозволяє прийняття рішень про втрату працездатності.

Основа метода це експертні знання, що відповідають правилам нечіткої логіки для прорахунку вхідних значень показників та прогнозування висновку обстеження, і можуть уточнюватись в процесі використання МІС. Розширення набору вимірюваних показників неструктурованими даними та знаннями потрібно для уточнення діагнозу та стану особи.

РОЗДІЛ 3

ОПИС ПРОГРАМИ

СКБД, які використовуються в сучасних умовах мають розвинутий інструментарій для роботи з базами даних. Однак в більшості випадків користувачу не потрібен такий потужний засіб, який ще й вимагає додаткових знань для використання. Тому виникають ситуації, при яких система не виправдовує очікувань хоча вона технічно ідеально спроектована. Найчастіше це пов'язано з недостатнім врахуванням в процесі проектування людського фактору. Вирішенням подібних задач є створення "вузьконаправлених" ПЗ для вирішення локальних задач в можливість використання персоналом з базовою комп'ютерною підготовкою, бажано з можливістю роботи у вибірково-опитувальному форматі.

Інтерфейс користувача (ІК) є важливою частиною будь-якого програмного продукту, що призначений для обробки даних в інтерактивному режимі. Цей модуль об'єднує всі елементи і компоненти пов'язані з введенням/виведенням даних, та впливають на взаємодію програмного засобу та користувача.

3.1 Початок роботи, введення даних

Програма розроблена на мові програмування Delphi, яка має кросплатформенну сутність, що дозволить її використання на різних пристроях.

Для полегшення, та спрощення роботи по використанню програмного забезпечення (ПЗ) розроблено графічний інтерфейс.

Він включає в себе наступні елементи:

- засоби, пристрої та технології для відображення інформації та введення даних;
- тип, формати інформація вводу/виводу, її коди;
- режими керування, користувач-інтерфейс;
- сценарії взаємодії користувача та програми;
- безпосередньо діалоги та форма зворотного зв'язку з користувачем;

- підтримка формування рішення для визначеної предметної області;
- документація та порядок експлуатації програми.

Для збереження інформації по роботі закладу використовується база даних InterBase.

ПЗ має модульну структуру з використанням основного тіла та підпрограм (класів).

Клас – це проект об'єкта. Головна мета в об'єктно-орієнтованому програмуванні це використання класів для створення об'єкту, і програмування стосується не об'єктів, а класів. Клас описує яким буде об'єкт, а не безпосередньо сам об'єкт. В результаті об'єкти та класи відображають одне одного.

Для реалізації функції створення електронних карток пацієнта, їх перегляду та редагування було створено клас, який містить відповідні до цілі методи.

Також окремі методи були створені і для класів адміністрування по роботі з даними пацієнтів, бази в цілому, а також формування звітів по роботі закладу.

В основному методі *NDI_Expert* відбувається підключення модулів-підпрограм (Unit & Forms), завантаження інформації з бази даних (БД), що складається з трьох основних таблиць, і декількох додаткових що містять довідкову інформацію та запуск основної форми (рис. 3.1):

```
program NDI_Expert;
```

```
uses
```

```
  Forms,
```

```
  frmMain in 'FORMS\frmMain.pas' {fmMain},
```

```
  hlIniFile in '..\components_01\HL\SOURCE\HLINIFILE.PAS',
```

```
  hlReport in '..\components_01\HL\SOURCE\HLREPORT.PAS',
```

```
  frmOptions in 'FORMS\frmOptions.pas' {fmOptions},
```

```
  okcancel_frm in
```

```
  '..\components_01\HL\SOURCE\FRAMES\OKCANCEL_FRM.PAS'
```

```
{OkCancel_frame: TFrame},
```

```
  frmGridFieldsVisiblity in 'FORMS\FRMGRIDFIELDSVISIBILITY.PAS'
```

```
{fmGridFieldsVisiblity},
```



```

    chackuncheck_frame in
'..\components_01\HL\SOURCE\FRAMES\CHACKUNCHECK_FRAME.PAS'
{checkuncheck_frame: TFrame},
    frmLogViewer in 'FORMS\frmLogViewer.pas' {fmLogViewer},
    frmEmployeTicket in 'FORMS\frmEmployeTicket.pas'
{fmEmployeTicket},
    hlApp in '..\components_01\HL\SOURCE\HLAPP.PAS',
    frmExportInDB in 'FORMS\frmExportInDB.pas' {fmExportInDB},
    frmDBView in 'FORMS\frmDBView.pas' {fmDBView},
    frmForma14 in 'FORMS\frmForma14.pas' {fmForma14},
    frmJornal in 'FORMS\frmJornal.pas' {fmJornal},
    frmDviyniky in 'FORMS\frmDviyniky.pas' {fmDviyniky},
    frmExport2008 in 'FORMS\frmExport2008.pas' {fmExport2008},
    frmExport2012 in 'FORMS\frmExport2012.pas' {fmExport2012},
    frmDviyniki in 'FORMS\frmDviyniki.pas' {fmDviyniki},
    frmIPRTicket in 'FORMS\frmIPRTicket.pas' {fmIPRTicket},
    frmNewPeriod in 'FORMS\frmNewPeriod.pas' {fmNewPeriod},
    frmGlueBase in 'FORMS\frmGlueBase.pas' {fmGlueBase},
    frmIPRMain in 'FORMS\frmIPRMain.pas' {fmIPRMain},
    frmF_SUM in 'FORMS\frmF_SUM.pas' {fmF_SUM},
    frmNDI in 'FORMS\frmNDI.pas' {fmNDI},
    frmNDI_1 in 'FORMS\frmNDI_1.pas' {fmNDI_1},
    frmNDI_In in 'FORMS\frmNDI_In.pas' {fmNDI_In},
    frmArhivView in 'FORMS\frmArhivView.pas' {fmArhivView},
    frmArhivExport in 'FORMS\frmArhivExport.pas' {fmArhivExport},
    frmExportTestBase in 'FORMS\frmExportTestBase.pas'
{fmExportTestBase};
begin
    hlLogger.AddString(false, '');
    hlLogger.AddString('NEW SESSION');
    Application.Initialize;
    Application.Title := 'NDI_Expert';
    Application.CreateForm(TfmMain, fmMain);
    Application.Run;
end.

```

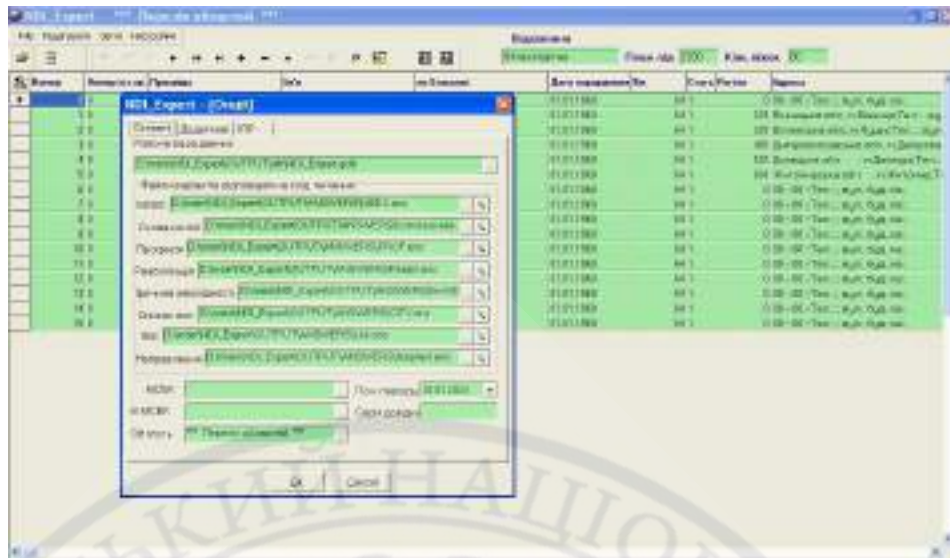


Рисунок 3.1 – Екран програми

Оскільки використання програми передбачається персоналом, який володіє лише базовими навиками роботи з комп'ютерами, то програмна оболонка повинна мати простий інтуїтивний інтерфейс.

Для спрощення використання команд розроблене графічне середовище з максимально можливим використанням піктограм для вводу команд за допомогою маніпулятора "мишка", в той же час, вони доступні для активації з клавіатури.

Ведення даних відбувається в інтерактивних формах, після закриття яких інформація зберігається у базі даних в новому або редагованому записі. Оновлення даних можливо як по командам меню, так і при відкритті нового чи редагуванні існуючого запису, або при завершенні роботи програми.

Налаштування доступу до довідкових таблиць і режимів роботи відбувається в МЕНЮ "Налаштування"- "Опції" (див. рис. 3.1).

3.2 Графічний інтерфейс

Як уже зазначалось, використання програми реалізовано у вигляді GUI. Він містить головний екран (див. рис. 3.1), що складається із стрічки "МЕНЮ" (MenuBar), стрічки піктограм основних команд (ToolDock), поля записів бази даних (DBGrid) та статусної стрічки (StatusBar).

Головне меню містить 4 розділи, кожен із розділів містить додаткові

підпункти:

- **FILE** – робота з базою даних в цілому: створення нового файлу БД, завантаження існуючої БД, об'єднання БД окремих відділень в загальну базу, "склейка" частин БД набраних різними користувачами в єдину базу відділення;
- **РЕДАГУВАННЯ** – команди для окремих записів: створити, редагувати, видалити, копіювати запис, переміститись по базі даних "вверх"/"вниз" на один/десять записів або в початок/кінець, оновити БД, пошук/фільтрація даних;
- **ЗВІТИ** – формування звітної та аналітико-статистичної інформації;
- **НАЛАШТУВАННЯ** – зміна параметрів роботи програми, значень довідників.

Як уже зазначалось, основні часто використовувані команди винесені у вигляді піктограм на панелі меню.

В статусній стрічці відображається інформація про використовувану базу даних, її фізичне розміщення на диску, загальна кількість записів та номер активного запису.

В основному полі вікна розміщається інформація бази даних у вигляді таблиці записів. Вони доступні для перегляду і переміщенню як по окремим коміркам (полям) запису так і по записам в цілому за допомогою стрілок переміщення на клавіатурі, або маніпулятором "мишка", або меню команд.

Формування нового запису електронної карточки пацієнта чи виклик існуючої виконується відповідною командою. При цьому відбувається запуск підпрограми відображення форми вводу (рис. 3.2). Для вводу однотипних даних використовуються модулі типу "ComboBOX", "RadioBUTON", "CheckBOX", "DateTimePicker". Це дозволяє вибирати інформацію із готових варіантів, що збільшує ефективність роботи, зменшує варіанти помилок і занесення тої самої інформації але в іншому вигляді (великі/малі літери, скорочення, українські/англійські літери тощо).

а) б) в) г) д) е)

Рисунок 3.2 – Сторінки форми введення даних

Карточка вводу складається з шести сторінок, інформація на яких згрупована у відповідності до типу даних:

- паспортно-анкетні дані;
- попередні медичні дані сторони яка направила пацієнта;

- результати клінічних досліджень;
- симптоми захворювань та стану пацієнта;
- результати-висновок обстеження у відділенні;
- епікриз при виписуванні, на якому розміщаються кнопки формування результируючих документів затверджених Міністерством охорони здоров'я форм – епікризу, виписки, заключення

Дані у форми можна заносити в довільному порядку. Допускається можливість доповнення вже існуючих записів.

Як вже відмічалось, БД складається з трьох таблиць в якості ключових елементів в яких виступає РНОКПП (раніше позначався як ПІН) особи (для таблиці анкетних даних – первинний). В компоненті DBGrid ці таблиці об'єднуються для відображення в табличній формі і для індексації використовується поле номеру карточки пацієнта який є унікальним і обчислювальним та складається з даних року і порядкового номеру. цей ключ є первинним для таблиць медичних обстежень та реабілітаційних заходів.

3.3 Попередня експертна оцінка стану пацієнта

Для прискорення проведення експертизи особи яка направлена на встановлення інвалідності, та зменшення можливих некоректних висновків в програмі передбачено попередній аналіз стану особи в залежності від показників клінічних досліджень та оцінки його стану.

Внесення результатів аналізів (рис. 3.3) проводиться у відповідних полях. Деякі параметри вибираються із можливих варіантів, що пов'язано із стандартизацією цих параметрів та однозначно термінів (наприклад: колір – світло-жовтого, жовтий, янтарного темно-жовтий, коричневий; запах – сильний, виражений, не різкий, слабкий), інші вводяться безпосередньо у вікні редагування. В цьому випадку для уникнення помилок внесення інформації відбувається у відповідності із вбудованим шаблоном.

Карточка пацієнта

Анкетні дані | Попередній огляд | **Результати аналізів** | Результат обстежень | Епікриз

ЗАС

Колір

Білок

Цукор

Лейк.

Ерит.

Солі

№ та Дата аналізу

1 січня 2014 р. ▼

Рентген

Гінеколог

☒ Обстежено серологічно

ЗАК

ШОЕ

Лейк.

Ерит.

КП

НВ

е

п/я

с/я

м

тр

№ та Дата аналізу

1 січня 2014 р. ▼

Примітки

Дата прийому: 12 січня 2024 р. ▼ Дата огляду: 12 січня 2024 р. ▼ Лікар:

Дата виписки: 12 січня 2024 р. ▼

Рисунок 3.3 – Сторінка результатів клінічних досліджень

Стан пацієнта описується на відповідній сторінці з використанням тестових відповідей (рис. 3.4).

Інформація на цій сторінці згрупована в блоки у відповідності до нозологій захворювань. Потрібні значення вибираються із переліку варіантів які представлені у вигляді розкривного переліку стрічок "checkLastBox". Така реалізація надає користувачу можливість вибирати потрібні відповіді у будь-якій сукупності. Ця можливість доступна як для окремої нозології, так і при виборі інформації що належить різним типам захворювань. Така необхідність пов'язана з тим що ряд захворювань мають симптоматику в різних органах людини.

Карточка пацієнта

Анкетні дані | Попередній огляд | Результати аналізів | **Симптоми** | Результат обстежень | Епікриз

Офтальм 1

- ☐ сліпота праве око - 0,05
- ☐ сліпота праве око - 0,05-0,1
- ☐ сліпота праве око - 0,1
- ☐ стійкий повний птоз на од
- ☒ стійкий повний птоз на од

Офтальм 2

- ☒ осередки крововиливу
- ☐ підвищена чутливість до світла
- ☐ погіршення сутінкового зору
- ☒ порцшення відчуття кольор
- ☒ поява блискавок і відблиск

Хвороби вуха

- ☐ двобічна глухота повна
- ☐ двобічна глухота понад 20 дБ
- ☐ двобічна глухота понад 50 дБ
- ☐ двобічна глухота понад 70 дБ
- ☐ двобічна глухота понад 90 дБ

Кардіо

- ☐ анемія
- ☐ гіпертонічна хвороба
- ☐ гостра ревматична гарячка
- ☐ інфаркт міокарда
- ☐ ішемічна хвороба

Нервова системи / Психіка

- ☐ дегенеративні хвороби
- ☐ епілепсія
- ☐ поліневропатія
- ☐ розсіяний склероз
- ☐ хвороба Паркінсона

Внутрішні органи

- ☐ виразка шлунка
- ☐ відсутності однієї легені
- ☐ гастро-езофагеальний рефлюкс
- ☐ грижа
- ☐ хвороби ротової порожнини

Інфекційні хвороби

- ☐ ВІЛ з імунною декомпенсацією
- ☐ вірусний гепатит А
- ☐ вірусний гепатит В
- ☐ вірусний гепатит С
- ☐ мінінгококова інфекція

Новоутворення

- ☐ новоутворення головного мозку
- ☐ новоутворення гортані
- ☐ онкологія крові
- ☐ онкологія стравоходу
- ☐ пухлина щитовидної залози

Ендокринна система

- ☐ синдром вродженої йодної недостатності
- ☐ синдром Нельсона
- ☐ форми гіперпаратиреозу
- ☐ форми гіпотиреозу
- ☐ хвороби щитоподібної залози

Травми

- ☐ дефекти будови поздовжньої осі
- ☒ контузія
- ☐ травми внутрішніх органів
- ☐ травми ока
- ☐ травми хребта

Ампутації

- ☐ ампутація лівого передпл'яччя
- ☐ ампутація лівої кисті
- ☐ ампутація правої кисті
- ☐ ампутація правої кисті
- ☐ кукса лівого стегна

Вроджені аномалії

- ☐ агенезія
- ☐ аплазія
- ☐ гетеротопія
- ☐ гіперплазія або гіпертрофія
- ☐ гіпоплазія або гіпотрофія

Примітки

Дата прийому: 16 травня 2025 р. | Дата огляду: 16 травня 2025 р. | Лікар: ...

Дата виписки: 16 травня 2025 р.

Протокол | Відмінити

Рисунок 3.4 – Сторінка симптоматичних ознак

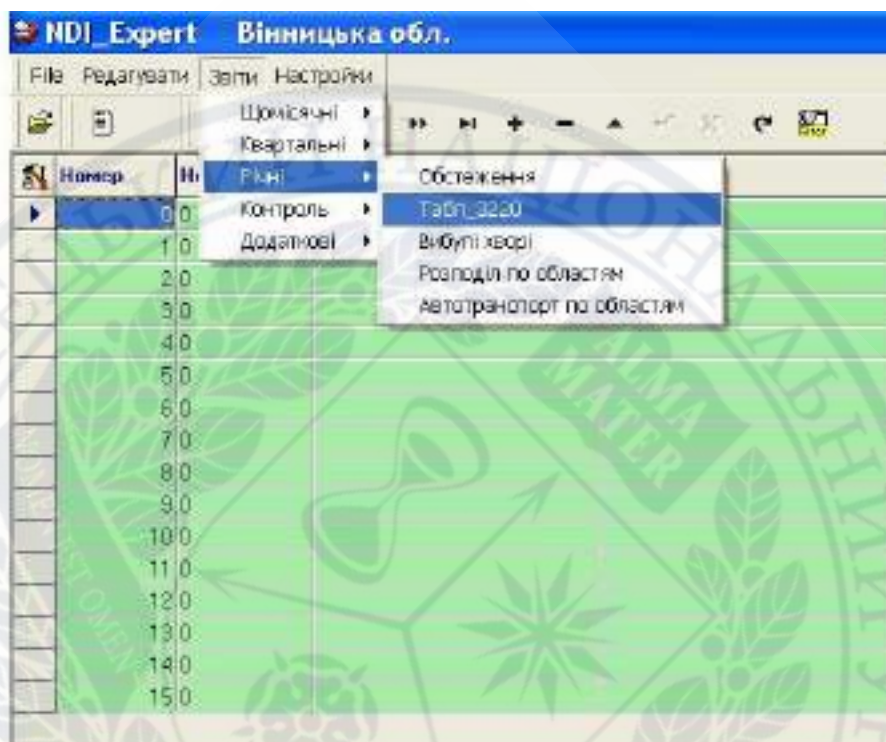
Наприклад для цукрового діабету характерне порушення зору – діабетична ретинопатія, а також захворювання нижніх кінцівок – діабетична стопа і ампутації пальців ноги. А це і свою чергу веде до наявності ускладненого діагнозу, а також до можливого підвищення групи інвалідності.

3.4 Формування звітів, статистичної інформації, фільтри даних

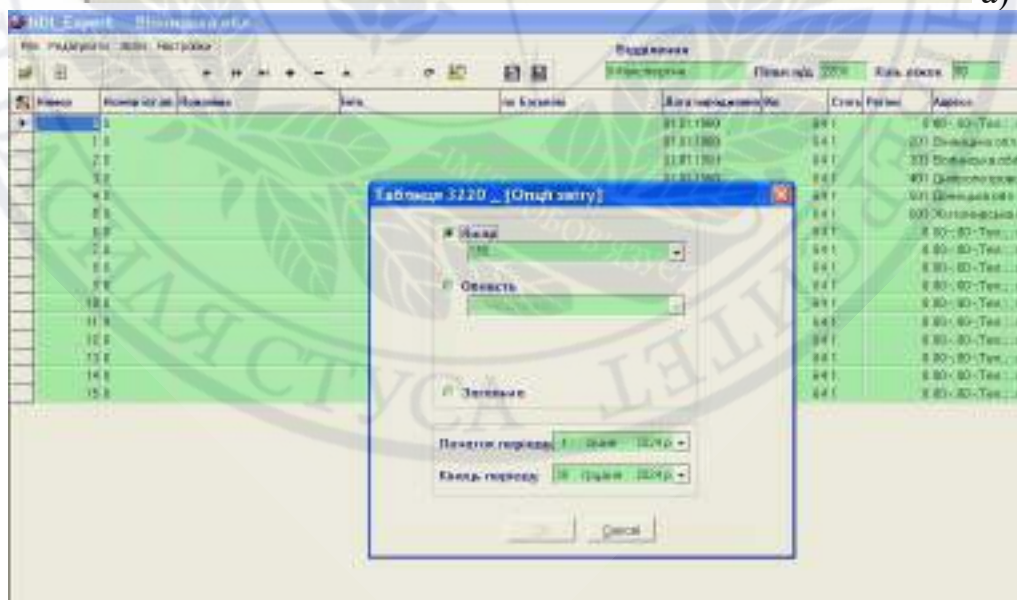
Оскільки в роботі відділення необхідно оформляти ряд звітних документів, вихідних форм, то в програмі передбачено проведення обчислень для автоматичного формування потрібних звітів, реєстрів, журналів (рис. 3.5). Для зручності перегляду і подальшої обробки цієї інформації вона надається у форматі файлу офісного пакету MS Office програми Excel (*.xls). Цей файл

електронних таблиць можна відкрити як в оригінальній програмі, так і в багатьох пакетах вільного поширення.

Також в програмі реалізована можливість відбору записів з необхідними даними (фільтрація, групування, рис. 3.6) та пошук значень в БД за вмістом окремих полів (рис. 3.7).



а)



б)

Рисунок 3.5 – Формування звітної інформації

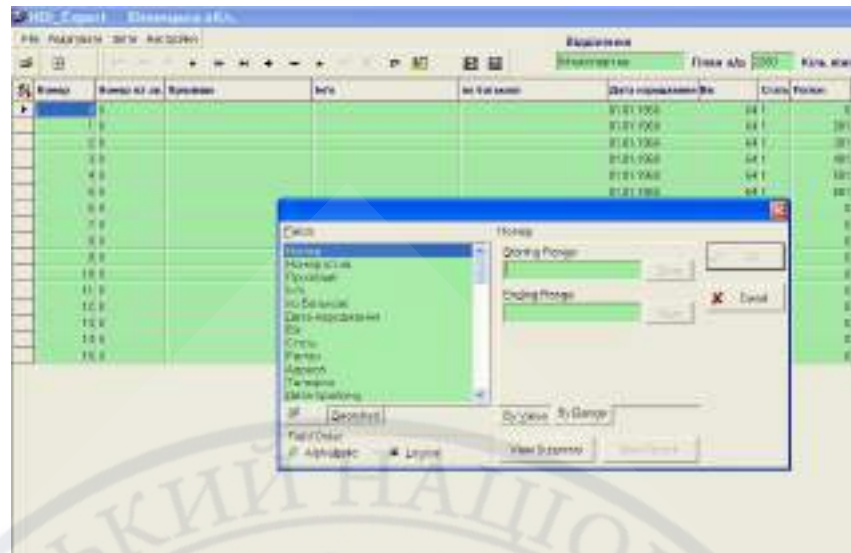


Рисунок 3.6 – Використання фільтру для бази даних

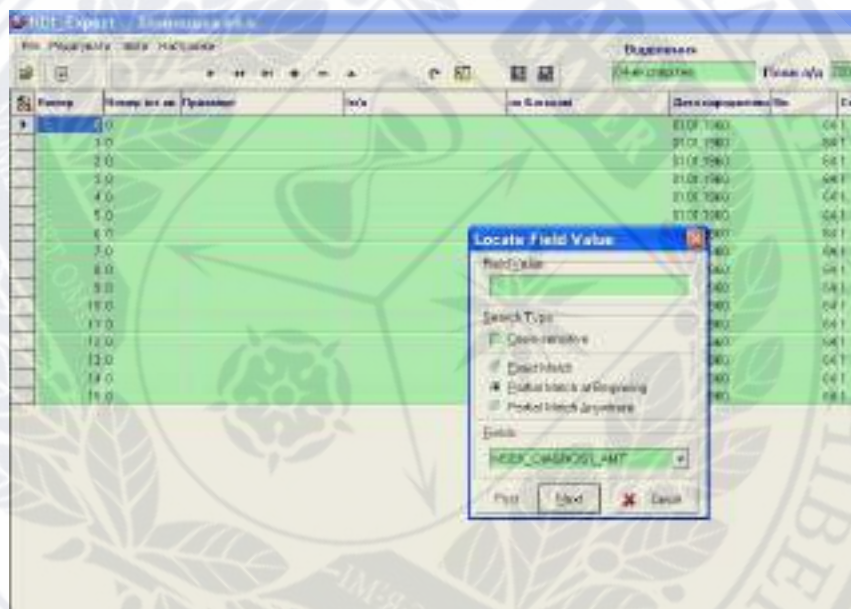


Рисунок 3.7 – Пошук інформації в базі даних

Для відслідковування дій користувача, до відповідних елементів прикріплені "слухачі", методи яких запускаються у відповідності до змiну стану об'єкта (натискання кнопки, введення тексту, наведенні чи натискання "мишки", тощо).

3.5 Керівництво користувача

Програма реалізована у вигляді графічної оболонки (GUI інтерфейс) побудованої за методом вікон: основного і допоміжних, які відкриваються за потреби.

За допомогою інсталятора програма розміщується на одному із розділів жорсткого диску, наприклад **d:**. При установці програми створюється робочий каталог **\NDI_Expert** в якому створюється чотири допоміжні підкаталоги:

d: \NDI_Expert \ANSWERS
d: \NDI_Expert \DB
d: \NDI_Expert \FORMY
d: \NDI_Expert \SYS

В головному каталозі розміщаються програма **NDI_Expert .exe**, файли ініціалізації (налаштувань) ***.ini** та ***.ini** і журнал роботи ***.log**. В каталозі **\ANSWERS** знаходяться файли довідників. Каталог **\FORMY** передбачений для зберігання шаблонів вихідних форм у форматі шаблонів EXCEL-файлів. Каталог **\SYS** це каталог шаблонів таблиць бази даних та допоміжних таблиць які використовуються при роботі програми. Сховище даних розміщується в каталозі **\DB**.

Запуск програми може бути виконаний за допомогою ярлика з Робочого Столу (DeskTop), або безпосередньо з каталогу **\NDI_Expert**. При першому запуску програми необхідно налаштувати основні параметри (див. рис. 3.1) про що буде повідомлено у **MessageBox** (рис. 3.8).

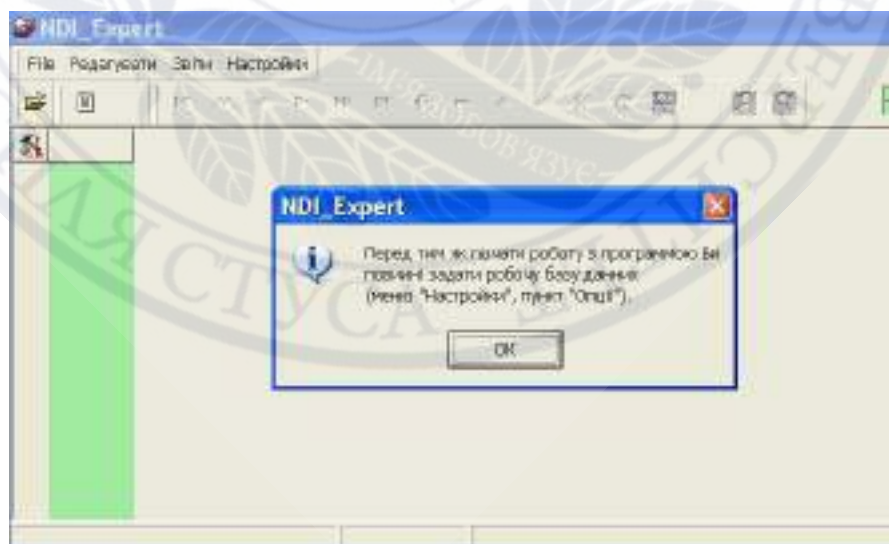


Рисунок 3.8 – Перший запуск, запит параметрів

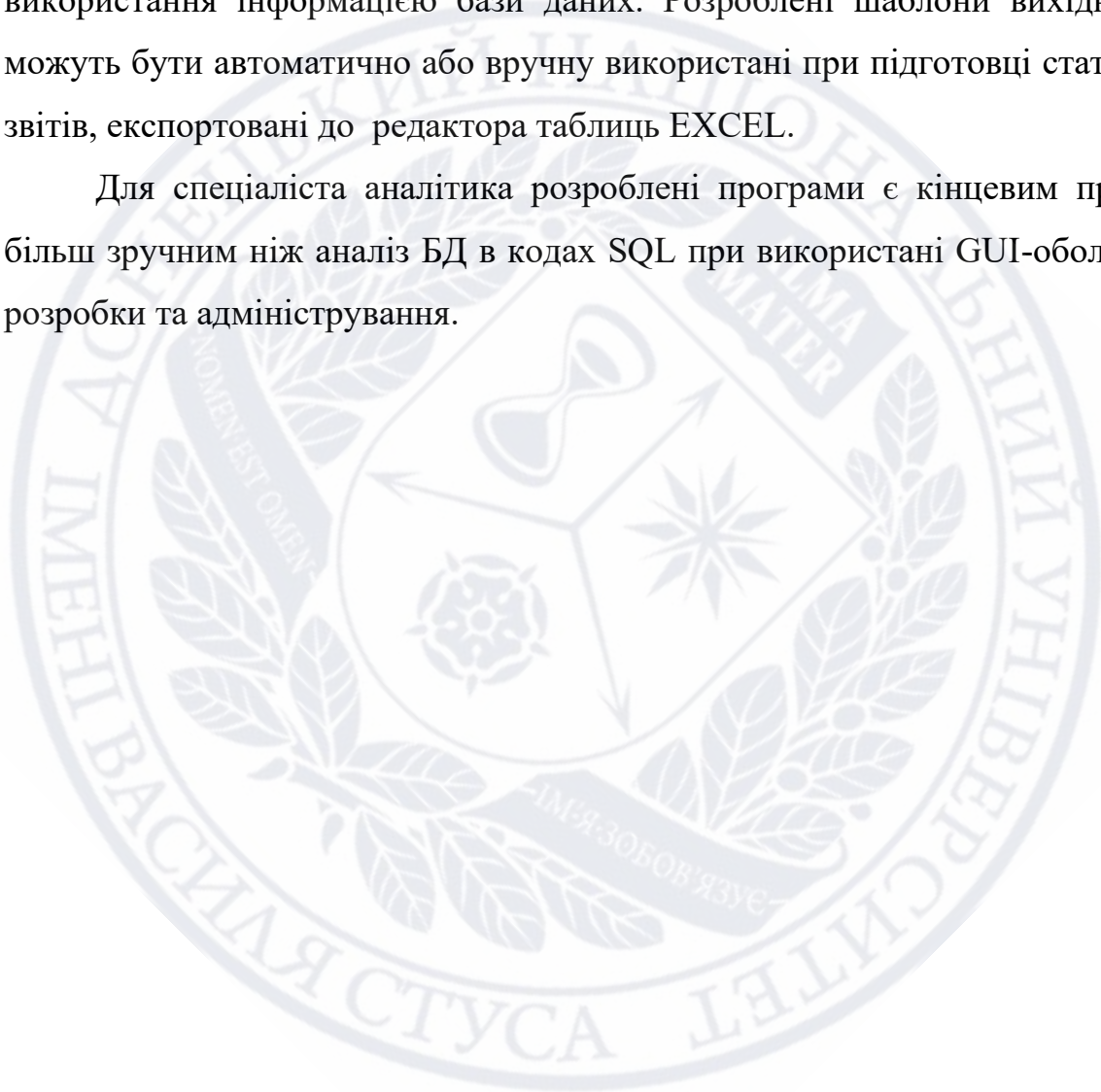
Підтвердивши ознайомлення з повідомленням отримуємо доступ до меню програми і вибираємо пункт **Опції**. Налаштування параметрів виконується в

використовують доступ до бази даних на мові запитів **SQL**. Це надає можливість швидко та з мінімальними ресурсами формувати результат. Відкрита архітектура БД та вміння використовувати SQL-запити дозволяє отримати довільну вибірку з існуючої інформації бази даних.

Лістинг головної програми та основного модуля надано в Додатку Б.

Отже, програмне середовище є оптимальним для управління та використання інформацією бази даних. Розроблені шаблони вихідних форм можуть бути автоматично або вручну використані при підготовці статистичних звітів, експортовані до редактора таблиць EXCEL.

Для спеціаліста аналітика розроблені програми є кінцевим продуктом, більш зручним ніж аналіз БД в кодах SQL при використанні GUI-оболонок, для розробки та адміністрування.



ВИСНОВКИ

Успіх бізнесу безпосередньо залежить від налагодження автоматизації процесів. Програма автоматизації підвищує ефективність роботи всього персоналу відділення.

Для визначення набору цілей та досягнення запланованого результату був проведений аналіз мети бакалаврської роботи та методів і напрямку дослідження. Кроки які були здійснені при виконання роботи, були направлені на реалізацію цього завдання. Наприкінці досліджень був розроблено програмне забезпечення.

Застосування гнучкого методу розробки програми з використанням мови програмування Delphi та бази даних типу InterBase також виявилось корисним для створення програми. Це середовище дозволяє корегувати помилки при експлуатації починаючими користувачами і у той же час надає потужний інструмент отримання різноманітних звітів та вибірок досвідченим програмістам. Розвинений інтуїтивний інтерфейс, шаблони і програми-надбудови дозволяють вирішувати широке коло питань.

В процесі виконання роботи постійно виконувався моніторинг можливостей на основі задіяних методів в переглядався результат з урахуванням прогресу та корегувань.

Реалізований проект повністю задовольнив функціональні вимоги бажаної інформаційної системи. Дані вимоги мають основний пріоритет і визначають найбільш вагомні функції відповідно до ТЗ.

Розроблене програмне забезпечення проходить апробацію на базі Університетської лікарні Вінницького національного медичного університету ім. М. І. Пирогова.

Методи та принципи побудови МІС були апробовані на ряді міжнародних конференцій на протязі 2022÷2025 р.р. та в публікаціях у наукових виданнях.

Отримані результати аналізу накопичених баз даних були використані для підготовки аналітичних довідників за 2021÷2023 р.р.

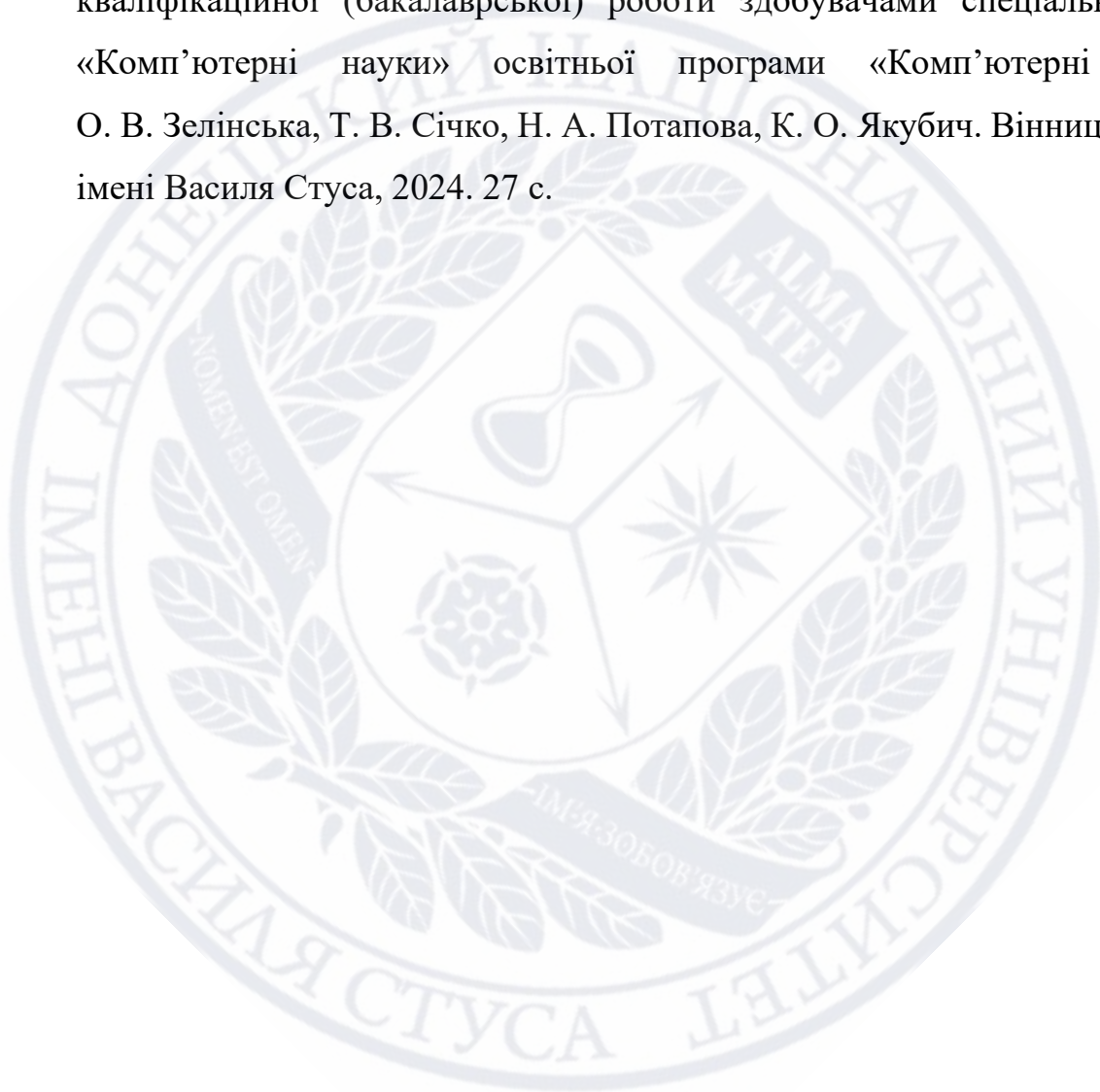
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Медичні інформаційні системи. URL: <https://naurok.com.ua/medichni-informaciyni-sistemi-254293.html> (дата звернення – 14.04.2025 р.).
2. Медичні інформаційні системи – впроваджуємо у вашому медичному закладі. URL: <https://www.medsprava.com.ua/article/544-qqq-17-m4-10-04-2017-medichn-nformatsyn-sistemi-vprovadjumo-u-vashomu-medichnomu-zaklad> (дата звернення – 14.04.2025 р.).
3. Злепко С.М., Овчарук Т.І., Овчарук А.А. Огляд медичних інформаційних систем. Інформаційні технології в медицині, економіці та екології. Збірник наукових праць Харківського університету Повітряних Сил. 2011. № 3. С.189–192.
4. Медична інформаційна система. URL: https://uk.wikipedia.org/wiki/Медична_інформаційна_система.html (дата звернення – 14.04.2025 р.)
5. Панасюк О. І. Особливості побудови медичної інформаційної системи для поліклініки [Текст] / О. І. Панасюк, В. Л. Плєскач // Сучасні електромеханічні та інформаційні системи : монографія / за заг. ред. І. В. Панасюка. - Київ : КНУТД, 2021. - С. 61-89. URL: <https://er.knutd.edu.ua/handle/123456789/19933> (дата звернення – 14.04.2025 р.)
6. Panasiuk O., Pleskach V., Statsenko V., Khomaziuk V. DEVELOPMENT OF MEDICAL INFORMATION SYSTEM FOR PRIMARY MEDICAL INSTITUTIONS. Technologies and Engineering, 2022 #6, P.9-18. doi.org/10.30857/2786-5371.2021.6.1
7. Медична інформаційна система «Доктор Елекс»: основи роботи: Навчальний посібник / під. ред. І. Березовської, Ю. Триуса. Львів: Ліга Прес. 2018. 186 с.
8. Електронна система охорони здоров'я. URL: <https://ehealth.gov.ua/> (дата звернення: 14.04.2025 р.)
9. Батюк А.Є., Батюк С.А., Пилипчик С.І., Цмоць І.Г. Компоненти інформаційних технологій для медицини: Тези міжнародної наукової

- конференції “Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту”. Т.1. Євпаторія, 2009. С. 135-137
10. Медичні інформаційні системи: огляд можливостей і приклади використання. URL: <https://evergreens.com.ua/ua/articles/medical-information-systems.html> (дата звернення – 14.04.2025 р.)
 11. Семенюк А. М. Створення реєстраційної системи МСЕК. Матеріали ЛІІ науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2023): Вінниця, 21-23 червня 2023 р., С. 744-746
 12. Семенюк А.М. SOFTWARE COMPLEX OF THE EXPERT DEPARTMENT Міжнародна науково-практична конференція «Молодіжна наука заради миру та розвитку», яка відбулась 9–11 листопада 2022 року, Чернівці. Збірник наукових праць Чернівецького національного університету ім. Ю. Федьковича, 2022. 656 с., С. 574-577, ISBN 978-966-423-755-7
 13. Бази даних. Поняття СУБД URL: <https://klona.ua/uk/blog/artificial-intelligence-uk/bazy-danyh-ponyattya-subd> (дата звернення – 14.04.2025 р.)
 14. Семенюк А.М., Ніколюк П.К. Моделювання систем прийняття рішень медичної експертизи. Прикладні аспекти сучасних міждисциплінарних досліджень. Вінниця. 2024. С. 45-47.
 15. Ротштейн А. П. Проектування нечітких баз знань: лабораторний практикум та курсове проектування : [навчальний посібник] / А. П. Ротштейн, С. Д. Штовба. Вінниця : ВДТУ, 1999. 65 с.
 16. Нескородева Т. В., Федоров Є. Є., Січко Т. В., Нескородева А. Р. Експертні та рекомендаційні системи: навч. посіб. для здобувачів вищої освіти спеціальностей 122 «Комп’ютерні науки», 125 «Кібербезпека», 113 «Прикладна математика». Вінниця: ДонНУ імені Василя Стуса, 2023. 224 с.
 17. Хорозов О. А. Застосування методів нечіткої логіки для телемедичних систем // Кибернетика и вычислительная техника. 2017. № 2. С. 36-48. URL: http://nbuv.gov.ua/UJRN/Kivt_2017_2_3.
 18. Andriy Semeniuk STATISTICAL RESEARCH OF REHABILITATION

MEASURES PERSONS WITH DISABILITIES INTERNATIONAL STUDENT CONFERENCE "STUDMATH-IT", 21-22 November 2024 p., University of Arad ROMANIYA, p.59. PP 33-35

19. Деякі питання запровадження оцінювання повсякденного функціонування особи. Постанова КМУ № 1338 від 15 листопада 2024 р., Київ
20. Методичні рекомендації до виконання, оформлення та захисту кваліфікаційної (бакалаврської) роботи здобувачами спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки». О. В. Зелінська, Т. В. Січко, Н. А. Потапова, К. О. Якубич. Вінниця: ДонНУ імені Василя Стуса, 2024. 27 с.



ДОДАТКИ



Структура бази даних "NDI_Expert"

SQL-послідовність створення бази даних

```
drop table ANKETA;
drop table AKT;
drop table IPR;
```

```
/*=====*/
/* Table: ANKETA */
/*=====*/
create table ANKETA(
  NN                INTEGER                not null,
  LastName          CHAR(25),
  Name              CHAR(25),
  PatronymicName    CHAR(25),
  BirthdayDate      DATE,
  IndNomer          CHAR(10),
  Address           CHAR(255),
  AdressCity        SMALLINT,
  SocStat           SMALLINT,
  Sex               SMALLINT,
  Profession         SMALLINT,
  Post              CHAR(50),
  Education          SMALLINT,
  TimeInv           DATE,
  Exam_First        SMALLINT,
  Exam_Lost         SMALLINT,

  constraint PK_ANKETA primary key (IndNomer)
);

/*=====*/
/* Table: AKT */
/*=====*/
create table AKT (
  Akt              INTEGER                not null,
  ExamBegin        DATE,
  Sex              SMALLINT,
  Post             CHAR(50),
  Exam_First       SMALLINT,
  Exam_WasGroup    SMALLINT,
  Exam_Lost        SMALLINT,
  MEDICAL          SMALLINT,
  TMP_DIS          SMALLINT,
  MSEK_DIAGNOS1_CODE INTEGER,
  MSEK_DIAGNOS2_CODE INTEGER,
  MSEK_DIAGNOS3_CODE INTEGER,
  MSEK_DIAGNOS1_TEXT CHAR(255),
  MSEK_DIAGNOS2_TEXT CHAR(255),
  MSEK_DIAGNOS3_TEXT CHAR(255),
```


Disability_Group SMALLINT,
 Disability_Setup DATE,
 Undiri_DisabilityGroup SMALLINT,
 Undiri_Lost SMALLINT,
 Lost SMALLINT,
 LKK SMALLINT,
 Defektura CHAR(15),
 Exam_Date DATE,
 Exam_EndDate DATE,
 Controllable SMALLINT,
 TreatmentReferral SMALLINT,
 Consult SMALLINT,
 Disability_SetupNoLimit SMALLINT,
 DisabilityReason SMALLINT,
 BusinessNum INTEGER,
 constraint PK_EMPLOYE primary key (Akt)
);

```

/*=====*/
/* Table: IPR */
/*=====*/
create table IPR(
    Akt INTEGER not
    null,
    IPRDate DATE,
    IPRTYPE SMALLINT,
    TimeInv DATE,
    LifeLim13_1 SMALLINT,
    LifeLim13_2 SMALLINT,
    LifeLim13_3 SMALLINT,
    LifeLim13_4 SMALLINT,
    LifeLim13_5 SMALLINT,
    LifeLim13_6 SMALLINT,
    LifeLim13_7 SMALLINT,
    LifeLim13_71 SMALLINT,
    LifeLim13_72 SMALLINT,
    LifeLim0 SMALLINT,
    /*
    Posl1_01 INTEGER,
    Posl2_02 INTEGER,
    Posl3_03 INTEGER,
    Posl4_04 INTEGER,
    Posl5_05 INTEGER,
    Posl6_06 INTEGER,
    Posl7_07 INTEGER,
    /*
    TehV1_01 INTEGER,
    TehV1_02 INTEGER,
    TehV1_03 INTEGER,
    TehV1_04 INTEGER,
    /*
    MedV_01 INTEGER,
    MedV_02 INTEGER,
  
```

MedV_03	INTEGER,	
MedV_04	INTEGER,	
MedV_05	INTEGER,	
MedV_06	INTEGER,	
MedV_07	INTEGER,	
MedV_08	INTEGER,	
MedV_09	INTEGER,	
MedV_10	INTEGER,	
/*		*/
RehabilPot	SMALLINT,	
RehabilTarget	CHAR(15),	
InterviewDate	DATE,	
AddToIPR	CHAR(50),	
EndIPRDate	DATE,	
RehabilRez21_1	SMALLINT,	
RehabilRez21_2	SMALLINT,	
RehabilRez21_3	SMALLINT,	
RehabilRez21_4	SMALLINT,	
RehabilRez21_5	SMALLINT,	
RehabilRez21_6	SMALLINT,	
RehabilRez21_7	SMALLINT,	
RehabilRez21_0	SMALLINT,	
ReportRehabil	CHAR(50),	
ReportDate	DATE,	
constraint PK_IPR	primary key (Akt)	
);		

Лістинг програми

```

program NDI_Expert;
uses
  Forms,
  frmMain in 'FORMS\frmMain.pas' {fmMain},
  hlIniFile in '..\components_01\HL\SOURCE\HLINIFILE.PAS',
  hlReport in '..\components_01\HL\SOURCE\HLREPORT.PAS',
  frmOptions in 'FORMS\frmOptions.pas' {fmOptions},
  okcancel_frm in '..\components_01\HL\SOURCE\FRAMES\OKCANCEL_FRM.PAS' {OkCancel_frame: TFrame},
  frmGridFieldsVisiblity in 'FORMS\FRMGRIDFIELDSVISIBILITY.PAS' {fmGridFieldsVisiblity},
  chackuncheck_frame in '..\components_01\HL\SOURCE\FRAMES\CHACKUNCHECK_FRAME.PAS'
{checkuncheck_frame: TFrame},
  frmLogViewer in 'FORMS\frmLogViewer.pas' {fmLogViewer},
  frmEmployeeTicket in 'FORMS\frmEmployeeTicket.pas' {fmEmployeeTicket},
  hlApp in '..\components_01\HL\SOURCE\HLAPP.PAS',
  frmJornal in 'FORMS\frmJornal.pas' {fmJornal},
  frmIPRTicket in 'FORMS\frmIPRTicket.pas' {fmIPRTicket},
  frmNewPeriod in 'FORMS\frmNewPeriod.pas' {fmNewPeriod},
  frmGlueBase in 'FORMS\frmGlueBase.pas' {fmGlueBase},
  frmIPRMain in 'FORMS\frmIPRMain.pas' {fmIPRMain},
  frmF_SUM in 'FORMS\frmF_SUM.pas' {fmF_SUM},
  frmNDI in 'FORMS\frmNDI.pas' {fmNDI},
  frmNDI_1 in 'FORMS\frmNDI_1.pas' {fmNDI_1},
  frmNDI_In in 'FORMS\frmNDI_In.pas' {fmNDI_In},
  frmArhivView in 'FORMS\frmArhivView.pas' {fmArhivView},
  frmArhivExport in 'FORMS\frmArhivExport.pas' {fmArhivExport},
  frmExportTestBase in 'FORMS\frmExportTestBase.pas' {fmExportTestBase};
begin
  hlLogger.AddString(false, '');
  hlLogger.AddString('NEW SESSION');
  Application.Initialize;
  Application.Title := 'NDI_Expert';
  Application.CreateForm(TfmMain, fmMain);
  Application.Run;
end.

-----
unit frmMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, IBSQLError, IBCustomDataSet, IBQuery, DB, IBDatabase, IBTable,
  Provider, DBClient, DBLocal, DBLocalI, IBEvents, Grids, DBGrids,
  ActnList, TB2Item, TB2Dock, TB2ToolBar, ComCtrls, Placemnt, ImgList,
  Wwdbgrid, Wwdbgrid, Wwdatsrc, AppEvnts, wwSpeedButton, wwDBNavigator,
  ExtCtrls, wwclearpanel, TB2ToolWindow, Buttons, frmLogViewer, StdCtrls,
  frmEmployeeTicket, frmIPRTicket, wwDataInspector, comObj, IBSQL;

const

  q_vse='select count(*) from employee where AKT >=0';           {всі}
  q_vse0='select count(*) from employee where SEX=0';           {всі жінки}
  q_vse1='select count(*) from employee where SEX=1';           {всі чоловіки}
  q_sum_kd='select sum(KIL_DAY) from employee where AKT >=0';

  q_sif=' and (KONTROL like ''1%'')'; {всі сифіліс}
  //q_vibuli=' and (KONTROL >= ''0002'' and KONTROL <= ''0009'')'; {всі вибулі}
  //q_vibuli=' and (KONTROL between (''0002'') and (''0009''))'; {всі вибулі}
  q_vibuli=' and (KONTROL not like ''0%'') and (KONTROL not like ''1%'')'; //всі вибулі
  q_vipisan=' and ((KONTROL like ''2%'') or (KONTROL like ''3%''))'; //всі виписані
  q_pereved=' and ((KONTROL like ''4%'') or (KONTROL like ''5%''))'; //всі переведені
  q_letal=' and ((KONTROL like ''8%'') or (KONTROL like ''9%''))'; //всі летальні

  w_d14=' and (EXAM_DATE-BIRTHDAYDATE)/365 < 14';           // діти до 14 років (<14)
  w_n1=' and ((EXAM_DATE-BIRTHDAYDATE)/365 >= 14 and (EXAM_DATE-BIRTHDAYDATE)/365 < 30)'; //Вік від
14 до 30 років
  w_n4=' and (Exam_Begin-BIRTHDAYDATE)/365 >= 60 '; //Вік від 60 років

  q_zag=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) = 1';           // загального
  q_dit=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) = 2';           // дитинства
  q_tr=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) in (3, 4)';           // трудове +
  q_chaes=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) = 5';           // чорнобильці
  q_covid=' and (NDI_DIAGNOS1_INV/100)=1';           // постковідні
  q_arm2=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) = 8';           // військові афганці

```



```

q_arm3=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) = 9';           // військові АТО
q_rg=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) = 10';           // Революції Гідності
q_mvs1=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) = 11';         // МВС
q_mvs2=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) = 12';         // Поліція
q_arm4=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) = 13';         // військовослужбовці
q_tr3=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) = 14';         // виконання
служб.обов.
q_arm5=' and (NDI_DIAGNOS1_INV-(NDI_DIAGNOS1_INV)/100*100) in (16, 17, 18)'; // військові
"війна укр\прос"

q_region=' and REGION/100 = ';           // область
q_m_s=' and PLACE_TYPE = 0';           // місто/село

q_konf0=' and TYP_OBS in (1,2,3,4)'; // конфлікт
q_konf1=' and TYP_OBS = 1';           // конфлікт загальний
q_kons1=' and TYP_OBS = 5';           // консультація загальний
q_kons2=' and TYP_OBS = 6';           // консультація група
q_kons3=' and TYP_OBS = 7';           // консультація причини
q_obst=' and TYP_OBS in (9,10)';      // обстеження всі = експертиза
q_po0=' and TYP_OBS = 9';             // попередній огляд
q_obst0=' and TYP_OBS = 10';          // обстеження
q_tem0=' and TYP_OBS = 11';           // тематичне
q_lik0=' and TYP_OBS = 12';           // лікування
q_kontr0=' and TYP_OBS = 13';         // контроль

q_grmSEK=' and (MSEK_GROUP)=';
q_grmSEK0=' and (MSEK_GROUP)=0';
q_grNDI2=' and (NDI_GROUP)=20';
q_grNDI9=' and (NDI_GROUP)=-1';

q_groupDOWN=' and (MSEK_GROUP) < (NDI_GROUP)'; // знижено рішення
q_groupUP=' and (MSEK_GROUP) > (NDI_GROUP)';   // підвищено рішення
q_amtDOWN=' and (MSEK_DIAGNOS1_AMT) > (NDI_DIAGNOS1_AMT)'; // знижено рішення
q_amtUP=' and (MSEK_DIAGNOS1_AMT) < (NDI_DIAGNOS1_AMT)';   // підвищено рішення

q_Auto=' and ((MSEK_DIAGNOS1_AMT)>-1 or (NDI_DIAGNOS1_AMT)>-1)';

q_MSEK_Auto=' and (MSEK_DIAGNOS1_AMT)>-1';

q_bet=' and Profil_BET like ';           //
q_bet_DU01=' and ((Profil_BET like ';           //
q_bet_DU02=' and ((Profil_BET like ';           //
q_bet_DU03=' and ((Profil_BET like ';           //
q_bet1=' and Profil_BET like '%1-%''';           // невралгія
q_bet2=' and Profil_BET like '%2-%''';           // терапія
q_bet1_2=' and ((Profil_BET like '%1-%') or (Profil_BET like '%2-%'))'; //
невралгія+терапія
q_bet3=' and Profil_BET like '%3-%''';           // хірургія

q_amp=' and Note not like '%Амп:0000%''';
q_ampH=' and (Note not like '%Амп:___0%' or Note not like '%Амп:___0%')';
q_ampHH=' and (Note not like '%Амп:___0%' and Note not like '%Амп:___0%')';
q_ampF=' and (Note not like '%Амп:___0%' or Note not like '%Амп:___0%')';
q_ampFF=' and (Note not like '%Амп:___0%' and Note not like '%Амп:___0%')';
q_amp_11=' and (Note like '%Амп:___1%' or Note like '%Амп:___1%')';
;
q_VPO=' and Note like '%ВПО:1%''';

q_konf='select count(*) from employe where EXAM_FIRST=2 and EXAM_LOST/10=6'; { оскарження}
q_kont='select count(*) from employe where EXAM_FIRST=2 and EXAM_LOST/10=7'; { контроль}
LOST/10=17'; { консультація додаткова (первичная) в кварталную(все МСЕК) }
q_travma1=' and (LOST/10)=1'; //виробнича
q_travma2=' and (LOST/10)=2'; //побутова
q_travma3=' and (LOST/10)=3'; //автопобутова

// q_zag=' and DISABILITYREASON in (0,1,2)'; // загального
q_zag_chern=' and DISABILITYREASON in (0,1,2,15,16,17)'; // загального+чернобыльцы
q_trud=' and DISABILITYREASON in (7)'; // трудовое
q_prof=' and DISABILITYREASON in (8)'; // профболезни +9
q_tr_pr=' and DISABILITYREASON in (7,8)'; // труд+профболезни +10
q_voen=' and DISABILITYREASON in (10,11,12,14)'; // военные
q_voen_3=' and DISABILITYREASON in (12)'; // военные - интернационалисты
q_ivv=' and DISABILITYREASON in (2,5,13)'; //ИВВ
q_ivv_1=' and DISABILITYREASON in (13)'; //ИВВ без "Дети" и загального
q_deti=' and DISABILITYREASON in (4,5,18)'; // "Дети" - т.е. инвалиды с детства
q_deti_2=' and DISABILITYREASON in (5)'; // "Дети" - т.е. инвалиды с детства по боевым
действиям
q_deti_3=' and DISABILITYREASON in (18)'; // "Дети" - т.е. инвалиды с детства от
чернобыльцев
q_chern=' and DISABILITYREASON in (15,16,17)'; // "Чернобыльцы" - все

```

```

q_chern_1=' and DISABILITYREASON in (15)'; // "Чернобыльцы" - ликвидаторы
q_chern_2=' and DISABILITYREASON in (16)'; // "Чернобыльцы" - эвакуированные
q_chern_3=' and DISABILITYREASON in (17)'; // "Чернобыльцы" - проживающие

q_gos=' and SOCSTAT in (0,1)'; // Госпредприятия
q_vrachi=' and (PROFESSION-(PROFESSION/100)*100)=26'; // врачи
q_vrachi_2=' and (PROFESSION-(PROFESSION/100)*100)=27'; // врачи средний мед.перс.
q_j_road=' and (PROFESSION-(PROFESSION/100)*100)=22'; // железнодорожники
q_mil=' and (PROFESSION-(PROFESSION/100)*100) in (43,44,45,46,47,48)'; // все военнослужащие
q_ATO=' and (PROFESSION/10000)=1'; // все АТО
q_ATO_1=' and POST like '%ATO-01%'

q_arb=' and POST <> 'Не працює'''; // работающие

gr=' and (DISABILITY_GROUP/1000) between 1 and 3'; // установлена любая группа
gr0=' and DISABILITY_GROUP=0'; // не признан
gr1=' and (DISABILITY_GROUP/1000)=1'; // 1 группа
gr1a=' and (DISABILITY_GROUP/100)=10'; // 1А группа
gr1b=' and (DISABILITY_GROUP/100)=11'; // 1Б группа
gr2=' and (DISABILITY_GROUP/1000)=2'; // 2 группа
gr3=' and (DISABILITY_GROUP/1000)=3'; // 3 группа
gr4=' and (DISABILITY_GROUP/1000)=4'; // больничный лист
gr5=' and (DISABILITY_GROUP/1000)=5'; // проценты потери трудоспособности
gr0_4=' and (DISABILITY_GROUP/1000) in (0,4)'; // не инвалид
gr0_4_5=' and (DISABILITY_GROUP/1000) not BETWEEN 1 and 3'; // нет группы
gr5_Y=' and (DISABILITY_GROUP-(DISABILITY_GROUP/100)*100)>0'; // есть проценты
q_ipr_v1=' and (MEDICAL/10-(MEDICAL/100)*10)=1'; // полностью выполнено ИПР
q_ipr_v2=' and (MEDICAL/10-(MEDICAL/100)*10)=2'; // частично выполнено ИПР
q_ipr_v3=' and (MEDICAL/10-(MEDICAL/100)*10)=3'; // невыполнено ИПР
q_ipr_v0=' and (MEDICAL/10-(MEDICAL/100)*10)=0'; // нет данных ИПР

q_ipr_l1=' and (MEDICAL/100-(MEDICAL/1000)*10) in (1,2,3,4)'; // направленные
на лечение
q_ipr_l2=' and (MEDICAL/100-(MEDICAL/1000)*10) in (1,2,3,4,5)'; // получен
ответ и снова направленные на лечение
q_ipr_l3=' and (MEDICAL/100-(MEDICAL/1000)*10)=5'; // пролечено
q_ipr_l4=' and (MEDICAL/100-(MEDICAL/1000)*10)=4'; // направленные на лечение
- санаторно-курорт
q_ipr_p1_3=' and (MEDICAL/1000-(MEDICAL/10000)*10)>0'; // працевлаштовані
q_ipr_p1=' and (MEDICAL/1000-(MEDICAL/10000)*10)=1'; // працевлаштовані
q_ipr_p2=' and (MEDICAL/1000-(MEDICAL/10000)*10) in (2,3)'; // перевірено
q_ipr_p3=' and (MEDICAL/1000-(MEDICAL/10000)*10)=3'; // перевірено не
раціонально
q_ipr_p0=' and (MEDICAL/1000-(MEDICAL/10000)*10)=0'; // не працевлаштовані
q_ipr_pb=' and (MEDICAL/10000)=1'; // бажають працювати
q_r_med=' and (OCCUPATION-OCCUPATION/10*10)>0'; // медицинская реабилитация
q_r_med1=' and (OCCUPATION-OCCUPATION/10*10)=1'; // медицинская - відновна
терапія
w=' and (((EXAM_DATE-BIRTHDAYDATE)/365 >= 18) and (((EXAM_DATE-BIRTHDAYDATE)/365 < 55 and SEX=0)
or ((EXAM_DATE-BIRTHDAYDATE)/365 < 60 and SEX=1)))'; // трудоспособные
w_0=' and ((EXAM_DATE-BIRTHDAYDATE)/365 < 55 and SEX=0)'; // трудоспособные женщины
w_d=' and (EXAM_DATE-BIRTHDAYDATE)/365 >= 18'; // взрослое население
w_p1=' and ((EXAM_DATE-BIRTHDAYDATE)/365 < 40)'; // предел для т.4000
w_p2=' and ((EXAM_DATE-BIRTHDAYDATE)/365 >= 40)'; // предел для т.4000
w_m=' and ((EXAM_DATE-BIRTHDAYDATE)/365 >= 18 and (EXAM_DATE-BIRTHDAYDATE)/365 < 45)'; // Возраст
от 18 по 44 года
w_m1=' and ((EXAM_DATE-BIRTHDAYDATE)/365 >= 45 and (EXAM_DATE-BIRTHDAYDATE)/365 < 61)'; // Возраст
от 45 по 60 года
w_m2=' and (EXAM_DATE-BIRTHDAYDATE)/365 >= 61'; // Возраст от 61 года

// NewRange='01.01.2013'; // новый период

qf_temp='db\Tempory.gdb';
qf_dv='db\Dviyniky.gdb';

```

type

```

TfmMain = class(TForm)
  IBDatabase: TIBDatabase;
  IBTicketQuery: TIBQuery;
  IBSQLMonitor: TIBSQLMonitor;
  IBEvents: TIBEvents;
  IBTransaction: TIBTransaction;
  ActionList: TActionList;
  StatusBar: TStatusBar;
  TBDock1: TTBDock;
  TBToolBar1: TTBToolBar;
  TBSubmenuItem1: TTBSubmenuItem;

  wwNavButton15: TwwNavButton;
  aEdit_ClearDB: TAction;

```



```

SaveDialog: TSaveDialog;
aFile_ExportInDB: TAction;
TBSeparatorItem3: TTBSeparatorItem;
TBSubmenuItem3: TTBSubmenuItem;
TBItem10: TTItem;
wwTicketDataSource: TwwDataSource;
IBTicketTable: TIBTable;

TBSeparatorItem4: TTBSeparatorItem;
TBSubmenuItem4: TTBSubmenuItem;
TBSubmenuItem5: TTBSubmenuItem;
F14_1000: TAction;
PerRab: TTItem;
IBSumDatabase: TIBDatabase;
IBSumTransaction: TIBTransaction;
TBItem21: TTItem;
TBSubmenuItem10: TTBSubmenuItem;
DviynFond: TTItem;
IBIprTableTIMEINV: TDateField;
IBIprTableLIFELIM13_1: TSmallintField;

procedure aFile_ExitExecute(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure ApplicationEventsException(Sender: TObject; E: Exception);
procedure ApplicationEventsActivate(Sender: TObject);
procedure ApplicationEventsDeactivate(Sender: TObject);
procedure ApplicationEventsMinimize(Sender: TObject);
procedure ApplicationEventsRestore(Sender: TObject);
procedure ApplicationEventsActionExecute(Action: TBasicAction;
  var Handled: Boolean);
procedure ApplicationEventsHint(Sender: TObject);
procedure IBEventsError(Sender: TObject; ErrorCode: Integer);
procedure IBSQLMonitorSQL(EventText: String; EventTime: TDateTime);
procedure aFile_OpenExecute(Sender: TObject);
procedure IBTableAfterClose(DataSet: TDataSet);
procedure IBTableAfterOpen(DataSet: TDataSet);
procedure aFile_DisconnectExecute(Sender: TObject);
procedure aFile_ConnectExecute(Sender: TObject);
procedure aOptions_SettingsExecute(Sender: TObject);
procedure wwDBGGridIBButtonClick(Sender: TObject);
procedure aDebug_LogViewerExecute(Sender: TObject);
procedure FormStorageStoredValues5Restore(Sender: TStoredValue; var Value: Variant);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure wwDBGGridDb1Click(Sender: TObject);
procedure IBTableAfterEdit(DataSet: TDataSet);
procedure IBTableAfterInsert(DataSet: TDataSet);
procedure wwNavButton14Click(Sender: TObject);

procedure Opcii_Form140Open(CaptionForm: String; var q_r, q_t, q_t1: string; d_b, d_e:
TDateTime);

procedure aFile_NewExecute(Sender: TObject);
procedure F_Nz2(Sender: TObject);
procedure F14_SUM(Sender: TObject);
procedure F37_SUM(Sender: TObject);
procedure FLL_SUM(Sender: TObject);

procedure K_Gospital(Sender: TObject);
procedure K_Analiz(Sender: TObject);
procedure K_Oglyad(Sender: TObject);

procedure F_Obstejenya(Sender: TObject);
procedure F_Tabl_3220(Sender: TObject);
procedure F_Vibuli(Sender: TObject);
procedure F_Rozpodil(Sender: TObject);
procedure F_Rozpodil_Auto(Sender: TObject);

procedure NDI(Sender: TObject);
procedure ArhivView(Sender: TObject);
procedure ArhivAdd(Sender: TObject);
procedure VidilSave(Sender: TObject);
procedure Robota_Kvartal(Sender: TObject);
procedure Robota_Mis(Sender: TObject);

private
{ Private declarations }

{Процедура открытия рабочей базы /процедура для многоразового использования/}
procedure OpenWorkDB(FileName: String);
procedure CreateEditMenu;

```



```

procedure EditMenuClick(Sender: TObject);
procedure SetEnableEdit;

public
{ Public declarations }
fmLogViewer: TfmLogViewer;

procedure FromEmployeeTicket(fmEmployeeTicket: TfmEmployeeTicket); {Из талона в Базу}
procedure ToEmployeeTicket(fmEmployeeTicket: TfmEmployeeTicket); {Из Базы в талон}

end;

var
fmMain: TfmMain;
exApp, imApp: variant;
test,f_nz,f_ipr, NewAkt_01: integer;
qi_r, qqi_r: integer;
Year, Month, Day: word;

End_Rees, Noz_01: Boolean;
Flag_01, Flag_02, Flag_R, Flag_J_01, Flag_CopyT_01: Boolean;
IPR_ini, NoAkt, F14_Dt, F14_Dt_ZV, F14_C50, F14_J: Boolean;
Flag_NoAuto, Flag_Auto: Boolean;

smJornal, Amput: integer;
Amput_S: string;

CopyT_01: array[0..50] of variant;

NewRange, NewY_01, NewY_02: TDateTime;
Def_Reg, Def_MSEK, Def_TypMSEK, MaxAKT: integer;
Def_Reg_S, Def_MSEK_S, Def_TypMSEK_S, N_File_01: string;
SerNomDov: string;

TimeInv: variant;
Sp_Reg: array[0..30] of string;
bd_Tal, bd_IPR, bd_T_I: string;
Per_Rayon: array[0..30, 0..40] of string;

VidN: string; //Дані про відділення: назва
VidB, VidBD: integer; //Кількість ліжок, план ліжко/днів

implementation

uses hlIniFile, hlReport, frmOptions, frmGridFieldsVisiblity, hlApp,
frmExportInDB, frmForma14, frmJornal, frmDBView, frmDviyniky, frmExport2008,
frmExport2012, frmExportTestBase, frmNewPeriod, frmGlueBase, frmDviyniki, frmIPRMain, frmF_SUM,
frmNDI, frmNDI_1, frmNDI_In, frmArhivView, frmArhivExport;

{$R *.dfm}

procedure TfmMain.aFile_ExitExecute(Sender: TObject);
begin
Close;
end;

procedure TfmMain.FormCreate(Sender: TObject);
var
s_01: string;
F_Reg: TextFile;
j: integer;
S_Kom_f: TextFile;
begin
Caption := Application.Title;
fmLogViewer := nil;
FormStorage.IniFileName := PrepareIniFileName;
wwDBGrid.IniAttributes.FileName := FormStorage.IniFileName;
wwDBGrid.LoadFromIniFile;
CreateEditMenu;
Flag_01:= False; //Не обходит вставку новых записей в базу, используется при конвертации баз
прежних версий
F14_Dt:=False; //В ф-14 не дети
Flag_J_01:=True; //В журналах протоколов не "выборка"
Flag_R:=False; //Выборка по районам, а не областям
Flag_NoAuto:=True; //Выборка без автотранспорта, Flag_NoAuto:
Flag_Auto:=True; //Выборка автотранспорта, Flag_Auto:
//NewRange:=StrToDate('21.12.2012'); // новый период, затем
считывается с ini-файла и опций
MaxAKT:=99999;
SerNomDov:='';

```

```

end;

procedure TfmMain.ApplicationEventsException(Sender: TObject;
E: Exception);
begin
  hlLogger.AddString(E.Message);
end;

procedure TfmMain.ApplicationEventsActivate(Sender: TObject);
var
  s_01, s_02, s_03, s_04: string;
  F_Reg, S_Kom_F: TextFile;
  i, j: integer;
begin
  hlLogger.AddString('ACTIVATE');
  /** Прочитали начальные данные по области
  Def_TypMSEK_S := FormStorage.StoredValue['DEFAULT_TYPMSEK'];
  Def_MSEK_S := FormStorage.StoredValue['DEFAULT_MSEK'];
  s_01 := copy(Def_TypMSEK_S,1,2);
  Val(s_01, Def_TypMSEK, j);
  if j <> 0 then Def_TypMSEK:=0; //помилка перетворення, вибираємо початкове значення
  Def_TypMSEK
  Def_Reg := FormStorage.StoredValue['DEFAULT_REG'];
  s_01:=FormStorage.StoredValue['CITIES_ANS'];
  Insert('2', s_01, Pos('.ans',s_01));
  AssignFile(F_Reg, s_01);
  Reset(F_Reg);
  j:=0;
  while not Eof(F_Reg) do
  begin
    Readln(F_Reg, Sp_Reg[j]);
    j:=j+1;
  end;
  CloseFile(F_Reg);
  Def_Reg_S:= Sp_Reg[Def_Reg];
  // Caption := Application.Title + ' ' + FormStorage.StoredValue['DEFAULT_REG'] + ' ' +
  FormStorage.StoredValue['DEFAULT_TYPMSEK'] + ' ' + FormStorage.StoredValue['DEFAULT_MSEK'];
  Caption := Application.Title + ' ' + Def_Reg_S + ' ' +
  FormStorage.StoredValue['DEFAULT_TYPMSEK'] + ' ' + FormStorage.StoredValue['DEFAULT_MSEK'];

  s_01:=FormStorage.StoredValue['CITIES_ANS'];
  Delete(s_01, Pos('.ans',s_01), 4);
  s_01:=s_01+'.xls';
  exApp:=CreateOLEObject('Excel.Application');
  exApp.Workbooks.Open(s_01);
  s_01:=exApp.WorkSheets['00'].Cells[1,1].Value;
  for i:=1 to StrToInt(s_01) do
  begin
    s_02:=IntToStr(i);
    s_03:=exApp.WorkSheets[s_02].Cells[1,1].Value;
    Per_Rayon[i,0]:=s_03;
    for j:=1 to StrToInt(s_03) do
      Per_Rayon[i,j]:=exApp.WorkSheets[s_02].Cells[j+3, 2].Value;
    end;

    exApp.DisplayAlerts:=False;
    exApp.Workbooks.Close;
    //exApp.quit;

  AssignFile(S_Kom_F, (AppPath + 'ANSWERS\Viddilennya.ans'));
  Reset(S_Kom_F);
  Readln(S_Kom_F, VidN);
  Readln(S_Kom_F, VidBD);
  Readln(S_Kom_F, VidB);
  CloseFile(S_Kom_F);

  VidName.Text:=VidN;
  VidPlan.Text:=IntToStr(VidBD);
  VidBed.Text:=IntToStr(VidB);

end;

procedure TfmMain.ApplicationEventsDeactivate(Sender: TObject);
begin
  hlLogger.AddString('DEACTIVATE');
end;

procedure TfmMain.ApplicationEventsMinimize(Sender: TObject);

```

```

begin
    hlLogger.AddString('MINIMIZE');
end;

procedure TfmMain.ApplicationEventsRestore(Sender: TObject);
begin
    hlLogger.AddString('RESTORE');
end;

procedure TfmMain.ApplicationEventsActionExecute(Action: TBasicAction;
    var Handled: Boolean);
begin
    hlLogger.AddString(Action.Owner.Name + '.' + Action.Name + '.Execute');
end;

procedure TfmMain.ApplicationEventsHint(Sender: TObject);
begin
    StatusBar.Panels[0].Text := Application.Hint;
end;

procedure TfmMain.IBEventsError(Sender: TObject; ErrorCode: Integer);
begin
    hlLogger.AddString('IBIventError: ErrorCode = ' + IntToStr(ErrorCode));
end;

procedure TfmMain.IBSQLMonitorSQL(EventText: String; EventTime: TDateTime);
begin
    hlLogger.AddString('IBSQLMonitorSQL: ' + EventText);
end;

procedure TfmMain.aFile_OpenExecute(Sender: TObject);
var numer: integer;
begin
    with OpenFileDialog do
        begin
            aFile_Disconnect.Execute;
            if (not IBDatabase.Connected) and (Execute) then
                begin
                    OpenWorkDB(FileName);
                end;
        end;

        IBTicketQuery.Active:=FALSE;
        IBTicketQuery.SQL.Clear;
        IBTicketQuery.SQL.Add('select count(*) from employee');
        IBTicketQuery.Active:=TRUE;
        numer:=IBTicketQuery.FieldByName('count').AsInteger;
        StatusBar.Panels[1].Text :=IntToStr(IBTable.RecNo) + '/' +IntToStr(numer);

        StatusBar.Panels[2].Text :=IBTable.DataBase.DataBaseName;
end;

procedure TfmMain.IBTableAfterClose(DataSet: TDataSet);
begin
    hlLogger.AddString('IBTable: Close');
end;

procedure TfmMain.IBTableAfterOpen(DataSet: TDataSet);
begin
    hlLogger.AddString('IBTable: Open');
    SetEnableEdit;
end;

procedure TfmMain.aFile_DisconnectExecute(Sender: TObject);
begin
    IBDatabase.Connected := false;
end;

procedure TfmMain.aFile_ConnectExecute(Sender: TObject);
begin
    IBTable.Active := true;
    IBTicketTable.Active := true;
end;

procedure TfmMain.aOptions_SettingsExecute(Sender: TObject);
var
    i01: integer;
    s01,s02: variant;
    str01: string;
begin

```



```

with FormStorage, TfmOptions.Create(Self) do
try
    hlWorkDB_FileNameEdit.FileName := StoredValue['WORKDB_PATH'];
    hlPreviewEditBoxMSEK.Text := StoredValue['MSEK_ANS'];
    hlPreviewEditBoxKom.Text := StoredValue['KOM_ANS'];
    hlPreviewEditBoxProfession.Text := StoredValue['PROF_ANS'];
    hlPreviewEditBoxOccupation.Text := StoredValue['OCUP_ANS'];
    hlPreviewEditBoxDisReason.Text := StoredValue['DISRIS_ANS'];
    hlPreviewEditBoxSITIES.Text := StoredValue['CITIES_ANS'];
    hlPreviewEditBoxLKK.Text := StoredValue['LKK_ANS'];
    hlPreviewEditBoxReferral.Text := StoredValue['REFERRAL_ANS'];
    hlDefaultMSEKComboBox.Text := StoredValue['DEFAULT_MSEK'];
    hlProfilMSEKComboBox.Text := StoredValue['DEFAULT_TYPMSEK'];
    SerDov.Text := StoredValue['SERDOV'];
//    hlDefaultREGComboBox.ItemIndex := StoredValue['DEFAULT_REG'];
//    i01:= StoredValue['DEFAULT_REG'];
    if FileExists(hlPreviewEditBoxSITIES.Text) then hlDefaultREGComboBox.ItemIndex :=
StoredValue['DEFAULT_REG']
        else hlDefaultREGComboBox.Text := 'Задайте область';
    s01:= StoredValue['DEFAULT_YEAR'];
    str01:= s01;
    if str01 > ''
        then ThisYear.Date:=s01;

NewRange:=ThisYear.Date; // новый период
if ShowModal = mrOk then
begin
    StoredValue['WORKDB_PATH'] := hlWorkDB_FileNameEdit.FileName;
    OpenWorkDB(StoredValue['WORKDB_PATH']);

    StoredValue['MSEK_ANS'] := hlPreviewEditBoxMSEK.Text;
    StoredValue['KOM_ANS'] := hlPreviewEditBoxKOM.Text;
    StoredValue['PROF_ANS'] := hlPreviewEditBoxProfession.Text;
    StoredValue['OCUP_ANS'] := hlPreviewEditBoxOccupation.Text;
    StoredValue['DISRIS_ANS'] := hlPreviewEditBoxDisReason.Text;
    StoredValue['CITIES_ANS'] := hlPreviewEditBoxSITIES.Text;
    StoredValue['LKK_ANS'] := hlPreviewEditBoxLKK.Text;
    StoredValue['REFERRAL_ANS'] := hlPreviewEditBoxReferral.Text;
    StoredValue['DEFAULT_MSEK'] := hlDefaultMSEKComboBox.Text;
    StoredValue['DEFAULT_TYPMSEK'] := hlProfilMSEKComboBox.Text;
    StoredValue['DEFAULT_YEAR'] := ThisYear.Date;
    StoredValue['DEFAULT_REG'] := hlDefaultREGComboBox.ItemIndex;
    StoredValue['SERDOV'] := SerDov.Text;

    Def_TypMSEK := hlProfilMSEKComboBox.ItemIndex ;
    Def_TypMSEK_S := hlProfilMSEKComboBox.Text ;
    Def_Reg := hlDefaultREGComboBox.ItemIndex ;
    Def_Reg_S:= hlDefaultREGComboBox.Text ;
    NewRange:= ThisYear.Date; // новый период
    SerNomDov:= SerDov.Text;

    fmlMain.Caption := Application.Title + ' ' + Def_Reg_S + ' ' + Def_TypMSEK_S + '
' + hlDefaultMSEKComboBox.Text;

    end
else
begin

    end;
finally
    Free;
end;
end;

end;

procedure TfmMain.wvDBGGridIButtonClick(Sender: TObject);
var
    i: Integer;
begin
    with TfmGridFieldsVisiblity.Create(Self) do
        try
            with IBTable do
                begin
                    iCheckCount := 0;
                    for i := 0 to FieldCount - 1 do
                        begin
                            CheckListBox.Items.Add(Fields[i].DisplayName);
                            CheckListBox.Checked[CheckListBox.Items.Count - 1] := Fields[i].Visible;
                            if Fields[i].Visible then

```

```

        Inc(iCheckCount);
    end;
end;
if ShowModal = mrOk then
begin
    with IBTable do
    begin
        for i := FieldCount - 1 downto 0 do
        begin
            Fields[i].DisplayLabel := CheckListBox.Items[i];
            Fields[i].Visible := CheckListBox.Checked[i];
//            if Fields[i].Visible then
//                wwDBGrid.Columns[i].DisplayLabel := CheckListBox.Items[i];
        end;
    end;
end
else
begin
    end;
finally
end;
end;

procedure TfmMain.aDebug_LogViewerExecute(Sender: TObject);
begin
    if fmLogViewer = nil then
        fmLogViewer := TfmLogViewer.Create(Self);
        fmLogViewer.Show;
    end;

procedure TfmMain.FormStorageStoredValues5Restore(Sender: TStoredValue; var Value: Variant);
var
    s_01: string;
    F_Reg: TextFile;
    j: integer;
begin
    if not FileExists(Value) then
        MessageBox(Handle, PChar('Перед тим як почати роботу з програмою Ви '#13 + 'повинні задати
робочу базу даних'#13 + '(меню' +
        ' "Настройки", пункт "Опції").'),
        PChar(Application.Title), MB_ICONINFORMATION)
    else
        begin
            OpenWorkDB(Value);
        end;
end;

procedure TfmMain.Opcii_Form14Open(CaptionForm: String; var q_r, q_t, q_t1: string; d_b, d_e:
TDateTime);
var
    q,q0,q1,q_tmp,q_tmp1: string;
    gr,gr0,gr1,gr2,gr3,gr4,gr0_4: string;
//    w,w_0,w_1: string;
//    t_b,t_e: string;
    kol: variant;
    numer, numer1, tmp1, tmp2: integer;
    i,j, ryad: integer;
    t_date: TDateTime;

begin
    with FormStorage, TfmForma14.Create(Self) do
        try

            Caption := CaptionForm + ' _ {' + Caption + '}';
            cbSelectMsek.Items.LoadFromFile(fmMain.FormStorage.StoredValue['MSEK_ANS']);
            cbSelectMsek.Text := fmMain.FormStorage.StoredValue['DEFAULT_MSEK'];
            cbSelectCountry.Items.LoadFromFile(fmMain.FormStorage.StoredValue['CITIES_ANS']);
            begin // для звіту НДІ
                GroupBox1.Enabled:= False;
                GroupBox1.Visible:= False;
                OkCancel_frame.aOk.Enabled := true;
            end; // begin
            if ShowModal = mrOk then
                begin
                    d_b:=BgDate.DateTime;
                    d_e:=EndDate.DateTime;
                    t_date:=BgDate.Date;

```

```

q_t:=DateTimeToStr(BgDate.Date);
t_b:=''+q_t+'';
q_e:=DateTimeToStr(EndDate.Date);
t_e:=''+q_e+'';
q_t:=t_b+' --- '+t_e;
//
q_r:=' and (EXAM_ENDDATE>=''+'+t_b+'' and EXAM_ENDDATE<=''+'+t_e+'') and
CONTROLLABLE>=10 and CONTROLLABLE- ((CONTROLLABLE/10)*10)!=1'; {если не вводится с запроса код района
или код МСЕК или "Дети" - то все}
q_r:=' and (EXAM_ENDDATE>=''+'+t_b+'' and EXAM_ENDDATE<=''+'+t_e+'')'; // and
CONTROLLABLE=1';
q_t1:=fmMain.FormStorage.StoredValue['DEFAULT_MSEK'];

    case qi_r div 1000 of
0: begin
    tmp1:=1; {если =0 (районы или дети) то не
считаются заседания}
    end;
1: begin
    qi_r:=cbSelectMsek.ItemIndex; {код МСЕК}
    str(qi_r,q);
    q_r:=q_r+' and MSEK='+q;
    q_t1:=cbSelectMsek.Text;
    tmp1:=1;
    end;
2: begin
    qi_r:=cbSelectCountry.ItemIndex; {код района}
    str(qi_r,q);
    str(qi_r+Def_Reg*100,q0); //Для коррекции района
с областью
    q_r:=q_r+' and (ADRESSCITY='+q+' or ADRESSCITY='+q0+' )';
    q_t1:=q_t1+' --> '+cbSelectCountry.Text;
    tmp1:=0;
    end;
3: begin
    qi_r:=cbSelectOther.ItemIndex; {код }
    case qi_r of
0: begin
    q_r:=q_r+q_deti; // "Дети" - т.е. инвалиды с
детства
    q_t1:=cbSelectMsek.Text+' --> '+'Діти';
    tmp1:=0;
    end;
1: begin
    q_r:=q_r+q_Chern; //"Чернобыльцы" -
    q_t1:=cbSelectMsek.Text+' --> '+'Чорнобильці';
    tmp1:=0;
    end;
2: begin
    q_r:=q_r+q_Chern_1; //"Чернобыльцы"
    q_t1:=cbSelectMsek.Text+' --> '+'Ліквідатори';
    tmp1:=0;
    end;
3: begin
    q_r:=q_r+q_Chern_2; //"Чернобыльцы"
    q_t1:=cbSelectMsek.Text+' --> '+'Евакуйовані';
    tmp1:=0;
    end;
4: begin
    q_r:=q_r+q_Chern_3; //"Чернобыльцы"
    q_t1:=cbSelectMsek.Text+' --> '+'Переселенці';
    tmp1:=0;
    end;
5: begin
    q_r:=q_r+q_Chern_3; //"Чернобыльцы"
    q_t1:=cbSelectMsek.Text+' --> '+'Мешканці';
    tmp1:=0;
    end;
6: begin
    q_r:=q_r+q_vrachi; //"Врачи"
    q_t1:=cbSelectMsek.Text+' --> '+'Лікарі';
    tmp1:=0;
    end;
7: begin
    q_r:=q_r+q_j_road; //"Железнодорожники"
    q_t1:=cbSelectMsek.Text+' --> '+'Залізничники';
    tmp1:=0;
    end;
    end; //case

```



```

        end;
        end; {case}
    end
    else
    begin
        Exit;
    end;
    tmp2:=cbSelectMSEK.ItemIndex;
finally
    Free;
end;

end; // Opcii_Form140open

procedure TfmMain.FromEmployeeTicket(fmEmployeeTicket: TfmEmployeeTicket); {Из талона в Базу}
var
    s, s1, q_tmp:      string;
    Protokol_1, i1:    integer;
    s_kontr:           integer;//Longword;
    i: word;
    MyDate: TDateTime;
    Year, Month, Day, Year_2, Month_2, Day_2, Hour, Min, Sec, MSec: Word;
    //i1: Longword;
begin
    with FormStorage, IBTable, fmEmployeeTicket do
    begin
        //***** PAGE "Анкетні дані"

        FieldByName('MSEK').AsInteger := hlMSEKComboBox.ItemIndex;

        FieldByName('AKT').AsInteger := hlAKTIntEdit.Value;
        FieldByName('BUSINESS_NUM').AsString := hlBusinessNumIntEdit.Text;

        FieldByName('LASTNAME').AsString := hlEditLastName.Text;
        FieldByName('NAME').AsString := hlEditName.Text;
        FieldByName('PATRONYMICNAME').AsString := hlEditPatronymicName.Text;
        FieldByName('BIRTHDAYDATE').AsDateTime := dtpBirthDay.DateTime;
        MyDate:=ExamDateTime.Date; //визначили вік особи від дати прийому
        DecodeDate(MyDate, Year, Month, Day);
        MyDate:=dtpBirthDay.Date;
        DecodeDate(MyDate, Year_2, Month_2, Day_2);
        i1 := (Year - Year_2);
        FieldByName('VIK').AsInteger := i1;

        if rbMale.Checked then FieldByName('SEX').AsInteger := 1 // чоловік
        else FieldByName('SEX').AsInteger := 0; // жінка

        FieldByName('IND_CODE').AsString := hlIdentKod.Text;

        s:=cbAddressReg.Text;
        if cbAddressReg.ItemIndex < 1 then s:='00 -' // Не введена область
        else s:=cbAddressReg.Text;
        if cbAddressCity.ItemIndex < 1 then
            begin
                s:=s + ' '; ' + '00 -'; // Не введений район
                FieldByName('REGION').AsInteger :=
cbAddressReg.ItemIndex*100;
            end
        else
            begin
                s:=s + ' '; ' + cbAddressCity.Text;
                FieldByName('REGION').AsInteger :=
cbAddressReg.ItemIndex*100 + cbAddressCity.ItemIndex;
            end;
        s:=s + 'Тел.: ' + hlTelephone.Text + ' ';
        FieldByName('TELEPHONE').AsString := hlTelephone.Text;
        s:=s + hlRestAddressEdit.Text + ' ';
        s1:= Copy(hlRestAddrStrEdit.Text, 1, 50); //чого не вся ??????
        s:=s + 'вул.' + s1 + ' ';
        s:=s + 'буд.' + hlRestAddrBEEdit.Text + ' ';
        s:=s + 'кв.' + hlRestAddrOfEdit.Text + ' ';
        FieldByName('ADDRESS').AsString := s;
        if rbCity.Checked then FieldByName('PLACE_TYPE').AsInteger := 1 // місто
        else FieldByName('PLACE_TYPE').AsInteger := 0; // село

        FieldByName('OSVITA').AsInteger := cbEducation.ItemIndex;
        FieldByName('PROFESSION').AsInteger := cbProfession.ItemIndex;
    end;
end;

```

```

case rgWork.ItemIndex of
  0: s1:= cbPost.Text;           //FieldByName('POST').AsString := cbPost.Text;
  1: s1:= 'Не працює';          //FieldByName('POST').AsString := 'Не працює';
  2: s1:= 'Тимчасово не працює'; //FieldByName('POST').AsString := 'Тимчасово не
  працює';
end; //case
FieldByName('POSADA').AsString := s1;

//***** PAGE "Попередній огляд"

//FieldByName('MSEK_GROUP').AsInteger := cbExamWasGroup.ItemIndex;
case cbExamWasGroup.ItemIndex of
  0: FieldByName('MSEK_GROUP').AsInteger := 0;
  1: FieldByName('MSEK_GROUP').AsInteger := 10;
  2: FieldByName('MSEK_GROUP').AsInteger := 11;
  3: FieldByName('MSEK_GROUP').AsInteger := 20;
  4: FieldByName('MSEK_GROUP').AsInteger := 30;
  5: FieldByName('MSEK_GROUP').AsInteger := 40;
  9: FieldByName('MSEK_GROUP').AsInteger := -1;
end; //case cbExamWasGroup.ItemIndex

FieldByName('MSEK_VIDSOTOK').AsInteger := seExamLost.Value;
FieldByName('MSEK_DATE').AsDateTime := DateInvalid.DateTime;
FieldByName('MSEK_DIAGNOS1_CODE').AsString := eMSEKDiagnosCode.Text;
FieldByName('MSEK_DIAGNOS1_TEXT').AsString := eMSEKDiagnos.Text;
FieldByName('MSEK_DIAGNOS1_TRAVMA').AsInteger := cbExamTrauma.ItemIndex;
FieldByName('MSEK_DIAGNOS1_AMT').AsInteger := cbAmtMsek.ItemIndex;
//FieldByName('KATEGORIYA').AsInteger := cbEditAddSurvey.ItemIndex;
i1 := cbEditAddSurvey.ItemIndex;
s1 := IntToStr(i1) + '-' + cbEditAddSurvey.Text;
FieldByName('KATEGORIYA').AsString := s1;
FieldByName('TYP_OBS').AsInteger := cbTypObst.ItemIndex;
//FieldByName('PROFIL_BET').AsInteger := hlComboBox3.ItemIndex;
i1 := cbBet.ItemIndex;
s1 := IntToStr(i1) + '-' + cbBet.Text;
FieldByName('PROFIL_BET').AsString := s1;
FieldByName('BONUS').AsInteger := cbBonus.ItemIndex;
FieldByName('SKARGI').AsString := eSkargi.Text;
s1 :=
eRist.Text+';'+eVaga.Text+';'+eTisk1.Text+';'+eTisk2.Text+';'+ePuls1.Text+';'+IntToStr(cbPuls2.ItemI
ndex)+';'+eRangVC.Text+';';
s1 :=
s1+eDihanya1.Text+';'+IntToStr(cbDihanya2.ItemIndex)+';'+IntToStr(cbDihanya3.ItemIndex)+';'+IntToStr
(cbDihanya4.ItemIndex)+';'+IntToStr(cbShkira.ItemIndex)+';'+IntToStr(cbYazik.ItemIndex)+';';
FieldByName('OGLYAD').AsString := s1;

//***** PAGE "Результати аналізів"

FieldByName('RENGEN').AsString := hlRentgen.Text;
FieldByName('ANALIZ_INSHI').AsString := hlGinekilog.Text;

//***** PAGE "Результати обстежень"

FieldByName('NDI_DIAGNOS1_CODE').AsString := eMainDiagnosCode.Text;
FieldByName('NDI_DIAGNOS1_TEXT').AsString := eMainDiagnos.Text;
FieldByName('NDI_DIAGNOS2_TEXT').AsString := eComplicationsDiagnos.Text;
if rbNoInvalid.Checked then FieldByName('NDI_GROUP').AsInteger := 0; //ZAMINNITI NA ----
NDI_GROUP
if rbDisGr1.Checked then FieldByName('NDI_GROUP').AsInteger := 10;
if rbDisGr1b.Checked then FieldByName('NDI_GROUP').AsInteger := 11;
if rbDisGr2.Checked then FieldByName('NDI_GROUP').AsInteger := 20;
if rbDisGr3.Checked then FieldByName('NDI_GROUP').AsInteger := 30;
if rbLost.Checked then FieldByName('NDI_GROUP').AsInteger := 40;
if rbIsInvalid.Checked then FieldByName('NDI_GROUP').AsInteger := 50;
if rbNoRishenya.Checked then FieldByName('NDI_GROUP').AsInteger := -1;
FieldByName('NDI_VIDSOTOK').AsInteger := seLost.Value; //ZAMINNITI NA ---- NDI_VIDSOTOK

Amput := 0;
Amput:= Amput + cbHandR.ItemIndex;
Amput:= Amput + cbHandL.ItemIndex*10;
Amput:= Amput + cbFootR.ItemIndex*100;
Amput:= Amput + cbFootL.ItemIndex*1000;
Amput_S := IntToStr(Amput);
if Amput < 1000
then
begin
Amput_S := '0'+Amput_S;
if Amput < 100
then
begin

```

```

        Amput_S := '0'+Amput_S;
        if Amput < 10
            then
                Amput_S := '0'+Amput_S;
        end;
end;

if chkCovid.Checked
    then FieldByName('NDI_DIAGNOS1_INV').AsInteger := 100+cbDisabilityReason.ItemIndex
    else FieldByName('NDI_DIAGNOS1_INV').AsInteger := cbDisabilityReason.ItemIndex;
FieldByName('NDI_DIAGNOS1_Nozol').AsInteger := cbNozologii.ItemIndex;
FieldByName('NDI_DIAGNOS1_AMT').AsInteger := cbAmtNdi.ItemIndex;

FieldByName('EXAM_DATE1').AsDateTime:= ControlDateGospital.Date;
FieldByName('EXAM_DATE2').AsDateTime:= ControlDateAnaliz.Date;
FieldByName('EXAM_DATE3').AsDateTime:= ControlDateOglyad.Date;

//***** PAGE "Епікриз"

FieldByName('NDI_DIAGNOS1_EPIKRIZ').AsString := hlEpycriz.Text;

s:='';
s_kontr:=0;

if chkControl.Checked then s_kontr:=s_kontr or 1;
//if chkVibuv.Checked then s_kontr:=s_kontr or 2;
//if RezLik.ItemIndex=0 then s_kontr:=s_kontr or 2;
    case RezLik.ItemIndex of
        0: s_kontr:=s_kontr or 2;
        1: s_kontr:=s_kontr or 4;
        2: s_kontr:=s_kontr or 8;
    end; //case RezLik.ItemIndex of
    case RezLik2.ItemIndex of
        0: s_kontr:=s_kontr+10;
        1: s_kontr:=s_kontr+20;
    else s:=s+'0'; //s_kontr:=s_kontr+40; //без змін, не внесено
    end; //case RezLik.ItemIndex of

if chkSif.Checked then s:='10'+s+IntToStr(s_kontr)
    else s:='00'+s+IntToStr(s_kontr);

//s:=s+IntToStr(s_kontr);

FieldByName('KONTROL').AsString := s;

//*****

//FieldByName('NOTE').AsString := 'Амп:' + Amput_S + '; ' + 'Л:' + IntToStr(hlDoktor.ItemIndex) + ';
'+ 'А:' + PerAnaliz.Text + '; ' + 'ВПО:0' + hlNote.Text; // Примечания
s := 'Амп:' + Amput_S + '; ' + 'Л:' + IntToStr(hlDoktor.ItemIndex) + '; ' + 'А:' + PerAnaliz.Text + '; '
if chkVPO.Checked
    then s := s + 'ВПО:1' + '; ' + hlNote.Text
    else s := s + 'ВПО:0' + '; ' + hlNote.Text;
FieldByName('NOTE').AsString := s;

//      FieldByName('NOTE').AsString := 'А:' + PerAnaliz.Text + '; ' + hlNote.Text; // Примечания
FieldByName('EXAM_BEGIN').AsDateTime := InputDokDateTimePicker.DateTime;
FieldByName('EXAM_DATE').AsDateTime := ExamDateTimePicker.DateTime;
FieldByName('EXAM_ENDDATE').AsDateTime := ExamEndDateDateTimePicker.DateTime;
FieldByName('KIL_DAY').AsVariant := ExamEndDateDateTimePicker.DateTime-
InputDokDateTimePicker.DateTime;

{

    FieldByName('EXAM_LOST').AsInteger := 0;
    // Добавление коэффициента при осмотре "МСЕК + 0"; "Выезд + 10"; "Стационар + 20"; "На дому + 30";
    //      FieldByName('EXAM_LOST').AsInteger :=
FieldByName('EXAM_LOST').AsInteger+cbExamPlace.ItemIndex;
    // Добавление коэффициента ЦЕЛЬ "группа + 10"; ...
    //      FieldByName('EXAM_LOST').AsInteger :=
FieldByName('EXAM_LOST').AsInteger+10*(1+cbTask.ItemIndex);
    //      FieldByName('MSEK_DIAGNOS3_CODE').AsString :=cbTask.Text;
    // Добавление коэффициента ДОПОЛНИТЕЛЬНЫЙ ОСМОТР "страховка + 0"; "конфликт + 1000"; "консультация +
2000"; "путевка + 3000"; "подтверждение лечения + 4000"
    //      FieldByName('EXAM_LOST').AsInteger :=
FieldByName('EXAM_LOST').AsInteger+1000*cbEditAddSurvey.ItemIndex;

```



```

//      FieldByName('MSEK_DIAGNOS3_CODE').AsString :=cbEditAddSurvey.Text; //пропечатали доп.осмотр
для информации

        case cbExamWasGroup.ItemIndex of
0:      FieldByName('EXAM_WASGROUP').AsInteger := 0;
1:      FieldByName('EXAM_WASGROUP').AsInteger := 1000;
2:      FieldByName('EXAM_WASGROUP').AsInteger := 1100;
3:      FieldByName('EXAM_WASGROUP').AsInteger := 2000;
4:      FieldByName('EXAM_WASGROUP').AsInteger := 3000;
5:      FieldByName('EXAM_WASGROUP').AsInteger := 4000;
6:      FieldByName('EXAM_WASGROUP').AsInteger := 5000;
        end; //case
FieldByName('EXAM_WASGROUP').AsInteger := FieldByName('EXAM_WASGROUP').AsInteger +
seExamLost.Value;

FieldByName('LOST').AsInteger := 10*cbExamTrauma.ItemIndex;

//      FieldByName('MSEK_DIAGNOS3_TEXT').AsString
:='Инд.№'+hlIdentKod.Text+';'+FieldByName('MSEK_DIAGNOS3_TEXT').AsString; //Добавить Инд.Номер
(перенесли в конец)

//*****
//***** ОCMOTП *****

if rbNoInvalid.Checked then
begin
FieldByName('DISABILITY_GROUP').AsInteger := 0;
FieldByName('DISABILITY_SETUPNOLIMIT').AsInteger:= -1;
cbDisabilityReason.ItemIndex := -1;
seLost.Value:=0
end;
if rbDisGr1.Checked then
begin
FieldByName('DISABILITY_GROUP').AsInteger := 1000;
if cbDisabilityReason.ItemIndex = -1 then
cbDisabilityReason.ItemIndex := 0;
end;
if rbDisGr1b.Checked then
begin
FieldByName('DISABILITY_GROUP').AsInteger := 1100;
if cbDisabilityReason.ItemIndex = -1 then
cbDisabilityReason.ItemIndex := 0;
end;
if rbDisGr2.Checked then
begin
FieldByName('DISABILITY_GROUP').AsInteger := 2000;
if cbDisabilityReason.ItemIndex = -1 then
cbDisabilityReason.ItemIndex := 0;
end;
if rbDisGr3.Checked then
begin
FieldByName('DISABILITY_GROUP').AsInteger := 3000;
if cbDisabilityReason.ItemIndex = -1 then
cbDisabilityReason.ItemIndex := 0;
end;
if rbIsInvalid.Checked then
begin
FieldByName('DISABILITY_GROUP').AsInteger := 4000;
FieldByName('DISABILITY_SETUPNOLIMIT').AsInteger := -1;
cbDisabilityReason.ItemIndex := -1;
seLost.Value:=0
end;
if rbLost.Checked then
begin
FieldByName('DISABILITY_GROUP').AsInteger := 5000;

if seLost.Value = 0 then
seLost.Value := 5;
end;
FieldByName('DISABILITY_GROUP').AsInteger := FieldByName('DISABILITY_GROUP').AsInteger +
seLost.Value;

FieldByName('MSEK_DIAGNOS1_CODE').AsString := eMainDiagnosCode.Text;
FieldByName('MSEK_DIAGNOS1_TEXT').AsString := eMainDiagnos.Text;
if Length(eComplicationsDiagnosCode.Text) < 3 //Не введен или не полностью введен второй код
then FieldByName('MSEK_DIAGNOS2_CODE').AsString :=

```

```
FieldByName('MSEK_DIAGNOS1_CODE').AsString := eComplicationsDiagnosCode.Text;
else FieldByName('MSEK_DIAGNOS2_CODE').AsString := eComplicationsDiagnosCode.Text;
FieldByName('MSEK_DIAGNOS2_TEXT').AsString := eComplicationsDiagnos.Text;
FieldByName('UNDIRI_LOST').AsInteger := cbNozologii.ItemIndex;

if rbUnlimited.Checked then
    FieldByName('DISABILITY_SETUPNOLIMIT').AsInteger := 0;
if rbLimit.Checked then
    FieldByName('DISABILITY_SETUPNOLIMIT').AsInteger := 1;
if rbOneTime.Checked then
    FieldByName('DISABILITY_SETUPNOLIMIT').AsInteger := 2;
FieldByName('DISABILITY_SETUP').AsDateTime := DisSetupDateTimePicker.DateTime;
FieldByName('DISABILITYREASON').AsInteger := cbDisabilityReason.ItemIndex;

FieldByName('DEFEKTURA').AsString := h1BlankSer.Text + '-' + h1BlankNum.Text;

FieldByName('EXAMBEGIN').AsDateTime := ExamBeginDateTimePicker.DateTime;
FieldByName('EXAM_DATE').AsDateTime := ExamDateTime.DateTime;
FieldByName('EXAM_ENDDATE').AsDateTime := ExamEndDateTimePicker.DateTime;
s:=DateToStr(ResultDateTimePicker.DateTime); // Дата прийняття рішення
FieldByName('MSEK_DIAGNOS3_TEXT').AsString := 'ДатаПР:' + s + ';
'+FieldByName('MSEK_DIAGNOS3_TEXT').AsString; //Дата в "Примечание"
s:=DateToStr(InputDokDateTimePicker.DateTime); // Дата прийому документів
FieldByName('MSEK_DIAGNOS3_TEXT').AsString := 'ДатаПД:' + s + ';
'+FieldByName('MSEK_DIAGNOS3_TEXT').AsString; //Дата в "Примечание"

if chkControl.Checked then
    Protokol_1 := 1+Protokol
else
    Protokol_1 := 0+Protokol;
if chkIPR.Checked then
    Protokol_1 := 100+Protokol_1
else
    Protokol_1 := 0+Protokol_1;
FieldByName('CONTROLLABLE').AsInteger := Protokol_1;

FieldByName('MSEK_DIAGNOS3_TEXT').AsString := 'ДатаПІ:' + s + '; '+'МсекПІ:' + s1 +';
'+FieldByName('MSEK_DIAGNOS3_TEXT').AsString; //Дата первой инвалидности добавили в "Примечание"
FieldByName('MSEK_DIAGNOS3_TEXT').AsString := 'Инд.№' + h1IdentKod.Text +';
'+FieldByName('MSEK_DIAGNOS3_TEXT').AsString; //Инд.номер добавили в "Примечание"
}

end; // with

end;

procedure TfmMain.Robota_Mis(Sender: TObject);
var
    q,q0,q1,q_r,q_tmp,q_tmp1:      string;
    q_t,q_t1:                      string;
//    gr,gr0,gr1,gr2,gr3,gr4,gr0_4:   string;
//    w,w_0,w_1:                   string;
    t_b,t_e:                      string;
    kol:                          variant;
    numer, numer1, tmp1, tmp2:     integer;
    numer_s:                      string;
    i,j, ryad, KilDniv:            integer;
    Koef:                         real;
    t_date, d_b, d_e:             TDateTime;

begin
kol:=0;

exApp:=CreateOLEObject('Excel.Application');
exApp.Workbooks.Add(AppPath + 'formy\ArbaitV-M_.xlt');

//***** Ввод параметров

Opcii_Forml4Open('ЗБИТ ЗА МІСЯЦЬ', q_r, q_t, q_t1, d_b, d_e);
//KilDniv:=int(d_e-d_b);
t_b:= Copy(q_t, 1, 10);
d_b:= StrToDate(t_b);
t_e:= Copy(q_t, 18, 10);
d_e:= StrToDate(t_e);
KilDniv:= Trunc(d_e-d_b);

{*****#1*****}
```

```

exApp.Cells[2,5].Value:=VidN;
exApp.Cells[3,5].Value:=q_t;

ryad:=7;
q1:=q_vse + q_r;
q:=q1;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;

q:=q1 + ' and SEX=1';
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,6].Value:=number;
q:=q1 + ' and SEX=0';
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,8].Value:=number;

ryad:=ryad+1;
q:=q1 + w_d18;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,6].Value:=number;
q:=q1 + w_d;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,8].Value:=number;

ryad:=ryad+1;
q:=q1 + q_VP0;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,8].Value:=number;

ryad:=15;
q1:=q_vse + q_r + q_vibuli; //' and (KONTROL > ''00001'' and KONTROL < ''00004'')'; {всі вибулі}
q:=q1;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;

q:=q1 + q_VP0;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,8].Value:=number;

ryad:=ryad+1;
q:=q1+q_exp;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;

```



```

exApp.Cells[ryad,4].Value:=number;
q:=q1+q_tem0;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;

ryad:=ryad+1;
q:=q1+q_lik0;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;
q:=q1+q_konf0;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;

ryad:= 24;
q:=q_sum_kd + q_vibuli + q_r;           //summa KIL_DAY

IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:= IBTicketQuery.Fields[0].AsInteger;
exApp.Cells[ryad,4].Value:=number;

//exApp.Cells[ryad,10].Value:=KilDniv;
if (KilDniv mod 30) > 16
    then Koef:=(KilDniv div 30)+1           //2 половина місяця
    else if (KilDniv mod 30) > 13           //середина місяця
        then Koef:=(KilDniv div 30)+0.5
        else if (KilDniv div 30) = 0
            then Koef:=KilDniv/30           //1 половина 1-го місяця виключити -->
Koef:=0
        else Koef:=(KilDniv div 30);       //1 половина місяця

ryad:=ryad+1;
number:= Round(VidBD*Koef);
exApp.Cells[ryad,4].Value:=number;

//exApp.Cells[ryad+1,4].Value:=VidBD/360*KilDniv //на кількість днів періоду;

ryad:=29;
q:=q1 + q_amp;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;

ryad:=30;
q:=q1 + q_ampH;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
number_s:=IntToStr(number);
q:=q1 + q_ampHH;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
number_s:=number_s + ' / o - '+IntToStr(number);
exApp.Cells[ryad,3].Value:=number_s;

ryad:=40;
q:=q1 + q_ampF;
IBTicketQuery.Active:=FALSE;

```

```

IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
number_s:=IntToStr(number);
q:=q1 + q_ampFF;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
number_s:=number_s + ' / o - ' + IntToStr(number);
exApp.Cells[ryad,3].Value:=number_s;

```

```

ryad:=31;
q:=q1 + q_amp__11;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,3].Value:=number;

```

```

ryad:=34;
q:=q1 + q_amp__44;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,3].Value:=number;

```

```

ryad:=37;
q:=q1 + q_amp__77;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,3].Value:=number;

```

```

ryad:=41;
q:=q1 + q_amp11;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,3].Value:=number;

```

```

ryad:=44;
q:=q1 + q_amp44;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,3].Value:=number;

```

```

ryad:=47;
q:=q1 + q_amp77;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,3].Value:=number;

```

```

ryad:=31;
q:=q1 + q_amp__1;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;
q:=q1 + q_amp__1;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;

```

```
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;
```

```
ryad:=ryad+1;
q:=q1 + q_amp__2;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;
q:=q1 + q_amp__2;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;
```

```
ryad:=ryad+1;
q:=q1 + q_amp__3;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;
q:=q1 + q_amp__3;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;
```

```
ryad:=ryad+1;
q:=q1 + q_amp__4;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;
q:=q1 + q_amp__4;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;
```

```
ryad:=ryad+1;
q:=q1 + q_amp__5;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;
q:=q1 + q_amp__5;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;
```

```
ryad:=ryad+1;
q:=q1 + q_amp__6;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;
q:=q1 + q_amp__6;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
```



```
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
numer:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=numer;
```

```
ryad:=ryad+1;
q:=q1 + q_amp___7;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
numer:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=numer;
q:=q1 + q_amp___7_;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
numer:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=numer;
```

```
ryad:=41;
q:=q1 + q_amp_1___;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
numer:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=numer;
q:=q1 + q_amp1___;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
numer:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=numer;
```

```
ryad:=ryad+1;
q:=q1 + q_amp_2___;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
numer:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=numer;
q:=q1 + q_amp2___;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
numer:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=numer;
```

```
ryad:=ryad+1;
q:=q1 + q_amp_3___;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
numer:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=numer;
q:=q1 + q_amp3___;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
numer:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,4].Value:=numer;
```

```
ryad:=ryad+1;
q:=q1 + q_amp_4___;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
numer:=IBTicketQuery.FieldName('count').AsInteger;
exApp.Cells[ryad,7].Value:=numer;
q:=q1 + q_amp4___;
IBTicketQuery.Active:=FALSE;
```

```
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldByName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;
```

```
ryad:=ryad+1;
q:=q1 + q_amp_5__;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldByName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;
q:=q1 + q_amp5__;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldByName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;
```

```
ryad:=ryad+1;
q:=q1 + q_amp_6__;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldByName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;
q:=q1 + q_amp6__;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldByName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;
```

```
ryad:=ryad+1;
q:=q1 + q_amp_7__;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldByName('count').AsInteger;
exApp.Cells[ryad,7].Value:=number;
q:=q1 + q_amp7__;
IBTicketQuery.Active:=FALSE;
IBTicketQuery.SQL.Clear;
IBTicketQuery.SQL.Add(q);
IBTicketQuery.Active:=TRUE;
number:=IBTicketQuery.FieldByName('count').AsInteger;
exApp.Cells[ryad,4].Value:=number;
```

```
exApp.Visible:=true; {vizualization EXEL}
```

```
{*****
{*****
{*****
*****}
```

```
end;
```

```
end.
```

Шаблони звітної інформації "NDI_Expert"

З В І Т					
про роботу		відділення			
в період:					
Кіл.ліжок		План л-д			
1. Всього виписано експертиза працездатності	хворих.,	тематичні	хворих.,		
лікувальні	хворих.,	по контролю	хворих.,		
Із загальної кількості осіб із інвалідністю склали:					
1	Загальне захворювання				чол.
2	Дитинства				чол.
3	Трудове каліцтво				чол.
4	Професійне захворювання				чол.
5	Особи з інвалідністю - ЧАЕС				чол.
6	Постковідні				чол.
7					чол.
8	Особи з інвалідністю - УБД "афганці"				чол.
9	Особи з інвалідністю - УБД АТО/ООС				чол.
10	Особи з інвалідністю - внаслідок Революції Гідності				чол.
11	Особи з інвалідністю - внаслідок виконання служб.обов'язків				чол.
12	Особи з інвалідністю - МВС				чол.
13	Особи з інвалідністю - Поліція				чол.
14	Особи з інвалідністю - військовослужбовці				чол.
15					чол.
16	ВПО				чол.
2. Виконання плану		ліжко-днів			
при плані		ліжко-днів	відсоток виконання	#ДЕЛ/0!	
3. Середнє перебування хворого на ліжку					
загальний	0 хворих.,	експертний	0 хворих.,		
тематичний	0 хворих.,	лікувальний	0 хворих.,		
4. Зайнятисть ліжка	#ДЕЛ/0!				
5. Оборот ліжка	#ДЕЛ/0!				
6. Кількість операцій		наявність укладень			
7. Кількість прокурованих хворих за					
Зав. відділенням:					
Дата					

Склад хворих у стаціонарі, строки та результати лікування									
			в період:						
			в відділенні:						
Таблиця 3220									
№ рядка	Найменування хвороби	І. Дорослі від 18 років і старші			ІІ. Діти від 0 до 17 років включно				
		Виписано хворих	Проведено виписаними ліжко-днів	Померло	Виписано хворих	У тому числі у віці до 1 року	Проведено виписаними ліжко-днів	Померло	У тому числі у віці до 1 року
	* Усього								
	у тому числі								
	2.0 Деякі інфекційні та паразитарні хвороби								
	2.1 З них: кишкові інфекції								
	2.2 туберкульоз органів дихання								
	2.3 міннігіокова інфекція								
	2.4 септицемії								
	2.5 вірусний гепатит								
	2.6 хронічний вірусний гепатит В								
	2.7 хронічний вірусний гепатит С								
	3.0 Новоутворення								
	3.1 з них: злоякісні новоутворення								
	3.2 в тому числі: злоякісні новоутворення лімфатичної, кровотворної та споріднених з ними тканин								
	4.0 Хвороби крові, кровотворних органів і окремі порушення із залученням імунного механізму								
	4.1 з них: анемії								
	4.2 порушення згортання крові, пурпура, інші геморагічні стани								
	5.0 Хвороби ендокринної системи, розладу харчування, порушення обміну речовин								
	5.1 з них: дифузний зоб II- III ступенів								
	5.2 набутий гіпотиреоз та інші форми гіпотиреозу								
	5.3 Цукровий діабет								
	6.0 Розлади психіки та поведінки								
	7.0 Хвороби нервової системи								
	7.1 з них запальні хвороби центральної нервової системи								
	7.2 транзиторні, церебральні, ішемічні напади та споріднені синдроми								
	7.3 церебральний параліч та інші паралітичні синдроми								
	8.0 Хвороби ока та придаткового апарату								
	9.0 Хвороби вуха та соскоподібного відростка								
	9.1 з них гострий отит середнього вуха								
	10.0 Хвороби системи кровообігу								
	10.1 з них гостра ревматична гарячка								
	10.2 хронічні ревматичні хвороби серця								
	10.3 гіпертонічна хвороба (без згадування про ішемічну хворобу серця та судинні хвороби ураження мозку)								
	10.4 ішемічна хвороба серця								
	10.5 у т.ч.: а) стенокардія								
	10.6 б) гострий інфаркт міокарда								
	10.7 в) інші форми гострої ішемічної хвороби серця								
	10.8 церебральні хвороби								
	10.9 з них: а) внутрішньочерепний крововилив								
	10.10 б) інфаркт головного мозку								
	10.11 в) інсульт, не уточнений як крововилив або інфаркт головного мозку								
	11.0 Хвороби органів дихання								
	11.1 з них: пневмонії								
	11.2 інші обструктивні хвороби легень								
	11.3 бронхіт хронічний обструктивний								
	11.4 бронхіальна астма								
	11.5 бронхоекстрінхна хвороба								
	12.0 Хвороби органів травлення								
	12.1 з них: хвороби ротової порожнини, залоз та щелеп								
	12.2 виразка шлунка та 12-палої кишки								
	12.3 у т.ч.: а) проривна виразка шлунка та 12-палої кишки								
	12.4 гастрит та дуоденіт								
	12.5 у т. ч.: гострий геморагічний та інші гострі гастрити								
	12.6 гострий апендицит								
	12.7								

РОЗПОДІЛ ХВОРИХ ЗА МІСЦЕМ ПРОЖИВАННЯ

		в період:							
		в відділенні:							
№ п/п	Області	Всього	Конфлікт		Консультація		Лікув.	Темат.	Контр.
			всього	в т.ч. АМТ	всього	в т.ч. АМТ			
1	АР Крим								
2	Вінницька								
3	Волинська								
4	Дніпропетровська								
5	Донецька								
6	Житомирська								
7	Закарпатська								
8	Запорізька								
9	Івано-Франківська								
10	Київська								
11	Кіровоградська								
13	Львівська								
14	Луганська								
15	Миколаївська								
16	Одеська								
17	Полтавська								
18	Рівненська								
19	Сумська								
20	Тернопільська								
21	Харківська								
22	Херсонська								
23	Хмельницька								
24	Черкаська								
25	Чернігівська								
25	Чернівечька								
26	м.Київ								
27	м. Севастопіль								
<i>Разом</i>			-	-	-	-	-	-	-
	<i>По відділенню</i>		-						
	<i>По клініці</i>								

Статистична інформація програми "NDI_Expert"

Вибір параметрів НДІ

МСЕК: Область: Період: 2014

Інвалідність по нозологіям: При жовробак ендокринної системи: E00-E99

Послуги: Зберегти дані: Ok Cancel

Очистити дані: Перегляд даних: ☐

Види та форми реабілітаційних заходів	Усього	Від кільк	Вік	18-44	45-60	61 i >	Група II	I гр.	II гр.	III гр.	Усього	Від кільк	I гр.	II гр.	III гр.
I. Послуги	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1. Медична реабілітація	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
відновна терапія	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
профілактичні заходи	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
реконструктивна хірургія	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
санаторно-курортне лікування	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
психіатрична допомога	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
слухопротезування	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
кохлеарна імплантація	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
медичне спостереження	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2. Психолого-педагогічна реабілітація	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
консультавання	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
психолого-педагогічна діагностика	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
психолого-педагогічний патронаж	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
психологічна та педагогічна корекції	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
освітні послуги:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
колективна форма навчання, ч.т. ч. II	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
індивідуальна форма навчання	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
дистанційна форма навчання, екстерні	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3. Фізична реабілітація	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
консультавання	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
еотерапія	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Параметри	Усього	Від кільк	I гр.	II гр.	III гр.
1. Обмеження життєдіяльності відносно:	0	0	0	0	0
- повністю	0	0	0	0	0
- частково	0	0	0	0	0
- не відновлено	0	0	0	0	0
- погіршено	0	0	0	0	0
- немає даних	0	0	0	0	0
2. Побутова активність відновлена:	0	0	0	0	0
- повністю	0	0	0	0	0
- частково	0	0	0	0	0
- без динаміки	0	0	0	0	0
- погіршилась	0	0	0	0	0
- немає даних	0	0	0	0	0
3. Технічні засоби компенсації об'ємів:	0	0	0	0	0
- в повному обсязі	0	0	0	0	0
- частково	0	0	0	0	0
- не компенсують	0	0	0	0	0
- немає даних	0	0	0	0	0
4. Рекомендації щодо подальшої реабілітації:	0	0	0	0	0
- не потребує	0	0	0	0	0
- потребує	0	0	0	0	0
- немає даних	0	0	0	0	0
5. ІПР виконана:	0	0	0	0	0
- ч. повного обсязі	0	0	0	0	0

