

РЕФЕРАТ/ABSTRACT

Кваліфікаційна робота магістра містить: с. – 53, рис. – 17, табл. – 1, джерел – 11.

МЕДИЧНІ ПОСЛУГИ, АВТОМАТИЗОВАНА СИСТЕМА ОБЛІКУ,
БАЗА ДАНИХ

У процесі виконання кваліфікаційної роботи впроваджено автоматичну систему обліку медичних послуг. Розроблено інтерфейс програми та окремі функціональні елементи. Проведено тестування, аналіз та дослідження розробленої системи, яка є гнучкою та забезпечує достатню інформаційну підтримку.

ПЗ може бути змінене та доповнене в наступних версіях.

Загалом нові технології та методи, що використовуються в програмних продуктах, виявилися дуже ефективними в автоматизації робочих місць.

MEDICAL SERVICES, AUTOMATED ACCOUNTING SYSTEM,
DATABASE

In the process of performing the qualification work, an automatic system of accounting for medical services was introduced. The program interface and individual functional elements have been developed. The developed system, which is flexible and provides sufficient information support, has been tested, analyzed and researched.

The software may be modified and supplemented in future versions.

In general, the new technologies and methods used in software products have proven to be very effective in automating workplaces. Economic efficiency/significance of work - the application of the algorithm will allow to automatically identify anomalies in medical data, which will reduce the specialist's time spent on this procedure.

ЗМІСТ

ВСТУП.....	7
1. АНАЛІТИЧНА ЧАСТИНА	9
1.1 Призначення та область застосування	9
1.2 Огляд способів і засобів вирішення поставлених задач	10
1.3 Переваги і недоліки	11
1.4 Аналіз програмно-апаратного забезпечення	15
1.5 Постановка задачі	17
2. РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	20
2.1. Призначення та область застосування	20
2.2. Відбір записів для аналізу	24
2.3. Основне меню	33
3. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	41
3.1 Охорона праці.....	41
3.2 Безпека в надзвичайних ситуаціях.....	44
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТКИ.....	53

ВСТУП

На сьогодні зауважується значний розвиток інформаційних технологій а саме зростання інформатизації та формування глобального інформаційного простору. Це стимул не лише для розвитку світового господарства, а й для інших сфер життєдіяльності людства.

В медичній галузі, застосування МІС значно підвищить рівень діяльності медичних установ. При наданні медичної послуги перевагу має оперативність доступу користувачів до потрібної інформації, автоматизація процесу статистичної обробки отриманих даних, конфіденційність зберігання їх.

Важливе значення в охороні здоров'я має впровадження нових технологій, пов'язаних з розвитком наукових та технічних досягнень, зокрема автоматизації робочих місць, адаптованого для конкретних аспектів діяльності медичної установи програмного забезпечення [1].

Застосування нових ІТ у сучасних медичних закладах полегшує облік усіх наданих послуг. Крім того здійснюється заповнення електронних амбулаторних медичних карт та історій хвороб, формування звітів, тощо.

Автоматизація медичного закладу – це єдиний інформаційний простір, в межах якого створюються АРМ лікарів, здійснюється організація роботи статистичних відділень, створюються БД, ведеться електронна медична документація та об'єднуються в одне ціле процеси, які задіяні в роботі медичного закладу. Результатом є спрощення роботи медичного персоналу та підвищення якості виконуваних робіт.

Перевагами автоматизації є підвищення ефективності процесу, зниження помилкових рішень, які виникають внаслідок неуважності персоналу та суб'єктивного підходу при виконанні завдань [2].

У контексті економічної реформи потрібно зосередити увагу на

удосконаленні організації ефективної роботи обліково-аналітичної системи надання медичних послуг та налагодити їх відповідний облік.

Важливість поставленої проблеми зумовило актуальність її [3].

Метою роботи є розроблення системи обліку надання медичних послуг для проведення контролю роботи, пов'язаної з ними.

Для досягнення поставлених цілей необхідно:

- Проведення аналітичних оглядів діяльності закладу охорони здоров'я.
- Розробити структурно-логічну схему системи.
- Розробити алгоритмічне та програмне забезпечення роботи з відповідними модулями.

На даний час білінг медичних послуг в Україні здійснюється за принципами єдиної методології з урахуванням деталей діяльності закладів, установ, організацій та комп'ютерних технологій обробки білінгових даних, тому дане дослідження є актуальним. Для забезпечення дієвості, ефективності та результативності роботи персоналу потрібні спеціальні програмні системи та сучасні інформаційні технології.

Використовуючи цей програмний продукт, система є дуже зручною, швидкою, не потребує спеціального обслуговування та економічно ефективнішою, ніж аналогічні продукти, що дозволяє заощадити робочий час персоналу та кошти установ. Дана робота спрямована на вирішення поставлених завдань автоматизації обліку медичних послуг.

1. АНАЛІТИЧНА ЧАСТИНА

1.1 Призначення та область застосування

На сьогодні ІТ все ширше використовуються в медичній сфері. Комп'ютери стали невід'ємною частиною розвинутих клінік і приватних практик. Керівникам медичних закладів відомо, що без використання сучасних методів управління бізнесом неможливо вести успішний бізнес у висококонкурентному світі. Це передбачає значні зміни, пов'язані з корективами у підготовці медичних працівників [3].

Є розроблено безліч інструментів, які дозволяють закладам охорони здоров'я та компаніям швидко та ефективно керувати своєю діяльністю, найчастіше за допомогою ПК та відповідного ПЗ. Сучасні програмні додатки та розробки значно підвищують ефективність, швидкодію та зручність роботи, спрощують ведення обліку та розроблення звітної документації.

Впровадження інформаційно-управлінської системи в медичному закладі дозволить автоматизувати роботу медзакладу, звільнивши адміністративний та медичний персонал від паперової роботи та давши можливість проводити аналіз роботи медичної установи. Форми звітності щодо наданих медичних послуг дозволяють здійснювати моніторинг та аналіз роботи медичних закладів [4].

ПЗ необхідне закладам охорони здоров'я, повинне відповідати певним зазначеним вимогам, які визначаються фахівцями. По суті, це забезпечує ефективну організацію та керування фінансово-господарською діяльністю і забезпечує оперативний доступ до оновленої, максимально повної та достовірної інформації (відомості про хворих, амбулаторні карти, облік наданих послуг тощо). Крім того, при цьому створюються умови для дотримання конфіденційності відповідно до медичної етики, полегшення

роботи медперсоналу.

Важливим для підвищення ефективності такої діяльності є використання систем автоматизованого обліку наданих послуг, зокрема статистичні відомості та бухоблік. Вимагається дотримання методологічних засад винесених у відповідних законодавчих документах із врахуванням профілю медичної установи та ПЗ необхідного для опрацювання даних, які надходять.

1.2 Огляд способів і засобів вирішення поставлених задач

В комп'ютерній індустрії завжди виділялися такі напрямки: числення, збереження, опрацювання інформації. Як відомо, поява комп'ютерів була зумовлена насамперед необхідністю проведення масштабних розрахунків для наукових цілей і військово-промислового комплексу. Обсяг необхідних обчислень перевищує те, що традиційні групи розробників можуть виконати за прийнятний час [5].

Так першими користувачами комп'ютерів і розробниками комп'ютерних програм стали математики. У цивільній економіці широкі розрахунки зазвичай не потрібні, за винятком таких галузей, як авіація та автомобілебудування. У більш загальних сферах приватної економіки (бухгалтерський облік, банківська справа, операції на біржах, системи бронювання квитків і готелів) основною проблемою завжди була необхідність збирати, забезпечувати надійний захист і своєчасно обробляти цю інформацію. Поява інформаційних систем була насамперед спрямована на вирішення вищезазначених проблем і виникла як відповідь комп'ютерної індустрії на потреби ділового світу.

В загальному, трактують інформаційну систему (ІС) як набір відповідних засобів зберігання та подальшого опрацювання даних для задоволення потреб користувачів. Таке означення є задовільним лише з найбільш загальної та

неформальної точки зору і потребує подальших уточнень [5].

Залежно від конкретної сфери застосування, ІС відрізняються за функціональністю, архітектурою та реалізацією. Проте всі інформаційні системи мають принаймні 2 спільні характеристики

Першою характеристикою є те, що усі ІС призначені для збирання, збереження та подальшого опрацювання інформації. Тому є середовище для збереження даних та доступ до них. Воно має забезпечити надійність збереження та ефективний доступ, у відповідності до призначення використання ІС. Основною вимогою до обчислювальної програми при вирішенні складних завдань – її швидкодія. Має бути забезпечена точність отримуваних результатів. При вирішенні серйозних обчислювальних проблем цей час може зайняти тижні або місяці, навіть на відносно потужних комп'ютерах. Тому програмісти завжди дуже скептично ставляться до збереження даних у зовнішньому ЗП. Зовнішня пам'ять зазвичай використовується для періодичного (а не частого) зберігання проміжних результатів обчислень. Це дозволяє програмі продовжувати роботу зі збереженої контрольної точки, навіть якщо комп'ютер виходить з ладу.

По-друге, ІС орієнтовані на кінцевого користувача, для якого термінал, персональний комп'ютер чи робоча станція – це допоміжний засіб. Тому, вимогою до ІС є простота та зручність інтерфейсу для забезпечення якісної роботи. У деяких випадках застосовують графічний інтерфейс з відповідним меню. [6].

На даний момент є системи, які використовують для автоматизації адміністративної, фінансової та господарської діяльності у медичній сфері. Деякі з них містять структурні особливості медичних закладів різних напрямків та використовують різні підходи та методи.

1.3 Переваги і недоліки

Аналіз відомих систем для підтримки управлінських процесів медичними закладами показав присутність неконтрольованої інформатизації медичних закладів, відсутність галузевих стандартів та ліцензованого ПЗ, єдиних вимог щодо об'єму та формату зберігання інформації, уніфікованості форми та ін. Сучасна звітність не забезпечує системність, повноту та послідовність інформатизації процесу. В Україні використовуються різноманітні медичні ІС, такі як «EMSIMED», ASMU «MEDI», «Meduchet», «Medialog», «LysMedap». Не зважаючи на їх високу ціну, відсутність спеціалістів, які можуть редагувати програми та змінювати інформацію та шаблони звітів відповідно до потреб закладу, встановлено, що для розробки власного ПЗ є всі підстави.

Напочаток проаналізуємо існуючі системи. За мету є визначити максимально вдале рішення та виявити недоліки та відмінності існуючих комплексів.

1.3.1 МІАС «EMSIMED»

Вона розроблена та використовується в ДУ «НП МЦ дитячої кардіології і кардіохірургії МОЗ України». Структура системи модульна, має інтегровані окремі модулі. В ній централізовано зберігається інформація, документальний реєстр, бази ЕМК, оцифровані медичні протоколи, реєстрограми та медичні зображення.

Модуль обліку надання медичних послуг забезпечує розробку відповідних звітних документів між відділами закладу, а також для автоматичного контролю. Автоматизація документів звітності про надані послуги, які потребують кількісного обліковування по суб'єктах в певному

часовому періоді. Цей модуль дозволяє відстеження кількості наданих послуг [7;8].

Базовими елементами ІС є українізований інтерфейс, логічне подання даних, навігація медичною картою, чітке розподілення рівнів доступу користувача.

Вона адаптована до вимог та специфіки закладу охорони здоров'я, а також доповнює та придатна до розширення або впровадження нових лікувально-діагностичних методів.

Систему можна «тонко» адаптувати індивідуально під кожного фахівця медзакладу. Модулі системи є взаємопов'язаними, а їх робота здійснюється з врахуванням єдиної схеми процесів в установі. Впроваджувати модулі можна послідовно, не порушуючи цілісність даних і не змінюючи звичайний інтерфейс користувача. Особливі вимоги щодо технічних характеристик, ПЗ, конфігурації відсутні [7].

В залежності від принципу роботи при одночасному вмиканні робочих станцій сервер є одно- або двопроцесорним на платформі Intel чи AMD із ОС Microsoft Windows Server 2003 і системою керування БД Microsoft SQL Server. Вони налаштовуються на базі офісних ПК з ОС Microsoft Windows XP Professional, яка працює з локальної мережі закладу (швидкість передачі даних до 1 Гбіт/с). Виконання кабельної мережі за технологією Star. Віддалений доступ надається через VPN. Шлюзування SMS-повідомлень здійснюється традиційним способом через оператора мобільного зв'язку.

1.3.2 Автоматизована медична білінгова система «МЕДІ»

АСМО «МЕДІ» – програмний комплекс, призначений для використання фахівцями медичних установ різного профілю. Вона складається із серії спеціалізованих модулів, реалізованих з урахуванням особливостей

відділення, спеціалізації персоналу та виконуваних функцій, для автоматизації роботи на окремих етапах технічного циклу медичних закладів. Кожен модуль є завершеним функціональним блоком, який функціонує незалежно і може працювати автономно з єдиною інформаційною базою в локальній мережі [9].

На практиці центральна робоча станція зазвичай розташована в лікарні, а робочі станції – в спеціалізованих відділеннях. Вони мають можливість обмінюватися даними одна з одною з використанням дискет або локальної мережі. Існує ряд звітів, необхідних для збору даних.

Основними перевагами та недоліками цієї системи є:

АСМО «МЕДІ» є гнучкою і адаптується до особливостей кожного медичного закладу, а формати звітів можна легко змінити при розширенні медичного закладу або впровадженні нових методів лікування чи дослідження. Крім того, є можливість конфігурувати модулі системи відповідно до індивідуальних потреб кожного спеціаліста.

Для встановлення програмного комплексу потрібна локальна мережа. До чотирьох робочих станцій можуть працювати з базою даних одночасно. Ви можете використовувати інтегровану однорангову мережу операційної системи Windows. Якщо у вашій мережі велика кількість ПК, вам слід використовувати мережу з виділеними серверами на основі технології NT (Windows NT/2000/XP). До 20 робочих станцій одночасно працюють з базою даних, або Novell (до 50 і більше).

Набір програм працює найефективніше, коли єдина база даних встановлена на сервері локальної мережі. При цьому доступні режими оперативного доступу до даних з різних модулів комплексу. Однак цей комплекс може працювати і на окремих комп'ютерах, не підключених один до одного в мережі. У цьому випадку збір даних з різних робочих місць здійснюється за допомогою пристроїв типу дискет або флеш-накопичувачів.

Для встановлення програмного комплексу вам знадобиться: сумісні

комп'ютери з принаймні 8 МБ оперативної пам'яті та 300 МБ вільного дискового простору, не менш потужні, ніж PC-486, і з операційною системою MS-DOS. Бажана конфігурація - P-II-300 з 64 МБ оперативна пам'ять та операційною системою Windows 95/98; ПК P-500/128 МБ і більш потужної операційної системи Windows 2000; ПК P-1000/256 МБ і більш потужної операційної системи Windows XP.

1. 4 Аналіз програмно-апаратного забезпечення

Програми, на високорівневих мовах, мають мало помилок, а робота програмістів є більш продуктивною за рахунок розроблення непроцедурних мов, які вказують, які дії мають бути виконані, але вони не описують детально, як виконуються дії. Використання їх дозволяє фундаментально переглянути наш погляд на створення орієнтованих на користувача програм, життєві цикли (ЖЦ) яких пропорційні очікуваному часу безвідмовної роботи. ЖЦ ПЗ — це постійний процес, від прийняття рішення про потребу ПЗ і до його «утилізації».

За останнє десятиліття з'явилися нові напрямки програмування: системи програмування CASE (ComputerAidedSoftwareSystemEngineering) і RAD (RapidApplicationDevelopment) [10]. Ця технологія характеризується аналізом, методом проектування, розробкою та підтримкою складених програмних систем, які комплексно підтримуються взаємопов'язаними автоматизованими засобами. Інструментарієм для є CASE, яка використовується в автоматизованому проектуванні та розробленні всіх етапів ЖЦ ПЗ [13].

Більшість з цих систем у своїй роботі ґрунтуються на структурному аналізі та методах проектування, в основі яких є візуальна побудова діаграм,

де графіки, діаграми, таблиці та діаграми використовуються для опису моделей спроектованої системи.

Розвиток КТ і необхідність ефективних засобів для розроблення ПЗ призвели до появи програмних засобів, зорієнтованих на т.з. «швидку розробку» RAD (RapidApplicationDevelopment). Його особливість в тому, що ним керує середовище розробки. Оскільки на нього припадає більшість генерації програмного коду, програмісту потрібно лише створити логічну частину програми та функції обробки подій. Оцінювання розміру програми відбувається з використанням функціональних елементів (повідомлення, звіт, файл тощо). Вони не мають залежності від мови програмування, використаної при розробці.

До основних принципів RAD належать: ітераційне розроблення програм, відсутність необхідності завершення роботи на кожному етапі ЖЦ, обов'язковість залучання користувачів до процесу розроблення ІС, використання інструментів керування конфігурацією для полегшення впровадження змін проекту та обслуговування готових систем, необхідне використання генератора коду, тестування та розробка проекту проводяться одночасно з розробкою. А технологія RAD дозволяє не тільки створити «правильний» продукт, а й забезпечити процес його створення. Їх метою є відокремлення дизайну програмного забезпечення від коду вання та наступних етапів розроблення і приховування всіх деталей розробки й функціонального середовища від розробника [12;5].

Окрім автоматизації CASE, технологія RAD забезпечує такі переваги: підвищується якість виробленого ПЗ за рахунок засобів автоматичного контролю (переважно проектного), дозволяє швидко створювати прототипи майбутніх систем і завчасно оцінювати потенційні проекти, прискорює процес проектування та розробки, звільняє розробників від повсякденних завдань і дозволяє їм повністю зосередитися на творчій частині розробки, підтримує

розвиток та супровід ПЗ, підтримує технологію повторного використання для компонентів розробки.

Останнім часом все більшої популярності набувають так звані «гнучкі» («живі») методології розробки ПЗ (гнучка розробка ПЗ). Найпопулярнішим з них є Extreme Programming (XP). Це простий (гнучкий) процес (або методологія), який передбачає той факт, що клієнти постійно змінюють свої вимоги до системи, і відповідні вимоги можуть не збігатися. Основними особливостями цієї технології є надзвичайно незначний дизайн і максимальний час програмування, досягнутий завдяки коротким і частим ітераціям. Замовники та розробники постійно спілкуються, щоб визначити нові вимоги та конкретизувати завдання [13].

Інструменти та технології розробки, перераховані вище, вимагають сучасного апаратного забезпечення, яке відповідає таким мінімальним вимогам: процесор - Intel Pentium 166 МГц або вище; RAM – 256 Мб або більше; принаймні 475 МБ дискового простору для середовища; монітор – роздільна здатність VGA або вище; маніпулятор "Мишка".

Ці вимоги базуються на вимогах багатозадачної операційної системи MS Windows. На основі досвіду представлені конфігурації, достатні для нормальної роботи інструменту розробки.

1.5 Постановка задачі

При здійсненні фінансово-господарської діяльності використовується велика кількість документів і звітів, які сьогодні не можуть бути оброблені без використання автоматизованих систем. Крім того, існуючі системи автоматизації намагаються охопити великий обсяг роботи, оскільки точна робота вимагає доступу до великої кількості баз даних і каталогів [1]. Претензії

на медичні послуги мають особливе значення в діяльності медичних установ. Недостатньо знати, скільки послуг надано. Має бути якісне керування ними. Визначення надлишкових послуг, оптимальне виконання їх, керування необхідними послугами тощо. Неякісно надані не тільки впливають на фінансові показники, але й можуть бути негативним соціальним фактором [2].

Зосереджуючись на проблемній зоні, ми бачимо, що сектор ринку інформаційно-технологічного забезпечення закладів охорони здоров'я на даний момент представлений великою кількістю різноманітних програмних продуктів. Однак не всі існуючі програмні засоби спеціально розроблені або адаптовані для конкретних аспектів діяльності організації охорони здоров'я. Перед початком розробки враховуються найбільш вдалі рішення, а також недоліки, виявлені в роботі існуючих комплексів, після аналізу існуючих систем.

Постановкою завдання є – розробка та побудова автоматизованої системи обліку медичних послуг на основі сучасних комп'ютерних та інформаційних технологій.

До основних функцій системи відносять: створення та керування БД; контроль доступу до даних та віддалений доступ в локальній мережі; уникнення втрати інформації; вести оперативний облік медичних послуг; внесення, зміна та видалення даних про надані послуги; ведення та друк карток обліку медичних послуг; розташування звітнього документа за обраним періодом або обраними критеріями.

Системи автоматизації повинні відповідати таким характеристикам: простота освоєння програми та простота роботи з програмою; організація зручної взаємодії між комп'ютерами та користувачами; якість дизайну екрану; відкритість до змін і доповнень через наступні версії та подальший розвиток; можливість перенесення даних з попередніх версій програми.

Оскільки простота введення є критичним фактором успіху програми,

інтерфейс користувача програми, а також власна структура бази даних і алгоритми програми повинні бути ретельно розглянуті та ретельно проаналізовані. Отже, щоб відповідати всім сучасним вимогам, програми потребують сучасного інтерактивного віконного інтерфейсу. Використання інтерактивного віконного інтерфейсу дає такі переваги: - фокусування уваги користувача на введені дані; вказує дії, які користувач може виконувати в даний момент; приховування неактивних елементів інтерфейсу на екрані.

2. РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Призначення програми

Програма обліку медичних послуг призначена для проведення бази даних клієнтів, результатів обстежень, історій хвороб, висновків. Отримані під час обстеження дані зберігаються у таблицях бази даних у текстовому, числовому, графічному форматах, соціальній та DOC-файлах. Доступні прості та розвинені засоби налаштування інтерфейсу користувача та обробки списків клієнтів, лікарів, відділень, обстежень, а також списки користувача з додатковою інформацією.

Стандартні інструменти роботи з таблицями включають застосування сортування, фільтрації, угруповання та пошуку по будь-якому полю. Є можливість створювати нові поля як зі збереженою, а й обчислюваної за формулами типу SELECT інформацією, і навіть створювати поля типу "картинка". Програма дозволяє будувати дерево даних по заданим полям, рахувати підсумки за заданими формулами, задавати правила колірного виділення рядків, визначати спадні списки і значення в них, визначати підлеглі таблиці з типом зв'язку "один-багатьом" або "багато-багатьом".

Будь-яку таблицю можна роздрукувати, експортувати до MS Excel, MS Word або текстового формату CSV. Є імпорт із інших джерел даних. І багато інших зручностей.

У програмі Таблиця це не лише форма подання даних у вигляді рядків та стовпців, це спеціальний об'єкт, який має внутрішнє ім'я, підпис або назву та інші параметри.

Пациєнти									
№	ІПБ	Дата народження	Стать	e-mail	Номер картки	Номер	Станов	Статус	Діл.
1	Малий Іван Іванович	31.12.1968	ч	iv@sample-so	6682/101	3865/201	Кіданс	амбулаторний	
2	Колодій Петро Васильович	24.07.1974	ч	paio@sample-so	3054/102	5830/202	Аскама	амбулаторний	
3	Лисий Ігор Ількович	24.08.1960	ч	iv2k@yandex	3711/103	3097/203	Аскама	амбулаторний	

Рисунок 2.1 – Приклад таблиці

Поле є об'єктом для зберігання одного виду даних. У таблиці полю відповідає стовпець з ім'ям поля (назвою). Також поле використовується при введенні даних за допомогою форми

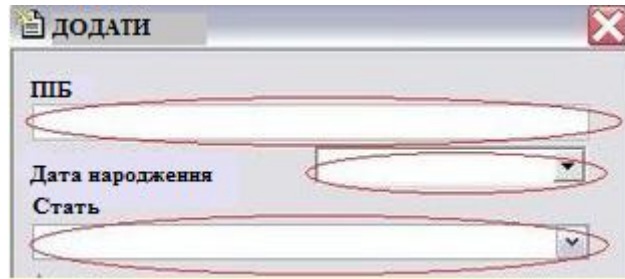


Рисунок 2.2 – Поле для введення даних

Запис – набір даних усіх полів по одній одиниці зберігання. У таблиці запису відповідає рядок, кожен запис має свій унікальний ID (число в полі ID).

Фільтр є умовою або набір умов, за якими програма відбирає відповідні записи. При увімкненому фільтрі відображаються лише записи, які відповідають заданій умові.

Зберігається уявлення є об'єкт на основі Таблиці, в якому є поля, що обчислюються. Використовується, якщо потрібно відобразити в таблиці поля, що обчислюють. Існують деякі обмеження роботи зі збереженими уявленнями. Оновлення прямо в таблиці не завжди можливе. Якщо неможливо, поряд з назвою таблиці написано "подання, що не оновлюється". У цьому випадку оновлення даних можливе лише через форму. Зберігання (якщо ще не створено) створюється автоматично при створенні першого обчислюваного поля і може бути відкрито з меню "Таблиці". Можна створити вкладку для збереження представлення, ввівши її внутрішнє ім'я Сервіс>Налаштування, як при створенні звичайних вкладок.

Форма є діалогове вікно, в якому вводяться або редагуються дані таблиці. Щоб увімкнути введення та редагування за допомогою форми, поставте галочку у відповідному полі в меню Сервіс>Налаштування.

У формі відображаються лише ті поля та в тому порядку, які відображаються в даний момент у таблиці. Якщо Ви створили багато полів і не всі з них видно у формі – збільшіть розмір вікна, розтягнувши його за нижній правий кут. Якщо Вас не влаштовує положення та розмір полів у формі, Ви можете змінити їх розмір у налаштуваннях. Шаблон є звичайним документом Word, в який вставлені закладки - це спеціальні маркери за якими "Облік клієнтів" визначає, в яке місце документа вставити дані з відповідного поля таблиці.

Створення висновків у Word за шаблоном

Передбачається, що Ви створили шаблони своїх висновків.

Виберіть потрібного Вам клієнта та встановіть курсор на запис про обстеження, за яким ви хочете зробити висновок.

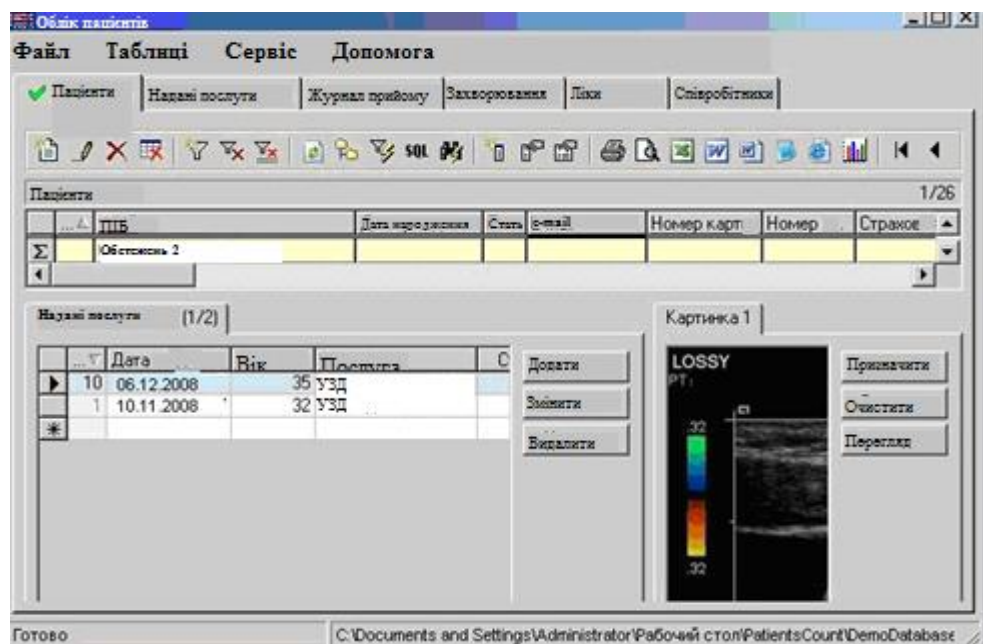


Рисунок 2.3 – Загальний вигляд вкладки «Клієнти»

Тепер натисніть кнопку  на панелі інструментів.

У діалоговому вікні вкажіть файл-шаблон.

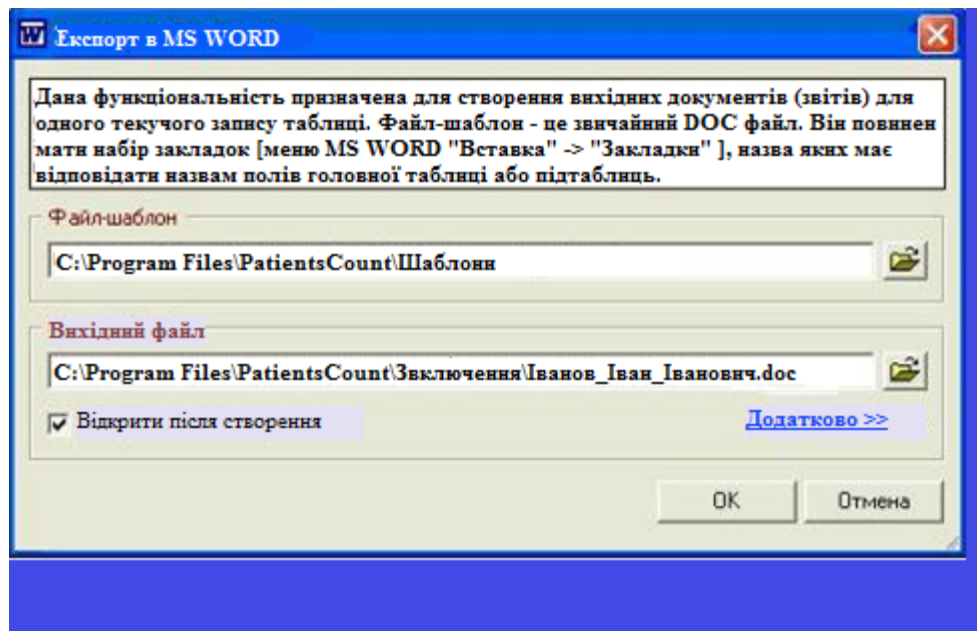


Рисунок 2.4 – Створення файлів-шаблонів

Якщо стоїть галочка на Відкрити після створення, то після натискання на ОК, висновок буде відкрито Word для редагування.

Після редагування висновку в Word необхідно зберегти зміни в документі як за нормальної роботи з текстом.

Додаткова інформація

- Ім'я файлу-висновку створюється автоматично використовуючи дані про клієнта (параметри автоматичного створення імені файлу змінити натиснувши на Додатково>>>), також шлях до файлу-висновку автоматично зберігається в спеціальному полі таблиці "Обстеження"

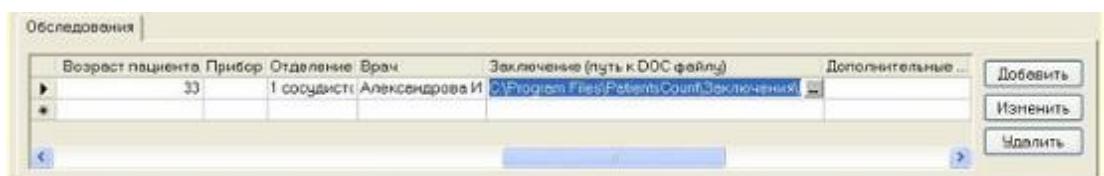


Рисунок 2.5 – ОФормування файлу-висновку

При повторному візиті клієнта як шаблон може бути використано попередній висновок, для цього потрібно вказати його у відповідному вікні як шаблон.

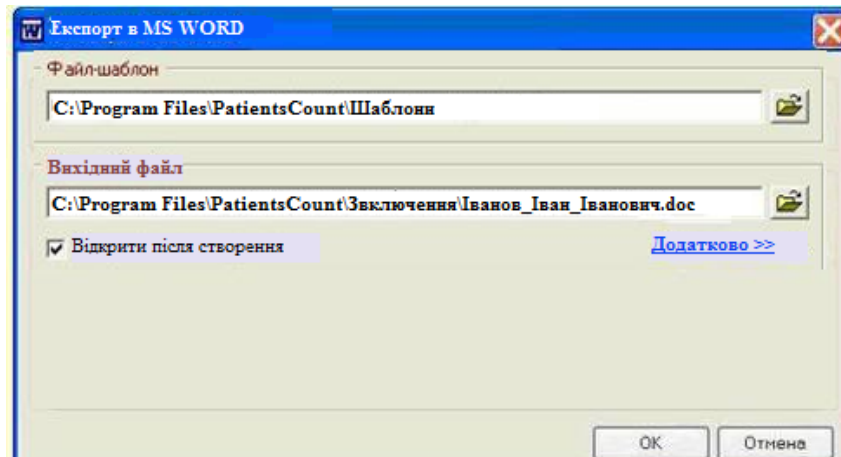


Рисунок 2.6 – Експорт в інший додаток

2.2 Відбір записів для аналізу

Вибір записів здійснюється за допомогою фільтрів.

Фільтри							Всього 2 (включено 2)
	Вкл.	Зв'язок	Відкрито	Поле	Умова	Значення	Закрито
	<input checked="" type="checkbox"/>			Дата народження	>=	28.07.1955	
	<input checked="" type="checkbox"/>	і		Стать	=	М	
*	<input type="checkbox"/>						

Рисунок 2.7 – Таблиця з фільтрами для зберігання умов відбору записів основної таблиці

Натисніть кнопку "Додати фільтр" на панелі інструментів і додайте фільтр по підлозі: "Пол = м". Натисніть кнопку "Оновити", зауважте, що загальна кількість показаних записів поменшала. Продовжуємо... "Дата

народження >= 01.01.1973", оновлюємо, і таке інше.

Використовуйте точку з комою як роздільник кількох значень для одного фільтра, наприклад "Відділ = Бухгалтерія; Фінансовий" або "Відділ <> Бухгалтерія; Фінансовий". Якщо вказати ще в кінці крапку з комою, додадуться записи з порожніми значеннями цього поля.

За допомогою оператора LIKE та спецсимволів *#? можна встановити діапазони значень. Якщо Вам це незнайомо, наведіть мишкою на оператор "LIKE", вибраний у полі "Умова" у таблиці фільтрів, зачекайте 2-3 секунди, і ви отримаєте вичерпну підказку щодо його використання. Але все ж таки, в більшості випадків можна обійтися і без нього, оскільки для текстових полів у програмі передбачені оператори "Утримує", "Не містить", "Починається з", "Не починається з", "Закінчується на", "Не закінчується на".

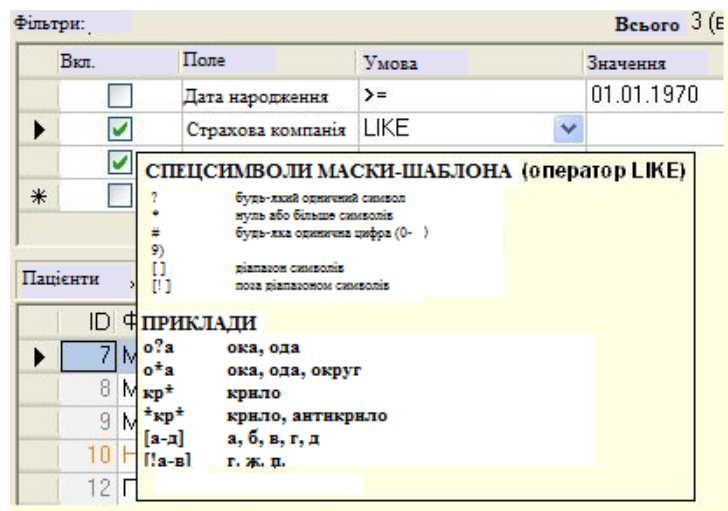


Рисунок 2.8 – Фільтр для зберігання умов відбору записів основної таблиці

Відібрані записи можна експортувати до Excel для подальшого аналізу. Експортуються лише поля, що відображаються в даний момент у таблиці, також виробляється експорт картинок.

Вибрані записи можна надрукувати.

Для зручнішої роботи з фільтрами можна увімкнути додаткові функції у налаштуваннях програми.

Використовуйте фільтри, щоб показувати потрібні записи та приховувати непотрібні. Занадто багато фільтрів типу LIKE знижує продуктивність запиту.

Якщо ви працюєте з великою кількістю фільтрів і різних баз даних, зберігайте фільтри у файл для зручного перенесення їх між базами даних.

Приховуйте непотрібні поля, щоб основна таблиця була не надто широкою, оскільки вони постійно відволікатимуть ваш погляд і таблицю незручно прокручувати зліва направо. Контекстне меню правого кліку на заголовках полів дозволяє приховувати/показувати поля.

Змінюйте назви полів відповідно до специфіки вашого бізнесу.

Введіть коментарі в полі "Нотатки" або створюйте нові збережені поля.

Створюйте поля, що обчислюються, на основі значень інших полів (наприклад, можна створити поле з формулою "[Поле 1] / [Поле 2]") або інших таблиць, використовуючи підзапит select... Наприклад, можна вивести телефон організації-постачальника задавши формулу (select [CompanyPhone] from [tblCompanies] where [tblCompanies].[CompanyName] = [tblMain].[Supplier]).

Видаляйте дубльовані записи, вибравши пункт меню. Якщо увійти в налаштування, з'явиться форма "Групове оновлення", де можна підкоригувати інструкцію SQL. Можливо, ви захочете не видаляти дубльовані записи, а лише відзначити їх як мертве (або як обране), мабуть, для подальшого видалення вручну. Якщо ви просунутий користувач, знайомий з мовою запитів SQL, ви можете самі писати і виконувати будь-які інструкції SQL, але будьте обережні.

Створюйте нові таблиці в базі даних, якщо це необхідно. Наприклад, "Контакти", куди заносити всіх відомих вам контактних осіб та їх телефонні номери. Потім можна буде додати обчислюване поле, яке виводитиме якесь поле цієї таблиці в основну таблицю.

Створюйте підлеглі таблиці для основної та інших таблиць, якщо потрібно. Наприклад, можна створити таблицю "Історія" та вказати її як підлеглу для основної таблиці. Далі для кожного запису основної таблиці можна буде додавати до цієї таблиці необмежену кількість записів з різною хронологічною інформацією.

Записи БД можна легко копіювати через буфер як завгодно і будь-куди, якщо відкрити базу даних за допомогою MS Access.

Ви можете вести базу даних, вводячи дані з клавіатури або копіюючи їх через буфер або імпортувати їх, для прикладу, з CSV-файлу.

З часом з якихось причин дані в базі даних можуть почати дублюватися, якщо ви не перевіряєте на повторюваність, коли додаєте, або, скажімо, після імпорту. Видалити дубльовані записи можна, вибравши пункт меню. Видається повідомлення "Дублікати записів із схожими значеннями полів "ПІБ" і "Дата народження" будуть видалені. Залишаться тільки їх останні версії (за більшим ID). Продовжити?" Ви можете продовжити або увійти в режим налаштування SQL, яка буде виконана. Також можна не видаляти дублі, а тільки відзначити їх як "Вибрані", ймовірно, для подальшого видалення вручну. У цьому випадку замініть SQL- інструкцію з

```
DELETEFROM [tblMain]T1 WHERE EXISTS(SELECT * FROM [tblMain]T2
WHERE (T2.Patient = T1.Patient) AND (T2.DateOfBirth =
T1.DateOfBirth) AND (T2.ID > T1.ID))
```

на цю:

```
UPDATE [tblMain]T1 SET [Favorite] = 1 WHERE EXISTS DELETE
FROM [tblMain]T1 WHERE EXISTS (SELECT * FROM [tblMain]T2 WHERE
(T2.Patient = T1.Patient) AND (T2.DateOfBirth = T1.DateOfBirth) AND (T2.ID > T1.ID))
```

Ви можете змінювати значення вручну для одного запису, а можна скористатися функціональністю "Групове оновлення". Допустимо, ви хочете змінити значення поля "Статус клієнта" з "амбулаторний" на "хворий" для всіх записів відразу. За допомогою цієї форми це легко зробити.

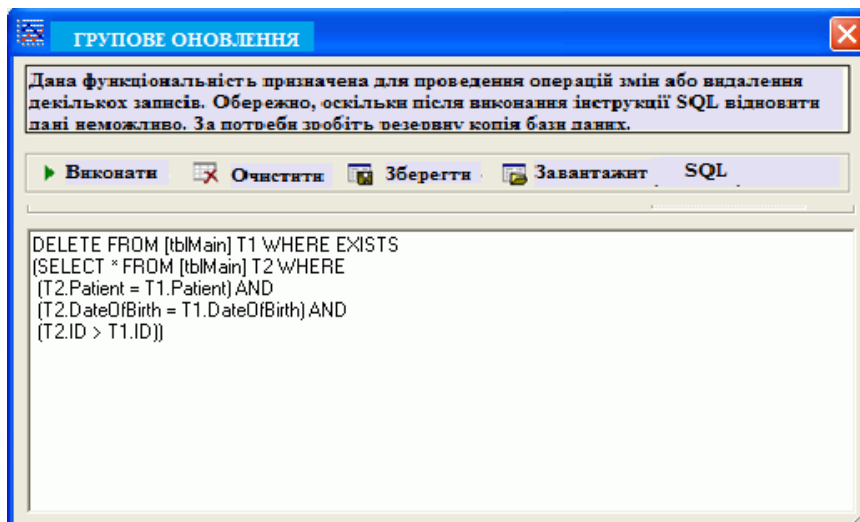


Рисунок 2.9 – Групове оновлення при змінах запису

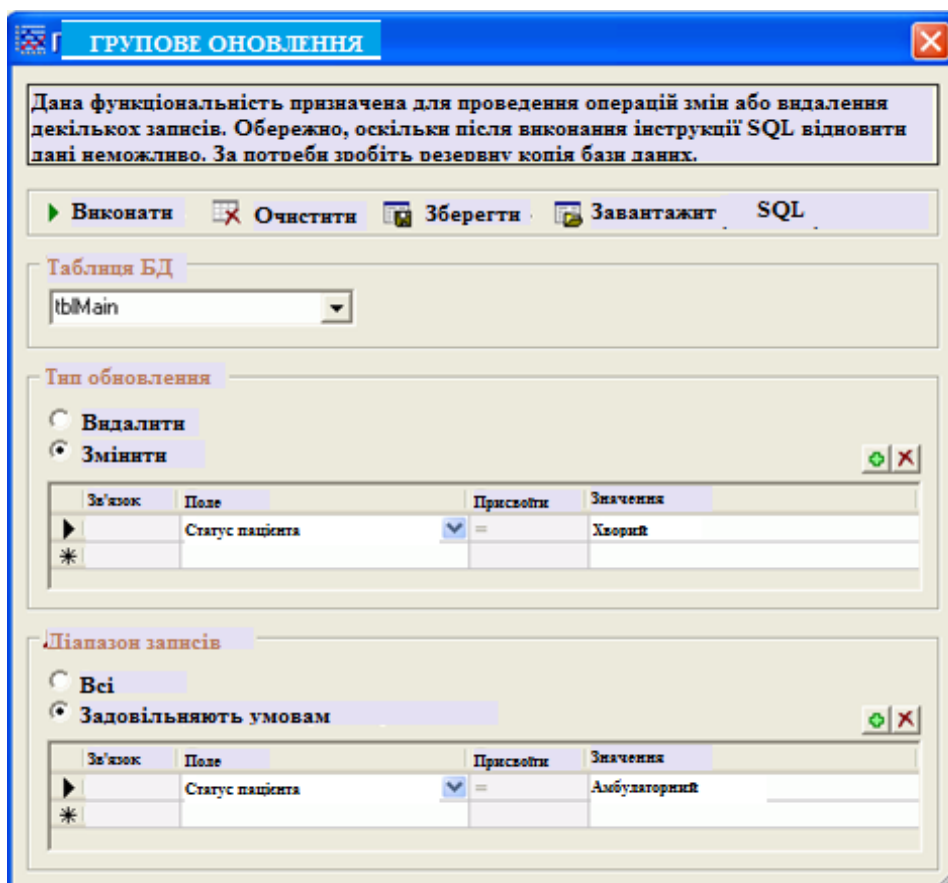


Рисунок 2.10 – Оновлення за критерієм

Імпорт даних можна проводити з інших джерел. Формат - текстовий CSV-файл, можна встановити роздільник полів. За замовчуванням розділювач - точка з комою. Імпортувати можна не лише клієнтів, а й будь-яку іншу інформацію. Ви самі вибираєте потрібну таблицю бази даних і задаєте відповідність полів на кшталт "звідки" - "куди". Під час зіставлення необхідно пам'ятати про тип полів, щоб не виходило, що ви намагаєтеся імпортувати в числове поле, наприклад, текст .

Для створення вихідних документів (звітів) для одного (поточного) запису таблиці можна зробити експорт у MS Word. Файл-шаблон повинен мати набір закладок (меню MS Word "Вставка" -> "Закладки"), назви яких повинні відповідати назвам полів у таблиці або підтаблицях а також інша текстова інформація, зображення тощо.

Експортувати також можна будь-яку таблицю. У MS Excel експорт виконується одразу без допоміжних форм. Зображення також експортуються. Для експорту в текстовий формат CSV видається форма, де ви вибираєте цільовий файл, задаєте роздільник полів.

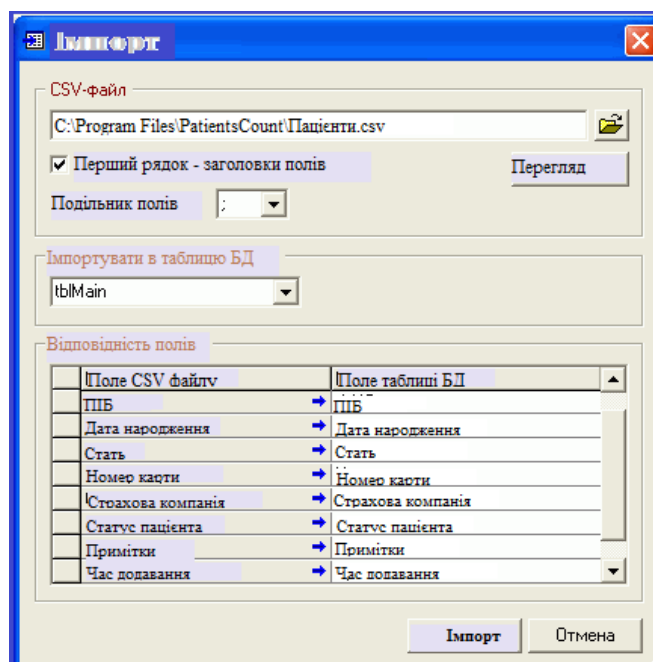


Рис. 2.11. Імпорт даних

Можна задавати будь-який символ-розділювач, а не тільки присутні у спадному списку. Також можна експортувати назви (заголовки) полів першим рядком. Експортуються лише видимі поля.

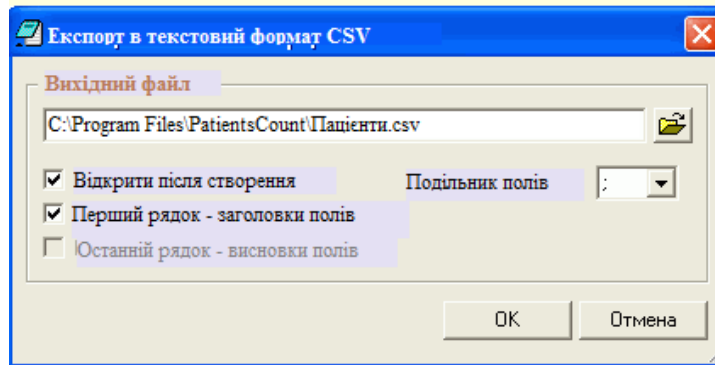


Рисунок 2.12 – Експорт у текстовий формат

Спецсимволи відіграють у процесі аналізу, побудови запитів, роботи з фільтрами, пошуку. За допомогою спецсимволів ви визначаєте діапазони (безліч) значень.

?	- будь-який одиночний символ
*	- нуль або більше символів
#	- будь-яка одиночна цифра (0-9)
[]	- Діапазон символів
[!]	- поза діапазоном символів

приклад

o?a	- Ока, ода
o*a	- ока, ода, пофарбована
кр*	- червоний
кр]	- червоний, антикрило, підкрилки
[а-д]	- а Б В Г Д
[!а-в]	- г, ж, п

a[!б-д]	-аж, апарат
---------	-------------

Існує деяка плутанина з приходом Microsoft технології доступу до даних ADO. Якщо ви самі пишете SQL-інструкції, то скрізь

замість *	- потрібно використовувати %
замість?	- потрібно використовувати _
місце #	- потрібно використовувати [0-9]

Облік клієнтів дозволяє прив'язувати до обстежень інформацію про зображення у форматі DICOM, які зберігаються в базі добре відомої програми для роботи з такими зображеннями – eFilm. Таким чином, можна з Обліку клієнтів давати команду eFilm відкрити зображення, що відповідають обраному обстеженню. Підтримується робота з версіями eFilm пізніше 1.8.

Для того, щоб Облік клієнтів міг відкривати в eFilm потрібні Вам зображення, необхідно зробити таке:

Створити в таблиці обстеження поле "Код для eFilm" з такими властивостями: тип даних "текстовий", унікальні значення "включено", значення за замовчуванням "SELECT MAX(ID) + 1 FROM tblSurveys" (вставити без лапок, для того, щоб унікальні значення генерувалися автоматично або залишити порожнім, якщо ви ведете свою нумерацію досліджень у приладі)

У полі "Код для eFilm" необхідно вводити таке ж значення, яке Ви вводите в пристрій як ID. (В eFilm це значення відображається у стовпці PatientID).

Створити кнопку на панелі інструментів. З наступними параметрами (вводити без лапок): найменування кнопки - "eFilm", шлях до файлу користувача - "script.vbs /[tblSurveys].[внутрішнє ім'я створеного вами поля "код для eFilm"]

Передбачається наступний порядок роботи.

Коли клієнт приходить на обстеження, Ви спочатку заводите його дані до Обліку клієнтів, додаєте обстеження - у цей момент у полі "Код для eFilm" генерується значення (або Ви його вводите самі).

У приладі вводиться значення "Код для eFilm", проводиться обстеження клієнта

В обліку клієнтів запроваджуються додаткові дані про обстеження (якщо потрібно), пишеться висновок.

Для перегляду зображень необхідно вибрати в Обліку клієнтів потрібне обстеження та натиснути кнопку eFilm панелі інструментів.

Буде поставлене запитання: де розташоване зображення? Натисніть "Локальний компютер" якщо база eFilm розташована на цьому компютері або "на віддаленому сервері", якщо Ви хочете відкрити зображення, яке зберігається на іншому комп'ютері.

Є можливість додавати на панель інструментів кнопки, які запускають будь-які інші програми. Для додавання кнопки потрібно:

Обрати вкладення, до якого потрібно додати кнопку.

Навести курсор на панель інструментів та натиснути на праву кнопку мишки.

Вибрати "Додати кнопку"

Ввести ім'я кнопки, яке буде показано при наведенні на неї мишею.

Ввести шлях до виконуваного файлу. (якщо ввести лише ім'я файлу типу file.exe, то він запускатиметься лише з робочого каталогу Обліку клієнтів)

Щоб видалити кнопку, зробіть таке:

Виберіть вкладку, з якої Ви бажаєте видалити кнопку.

Наведіть курсор на панель інструментів та натисніть на праву кнопку миші.

Виберіть "Видалити кнопку ..."

2.3 Основне меню

Основне меню знаходиться у верхній частині головного екрана. Меню містить:

–Файл

– Створити базу даних – на основі існуючої бази створюємо нову: копіюємо файл поточної БД та очищаємо всю інформацію в ньому, крім службової. Тому, неможливо створювати нові БД, не маючи жодної відкритої БД. Не потрібно створювати нові БД, маючи відкриту БД якоїсь старої версії. У дистрибутиві завжди є файл DemoDatabase.mdb гарантовано останньої версії.

Відкрити базу даних – відкриваємо вказану базу даних. Якщо БД старої версії, видається попередження. Зі старими БД Ви працюєте на свій страх і ризик. Як правило, перегляд інформації можливий, а ось зміна може десь не працювати.

Створити резервну копію – створюється резервна копія бази даних.

Імпорт - імпортуємо інформацію у обрану таблицю баз даних із зазначеного текстового CSV-файлу. Необхідно буде задати відповідність полів на кшталт "звідки" -> "куди".

– Новий екземпляр програми – запуск ще одного екземпляра цієї програми. Кількість запущених екземплярів не обмежена.

Таблиці

Назва таблиці – при натисканні на назву таблиці вона відкривається в окремому вікні для налаштування та редагування.

Додати таблицю – створення нової таблиці.

Видалити таблицю – видалення таблиці

- Сервіс

Видалити дублі - всі дубльовані записи з однаковим ПІБ клієнта та датою народження видаляються. Залишаються лише їхні останні версії (за більшим ID). При натисканні кнопки "Налаштування" відображається SQL-інструкція, яка буде виконана.

Групове оновлення - дана функціональність призначена для проведення операцій зміни або видалення відразу декількох записів. Наприклад, можна відзначити записи як обране, якщо страхова компанія клієнта "Кадіс".

Налаштування - глобальні налаштування програми, які є спільними для всіх баз даних та зберігаються у системному реєстрі.

– Допомога.

Вкладення «Клієнти» складається з наступних основних частин:

- панель інструментів, де розташовані всі кнопки з впливаючими підказками.

- Таблиця з фільтрами для зберігання умов відбору записів основної таблиці. Щоб показувати в повному обсязі записи таблиці, лише задовольняють певним критеріям, використовуються фільтри. Натисніть кнопку "Додати фільтр" на панелі інструментів і додайте фільтр «Стать». Натисніть кнопку "Оновити", зауважте, що загальна кількість показаних записів поменшала. «Дата народження >= 01.01.1973», оновлюємо, і т.д.

- основна таблиця, де відображаються усі об'єкти обліку

- Дерево для ієрархічного представлення даних основної таблиці на основі будь-якого поля (або полів).

Фактично використання дерева є видом фільтрації та сортування, тому допомагає швидко відобразити записи, що задовольняють будь-якій умові, наприклад, всіх клієнтів клієнтів однієї страхової компанії або хворих з однаковим діагнозом.

Дерево будується на основі даних основної таблиці. У властивостях таблиці ви задає поля для рівнів (гілок) дерева. Навряд чи потрібно будувати

дерево більш ніж 2-3 рівневе. Дані в полях для дерева мають бути повторюваними. Немає сенсу будувати дерево по полях з унікальними даними. При натисканні на гілки дерева відбувається автоматична фільтрація по цій гілці (і всім її батьківським гілкам).

Ця таблиця є повним списком обстежень всіх клієнтів. Передбачається, що ця таблиця використовується головним чином для статистичної обробки даних з усіх обстежень. А саме, для відбору за допомогою фільтрів обстежень, що задовольняють певній умові, наприклад, зроблені в заданий проміжок часу або одним лікарем або з певною патологією (якщо вона закодована в спеціальному полі) і т.д. Після додавання та налаштування фільтра(ів) потрібно натиснути на оновити і будуть відображені тільки записи, що задовольняють умові фільтра, кількість цих записів відображається у рядку стану вгорі у правій частині заголовка таблиці обстеження. Відібрані записи можна роздрукувати, експортувати до Excel, а потім, за потреби, до програми статистичної обробки. Таблиця «Надані послуги» містить ПІБ клієнта, що є обчислюваним полем, взятим з іншої таблиці. Таким чином, зміна записів на вкладці недоступна. При необхідності змінити дані про використовується вкладка "Клієнти" з підтаблицею «Послуги». За допомогою цих вкладок можна відсортовувати, фільтрувати, видаляти записи; додавати поле для обчислення чи збереження даних; проводити пошук, виведення, експорт даних в Word чи Excel.

Вкладка «Організації» має таку саму функціональність і ті самі можливості, як і основна вкладка. Ви можете сортувати, фільтрувати, додавати, змінювати, видаляти записи; додавати поля, що зберігаються або обчислюються; здійснювати пошук, друкувати, експортувати дані.

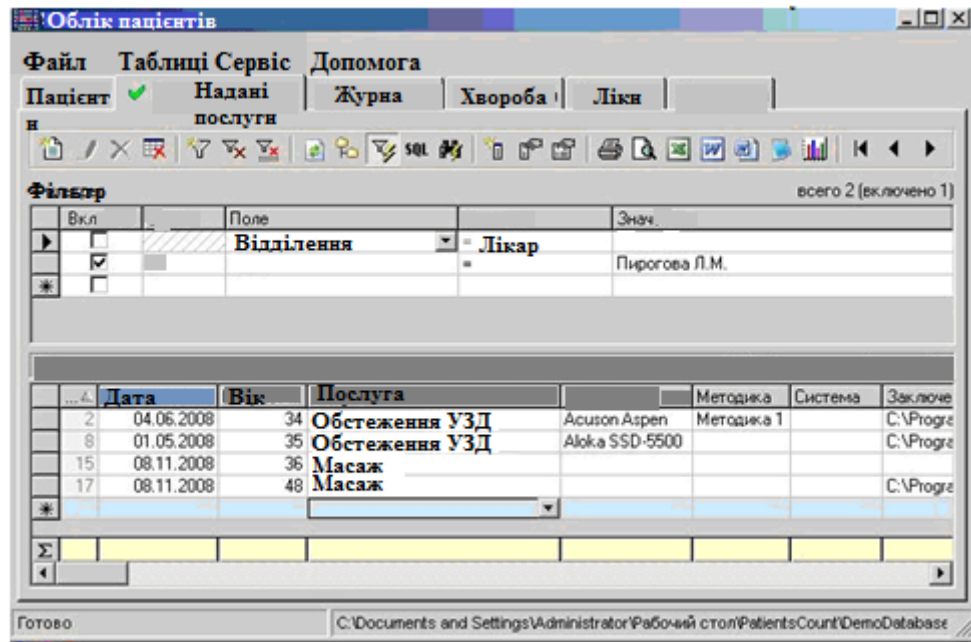


Рисунок 2.13 – Вкладка «Надані послуги»

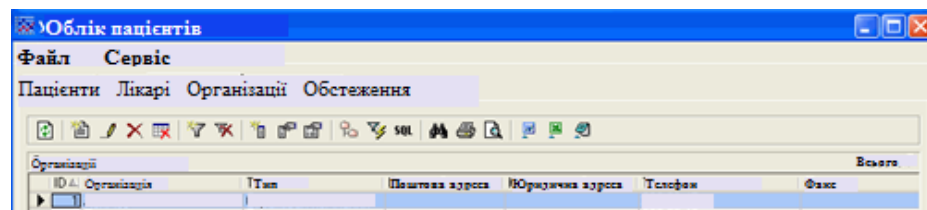


Рисунок 2.14 – Вкладка «Організації»

Ви можете створювати нові вкладки, які будуть відображатися в основній формі, на основі будь-якої таблиці бази даних або уявлення, що зберігається. Це робиться у формі "Параметри" натисканням кнопки "+" (Додати) для списку вкладок.

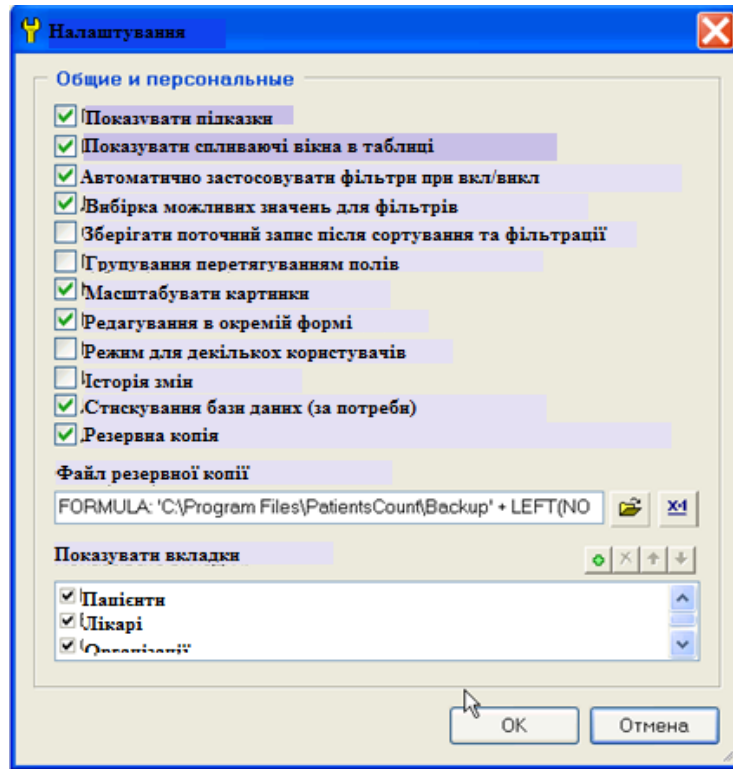


Рисунок 2.15 – Налаштування

Таблиці, що постачаються з програмою

tblMain	Основні таблиці з клієнтами.
tblDoctors	Таблиця "Лікарі". Дані з цієї таблиці також відображаються в спадному списку під час редагування поля "Лікар" таблиці "Обстеження".
tblDepartments	Таблиця "Відділення"
tblCompanies	Таблиця "Організації". Дані з цієї таблиці також відображаються у спадному списку під час редагування поля "Страхова компанія" основної таблиці "Клієнти".
tblSurveys	Таблиця "Обстеження"

Побудова нової таблиці

Увійдіть у меню Сервіс⇒Таблиці бази даних⇒Додати таблицю

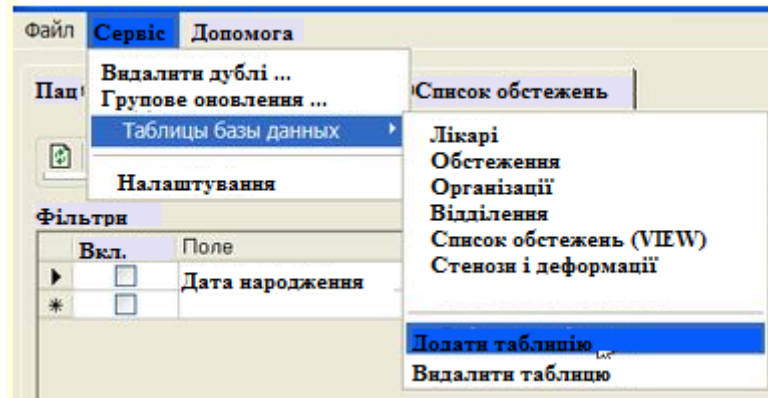


Рисунок 2.16 – Додавання таблиць

У діалоговому вікні вкажіть підпис (назва таблиці), внутрішнє ім'я таблиці генерується автоматично, але за необхідності ви можете ввести його самостійно.

Щоб відкрити нову таблицю, увійдіть у меню Сервіс⇒Таблиці бази даних⇒Назва таблиці.

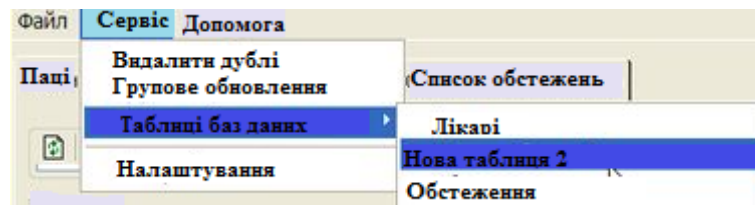


Рисунок 2.17 – Відкриття нової таблиці

Таблиця буде відкрита в окремому вікні, тепер ви можете додавати до неї нові поля та записи.

Видалення таблиці

Увійдіть у меню Сервіс>Таблиці бази даних>Видалити таблицю. Якщо у вас є пов'язані таблиці, спочатку потрібно видалити їх, а потім основну таблицю.

Можна створювати нові поля двох основних типів:

Збережені

Обчислювані

Зберігаючі зберігатимуть введену інформацію в собі. Обчислювані призначені, щоб обчислювати значення з урахуванням значень інших полів. Зберігаючі зберігають дані (інформацію) наступних типів:

Текстовий – до 255 символів

Великий текст – понад 255 символів

Числовий

Так/Ні

Дата та час

Картинка

Важливо правильно вибрати тип поля, щоб записи правильно сортувалися та фільтрувалися. Значення в текстових полях вирівнюються по лівому краю колонки (поля), у числових - по правому, Так/Ні - галочка по центру.

Налаштування відображення полів.

Якщо Ви хочете, щоб якісь поля не відображались і при цьому Вам не потрібно їх видаляти (наприклад, поля, що зберігають інформацію для зв'язку з підлеглими таблицями, або які Ви поки не заповнюєте, але можливо будете в майбутньому), то можна сховати непотрібні поля/ стовпці, для цього за потрібно зняти галочки з непотрібних полів за допомогою мишки.

Для того, щоб змінити порядок полів у таблиці (порядок стовпців), натисніть лівою кнопкою миші назву поля (підпис стовпця) і перетягніть вправо або вліво на потрібне місце.

Внутрішнє ім'я поля генерується автоматично, і його можна не змінювати. Внутрішнє ім'я використовується при створенні формул для обчислюваних полів, а також як імена закладок, при створенні шаблону для експорту в Word.

Назва поля (підпис) - те, що буде відображатися в назвах стовпців у

таблиці та як підписи до полів при введенні за допомогою форми. Обов'язкове для заповнення поле - при включенні цієї властивості, програма не буде працювати до заповнення.

3. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Питання охорони праці людини необхідно вирішувати на всіх стадіях трудового процесу незалежно від виду професійної діяльності.

Забезпечення безпечних і здорових умов праці в значній мірі залежить від правильної оцінки небезпечних, шкідливих виробничих факторів. Однакові по складності зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища, надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин.

3.1 Охорона праці

При розробці програмного забезпечення інформаційної системи управління підприємством використано приміщення загальною площею 20 м^2 , висотою стелі 3 м , де розташовано 6 робочих місць з ПК. Кожне робоче місце обладнане робочим столом площею $1,2 \text{ м}^2$, стільцем та персональним комп'ютером, що складається з монітора, системного блоку, клавіатури та миші. Слід відзначити, що площа одного робочого місця не повинна бути меншою за 6 м^2 , а об'єм не менший за 20 м^3 , тобто площі та об'єму даного приміщення вистачає для розташування шести робочих місць.

Аналіз умов праці показує, що на програміста можуть негативно впливати наступні фізичні та психофізіологічні фактори:

- підвищена або знижена температура повітря робочої зони;
- підвищена або знижена вологість повітря;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- підвищена іонізація повітря;

- підвищений рівень електромагнітних випромінювань;
- нервово-психічні перевантаження (розумова перенапруга, перенапруга аналізаторів);
- фізичні перевантаження (одноманітна поза викликає статичну втому).

Робота програміста за енерговитратами відноситься до категорії легких робіт Іа, Іб, тому повинні дотримуватися наступні вимоги згідно ДСН 3.3.6.042-99:

- оптимальна температура повітря – 22°C (допустима – 20-24°C);
- оптимальна відносна вологість – 40-60% (допустима – не більше 75%);
- швидкість руху повітря не більш 0,1 м/с.

Виміряні за допомогою приладів (психрометр Августа) температура та вологість у лабораторії відповідають вказаним у таблиці для теплого періоду року.

Розташовані у приміщенні 6 ПК є джерелами тепловиділень, крім того для підтримання у приміщенні в холодний період року оптимальних параметрів мікроклімату використовуються нагріті поверхні опалювальної системи. Нормованим показником ІЧВ є гранично допустима густина потоку енергії $J_{г.д}$, Вт/м², яка встановлюється в залежності від площі опромінюваної поверхні тіла людини ($S_{опр}$). Нормовані рівні складають: $J_{г.д}=35$ Вт/м² при $S_{опр} > 50\%$; $J_{г.д}=70$ Вт/м² при $S_{опр} \sim 25-50\%$; $J_{г.д}=100$ Вт/м² при $S_{опр} < 25\%$.

Нормованим параметром природного освітлення згідно ДБН В.2.5–28 – 2006 є коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт. Робота програміста відноситься до робіт середньої точності (IV розряд зорових робіт, мінімальний розмір об'єкту розрізнення складає 0,5-1,0мм), для яких при використанні бокового освітлення КПО=1,5%. Для штучного освітлення нормованим параметром виступає $E_{мін}$ – мінімальний рівень освітленості, та $Kп$ – коефіцієнт пульсації світлового потоку, який не повинний бути більшим ніж

20%. Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 300-500 лк. Для забезпечення штучного освітлення використано світильники типу ЛПО. Кожен світильник комплектується двома лампами. Тобто необхідно використовувати 6 світильників із 12 працюючими лампами в них.

Оскільки при розробці програмного забезпечення інформаційної системи управління підприємством використовуються ПК, кожен з яких устаткований монітором, вінчестером в системному блоці, трьома вентиляторами системи охолодження та клавіатурою. Крім того поряд працює периферійна техніка, то мають місце шуми механічного і аеродинамічного походження, широкосмугові із аперіодичним підсиленням (при роботі принтерів). [13].

Допустимі значення параметрів неіонізуючих електромагнітних випромінювань від монітору комп'ютера представлені в таблиці 4.1. Нормованим параметром невикористаного рентгенівського випромінювання виступає потужність експозиційної дози. На відстані 5 см від поверхні екрану монітору її рівень не повинен перевищувати 100 мкР/год. Максимальний рівень рентгенівського випромінювання на робочому місці програміста зазвичай не перевищує 20 мкР/год.

На відстані 5-10 см від екрана і корпусу монітора рівні напруженості можуть досягати 140 В/м по електричній складовій, що значно перевищує допустимі значення.

Таблиця 3.1 – Допустимі значення параметрів неіонізуючих ЕМВ

Найменування параметра	Допустимі значення
Напруженість електричної складової електромагнітного поля на відстані 50 см від поверхні відеомонітора	10 В/м
Напруженість магнітної складової електромагнітного поля на відстані 50 см від поверхні відеомонітора	0,3 А/м
Напруженість електростатичного поля не повинна перевищувати:	20 кВ/м

Таким чином, встановлено необхідність забезпечення вимог охорони праці та техніки безпеки на робочому місці програміста при розробці програмного забезпечення інформаційної системи управління підприємством.

3.2 Безпека в надзвичайних ситуаціях

3.2.1 Електробезпека. Статична електрика.

Приміщення лабораторії за небезпекою ураження електричним струмом можна віднести до 1 класу, тобто це приміщення без підвищеної небезпеки (сухе, без пилу, з нормальною температурою повітря, ізольованими підлогами і малим числом заземлених приладів).

На робочому місці програміста з усього устаткування металевим є лише корпус системного блоку комп'ютера, але тут використовуються системні блоки, що відповідають стандартів фірми ІВМ, у яких крім робочої ізоляції передбачений елемент для заземлення і провід з жилою, що заземлює, для приєднання до джерела живлення.

Основні причини ураження людини електричним струмом на робочому місці:

- дотик до металевих неструмоведучих частин (корпусу, периферії комп'ютера), що можуть виявитися під напругою в результаті ушкодження ізоляції;
- нерегламентоване використання електричних приладів;
- відсутність інструктажу співробітників з правил електробезпеки.

На протязі роботи на корпусі комп'ютера накопичується статична електрика. На відстані 5-10 см від екрана напруженість електростатичного поля складає 60-280 кВ/м, тобто в 10 разів перевищує норму 20 кВ/м.

Для попередження впровадження небезпечної техніки всі дисплеї повинні бути сертифіковані.

3.2.2 Електробезпека

Електробезпечність у приміщенні лабораторії пропонуємо забезпечити наступними технічними способами і засобами захисту:

- для зменшення накопичення статичної електрики застосовувати зволожувачі і нейтралізатори, антистатичне покриття підлоги;
- забезпечити приєднання металевих корпусів устаткування до жили, що заземлює. Заземлення корпуса ПК забезпечити підведенням жили, що заземлює, до розеток. Опір заземлення 4 Ом, згідно правил налаштування електроустановок (ПНЕ) для електроустановок з напругою до 1000 В.
- а також організаційними заходами: своєчасне проведення інструктажів з техніки безпеки; заборона використання непередбачених у лабораторії електричних приладів, таких як електричні чайники, обігрівачі.

3.2.3 Пожежна безпека

Ступінь вогнестійкості будинків приймається в залежності від їхнього призначення, категорії по вибухопожежній і пожежній небезпеці, по поверховості, площі поверху в межах пожежного відсіку згідно НАПБ Б.03.002-2007.

Будинок, у якому знаходиться лабораторія по пожежній небезпеці будівельних конструкцій відноситься до категорії К1 (малопожежонебезпечні), оскільки тут присутні займисті (книги, документи, меблі, оргтехніка і т.д.) і легкогорючі речовини (сейфи, різне устаткування і т.д.), що при взаємодії з вогнем можуть горіти без вибуху.

По конструктивних характеристиках будинок можна віднести до будинків з несучими і огорожуючими конструкціями із природних або штучних кам'яних матеріалів, бетону або залізобетону, де для перекриттів допускається використання дерев'яних конструкцій, захищених штукатуркою або важкогорючими листовими, а також плитними матеріалами.

Отже, ступінь вогнестійкості будинку можна визначити як третю (III). Приміщення лабораторії по функціональній пожежній небезпеці відноситься до класу Ф 4.2 – вищі навчальні заклади, установи підвищення кваліфікації.

3.2.4 Причини виникнення пожежі

Пожежа в лабораторії, може привести до дуже несприятливих наслідків (втрата кошовної інформації, псування майна, загибель людей і т.д.), тому необхідно:

- виявити й усунути всі причини виникнення пожежі;
- розробити план заходів для ліквідації пожежі в будинку; план евакуації людей з будинку.

Причинами виникнення пожежі можуть бути:

- несправності електропроводки, розеток і вимикачів які можуть привести до короткого замикання або пробоя ізоляції;
- використання ушкоджених (несправних) електроприладів;
- використання в приміщенні електронагрівальних приладів з відкритими нагрівальними елементами;
- виникнення пожежі внаслідок влучення блискавки в будинок;
- загоряння будинку внаслідок зовнішніх впливів;
- неакуратне поводження з вогнем і недотримання мір пожежної безпеки.

3.2.5 Профілактика пожежі

Для профілактики пожежі надзвичайно важлива правильна оцінка пожежонебезпеки будинку, визначення небезпечних факторів і обґрунтування способів і засобів пожежопередження і захисту.

Одне з умов забезпечення пожежобезпеки – ліквідація можливих джерел запалення. У лабораторії джерелами запалення можуть бути:

- несправне електроустаткування, несправності в електропроводці, електричних розетках і вимикачах. Для виключення виникнення пожежі з цих причин необхідно вчасно виявляти й усувати несправності, проводити плановий огляд і вчасно усувати всі несправності;

- несправні електроприлади. Необхідні міри для виключення пожежі містять у собі своєчасний ремонт електроприладів, якісне виправлення поломок, не використання несправних електроприладів;

- обігрівання приміщення електронагрівальними приладами з відкритими нагрівальними елементами. Відкриті нагрівальні поверхні можуть спричинити пожежу, тому що в приміщенні знаходяться паперові документи і довідкова література у виді книг, посібників, а папір – легкозаймистий матеріал. З метою профілактики пожежі пропоную не використовувати відкриті обігрівальні прилади в приміщенні лабораторії;

- коротке замикання в електропроводці. З метою зменшення імовірності виникнення пожежі внаслідок короткого замикання необхідно, щоб електропроводка була схованою;

- недотримання заходів пожежної безпеки і паління в приміщенні також може спричинити пожежу. Для усунення загоряння в результаті паління в приміщенні лабораторії пропоную категорично заборонити паління, а дозволити тільки в строго відведеному для цього місці.

З метою запобігання пожежі пропоную проводити з інженерами, що працюють у лабораторії, протипожежний інструктаж, на якому ознайомити

працівників із правилами протипожежної безпеки, а також навчити використанню первинних засобів пожежогасіння.

У випадку виникнення пожежі необхідно відключити електроживлення, викликати по телефоні пожежну команду, евакуювати людей із приміщення відповідно до плану евакуації, приведеному на рисунку 1 і приступити до ліквідації пожежі вогнегасниками. При наявності невеликого вогнища полум'я, можна скористатися підручними засобами з метою припинення доступу повітря до об'єкта загоряння.

ВИСНОВОК

Наслідки збільшення обсягу інформації та скорочення часу передачі не зовсім зрозумілі. Підвищення якості інформації, доступної під час прийняття рішень, дозволить керівництву приймати більш обґрунтовані та своєчасні рішення. Миттєва передача детальної інформації допомагає координувати діяльність просторово розділених підрозділів. Однак через велику кількість інформації, яка надходить, стає все важче знайти та вибрати необхідну інформацію. Скорочення часу, необхідного для надсилання інформації, означає, що адміністратори мають усе менше часу на її отримання та використання. Інформація є основним ресурсом продуктивності. Ефективніше використання інформації стає все більш важливим для забезпечення продуктивності всієї організації. Впровадження техніки та технологій у сферу обробки інформації призвело до зростання продуктивності, порівнянного з тим, що було досягнуто на початку промислової революції завдяки стандартизації та складанням ліній у виробництві. Так само як організації, які продовжують використовувати старі методи виробництва, не можуть конкурувати, організації, які не використовують інформаційні технології, не можуть конкурувати в інформаційному суспільстві.

Основним результатом роботи є створення функціональної СУБД, яка відповідає необхідному спектру завдань. Реалізація цього проекту здійснювалася без використання потужних інструментів для роботи з БД, але через їх універсальний характер і необхідні знання теорії було складно. Уся робота, необхідна для доступу та зміни збереженої інформації в БД, а також для впровадження методу підтримки цілісності їх, прихована всередині, залишаючи користувачеві вільне вирішення будь-яких пов'язаних проблем. Крім того, програмний інтерфейс максимально полегшує роботу.

Основним завданням цієї роботи було створення системи, що дозволяє автоматизувати оперативні процеси білінгу за медичні послуги. Результатом роботи стало впровадження автоматизованої системи обліку медичних послуг.

Програма оперативно реагує на всі запити користувачів і не містить помилок, які можуть знищити чи поставити під загрозу цілісність бази даних або мати потенційно фатальні наслідки для всієї комп'ютерної системи та, зрештою, для медичного закладу. Вона не включає помилки, які порушують систему. Програмний продукт підвищує ефективність діяльності медперсоналу та економить їх час. Програмні проекти прості у реалізації та не вимагають встановлення додаткового програмного забезпечення чи великих ресурсів. Розроблена система є гнучкою та забезпечує достатню інформаційну підтримку. Ця програма може бути змінена та доповнена в наступних версіях та подальшому розвитку.

Загалом нові технології та методи, що використовуються в програмних продуктах, виявилися дуже ефективними в автоматизації робочих місць. Під час роботи над програмним продуктом всі можливості та їх функціональність були протестовані кілька разів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Новини інформаційних технологій. [Електронний ресурс]. Режим доступу: <http://it-tehnolog.com/statti/informatsiyni-tehnologiyi-v-meditcini/>.
2. Вовчак І. С. Інформаційні системи та комп'ютерні технології в менеджменті. Навчальний посібник / І. С. Вовчак. – Тернопіль: Карт-бланш, 2001.
3. Антоненко В. М., Рогушина Ю.В. Сучасні інформаційні системи і технології. Навчальний посібник. - К.: КСУ МГІ, 2005. - 131 с.
4. Твердохліб М.Г. Система обробки облікової інформації на персональних ЕОМ / М.Г. Твердохліб. – К.:Техніка, 1993.
5. Лекції з курсу «Сучасні технології програмування» / Укладач Авраменко В.С. – Черкаси, ЧНУ. – 81 с.
6. Якобсон А., Буч Г., Рамбо Дж. Уніфікований процес розробки програмного забезпечення / А. Якобсон, Г. Буч, Дж. Рамбо. – Київ, 2022.
7. Литвинов В., Голуб С., Григор'єв К., Жигульська В. Об'єктно-орієнтоване моделювання при проектуванні вбудованих система і систем реального часу. – Черкаси, ЧНУ, 2011. - 375 с.
8. Дейт К. Дж. Вступ до систем баз даних / Дж. К. Дейт. Пер. с англ. – К., 2021. –1072 с.
9. Карвин Б. Програмування баз даних SQL. Типові помилки та їх уникнення / Б. Карвин. – К., 2020.
10. Борніцький Р.І., Яворська Є.Б., Андрійчук Н.Є. Парадигма інформатизації в медицині Матеріали XI науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 13-14 грудня 2023 р.).

– Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2023. – 257 с.

11. Стручок В.С. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної бо та заочної (дистанційної) форм навчання / В.С.Стручок. — Тернопіль: ФОП Паляниця В. А., 2022. — 156 с.

ДОДАТКИ

ДОДАТОК А

Код основних елементів програми

```
procedure TFNewSotr.Button1Click(Sender: TObject);
begin
with QNewSotr do
begin
if EFio.Text='' then
begin
ShowMessage();
exit;
end;
if cbSex.Text ='' then
begin
ShowMessage();
exit;
end;
if EBirth.Text<>'' then
try
StrToDate(EBirth.Text);
except
ShowMessage();
exit;
end else
begin
ShowMessage();
exit;
end;
ParamByName('SotrFIO').asString := EFio.Text;
ParamByName('BirthData').asDate := StrToDate(EBirth.Text);
ParamByName('Sex').asString := cbSex.Text;
if cbPNPriv.Text='' then ParamByName('PNPriv').asInteger :=0
else ParamByName('PNPriv').asInteger
```

```

:=StrToInt (cbPNPriv.Text);
    ParamByName('IsPens').asBoolean := cbIsPens.Checked;
    ParamByName('Childs').asSmallInt := udChilds.Position;
    ExecSQL;
    QLastSotr.Open;
    SotrID := QLastSotrLastSotr.asInteger;
end;
begin
    DayCount: = 0;
    DecodeDate(Data, Year, Mon, Day);
    CurrDate: = EncodeDate (Year, Mon, 1);
    LastDate := IncMonth(CurrDate, 1)-1;
    DMState.QHolydays.Params.ParamByName('Data1').asDate :=
CurrDate;
    DMState.QHolydays.Params.ParamByName('Data2').asDate :=
LastDate;
    DMState.QHolydays.Open;
    DecodeDate(LastDate,Year, Mon, Days);
    case WorkDays of
        5 :for i := 0 to Days-1 do if (DayOfWeek(EncodeDate(Year,
Mon, i+1))<>1) and
            (DayOfWeek(EncodeDate(Year, Mon, i+1))<>7) then
                inc(DayCount);
        6 :for i := 0 to Days-1 do if (DayOfWeek(EncodeDate(Year,
Mon, i+1))<>1) then
                inc(DayCount);
        else DayCount: = -1;
    end;
    if DayCount=-1 then result:=-1 else
    Result := DayCount-DMState.QHolydaysCountOfData.asInteger;
    DMState.QHolydays.Close;
end;
procedure TFZarplMaster.FormShow(Sender: TObject);
var Year, Mon, Day :word;

```

```

begin
Screen.Cursor := crHourGlass;
try
FZarplMaster.NB1.ActivePage := 'PodrPage';
CalcedPodr: = 0;
Edit1.Text := '';
DecodeDate(Date, Year, Mon, Day);
CalcedDate := EncodeDate(Year, Mon, 1);
CalcedMon: = Mon;
Case CalcedMon of
1: CalcedMonStr := ' січень ' +IntToStr(Year);
2: CalcedMonStr := ' лютий '+IntToStr(Year);
3: CalcedMonStr := ' березень ' + IntToStr(Year);
4: CalcedMonStr:= ' квітень ' + IntToStr (Year);
5: CalcedMonStr := ' травень '+IntToStr(Year);
6: CalcedMonStr := ' червень '+IntToStr(Year);
7: CalcedMonStr := ' липень '+IntToStr(Year);
8: CalcedMonStr := ' серпень ' + IntToStr(Year);
9: CalcedMonStr := ' вересень ' + IntToStr(Year);
10: CalcedMonStr := ' жовтень '+IntToStr(Year);
11: CalcedMonStr := ' листопад ' + IntToStr(Year);
12: CalcedMonStr := ' грудень ' + IntToStr(Year);
end; //case
Button11.Enabled := false;
MMOT := FBuhMain.MMOT;
MRP: = FBuhMain.MRP;
DMMaster.QStateTypes.Open;
Button28.Enabled := false;
Button28_2.Enabled := false;
finally
screen.Cursor := crDefault;
end;
end;
Розрахунок прибуткового податку - модуль UPNFunc:

```

```

function NewPN(SotrID, Mon :integer) :real;
//функція CalcPN(Summa: Real): real;
implementation
uses UDMCalc, UDMMaster, UZarplMaster;
var NalCount :integer;
Nal: array[0..15,0..1] of Real;
Koeff: array[0..11] of real;
procedure LoadKoeffs;
var i,j: integer;
begin
with DMCalc do
begin
QNalog.Open;
NalCount:= QNalog.RecordCount;
for i := 0 to NalCount-1 do
begin
Nal[i,0]:= QNalogMaxSumma.asFloat;
Nal[i,1]:= QNalogPercent.asInteger;
QNalog.Next;
end;
QNalog.Close;
QNalKoeff.Open;
for j := 0 to 11 do
begin
Koeff[j] := QNalKoeffKoeff.asFloat;
QNalKoeff.Next;
end;
QNalKoeff.Close;
end;
end;
функція CalcPN(Summa: Real): real;
var i: integer;
begin
//процедура визначення суми ПН із розрахунку річного доходу

```

```

Result :=0;
if Summa <=0 then
begin
Result: = 0;
Exit;
end;
for i := 0 to NalCount do
if Summa <= Nal[i,0] then
if i=0 then
begin
Result := Summa*Nal[i,1]/100;
Exit;
end else
begin
Result := Result+(Summa-Nal[i-1,0])*Nal[i,1]/100;
Exit;
end else
if i=0 then Result := Result+Nal[i,0]*Nal[i,1]/100
else Result := Result+(Nal[i,0]-Nal[i-1,0])*Nal[i,1]/100;
end;
function NewPN(SotrID, Mon:integer) :real;
var YearBase, YearPN, NeedPN: real;
begin
LoadKoeffs;
with DMMaster do
begin
QSumForPN.Params.ParamByName('SotrID').asInteger := SotrID;
//Спочатку визначаємо базу - оподатковуваний дохід протягом
року
QSumForPN.Params.ParamByName('GroupID').asInteger := 6;
QSumForPN.Open;
YearBase: = QSumForPN.Summa.asFloat;
QSumForPN.Close;
//Тепер - вже сплачений ПН

```

```

QSumForPN.Params.ParamByName('GroupID').asInteger := 5;
QSumForPN.Open;
YearPN := -QSumForPNSumma.asFloat;
QSumForPN.Close;
//Розрахунок ПН з відрахуванням МРП
QSotrsChild.Params.PParamByName('SotrID').asInteger :=
SotrID;
QSotrsChild.Open;
NeedPN := CalcPN((YearBase-
FZarp1Master.MRP*QSotrsChildChilds.asInteger *Mon)*Koef[Mon-
1])/Koef[Mon-1]; // else
NeedPN := NeedPN*(100-QSotrsChildPNPriv.asInteger)/100;
QSotrsChild.Close;
end;
Result := NeedPN-YearPN;
end;
end.
Модуль розрахунку нарахувань та утримань:
begin
{Нарахування ПФ та ПН}
Screen.Cursor := crHourGlass;
try
with DMMaster, QSvmPdrEx do
begin
First;
FIndicator.Maxim := RecordCount;
FIndicator.Caption := 'Розрахунок основних утримань';
FIndicator.Show;
for i := 0 to RecordCount-1 do
begin
CurrSovm := QSvmPdrExNum.asInteger;
CurrSotr := QSvmPdrExSotrID.asInteger;
Kateg: = QSvmPdrExKategory.asInteger;
// Розрахунок пенсійних відрахувань

```

```

    QSumByGrMon.Params.PParamByName('StateType').asString:=
StateType;
    QSumByGrMon.Params.PParamByName('SovmID').asInteger:=
CurrSovm ;
    QSumByGrMon.Params.PParamByName('Mon').asInteger :=
CalcedMon;
    QSumByGrMon.Params.PParamByName('GroupID').asInteger := 1;
    QSumByGrMon.Open;
    //Внесено для поділу ПФ за основним окладом та внутрішнім
сумісництвом
    QCalcByType.Params.PParamByName('SovmID').asInteger :=
CurrSovm ;
    QCalcByType.Params.PParamByName('Mon').asInteger :=
CalcedMon;
    QCalcByType.Params.PParamByName('CalcType').asInteger := 24;
    QCalcByType.Open;
    QNCalcs.Params.PParamByName('SovmID').asInteger := CurrSovm;
    QNCalcs.Params.PParamByName('GroupID').asInteger := 2;
    // Всі утримання
    QNCalcs.Open;
    PN: = 0;
    for j := 0 to QNCalcs.RecordCount-1 do
begin
    Calc: = QNCalcsCalcType.asInteger;
    if QNCalcsAct.asBoolean=true then
    if QNCalcsKind.asString='NA' then // Довільні та складні
доплати
        case Calc of
            //Пенсфонд
            17: begin
                //Якщо пенсіонер - то до пенсійного фонду не відраховуємо
                if QSvmPdrExIsPens.asBoolean=false then
                AddCalc(Kateg, CurrSovm, CurrSotr, 17, ' Пенс . фонд ',
                -(QSumByGrMonSumma.asFloat-QCalcByTypeSumma.asFloat)*0.1);

```



```

end;
// Прибутковий податок
19: begin
PN := -NewPN(QSvmPdrExSotrID.asInteger, CalcedMon);
AddCalc(Kateg, CurrSovm, CurrSotr, 19, ' Прибутковий податок
', PN);
end;
// Профспілковий внесок
21: begin
if QSvmPdrExIsPens.asBoolean then
AddCalc(Kateg, CurrSovm, CurrSotr, 21, ' Проф . Внесок ',
-QSumByGrMonSumma.asFloat*0.01)
else
AddCalc(Kateg, CurrSovm, CurrSotr, 21, ' Проф . Внесок ',
-QSumByGrMonSumma.asFloat*0.009);
end;
// Аліменти
1: begin
AddCalc(Kateg, CurrSovm, CurrSotr, 1, ' Ісп . лист ',
-(QSumByGrMonSumma.asFloat-PN)*QSvmPdrExAlim.asInteger/100);
end;
// Аліменти - поштові витрати
2: begin
AddCalc(Kateg, CurrSovm, CurrSotr, 1, ' Ісп . лист - пошт .
витрати ',
-(QSumByGrMonSumma.asFloat-
PN)*QSvmPdrExAlim.asInteger*0.05/100);
end;
end //case
else
if QNCalcsKind.asString='P' then // Всі утримання відсотком
AddCalc(Kateg, CurrSovm, CurrSotr, Calc,
QNCalcsDescription.asString, -
QSumByGrMonSumma.asFloat*QNCalcsBase.asFloat/100)

```

```

else
  if QNCalcsKind.asString='S' then // Всі утримання сумою
    AddCalc(Kateg, CurrSovm, CurrSotr, Calc,
QNCalcsDescription.asString, -QNCalcsBase.asFloat)
  else
    if QNCalcsKind.asString='MR' then // Всі МРП - залежні
утримання
      AddCalc(Kateg, CurrSovm, CurrSotr, Calc,
QNCalcsDescription.asString, -MRP*QNCalcsBase.asFloat);
    QNCalcs.Next;
  end;
  QSumByGrMon.Close;
  FIndicator.OneStep(QSvmPdrExSotrFIO.asString,1);
  QNCalcs.Close;
  QCalcByType.Close;
next;
end;
First;
FIndicator.Close;
end;
VedMaking;
finally
  Screen.Cursor: = crDefault;
  NB1.ActivePage := 'EAddPage';
end;
end;

procedure TFZarplMaster.Button38Click(Sender: TObject);
var i, j, CurrSovm, CurrSotr, Kateg, TotDays, Calc: integer;
Oklad, DayKoef, DekanDayKoef :real;
s1: string;
begin
  with DMMaster, QForAdds do
  begin
    Params.PParamByname('PodrID').asInteger := CalcedPodr;

```

```

Params.PParamByName('StateType').asString := StateType;
Params.ParamByname('Mon').asInteger := CalcedMon;
Open;
FIndicator.Maxim := RecordCount;
FIndicator.Caption := ' Розрахунок основних нарахувань ';
FIndicator.Show;
FIndicator.OneStep(QForAddsSotrFIO.asString, 1);
for i := 0 to RecordCount-1 do
if QForAddsSovmType.asString<>' Внутрішній сумісник ' then
begin
s1 := QForAddsSovmType.asString;
CurrSotr := QForAddsSotrID.asInteger;
CurrSovm := QForAddsNum.asInteger;
Kateg: = QForAddsKategory.asInteger;
TotDays := WorkDaysOfMonth(CalcedDate,
QForAddsCalendar.asInteger);
DayKoeff := QForAddsDays.asInteger/TotDays;
DekanDayKoeff := QForAddsDekanDays.asInteger/TotDays;
Oklad:=QForAddsTarKoeff.asFloat*QForAddsStavka.asFloat*MMOT*D
ayKoeff;
QNCalcs.PParams.ParamByName('SovmID').asInteger :=
QForAddsNum.asInteger;
QNCalcs.PParams.ParamByName('GroupID').asInteger := 1;
QNCalcs.Open;
for j := 0 to QNCalcs.RecordCount-1 do
begin
Calc: = QNCalcsCalcType.asInteger;
if QNCalcsAct.asBoolean=true then
if QNCalcsKind.asString='NA' then // Довільні та складні
доплати
case Calc of
14: AddCalc (Kateg, CurrSovm, CurrSotr, Calc,
QNCalcsDescription.asString, Oklad);
10:

```

```

AddCalc (Kateg, CurrSovm, CurrSotr, Calc, QNCalcsDescription.asString
, Oklad*0.15*DekanDayKoef);
    56:
AddCalc (Kateg, CurrSovm, CurrSotr, Calc, QNCalcsDescription.asString
, Oklad*0.1*DekanDayKoef);
    31: begin
    QZona.Params.ParamByName ('SovmID').asInteger := CurrSovm;
    QZona.Open;
    AddCalc (Kateg, CurrSovm, CurrSotr, Calc,
QNCalcsDescription.asString,
    QNCalcsBase.asFloat*DayKoef*MMOT*LoadTarif (QZonaZonaT.asInte
ger)/100);
    QZona.Close;
    end;
    end else
    if QNCalcsKind.asString='P' then // Всі доплати відсотком
    AddCalc (Kateg, CurrSovm, CurrSotr, Calc,
QNCalcsDescription.asString, Oklad*QNCalcsBase.asFloat/100)
    else
    if QNCalcsKind.asString='S'
    then
    AddCalc (Kateg, CurrSovm, CurrSotr, Calc,
QNCalcsDescription.asString, QNCalcsBase.asFloat)
    else
    if QNCalcsKind.asString='MO' then // Всі MMOT - залежні
доплати
    AddCalc (Kateg, CurrSovm, CurrSotr, Calc, QNCalcsDescription.asSt
ring, QForAddsStavka.asFloat*DayKoef*MMOT*QNCalcsBase.asFloat)
    else
    if QNCalcsKind.asString='MR' then // Всі МРП - залежні
доплати
    AddCalc (Kateg, CurrSovm, CurrSotr, Calc,
QNCalcsDescription.asString, MRP*QNCalcsBase.asFloat);
    QNCalcs.Next;

```

```

end;
QNCalcs.Close;
Next;
FIndicator.OneStep(QForAddsSotrFIO.asString,1);
end else
begin
s1 := QForAddsSovmType.asString;
QMainID.Params.PParamByName('SotrID').asInteger:=
QForAddsSotrID.asInteger;
QMainID.Params.PParamByName('StateType').asString :=
StateType;
QMainID.Open;
CurrSovm := QMainIDNum.asInteger;
QMainID.Close;
TotDays := WorkDaysOfMonth(CalcedDate,
QForAddsCalendar.asInteger);
DayKoeff := QForAddsDays.asInteger/TotDays;
DekanDayKoeff := QForAddsDekanDays.asInteger/TotDays;
QNCalcs.Params.ParamByName('SovmID').asInteger :=
QForAddsNum.asInteger;
QNCalcs.Params.ParamByName('GroupID').asInteger := 1;
QNCalcs.Open;
Oklad: = 0;
for j := 0 to QNCalcs.RecordCount-1 do
begin
Calc: = QNCalcsCalcType.asInteger;
if QNCalcsAct.asBoolean=true then
if QNCalcsKind.asString='NA' case Calc of
14:
Oklad:=Oklad+QForAddsTarKoeff.asFloat*QForAddsStavka.asFloat*
MMOT*DayKoeff;
10: Oklad: = Oklad + Oklad * 0.15 * DekanDayKoeff;
56: Oklad: = Oklad + Oklad * 0.1 * DekanDayKoeff;
31: begin

```

```

QZona.Params.ParamByName('SovmID').asInteger := CurrSovm;
QZona.Open;
Oklad:=Oklad+QNCalcsBase.asFloat*DayKoeff*MMOT*LoadTarif(QZona
aZonaT.asInteger)/100;
QZona.Close;
end;
end //case
else
if QNCalcsKind.asString='P' then // Всі доплати відсотком
Oklad := Oklad + Oklad*QNCalcsBase.asFloat/100
else
if QNCalcsKind.asString='S' then // Всі доплати сумою
Oklad := Oklad + QNCalcsBase.asFloat
else
if QNCalcsKind.asString='MO' then // Всі ММОТ - залежні
доплати
// Ступінь
Oklad := Oklad + QForAddsStavka.asFloat *DayKoeff*MMOT
*QNCalcsBase.asFloat
else
if QNCalcsKind.asString='MR' then // Всі МРП - залежні
доплати
Oklad := Oklad + MRP * QNCalcsBase.asFloat;
QNCalcs.Next;
end;
QNCalcs.Close;
AddCalc(QForAddsKategory.asInteger, CurrSovm,
QForAddsSotrID.asInteger, 24, ' за поєднання ', Oklad);
if QForAddsIsPens.asBoolean=false then
AddCalc(QForAddsKategory.asInteger, CurrSovm,
QForAddsSotrID.asInteger, 52, ' Пенс . фонд - суміщення ', -
Oklad/10);
FIndicator.OneStep(QSovmsByPodrSotrFIO.asString,1);
Next;

```

```
end;  
Close;  
QsvmPdrEx.Close;  
QsvmPdrEx.Params.ParamByName('PodrID').asInteger :=  
CalcedPodr;  
QsvmPdrEx.Params.ParamByName('StateType').asString :=  
StateType;  
QsvmPdrEx.Open;  
QTimeTable.Close;  
FIndicator.Close;  
NB1.ActivePage := 'AddPage';  
end; end.
```

ДОДАТОК Б

Тези конференції

СЕКЦІЯ 4. ПРОГРАМНА ІНЖЕНЕРІЯ ТА МОДЕЛЮВАННЯ СКЛАДНИХ РОЗПОДІЛЕНИХ СИСТЕМ

УДК 004.94; 681.3

Борніцький Р.І., Яворська Є.Б., к.т.н., доц., Андрійчук Н.Є.

Тернопільський національний технічний університет імені Івана Пулюя, Україна
Відокремлений структурний підрозділ «Тернопільський фаховий коледж Тернопільського національного технічного університету ім. І.Пулюя», Україна

ПАРАДИГМА ІНФОРМАТИЗАЦІЇ В МЕДИЦИНІ

**R. Bornitskyi, E. Yavorska, Ph.D., Assoc. Prof., N. Andriychuk,
THE PARADIGM OF INFORMATIZATION IN MEDICINE**

Основні відмінності медицини від інших областей поза сумнівами медицина є однією з найскладніших областей людської діяльності. Вся складність медицини як галузі людської діяльності відбивається у тому числі й на функціонуванні медичних організацій та відповідно на проблематиці розробки та експлуатації медичних інформаційних систем.

Однією з проблем впровадження інформаційних технологій у медицину є те, що рівень оснащення системи охорони здоров'я сучасними інформаційно-комунікаційними технологіями вкрай неоднорідний, здебільшого обмежується використанням кількох комп'ютерів як автономні автоматизовані робочі місця. Ще однією проблемою є відсутність уніфікації програмно-апаратних платформ, що використовуються. Єдиний вид програмного забезпечення, встановлений практично повсюдно в закладах охорони здоров'я, – це розроблені програми обліку реєстрів послуг системи обов'язкового медичного страхування, а також компоненти інформаційних систем забезпечення пільговими лікарськими засобами. В результаті неможливість створення єдиного підходу до впровадження та використання інформаційно-комунікаційних технологій у медичних установах та організаціях, інтеграція існуючих програмних рішень дуже обмежена. Для розробника основною проблемою є підтримка великого різноманітності парку медичного обладнання, розробка архітектури відповідних програмних інструментів для оптимізації самого процесу розробки ПЗ (програмне забезпечення), його впровадження та подальшої ефективної експлуатації.

Моделювання діяльності медичних організацій та їх об'єднань потрібне як для створення інформаційної та функціональної моделей МІС на етапі проектування та розробки, так і для розробки концепцій та технічних завдань на побудову МІС, а також є надзвичайно корисним для підтримки процесу впровадження у конкретній МО.

На думку багатьох дослідників, подальший розвиток медичних інформаційних систем пов'язаний насамперед із розвитком можливостей систем із підтримки прийняття рішень. Наприклад, за класифікацією аналітиків компанії Gartner, наявні над ринком нині МІС переважно відносяться до четвертого покоління (MIS Colleague за класифікацією Gartner), і очікується перехід до МІС п'ятого покоління (MIS Mentor за класифікацією Gartner). Основною відмінністю цього покоління будуть функціональні можливості МІС підказувати лікарям оптимальні шляхи лікування та діагностики пацієнтів. Для ефективного вирішення завдання побудови МІС необхідний комплексний розгляд усіх зазначених аспектів, кожен з яких вимагає проведення досліджень та розробок за відповідним напрямом з урахуванням особливостей предметної галузі та вимог комплексної постановки завдання.

Література

1. Р.А. Ткачук, Г.Б. Цуприк, і Б.І. Яворський, "Підвищення інформативності та швидкості біотехнічних систем", Опт-ел. інф-енерг. техн., вип. 24, вип. 2, с. 81–85, Жов 2013.