

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
Савченко А.С.

“ _____ ” _____ 2020 р.

ДИПЛОМНА РОБОТА (ПОЯСНЮВАЛЬНА ЗАПИСКА)

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
“МАГІСТРА”**

**ЗА СПЕЦІАЛІЗАЦІЄЮ “ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА
ТЕХНОЛОГІЇ (ЗА ГАЛУЗЯМИ)”**

Тема: “Автоматизована система оцінки стану легенів людини”

Виконавець: Костянян Павло Костянтинович

Керівник: д.т.н., професор Воронін Альберт Миколайович

Нормоконтролер: _____ Райчев І.Е.

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології (за галузями)”

ЗАТВЕРДЖУЮ
Завідувач кафедри

Савченко А.С.

“ ” 2020 р.

ЗАВДАННЯ

на виконання дипломної роботи студента

Костяняна Павла Костянтинівича

1. Тема роботи: “Автоматизована система оцінки стану легенів людини”

Затверджена наказом ректора від “ 02 ”10.20 за № 1891/ст

2. Термін виконання роботи: 05.10.2020 р. по 31.12.2020 р.

3. Вихідні данні до роботи: електронні джерела про медицину, медична література.

4. Зміст пояснювальної записки: вступ; хвороби легень та способи їх діагностики; експертні системи; використані технології розробки; розробка експертної системи для визначення вірогідності захворювання легень.

5. Перелік обов'язкового ілюстративного матеріалу: таблиця, рисунки, діаграми, а також слайди презентації доповіді у PowerPoint.

6. Календарний план-графік

<i>№ з/п</i>	<i>Завдання</i>	<i>Термін виконання</i>	<i>Підпис керівника</i>
1.	Огляд літератури і джерел за темою дипломного проекту.	5.10.20– 09.10.20	
2	Створення плану та розробка розділу 1 – хвороби легень та їх діагностика	10.10.20– 20.10.20	
3	Розробка розділу 2: експертні системи	21.10.20– 30.10.20	
4	Розробка розділу 3: використані технології розробки	01.11.20– 08.11.20	
5	Проектування бази даних. Опис моделі даних у програмному коді	10.11.20– 18.11.20	
6	Розробка функціональної частини додатку	19.11.20– 02.12.20	
7	Оформлення інтерфейса експертної системи	03.12.20– 06.12.20	
8	Розробка розділу 4: розробка експертної системи для визначення вірогідності захворювання легень	07.12.20– 10.12.20	
9	Створення доповіді та слайдів до неї	11.12.20– 13.12.20	
10.	Оформлення та друк пояснювальної записки дипломної роботи	14.12.20– 20.12.20	

7. Дата видачі завдання: 05.10.2020 р.

Керівник дипломної роботи _____ Воронін А.М.

Завдання прийняв до виконання _____ Костанян П.К.

(підпис випускника)

(ПІБ)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи "Автоматизована система оцінки стану легенів людини" містить 97 сторінок, 18 рисунків, 2 таблиці, 21 наукове джерело.

Мета роботи: дослідити розповсюджені хвороби легень людини та їх симптоми, розробити додаток, який зможе оцінювати вірогідність хвороб легень на основі набору симптомів.

Об'єкт дослідження: захворювання легенів людини.

Методи дослідження: методи розробки бази знань, методи розробки експертних систем та методи розробки веб-додатків.

Результати магістерської роботи: розроблена експертна система оцінки вірогідності хвороб легенів, в основі якої лежить база знань, що включає в себе 11 найросповсюджених хвороб легень, а також 3 суміжні хвороби.

Ключові слова: ЛЕГЕНІ ЛЮДИНИ, ХВОРОБИ ЛЕГЕНЬ, ЕКСПЕРТНА СИСТЕМА, ВІРОГІДНІСТЬ, MVC.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	8
ВСТУП.....	9
РОЗДІЛ 1	10
ХВОРОБИ ЛЕГЕНЬ ТА СПОСОБИ ЇХ ДІАГНОСТИКИ.....	10
1.1 Захворювання легень.....	10
1.1.1 Астма	10
1.1.2 Хронічне обструктивне захворювання легень	11
1.1.3 Рак легенів	12
1.1.4 Інші захворювання легенів.....	12
1.2 Симптоми захворювання легенів та їх лікування.....	13
1.2.1 Симптоми астми та її лікування	15
1.2.2 Симптоми хронічного обструктивного захворювання легенів та його лікування	17
1.2.3 Симптоми рака легенів та його лікування.....	19
1.3 Способи зниження ризику захворювання легенів	21
1.4. Способи діагностики захворювання легенів та їх ризику	22
1.4.1. Бронхоскопія	22
1.4.2 Процедура грудної трубки	24
1.4.3 Комп'ютерна томографія	26
1.4.4 Комп'ютерна томографія – керована біопсія легенів	27
1.4.5 Ендобронхіальне ультразвукове дослідження (EBUS)	29
1.4.6 Тест на оксид азоту на видиху.....	30
1.4.7 Лобектомія	31

1.4.8 Тести функції легенів	33
1.4.9 Киснева терапія	35
1.4.10 Пульсоксиметр	36
ВИСНОВОК ДО 1 РОЗДІЛУ	37
РОЗДІЛ 2	38
ЕКСПЕРТНІ СИСТЕМИ	38
2.1 Призначення експертних систем	40
2.1.1 Компоненти експертної системи	42
2.1.2. База знань	43
2.1.3 Учасники розробки експертних систем	45
2.1.4. Процес побудови експертних систем	46
2.1.5. Переваги та обмеження експертних систем	57
2.2. Приклади експертної системи.....	59
2.2.1. MYCIN	59
2.2.2. DENDRAL.....	64
2.2.3. PXDES	67
2.2.4. CaDet.....	68
2.3. Приклад створення експертної системи CLIPS	68
ВИСНОВОК ДО РОЗДІЛУ 2	74
РОЗДІЛ 3	75
ВИКОРИСТАНІ ТЕХНОЛОГІЇ РОЗРОБКИ	75
3.1.1 Сфери використання мови С#.....	76
3.1.2 Переваги мови С#.....	77
3.2. Мова програмування JavaScript.....	79
3.2.1 Переваги та недоліки мови JavaScript.....	81

3.3. Архітектурний патерн MVC	82
ВИСНОВОК ДО РОЗДІЛУ 3	86
РОЗДІЛ 4	87
РОЗРОБКА ЕКСПЕРТНОЇ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ВІРОГІДНОСТІ ЗАХВОРЮВАННЯ ЛЕГЕНЬ	87
4.2 Розробка бази даних, алгоритму обчислення, програмного коду та інтерфейсу експертної системи.....	88
4.3. Приклад роботи експертної системи.....	89
ВИСНОВОК ДО РОЗДІЛУ 4	92
ВИСНОВКИ.....	93
Список літератури	95
ДОДАТОК А.....	98

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

ХОЗЛ – Хронічна обструктивна хвороба людини

КТ – комп'ютерна томографія

EBUS – Endobronchial Ultrasound (Ендобронхіальне ультразвукове дослідження)

CPAP – Constant Positive Airway Pressure (апарати для лікування порушень дихання під час сну)

ЕС – експертна система

ЕОМ – електронно-обчислювальна машина

ВСТУП

З розвитком технологій значно поліпшилися усі сфери життєдіяльності людини. У тому числі і медицина. Комп'ютеризація зробила медицину більш ефективною, діагнози більш точними. У більшості процедур для визначення тієї чи іншої хвороби застосовується не тільки знання та досвід лікарів, а і комп'ютерна техніка. Захворювання будь-яких людських органів потребують тонкого діагнозу, а найбільш смертельні хвороби пов'язані з серцем та легенями людини – особливо.

Захворювання легень є одними з найросповсюдженіших хвороб, якими можуть захворіти люди будь-якого віку, статі та способу життя. За статистикою, кожна шоста смерть у світі через хвороби легень. За роки, що минули з початку століття, переломити ситуацію з легневими захворюваннями не вдалося, і, ймовірно, їх вплив залишиться на колишньому рівні ще кілька десятиліть.

Захворювання легенів ведуть до інвалідності та передчасної смерті, що, в свою чергу, пов'язано з величезними витратами на першу допомогу, госпіталізацію та ліки. Крім цього, величезний збиток наносить втрата людьми працездатності і передчасні смерті в результаті хвороби.

Перший крок до визначення тієї чи іншої хвороби – аналіз симптомів. Це, звичайно, задача лікаря, але за лікаря це може зробити комп'ютеризована експертна система, в основі якої лежить база знань, що описує симптоми різних захворювань. База знань експертної системи і є сукупністю знань лікаря(або великої кількості лікарів), яка на основі запитів користувача видає найбільш підходящий результат. Таким чином користувач(пацієнт) може за допомогою експертної системи приблизно оцінити стан своїх легенів та дізнатися вірогідність тієї чи іншої хвороби.

РОЗДІЛ 1

ХВОРОБИ ЛЕГЕНЬ ТА СПОСОБИ ЇХ ДІАГНОСТИКИ

1.1 Захворювання легень

Хвороба легенів відноситься до розладів, що вражають легені, органи, що дозволяють нам дихати. Проблеми з диханням, викликані захворюваннями легенів, можуть перешкодити організму отримувати достатню кількість кисню.

Прикладами легеневих захворювань є:

- астма, хронічний бронхіт та емфізема;
- інфекції, такі як грип та пневмонія;
- рак легенів;
- саркоїдоз та легеневий фіброз.

Хвороби легенів є основною проблемою для жінок. Кількість жінок із діагнозом захворювання легенів у США зростає. Більше жінок також помирає від хвороб легенів.

Три найпоширеніші захворювання легенів у жінок – це астма, хронічна обструктивна хвороба легень (ХОЗЛ) та рак легенів.

1.1.1 Астма

Астма – це хронічне (постійне) захворювання дихальних шляхів у легенях, яке називається бронхи. Бронхіальні трубки переносять повітря в легені та з них. У людей з астмою стінки цих дихальних шляхів запалюються (набрякають) і надмірно чутливі. Дихальні шляхи надмірно реагують на такі речі, як дим, забруднення повітря, цвіль та багато хімічних спреїв.

Кафедра КІТ (47)				НАУ 20 11 93 000 ПЗ			
Виконав	Костанян П.К.			Хвороби легень та способи їх діагностики	Літ.	Арк.	Аркушів
Керівник	Воронін А.М.					10	28
Консульт.					УС-211М 122		
Н-контроль	Райчев І.Е.						

Вони також можуть дратуватися алергенами (наприклад, пилом та пиловими кліщами) та респіраторними інфекціями (як застуда). Коли дихальні шляхи надмірно реагують, вони звужуються. Це обмежує приплив повітря в легені та з них і спричиняє проблеми з диханням. Симптоми астми включають хрипи, кашель і стискання в грудях.

Жінки частіше за чоловіків хворіють на астму і частіше помирають від неї. У США зростає відсоток жінок, особливо молодих жінок, хворих на астму. Дослідники не впевнені, чому. Багато експертів вважають, що забруднення повітря та алергени відіграють певну роль у цьому збільшенні. Дихання тютюнового диму також пов'язане з підвищеним ризиком розвитку астми.

1.1.2 Хронічне обструктивне захворювання легень

Хронічна обструктивна хвороба легень (ХОЗЛ) відноситься до хронічного обструктивного бронхіту та емфіземи. Ці стани часто виникають разом. Обидві хвороби обмежують приплив повітря в легені та з них і ускладнюють дихання. ХОЗЛ зазвичай з часом погіршується.

У людини з ХОЗЛ постійно відбувається запалення бронхів, яке переносить повітря в легені та з них. Це подразнення спричиняє ріст клітин, які утворюють слиз. Надлишок слизу призводить до сильного кашлю. З часом подразнення призводить до потовщення стінок дихальних шляхів і утворення рубців. Дихальні шляхи можуть стати досить потовщеними, щоб обмежити приплив повітря до легенів і назад. Якщо це трапляється, стан називається хронічним обструктивним бронхітом.

При емфіземі легенева тканина слабшає, стінки повітряних мішків (альвеоли) руйнуються. Зазвичай кисень з повітря надходить у кров через ці стінки повітряних мішків. У людини з емфіземою зіпсовані стінки повітряних мішків означають, що менше крові може проникати в кров. Це спричиняє задишку, кашель та хрипи.

Зараз більше, ніж удвічі більше жінок, ніж чоловіків, діагностують хронічний бронхіт. Рівень емфіземи серед жінок за останні роки зріс на 5 відсотків, але серед

чоловіків зменшився. І щороку з 2000 року від ХОЗЛ помирає більше жінок, ніж чоловіків. Дослідники намагаються зрозуміти, чому. Серед жінок зросло куріння сигарет, яка є основною причиною ХОЗЛ. Одна з теорій полягає в тому, що сигаретний дим завдає більшої шкоди жінкам, ніж чоловікам.

1.1.3 Рак легенів

Рак легенів – це захворювання, при якому аномальні (злоякісні) клітини легенів розмножуються і ростуть без контролю. Ці ракові клітини можуть вторгнутися в сусідні тканини, поширитися в інші частини тіла або в обидва. Два основні види раку легенів названі за тим, як клітини виглядають під мікроскопом. Вони є:

- *дрібноклітинний рак легенів* – цей вид раку легенів має тенденцію до швидкого поширення;
- *недрібноклітинний рак легенів* – цей термін для декількох видів раку легенів, які діють подібним чином. Більшість видів раку легенів є недрібноклітинними. Цей вид раку легенів, як правило, поширюється повільніше, ніж дрібноклітинний рак легенів.

У США зараз більше жінок помирає від раку легенів, ніж від будь-якого іншого типу раку. Вживання тютюну є основною причиною раку легенів.

1.1.4 Інші захворювання легенів

До менш поширених проблем з легенями, які вражають жінок, належать:

- 1) *Легеневі емболії*. Це згустки крові, які надходять у легені з інших частин тіла і закупорюють кровоносні судини в легенях. Деякі фактори, що підвищують ризик, включають вагітність, нещодавно народження дитини, прийом протизаплідних таблеток або гормональну терапію в менопаузі. Легеневі емболи можуть впливати на кровотік в легенях і можуть зменшити надходження кисню в кров. Дуже великі емболії можуть спричинити раптову смерть.

2) *Легенева гіпертензія*. Це високий кров'яний тиск в артеріях, що приносить кров до легенів. Це може вплинути на приплив крові в легенях і може зменшити приплив кисню в кров.

3) *Саркоїдоз та фіброз легенів*. Ці запальні захворювання викликають жорсткість і рубцювання в легенях.

4) *ЛАМ (лімфангіолейоміоматоз)*. Це рідкісне захворювання легенів, яке в основному вражає жінок у середині 30-40-х років. М'язоподібні клітини виростають з-під контролю в деяких органах, включаючи легені.

5) *Грип (грип) та пневмонія*. Грип - це респіраторна інфекція, яка викликається вірусом і може пошкодити легені. Зазвичай люди добре одужують від грипу, але для деяких людей він може бути небезпечним і навіть смертельним. Серед тих, хто піддається більшому ризику, - люди похилого віку, маленькі діти, вагітні жінки та люди з певними захворюваннями, такими як астма. Пневмонія - це важке запалення легенів, яке може бути спричинене бактеріями, вірусами та грибками. Рідина накопичується в легенях і може зменшити кількість кисню, який кров може отримати з повітрям, яке вдихається. Люди, яким найбільше загрожує ризик, старші 65 років або молодше 2 років, або вже мають проблеми зі здоров'ям. Вакцини - найкращий захист від грипу та пневмонії.

1.2 Симптоми захворювання легенів та їх лікування

Експерти не знають причин усіх типів захворювань легенів, але вони знають причини деяких. До них належать:

1) *Куріння*. Дим від сигарет, сигар та люльок є причиною захворювання легенів номер один. Не починайте палити і не кидайте, якщо ви вже палите. Якщо ви живете або працюєте з курцем, уникайте пасивного куріння. Попросіть курців палити на відкритому повітрі. Пасивне куріння особливо шкідливо для немовлят та маленьких дітей.

2) *Радон*. Цей безбарвний газ без запаху присутній у багатьох будинках і є визнаною причиною раку легенів. Перевірити наявність радону можна за

допомогою набору, придбаного в багатьох будівельних магазинах. Радон може бути знижений у вашому домі, якщо ви дізнаєтесь, що рівень його високий.

3) *Азбест*. Це природне мінеральне волокно, яке використовується для утеплення, протипожежних матеріалів, автомобільних гальм та інших виробів. Азбест може виділяти дрібні волокна, які занадто малі, щоб їх можна було побачити, і їх можна вдихати. Азбест шкодить легенеvim клітинам, викликаючи рубцеві зміни легенів і рак легенів. Це може спричинити мезотеліому (MEZ-oh-THEE-lee-ON-tuh), яка являє собою рак, який утворюється в тканині, що покриває легені та багато інших органів тіла.

4) *Забруднення повітря*. Недавні дослідження показують, що деякі забруднювачі повітря, такі як вихлопні гази автомобілів, можуть сприяти розвитку астми, ХОЗЛ, раку легенів та інших захворювань легенів.

5) *Деякі захворювання*, що вражають легені, наприклад, грип, спричинені мікробами (бактеріями, вірусами та грибками).

Ранні ознаки захворювання легенів легко не помітити. Часто ранньою ознакою захворювання легенів є відсутність звичного рівня енергії.

Ознаки та симптоми можуть відрізнитися залежно від типу захворювання легенів. Загальними ознаками є:

- проблеми з диханням;
- задишка;
- відчуваєте, що вам не вистачає повітря;
- зниження здатності до фізичних вправ;
- кашель, який не зникне;
- кашель крові або слизу;
- біль або дискомфорт при вдиху або видиху.

Якщо у вас є якісь із цих симптомів, обов'язково зателефонуйте своєму лікарю.

1.2.1 Симптоми астми та її лікування

Діагностувати астму буває важко. Ознаки астми можуть здаватися ознаками ХОЗЛ, пневмонії, бронхіту, легеневої емболії, тривоги та серцевих захворювань.

Загальними симптомами астми є:

- кашель;
- хрипи;
- ущільнення в грудях;
- задишка.

Щоб діагностувати астму, лікар запитує про ваші симптоми та те, що, здається, викликає їх, переглядає історію вашого здоров'я та робить фізичний огляд.

Для підтвердження діагнозу лікар може зробити інші обстеження, такі як:

1) *Спірометрія*. Лікар використовує медичний апарат під назвою спірометр. Цей тест вимірює, скільки повітря ви можете вдихнути і вивести. Він також вимірює, наскільки швидко ви можете продати повітря. Лікар може також дати вам ліки, а потім повторно перевірити, чи покращуються ваші результати.

2) *Бронхопровокація*. Ваша функція легенів перевіряється за допомогою спірометрії, тоді як на легені надається більше напруги. Це може бути під час фізичних навантажень або після вдиху все більших доз спеціальної хімічної речовини або холодного повітря.

3) *Рентген грудної клітки або ЕКГ* (електрокардіограма). Ці тести можуть іноді з'ясувати, чи може інша хвороба або сторонній предмет викликати ваші симптоми.

4) *Інші тести*. Лікар може захотіти перевірити наявність інших проблем, які можуть спричинити симптоми. Сюди входить запасна шлункова кислота в горлі, проблеми з голосовими зв'язками або апное уві сні.

Астма – хронічне захворювання. Ліки можна використовувати для лікування астми, але вони не можуть їївилікувати. Ви можете допомогти контролювати свої симптоми, працюючи зі своїм лікарем, щоб створити та слідувати особистому плану

дій щодо астми. План включатиме можливі ліки та способи уникнення речей, які викликають вашу астму.

Ваш план дій щодо астми покаже:

- види ліків, які слід приймати;
- коли приймати ліки;
- як регулярно контролювати астму;
- способи уникнути того, що викликає у вас астму;
- коли зателефонувати своєму лікарю або піти в травмпункт;
- прийом ліків.

Ліки від астми діють, відкриваючи легені дихальних шляхів. Ліки, що використовуються для лікування астми, діляться на дві групи: довготривалий контроль та швидке полегшення.

Ліки для тривалого контролю слід приймати щодня, як правило, протягом тривалого періоду часу. Вони допомагають запобігти появі симптомів. Як тільки симптоми виникають, вони не дають швидкого полегшення. Ці ліки включають:

1) *Інгаляційні кортикостероїди*. Це найкращі ліки для тривалого контролю астми. Вони знімають запалення та набряки дихальних шляхів.

2) *Бета2-агоністи тривалої дії*. Ці інгаляційні ліки часто додають до низькодозованих інгаляційних кортикостероїдів для поліпшення довгострокового контролю астми.

3) *Модифікатори лейкотрієну*. Ці таблетки допомагають блокувати ланцюгову реакцію, яка викликає запалення в дихальних шляхах.

4) *Кромолін і недокроміл*. Ці ліки для інгаляцій можуть допомогти уникнути реакції дихальних шляхів у відповідь на тригер астми.

5) *Теофілін*. Це таблетка, яка допомагає відкрити дихальні шляхи.

Лікарські засоби для швидкого полегшення застосовуються лише за потреби. Сюди входять інгаляційні бета2-агоністи короткої дії та бронходилататори короткої дії, такі як альбутерол та пірбутерал. Лікарські засоби для швидкого полегшення часто знімають симптоми за лічені хвилини. Вони роблять це, швидко розслабляючи напружені м'язи навколо дихальних шляхів. Їх приймають, коли симптоми

погіршуються, щоб запобігти повноцінному нападу астми та зупинити напади, як тільки вони почалися.

Уникайте речей, які погіршують вашу астму. Поширеними причинами астми є тютюновий дим, лупа тварин, пилові кліщі, забруднення повітря, цвіль та пилок. Ви можете спробувати продукти, що не містять запахів, якщо ваша астма спровокована ароматами. Поговоріть зі своїм лікарем про алергічні знімки, якщо симптоми астми пов'язані з алергенами, яких вам не уникнути. Постріли можуть зменшити або запобігти симптомам, але астму не вилікувати. Ви можете зменшити вплив забруднення повітря, обмеживши діяльність на свіжому повітрі в дні, коли якість повітря у вашому районі погана.

1.2.2 Симптоми хронічного обструктивного захворювання легенів та його лікування

У людей з ХОЗЛ симптоми, які розвиваються дуже повільно протягом багатьох років. Як результат, багато людей ігнорують ці симптоми, поки їх хвороба не досягла глибокої стадії. ХОЗЛ можна легко діагностувати і ним можна управляти.

Симптоми ХОЗЛ включають:

- постійний кашель, який часто виробляє велику кількість слизу;
- задишка, особливо під час фізичних навантажень;
- хрипи;
- ущільнення в грудях.

Якщо у вас є деякі або всі ці симптоми, обов'язково поговоріть зі своїм лікарем.

Щоб з'ясувати, чи є у вас ХОЗЛ, лікар має:

- запитати про ваші симптоми;
- запитати про вашу історію хвороби, включаючи сімейну історію;
- запитати про вашу історію впливу речей, які можуть спричинити ХОЗЛ, таких як тютюновий дим, забруднення повітря або хімікати;

– зробити фізичний огляд, включаючи використання стетоскопа для прослуховування хрипів або інших ненормальних звуків у грудях.

Основним тестом для перевірки ХОЗЛ є спірометрія. Для цього тесту вам буде запропоновано глибоко вдихнути і подути якомога сильніше в трубку, підключену до спірометра. Ця машина вимірює, скільки повітря ви дихаєте і як швидко.

Пошкодження легенів не підлягає усуненню. Однак хворобу можна уповільнити, уникаючи певного впливу. Для курців найкращий підхід – кинути палити. Також слід обмежити вплив диму, пилу, випарів та подразнюючої пари вдома та на роботі. Також обмежте діяльність на свіжому повітрі під час оповіщення про забруднення повітря. Лікування може полегшити симптоми. Поширеними ліками є:

- 1) бронходилататори для відкриття повітряних проходів у легенях;
- 2) інгаляційні стероїди для полегшення симптомів шляхом зменшення запалення в легенях;
- 3) антибіотики для очищення інфекцій в легенях.

Пацієнтам із ХОЗЛ лікарі можуть також рекомендувати:

- 1) *Щеплення від грипу.* Грип (грип) може викликати серйозні проблеми у людей з ХОЗЛ.
- 2) *Постріли пневмонії.* Пневмококова вакцина (NOO-muh-kok-uhl) зменшує ризик деяких видів пневмонії.
- 3) *Легенева реабілітація.* Це лікування допомагає людям фізично та психічно справлятися з ХОЗЛ. Це може включати фізичні вправи, тренінги з лікування захворювання, поради щодо дієти та консультування.

4) *Киснева терапія.* Пацієнт отримує додатковий кисень через трубку або маску.

5) *Хірургія.* Іноді хірургічне втручання може допомогти людям з важкою формою ХОЗЛ почуватись краще. Хірургія трансплантації легенів стає все більш поширеною для людей з важкою емфіземою. Інша процедура, яка називається операцією зменшення об'єму легенів, також використовується для лікування деяких

пацієнтів з важкою формою ХОЗЛ типу емфіземи. У цій операції видаляється найбільш пошкоджена частина кожної легені.

1.2.3 Симптоми рака легенів та його лікування

Зазвичай немає попереджувальних ознак раннього рака легенів. На той час, як більшість людей, хворих на рак легенів, мають симптоми, рак став більш серйозним.

Симптоми рака легенів можуть включати:

- кашель, який не зникає або посилюється;
- проблеми з диханням, як задишка;
- кашель крові;
- біль у грудях;
- охриплість або хрипи;
- пневмонія, яка не зникає або проходить і повертається.

Крім того, ви можете відчувати сильну втому, втрату апетиту або незрозумілу втрату ваги. Якщо у вас є симптоми рака легенів, важливо поговорити зі своїм лікарем. Лікар запитає про історію вашого здоров'я, історію куріння та вплив шкідливих речовин. Він або вона також буде робити фізичний огляд і може запропонувати деякі тести.

Загальні тести для діагностики рака легенів включають:

1) *Рентген грудної клітки.* Рентген грудної клітки дозволяє лікарям «бачити» аномальні розростання в легенях.

2) *Комп'ютерна томографія (КТ).* КТ є потужнішим, ніж стандартні рентгенівські промені. На знімках можна виявити тонкі ознаки рака, які не виявляються на рентгенівських променях. Це може збільшити шанси виявити рак ще до його подальшого поширення.

3) *Цитологія мокротиння.* Зразок слизу, який ви кашляєте, вивчається, щоб побачити, чи є в ньому ракові клітини.

4) *Бронхоскопія*. Лікарі пропускають спеціальну трубку, яка називається бронхоскоп, через ніс або рот і опускає в легені. Вони можуть побачити легені і видалити невеликі шматочки тканини для тестування.

5) *Тонкогolgкова аспірація*. Лікарі проводять голку через грудну стінку в легені, щоб видалити невелику кількість тканини або рідини.

6) *Торакотомія* (thohr-uh-KOT-oh-mee). Лікарі розрізають грудну клітку і видаляють тканини з легенів.

Іноді лікування раку легенів використовують для того, щоб спробувативилікувати рак. В інших випадках застосовуються методи лікування, щоб зупинити розповсюдження раку та полегшити симптоми.

Ваше конкретне лікування буде залежати від:

- тип раку легенів;
- де рак і якщо він поширився на інші частини тіла;
- ваш вік і загальний стан здоров'я.

Ваш лікар може порекомендувати одне лікування або їх комбінацію.

Хірургічне втручання використовується для видалення легеневої тканини, яка має ракову пухлину. Іноді видаляється велика частина легені або вся вона. Коли рак не поширився, хірургічне втручання можевилікувати пацієнта.

Променева терапія використовує апарат для націлювання рентгенівських променів високої енергії на пухлину. Ця енергія вбиває ракові клітини. Променева терапія може полегшити біль і змусити людину почуватися краще.

Хіміотерапія використовує ліки для знищення ракових клітин. Ліки для хіміотерапії можна давати через вену або приймати у вигляді таблеток.

Цільова терапія використовує ліки для блокування росту та поширення ракових клітин. Її можна давати через вену або приймати як таблетку.

Щоб дізнатись про наукові дослідження щодо нових методів лікування раку легенів, відвідайте веб-сайт клінічних випробувань Національного інституту раку.

1.3 Способи зниження ризику захворювання легенів

Що можна зробити, щоб зменшити ризик захворювань легенів, є:

1) *Перестати палити.* Якщо ви палите, найголовніше, що ви можете зробити, - це зупинитися. Поговоріть зі своїм лікарем про найкращий спосіб кинути палити. Всі види куріння (сигарети, сигари, люльки та марихуана) можуть збільшити шанси на захворювання легенів.

2) *Уникайте пасивного куріння.* Якщо ви живете або працюєте з людьми, які палять сигарети, люльки чи сигари, попросіть їх палити надворі. Некурці мають право на робоче місце, яке не палить.

3) *Тест на вміст радону.* З'ясуйте, чи є високий вміст газового радону у вас вдома чи на робочому місці. Ви можете придбати набір для випробування радону в більшості будівельних магазинів. Американське агентство з охорони навколишнього середовища пропонує інформацію про те, як поводитися з радоном.

4) *Уникайте азбесту.* Вплив азбесту може спричинити утворення рубців на легенях, рак легенів та інші серйозні захворювання легенів. Азбест може особливо турбувати тих, чия робота з ним контактує. Сюди входять люди, які обслуговують будівлі, що мають ізоляцію або інші матеріали, що містять азбест, і люди, які ремонтують автомобільні гальма або зчеплення. Роботодавці, які працюють з азбестом, повинні пропонувати навчання з питань безпеки азбесту та повинні регулярно перевіряти рівень впливу. Вони також повинні передбачати способи обмеження впливу, наприклад, спеціальні дихальні маски, які фільтрують азбестовий пил з повітря.

5) *Захистіть себе від пилу та хімічних випарів.* Робота в запилених умовах та з хімічними речовинами може збільшити ризик захворювання легенів. І ризик полягає не тільки в промислових хімікатах. Багато продуктів, що використовуються вдома, наприклад, фарби та розчинники, можуть спричинити або посилити захворювання легенів. Прочитайте етикетки та уважно дотримуйтесь інструкцій із використання. По можливості уникайте використання продуктів, що викликають подразнення очей, носа або горла. Якщо вам не вдається їх уникнути,

використовуйте їх якомога менше і лише в добре провітрюваному приміщенні. Носіть захисні засоби, такі як спеціальна маска. Обов'язково знайте, який тип обладнання вам потрібен і як його носити.

6) *Харчуйтесь здорово.* Національний інститут раку зазначає, що дослідження показують, що вживання великої кількості фруктів або овочів може допомогти знизити ризик раку легенів. Звичайно, дієта не може відшкодувати шкоду, заподіяну нездоровою поведінкою, такою як куріння.

7) Попросіть свого лікаря, *чи слід робити тест на спірометрію.* Деякі групи рекомендують проводити планові спірометрійні дослідження людей, що перебувають у групі ризику, таких як люди старше 45 років, які палять, та особи, які піддаються впливу шкідливих для легенів речовин на роботі.

8) Попросіть свого лікаря про *захист від грипу та пневмонії щепленнями.*

Зверніться до свого лікаря, якщо у вас кашель, який не зникне, проблеми з диханням, біль або дискомфорт у грудях або будь-який інший із описаних тут симптомів.

1.4. Способи діагностики захворювання легенів та їх ризику

1.4.1. Бронхоскопія

Бронхоскопія – це техніка, яка використовується для перегляду повітряних проходів за допомогою маленької камери, яка розташована в кінці гнучкої трубки. Вони підключається до відеоекрану, на якому лікарі можуть переглядати та фотографувати ваші повітряні проходи. У трубці також є невеликий канал для збору зразків тканин з легенів, які можна використовувати для діагностики захворювань. Зазвичай це робиться амбулаторно, це означає, що ви можете повернутися додому того самого дня процедури. Бронхоскопія робиться в умовах «свідомої» седатії. Ви продовжуєте дихати самостійно, але не відчуваєте дискомфорту через наявність трубки у роті чи носі.

Ви не зможете нічого їсти чи пити після опівночі ввечері перед бронхоскопією. Однак лікар може дозволити вам приймати деякі ліки. Будь ласка, не забудьте запитати свого лікаря, як поводитись із щоденними ліками перед процедурою.

Застосований седативний препарат може залишатися у вашому тілі протягом декількох годин після завершення процедури. Можливо, вам знадобиться хтось, хто забере вас додому, оскільки після виписки вам не дозволять керувати автомобілем.

Лікар пояснить методику та потенційні ризики до отримання вашої підписаної згоди. Настав час задати будь-які конкретні запитання, які можуть виникнути у вас.

Медсестра вивчить вашу історію хвороби та вживання ліків. Для доставки будь-яких ліків під час процедури буде встановлено внутрішньовенну лінію (IV). Будуть встановлені монітори для постійної перевірки артеріального тиску, частоти серцевих скорочень та рівня кисню під час процедури. Також буде встановлена киснева маска або носові трубки, щоб забезпечити вас додатковим киснем.

Перед процедурою вам дадуть седативні ліки та потенційно оніміючі гелі або спреї.

Після того, як ви заспокоїтесь, лікар введе бронхоскоп через рот або ніс у легені. Якщо є ознаки дискомфорту, член команди буде вводити додаткові ліки, поки вам не буде комфортно. Зазвичай бронхоскопія займає близько 30-45 хвилин.

Після процедури лікар може обговорити з вами результати бронхоскопії та майбутні плани охорони здоров'я. Однак через ефект ліків від седатії ви можете не пам'ятати всіх деталей. Завжди рекомендується звертатися до лікаря до члена вашої родини або адвоката охорони здоров'я щодо результатів та будь-якого плану лікування.

Якщо була зібрана будь-яка проба для біопсії або промивання бронхів, деякі результати можуть бути доступні приблизно через два-чотири дні. Інші результати можуть зайняти тиждень або довше до остаточного звіту.

Рівень кисню в крові може знизитися під час процедури через седативну дію або в області повітря. Під час процедури команда буде вводити додатковий кисень.

Оскільки бронхоскоп вводиться через голосові зв'язки, під час та після процедури у вас може з'явитися кашель. Біль у горлі після процедури є загальним явищем, і його можна полегшити лише краплями від кашлю.

1.4.2 Процедура грудної трубки

Грудна трубка – це пластикова трубка, яка використовується для відводу рідини або повітря з грудної клітини. Повітря або рідина (наприклад, кров або гній), що накопичується в просторі між легень та грудною стінкою (плевральний простір), може спричинити колапс легені. Грудні трубки можна вставити в кінці хірургічної процедури, поки пацієнт все ще спить від наркозу або біля ліжка, використовуючи місцевий засіб для знеболення та деяку седацію. Нагрудні труби бувають найрізноманітніших форм і розмірів. Залежно від того, для чого вони потрібні, вони можуть мати діаметр від маленького, як шнурок, до великого, як підсвітка.

Грудні труби зазвичай підключені до дренажних систем, які збирають рідину і пропускають повітря з грудної клітки. Цим системам можна дозволяти пасивно стікати або застосовувати до них всмоктування.

Коли трубки грудної клітини поміщають у пацієнта, який не спить, пацієнти можуть розраховувати на отримання тієї чи іншої форми місцевого знеболюючого засобу, куди буде введена грудна трубка. Часто пацієнтам також дають ліки для полегшення тривоги. Хоча докладаються зусилля, щоб зробити процедуру більш стерпною, пацієнти все ще зазвичай відчувають певний дискомфорт. Деякі грудні трубки вставляються після того, як шкіра і м'язи грудної стінки розрізаються і м'яко розводяться. Деякі грудні трубки вставляються після того, як дріт вкладається в грудну клітку через голку, і дріт діє як доріжка, по якій трубка повинна йти. В обох випадках пацієнти часто повідомляють про певний дискомфорт після процедури, оскільки трубка лежить на ребрах і злегка рухається з кожним вдихом. На щастя, цей дискомфорт, як правило, тимчасовий.

Основна мета цієї процедури – дренування плеврального простору. Пацієнти можуть очікувати побачити або відчутти рідину або повітря, що виходять із грудної клітки. Часто пацієнти можуть відчутти, як розпалася легеня знову розширюється. Після процедури буде проведено рентген грудної клітки, щоб побачити, скільки повітря або рідини витекло, наскільки легеня знову розширилася, та визначити кінцеве положення грудної труби. Грудні труби залишаються на місці протягом різної кількості днів. Зазвичай, коли кількість рідини, що стікає з трубки, є низькою, або якщо через трубку більше не виходить повітря, її можна видалити.

Як і будь-яка хірургічна процедура, основним ризиком при розміщенні грудної трубки є кровотеча та інфекція. Практикуючі лікарі обережно уникають кровоносних судин, які проходять по нижній частині ребер під час розміщення. Щоб уникнути зараження інфекцією, всю процедуру проводять стерильно. Інші основні ризики пов'язані з пошкодженням інших структур у грудях, таких як легені та серце. Хоча травми цих конструкцій дуже рідкісні, вони можуть бути серйозними.

Грудні труби утримуються на місці швами і покриваються стерильною пов'язкою. Помірний дискомфорт у місці введення є загальним явищем. Якщо у вас сильний біль або утруднення дихання, негайно покличете допомогу. Тривалість, для якої потрібна грудна труба, варіюється, але зазвичай становить кілька днів. У певних ситуаціях пацієнтів можна відправити додому з грудною трубкою; однак у більшості випадків їх видаляють перед випискою з лікарні. Ваш медичний працівник видалить грудну трубку, розрізавши шви, які утримують її на місці. Під час видалення може виникнути легкий дискомфорт.

Які ризики? Біль під час введення та після розміщення грудної трубки: Хоча біль під час введення та легкий дискомфорт після установки є загальним явищем, ваш лікар може допомогти мінімізувати ці наслідки за допомогою знеболюючих препаратів.

- інфекція;
- неправильне розміщення;
- зсунута грудна труба;
- зникнення легені після видалення грудної трубки;

- кровотеча в місці введення;
 - кровотеча в плевральний простір (простір навколо легенів) або всередині живота;
 - травма легені;
 - травма інших органів, таких як серце, селезінка, печінка, діафрагма.
- Ваш медичний працівник вживатиме заходів для мінімізації ризику цих ускладнень. Серйозні ускладнення рідкісні.

1.4.3 Комп'ютерна томографія

Комп'ютерна томографія, яку частіше називають скануванням котів або КТ, – це діагностичний тест, який використовує серію комп'ютеризованих оглядів, зроблених під різними кутами, для створення детальних внутрішніх зображень вашого тіла. Комп'ютер збирає знімки та розміщує їх послідовно для вашого лікаря.

Порівняно з традиційними рентгенівськими променями, КТ створюють двовимірні зображення в поперечному перерізі, які надають більше інформації. Багато разів КТ замовляє лікар після того, як помітив на рентгені щось ненормальне.

Незважаючи на те, що КТ не може дати остаточного діагнозу, це корисно для оцінки захворювань легенів та таких станів, як пневмонія, рак, згустки крові або пошкодження, спричинені курінням.

Технік КТ дасть вам інструкцію лежати рівно на столі КТ, який швидко рухається через пристрій у формі пончика, який називається сканером. Вас попросять залишатися нерухомими і затримати дихання на кілька секунд. Хоча це може здатися схожим на МРТ, воно дещо відрізняється тим, що КТ-сканер відкритий і менш шумний, а КТ-дослідження займає набагато менше часу.

Залежно від того, де на вашому тілі проводиться сканування, можливо, вам доведеться зняти частину одягу та одягнути сукню, зняти окуляри та будь-які металеві предмети. Якщо ви думаєте, що можете бути вагітною, обов'язково повідомте про це лікаря та техніка.

Якщо КТ замовлено з контрастом, ви отримаєте внутрішньовенну (IV) ін'єкцію, під час якої ви можете відчути тимчасове відчуття тепла у всьому тілі. Перед скануванням вам можуть дати конкретні вказівки, які включають не їсти та не пити щонайменше за дві години до контрастного сканування, а також попередити свого лікаря або техніка, якщо у вас є проблеми з нирками та / або алергія на ліки або їжу, особливо йод.

Після комп'ютерної томографії ви зможете повернутися до звичайних занять. Складіть план зустрічі зі своїм лікарем для обговорення результатів, які, як правило, готові протягом декількох днів. Ваш лікар розгляне звіт та зображення, обговорить і порекомендує подальші дії, якщо такі є. Можливо, вам доведеться повторити КТ протягом певного періоду часу; однак, можливо, вас попросять пройти додаткове тестування або таку процедуру, як біопсія. Ви завжди можете попросити копію своїх зображень на КТ для власних записів.

Які ризики? КТ – це швидкий тест, а відкрите обладнання легше для людей з клаустрофобією. Найбільш значним ризиком КТ є опромінення. Обговоріть це зі своїм лікарем, але невеликі ризики цього випромінювання компенсуються перевагою інформації, отриманої при скануванні.

Якщо ваш лікар призначив КТ з контрастом, попередите свого лікаря та техніка, якщо у вас алергія на йод або контраст внутрішньовенно, оскільки вам може знадобитися приймати ліки перед скануванням. Цей тип КТ може вплинути на функцію нирок, тому лікар планує перевірити аналіз крові перед призначенням тесту.

1.4.4 Комп'ютерна томографія – керована біопсія легенів

Легеневі вузли – це ненормальні ділянки тіні на легенях, виявлені на рентгенологічному знімку або КТ грудної клітки. Лікарі не завжди можуть сказати, чи є вузол раком легенів, спираючись лише на ці типи візуалізації. Залежно від розміру вузлика та факторів ризику може знадобитися подальший аналіз, включаючи біопсію вузлика. Це дозволяє дослідити тканину всередині вузлика, щоб

визначити, чи містить вона рак. Одним із способів отримати цю тканину є те, що лікарі, які спеціалізуються на візуалізації (рентгенологи), використовують КТ, щоб провести голку крізь стінку грудної клітки та у вузлик легені.

У день процедури вас попросять не їсти і не пити принаймні за шість-вісім годин до цього. Також важливо повідомити свого лікаря, якщо ви приймаєте будь-які ліки, від яких кров розріджується. Перед початком процедури ви переодягнетесь в халат і ляжете на стіл сканера КТ, де вам дадуть ліки, які допоможуть розслабитися. Рентгенолог очистить ділянку шкіри, через яку пройде голка, покладе стерильну завісу навколо ділянки і зафіксує область. За допомогою КТ-сканера голку вводять через шкіру та в легені до вузлика та відбирають кілька зразків. Після процедури вас будуть спостерігати медсестри для рутинного спостереження.

Лікар, який спеціалізується на розгляді тканин під мікроскопом (патологоанатом), досліджуватиме тканини під час спостереження. Бувають випадки, коли патологоанатому не вдається поставити діагноз, що може вимагати подальших процедур для отримання тканини, таких як хірургічне втручання або бронхоскопія. Якщо це рак, вас, швидше за все, направлять до онкологічного фахівця (онколога) та / або до хірурга. Якщо виявлено іншу причину вузлика, таку як інфекція або запалення, лікар може призначити ліки для лікування основної причини.

Які ризики? Хоча біопсія легенів, проведена за допомогою КТ, є менш інвазивною, ніж інші процедури, що використовуються для отримання тканини з легеневого вузлика, проте вона не позбавлена певного ризику. Дуже мало пацієнтів можуть відчувати витік повітря через голку, що спричиняє отвір у легені. Зазвичай це заживає самостійно і не потребує подальших процедур. Але якщо витік повітря досить великий або у вас виникають симптоми через витік повітря, можливо, потрібно буде ввести трубку через шкіру та грудну стінку, щоб злити повітря з грудної порожнини. І оскільки процедура включає голку, яка проходить через шкіру, існує ймовірність того, що інфекція може потрапити в організм, що може вимагати лікування антибіотиками. Як і багато інвазивних процедур, існує невелика

ймовірність виникнення кровотечі під час процедури, що може призвести до необхідності переливання крові.

1.4.5 Ендобронхіальне ультразвукове дослідження (EBUS)

Бронхоскопія EBUS (ендобронхіальне ультразвукове дослідження) – це процедура, яка використовується для діагностики різних типів легеневих розладів, включаючи запалення, інфекції або рак. Виконуваний пульмонологом, бронхоскопія EBUS використовує гнучку трубку, яка проходить через рот і потрапляє в дихальну трубу та легені. Подібно до пристрою, що використовується під час колоноскопії, хоча і менше, приціл EBUS має відеокамеру з ультразвуковим зондом, прикріплену для створення локальних зображень легенів та прилеглих лімфатичних вузлів для точного визначення та оцінки ділянок, побачених на х-промені або скани, які потребують пильного розгляду.

Ваш лікар може отримати кров'яний аналіз до вашої процедури, а напередодні ввечері вас попросять не їсти та не пити після півночі, залежно від того, коли запланована процедура. У день процедури ви отримаєте внутрішньовенне введення для введення ліків, які забезпечать вам комфорт протягом усієї процедури. Іноді анестезію застосовують, щоб повністю заснути. Після того, як вам буде зручно або ви спите, ваш лікар вставить камеру через рот, щоб розпочати бронхоскопію EBUS.

За допомогою камери та ультразвуку ваш лікар досліджує та збирає зразки з легенів, які зазвичай беруть за допомогою маленької голки. Ви можете відчути легкий кашель та біль у горлі, але обидва вони зникають через день.

Бронхоскопія EBUS є амбулаторною процедурою, і вам, як правило, дозволяють повертатися додому після короткого періоду спостереження. Вас попросять, щоб хтось відвів вас додому з процедури.

Зразки, зібрані вашим лікарем, будуть надіслані для дослідження на наявність інфекції, запалення або раку, залежно від того, чому ви проходитье процедуру. Зазвичай це займає від трьох до п'яти днів, після чого вам зателефонує ваш лікар

або ви прийдете в кабінет для обговорення результатів та подальших кроків у догляді.

Які ризики? Бронхоскопія EBUS надзвичайно безпечна, але, як і при будь-якій медичній процедурі, існує невеликий ризик ускладнень, що може включати кровотечу з біопсії, інфекцію після процедури, низький рівень кисню під час або після процедури та дуже малий ризик колапсу легені. Всі ці ускладнення піддаються лікуванню, але, можливо, вам доведеться ненадовго потрапити до лікарні, а не повертатись додому того ж дня, що і процедура. Не забудьте повідомити свого лікаря, якщо раніше у вас були проблеми з анестезією або седативними препаратами.

1.4.6 Тест на оксид азоту на видиху

Тест на рівень оксиду азоту на видиху може допомогти діагностувати та лікувати астму. Він вимірює кількість оксиду азоту, який видихається при вдиху. Підвищений рівень оксиду азоту пов'язаний з набряком дихальних шляхів легенів. Цей тест можна використовувати для визначення того, чи хтось, хто лікується від астми, добре реагує на певні ліки.

Тест займає менше п'яти хвилин, щоб виконати його, і подібний до іншого тесту на дихання, який називається тестом легеневої функції або спірометрією. Там буде технік, який допоможе і направить вас.

Вам буде запропоновано:

- надягніть на ніс затискачі;
- на видиху або на вдиху повністю випорожніть легені;
- покладіть мундштук у рот і вдихніть, або вдихніть повільно і глибоко, щоб наповнити легені повітрям;
- видихайте повільно і рівномірно, поки не почуєте звуковий сигнал або не загориться індикатор. Ваш лікар може попросити вас спостерігати за монітором комп'ютера, який реєструє, скільки ви дихаєте, щоб ви могли підтримувати

рівномірний видих. Ймовірно, вам доведеться повторити тест кілька разів, щоб підтвердити свої результати.

Вимірювання використовуються для визначення:

- якщо вам слід приймати стероїди для лікування астми;
- якщо ви вже приймаєте стероїди, результати покажуть, наскільки ефективні ваші ліки;
- якщо ви дотримуетесь плану лікування астми;
- якщо вам слід змінити свій план терапії; Вас можуть попросити робити тест три-чотири рази на рік.

У деяких випадках результати використовуються для діагностики астми разом з результатами інших діагностичних тестів.

Вищий за норму рівень оксиду азоту означає, що у слизовій оболонці дихальних шляхів є запалення (набряк), або це може означати, що у вас алергічна астма. Запалення, як правило, добре реагує на терапію кортикостероїдами.

Чи безпечний цей тест? Цей тест швидкий і дуже безпечний. Рідко деякі люди можуть відчувати легку запаморочення, коли їм пропонують видихнути і вдихнути. У цьому випадку технік попросить вас посидіти і розслабитися протягом декількох хвилин, і якщо ви забажаєте, спробуйте тест ще раз.

1.4.7 Лобектомія

Лобектомія – це хірургічна процедура, при якій видаляється ціла частка легені з різних причин, які можуть включати діагностику раку легенів, інфекцію, ХОЗЛ або доброякісні пухлини. Є три частки вашої правої легені і дві частки вашої лівої легені. Процедура включає або кілька невеликих розрізів (малоінвазивних), або один довший розріз (торакотомія) збоку грудної клітки на основі вашого конкретного діагнозу. При будь-якому з цих підходів, поряд з часткою, сусідні лімфатичні вузли також видаляються через можливе поширення хвороби або раку.

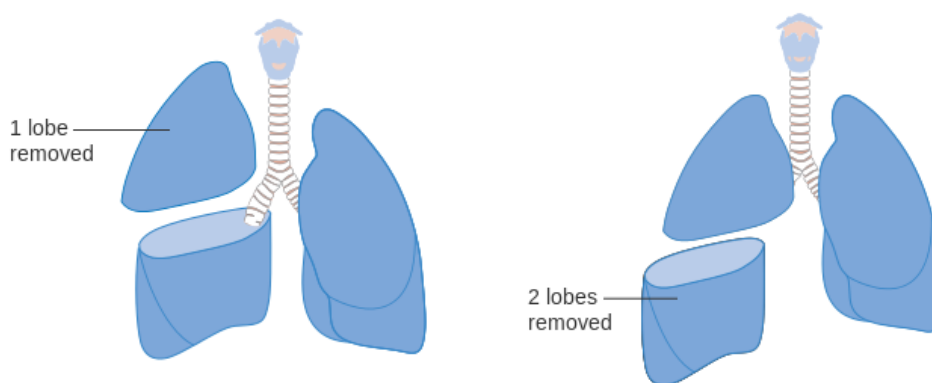


Рис. 1.1. Лобектомія

До вашої лобектомії у вас є аналізи крові та дихання (легеневі функції), а також анамнез та фізичний огляд, проведений вашим хірургом. Залежно від вашого віку та фізичного стану можуть бути рекомендовані подальші тести для перевірки вашого серця. Якщо ви палите, рекомендується кинути кинути якомога швидше, щоб зменшити ризик ускладнень під час та після операції. Перед операцією вас попросять не їсти і не пити до прибуття до лікарні. Під час операції ви будете повністю спати під загальною анестезією, а коли прокинетесь, у вас буде мати дренажну трубку, або грудну трубку, що виходить із грудей, щоб забезпечити видалення зайвої рідини та повітря.

Після операції ви, швидше за все, пробудете в лікарні два-чотири дні, залежно від уподобань хірурга та способу одужання. Очікується, що ви встанете і будете ходити, і вам буде запропоновано якомога більше залишатися з ліжка, починаючи з наступного дня після процедури. У вас буде деякий біль, який можна контролювати за допомогою ліків і зазвичай проходить протягом одного-двох тижнів. Після виписки вам дадуть конкретні вказівки щодо догляду за ранами та плани подальшого спостереження з хірургом. Результати вашої лабораторії та інші результати, отримані в результаті вашої процедури, будуть обговорені під час цього прийому, а також будь-яка подальша допомога, яка вам може знадобитися. Більшість пацієнтів повертаються до своїх звичних процедур через місяць після операції.

Які ризики? Як і будь-яка процедура, існує ризик кровотечі або інфікування. Вам будуть давати антибіотики приблизно під час операції, щоб запобігти

зараженню, і ризик кровотечі мінімальний при цій процедурі. Існує невеликий ризик того, що вам знадобиться дренажна трубка протягом тривалого періоду часу через течі в легені. Зазвичай це заживає самостійно і не потребуватиме подальших процедур. Як і будь-яка велика хірургічна операція, також існують ризики, пов'язані із загальною анестезією, такі як інфаркт, інсульт, згустки крові або пневмонія. Ваш хірург обговорить з вами кожен із цих ризиків на вашому передопераційному призначенні.

1.4.8 Тести функції легенів

Тести функції легенів (їх також називають тестами легеневої функції) включають різноманітні тести, які перевіряють, наскільки добре працюють легені. Найосновнішим тестом є спірометрія. Цей тест вимірює кількість повітря, яке можуть утримувати легені. Тест також вимірює, наскільки сильно можна випорожнити повітря з легенів.

Спірометрія використовується для скринінгу захворювань, які впливають на об'єм легенів. Він також використовується для скринінгу на захворювання, що вражають дихальні шляхи, такі як ХОЗЛ або астма.

Тестування об'єму легенів – це ще одне часто проводиться тестування функції легенів. Це точніше, ніж спірометрія, і вимірює об'єм повітря в легенях, включаючи повітря, що залишається в кінці нормального вдиху. Крім того, тест дифузійної здатності вимірює, наскільки легко кисень потрапляє в кров. Тестування на вправу допомагає оцінити причини задишки. Існують також тести, щоб з'ясувати, чи є астма, коли звичайні результати тестів на дихання нормальні.

Ці тести не болючі. Їх виконує технік легеневої функції, який вимагатиме від вас максимального зусилля для продування та вдихання повітря. Тести повторюють кілька разів, щоб переконатися, що результати є точними. Виконуючи тест, майте на увазі наступне:

- ви повинні приймати щоденні ліки перед тестуванням, якщо не сказано інше;

- не куріть принаймні шість годин до тестування;
- якщо ви використовуєте інгалятор короткої дії, який застосовується лише за потреби, не використовуйте його за шість-вісім годин до тестування, якщо це можливо;

- ваш лікар може дати вам інші вказівки щодо ліків;
- тест на вправи буде проводитись на велосипеді чи біговій доріжці, і ви повинні планувати носити вільний одяг, зручний одяг та спортивне взуття. Ви будете прикріплені до монітора серця та апарату для вимірювання артеріального тиску, щоб контролювати ваші життєві показники під час тесту. Вам буде надано додаткові вказівки щодо підготовки до цього тесту під час його замовлення.

Після тесту ви можете повернутися до звичної повсякденної діяльності. Нормальні значення обчислюються на основі віку, зросту та статі. Якщо значення ненормальне, може бути проблема з легенями. Іноді у пацієнта з нормальними легенями показник тесту на дихання може бути ненормальним. Ваш лікар пояснить, що означають результати тесту.

Які ризики? Вам потрібно зрозуміти і слідувати вказівкам для проведення тесту на функцію легенів. Тестування на вправи не слід робити тим, у кого були:

- серцевий напад або інсульт за останні три місяці;
- велика аневризма великої судини;
- велика травма голови;
- недавня операція на очах;
- спантеличеність.

Вживаються спеціальні заходи, щоб уникнути поширення мікробів між пацієнтами, які використовують одне і те ж обладнання для функціонування легенів. Технік мінятиме мундштуки та інше обладнання між пацієнтами. Спеціальні фільтри також використовуються для запобігання поширенню мікробів. Не забудьте задати питання перед початком тесту.

1.4.9 Киснева терапія

Кисень – газ, життєво важливий для життя людини. Це один із газів, який міститься у повітрі, яким ми дихаємо. Якщо у вас хронічне захворювання легенів, вам може знадобитися додатковий (додатковий) кисень для нормальної роботи ваших органів.

Ось деякі умови, які можуть вимагати додаткового кисню, тимчасово чи довгостроково:

- ХОЗЛ (хронічна обструктивна хвороба легень);
- легеневий фіброз;
- пневмонія;
- сильний напад астми;
- кістозний фіброз;
- апное сну.

Киснева терапія – це лікування, яке забезпечує вас додатковим або додатковим киснем. Хоча киснева терапія може бути поширеною в лікарні, її також можна застосовувати вдома. Існує кілька пристроїв, що використовуються для доставки кисню додому. Ваш медичний працівник допоможе вам вибрати обладнання, яке найкраще вам підходить. Зазвичай кисень доставляється через носові зубці (кисневу канюлю) або маску для обличчя. Кисневе обладнання можна приєднати до іншого медичного обладнання, такого як апарати СРАР та вентилятори.

Киснева терапія може допомогти вам почуватись краще і залишатися активними. Вивчення всього, що можна про кисневу терапію, може допомогти вам почуватись більш комфортно і впевнено. Скористайтеся посиланнями нижче, щоб дізнатися більше.

1.4.10 Пульсоксиметр

Якщо у вас є симптом задишки або відомий стан легенів або серця, лікар може застосувати пульсоксиметр. Пульсоксиметр, або Pulse Ox – це електронний пристрій, що вимірює насиченість кисню, що міститься в ваших еритроцитах. Пульсоксиметри можна прикріпити до пальців, лоба, носа, стопи, вух або пальців ніг. Потім пристрій може бути використаний повторно або утилізований. Якщо ви використовуєте це в домашніх умовах, вам слід попросити свого медичного працівника перед тим, як утилізувати імпульсний пристрій, оскільки це може бути дорогим та багаторазовим.

Пульсоксиметр, наприклад, пальцевий зонд, використовує холодне джерело світла, яке просвічує світло через кінчик пальця, завдяки чому кінчик здається червоним. Аналізуючи світло від джерела світла, яке проходить через палець, пристрій може визначити відсоток кисню в еритроцитах.

Ви можете очікувати простого, швидкого та безпечного вимірювання рівня насичення киснем у вашому тілі. Зонд буде встановлений, і протягом декількох секунд оксиметр забезпечить зчитування частоти серцевих скорочень та рівня насичення киснем, як на малюнку вище.

Якщо ви носите темний лак для нігтів, довгі штучні нігті або пальці нечисті, пульсоксиметр може працювати неправильно. У вимірюванні оксиметрії відсутні голки та біль. Деякі лікарні також використовують одноразові стрічкові зонди, які обмотують ваш палець, ніс або ногу.

Пульсоксиметр спостерігає швидке вимірювання рівня насичення киснем у вашому тілі без використання голок або взяття проби крові. Виміряна кількість, показана на екрані, відображає насиченість ваших еритроцитів киснем. Ця цифра дає вашим лікарям та медсестрам уявлення про те, яким буде ваше лікування. Рівень кисню також може допомогти визначити, чи потрібно отримувати додатковий кисень. Це число насичення (хороше число буде більше 90-92%) відрізняється від значення, яке називається pO_2 (хороше число буде більше 60-65), яке вимірюється

шляхом отримання крові з артерії. Ваш лікар може пояснити значення вашої цінності, пов'язаної з вашою конкретною ситуацією.

Які ризики? Немає відомих ризиків чи небезпек використання пульсоксиметра, коли значення переглядає та контролює компетентний медичний працівник.

ВИСНОВОК ДО 1 РОЗДІЛУ

Більшість хвороб легень є виліковними, але найроповсюдженіше захворювання легень(хронічне обструктивне захворювання легень) є невиліковною хворобою і супроводжує людину, що захворіла протягом усього її життя.

Головними причинами захворювання органів дихання є паління(у тому числі і пасивне), забруднене повітря та шкідливі випаровування, які пов'язані з деякими професіями. Найбільш розповсюдженішими симптомами хвороб легень є кашель(частіше вологий), задишка, утруднене дихання, хрипи при диханні, біль у грудях. При чому у деяких хвороб симптоми можуть бути майже ідентичними, а от серйозність захворювання – зовсім різною.

Поставити діагноз того чи іншого захворювання можна шляхом аналізу симптомів хворого, його історії хвороб та способу життя. Але через схожість симптомів такий діагноз не може бути точним. Тому для точного визначення хвороби застосовують такі процедури як рентген(є найменш точним приладом), бронхоскопія(фото дихальних шляхів з середини, а також збір тканин легень), процедура грудної трубки(використовується для відводу рідини або повітря з грудної клітини), комп'ютерна томографія(серію комп'ютеризованих оглядів, зроблених під різними кутами, для створення детальних внутрішніх зображень) тощо.

РОЗДІЛ 2

ЕКСПЕРТНІ СИСТЕМИ

На сьогоднішній день використання комп'ютерних технологій у галузі діагностики медицини значно зросло. Експертна система шукає інформацію у своїх користувачів, щоб дати рекомендації. Експертні системи призначені для вирішення складних проблем шляхом міркувань про знання, представлених переважно як правила IF-THEN, а не за допомогою звичайного процесуального кодексу. [12]

MYCIN була першою відомою медичною експертною системою, розробленою Шортліффом із Стенфордського університету, щоб допомогти лікарям, призначити протимікробні препарати при зараженні крові [13]. Деякі з дослідників розробили експертну систему діагностики ішемічної хвороби артерій із використанням міокардіальної перфузійної візуалізації [14]; та інтелектуальна медична система для діагностики захворювань кісток [15]. У цьому розділі ми представимо дизайн експертної системи для діагностики дилатаційної кардіоміопатії за допомогою CLIPS.

Експертна система – це інтерактивна та надійна комп'ютерна система прийняття рішень, яка використовує як факти, так і евристику для вирішення складних проблем прийняття рішень. Це вважається на найвищому рівні людського інтелекту та досвіду. Метою експертної системи є вирішення найскладніших питань у певній галузі.

Експертна система в AI може вирішити багато питань, які, як правило, потребують експерта-людини. Він базується на знаннях, отриманих від експерта. Штучний інтелект та експертні системи здатні висловлювати та міркувати щодо певної галузі знань. Експертні системи були попередниками сучасних систем штучного інтелекту, глибокого навчання та машинного навчання.[3]

Кафедра КІТ (47)				НАУ 20 11 93 000 ПЗ			
Виконав	Костянян П.К.			Експертні системи	Літ.	Арк.	Аркушів
Керівник	Воронін А.М.					38	37
Консульт.					УС-211М		122
Н-контроль	Райчев І.Е.						

На сьогоднішній день використання комп'ютерних технологій у галузі діагностики медицини значно зросло. Експертна система шукає інформацію у своїх користувачів, щоб дати рекомендації. Експертні системи призначені для вирішення складних проблем шляхом міркувань про знання, представлених переважно як правила IF-THEN, а не за допомогою звичайного процесуального кодексу [12].

MYCIN була першою відомою медичною експертною системою, розробленою Шортліффом із Стенфордського університету, щоб допомогти лікарям, призначити протимікробні препарати при зараженні крові [13]. Деякі з дослідників розробили експертну систему діагностики ішемічної хвороби артерій із використанням міокардіальної перфузійної візуалізації [14]; та інтелектуальна медична система для діагностики захворювань кісток [15]. У цьому розділі ми представимо дизайн експертної системи для діагностики дилатаційної кардіоміопатії за допомогою CLIPS.

Експертна система – це інтерактивна та надійна комп'ютерна система прийняття рішень, яка використовує як факти, так і евристику для вирішення складних проблем прийняття рішень. Це вважається на найвищому рівні людського інтелекту та досвіду. Метою експертної системи є вирішення найскладніших питань у певній галузі.

Експертна система в AI може вирішити багато питань, які, як правило, потребують експерта-людини. Він базується на знаннях, отриманих від експерта. Штучний інтелект та експертні системи здатні висловлювати та міркувати щодо певної галузі знань. Експертні системи були попередниками сучасних систем штучного інтелекту, глибокого навчання та машинного навчання [3].

2.1 Призначення експертних систем



Рис. 2.1. Призначення експертних систем

Нижче наведені важливі характеристики експертної системи у ШІ:

- 1) *Найвищий рівень знань*: система експертів в АІ пропонує найвищий рівень знань. Це забезпечує оперативність, точність та творче вирішення проблем.
- 2) *Пряма реакція за часом*: Експертна система зі штучного інтелекту взаємодіє з користувачем протягом дуже розумного періоду часу. Загальний час повинен бути менше часу, який витрачається експертом, щоб отримати найбільш точне рішення тієї ж проблеми.
- 3) *Хороша надійність*: Експертна система в ШІ повинна бути надійною і не допускати помилок.
- 4) *Гнучкий*: життєво важливо, щоб він залишався гнучким, оскільки він володіє системою експертів.
- 5) *Ефективний механізм*: Експертна система у галузі штучного інтелекту повинна мати ефективний механізм управління компіляцією наявних у ній знань.
- 6) *Здатний вирішувати складні рішення та проблеми*: Експертна система здатна вирішувати складні проблеми, що стосуються рішень, та надавати рішення.

Коли використовувати експертні системи? ES не підходять для всіх типів завдань. Спочатку багато розробників активно шукали проблеми, піддаються вирішенню ES, або намагалися вирішити всі проблеми за допомогою ES. У міру накопичення досвіду увагу стало більш зосереджено на проблемах, які необхідно вирішити, а не на техніці рішення. Зверніть увагу, що в цьому курсі ми фокусуємося на методах і інструментах системної інженерії і тому були і будемо дуже стурбовані технікою рішення.

Деякі проблеми можна описати за допомогою існуючих алгоритмів або за допомогою методу статистичної оцінки. Інші проблеми, які не так чітко визначені, погано структуровані і які в даний час вимагають допомоги людини-експерта, можуть бути належним чином вирішені за допомогою ES. Фактично, методи швидко стають, поряд з моделюванням і іншим традиційним програмуванням, важливими інструментами, доступними для вирішення широкого кола проблем. Неповнота інформації характерна для задач, які можна вирішити за допомогою ES.

«Телефонний тест» часто може використовуватися, щоб допомогти визначити, чи можна вирішити проблему, яку неможливо вирішити традиційними методами, за допомогою ES. Якщо експерт в предметній області може вирішити проблему через телефонну станцію з кінцевим користувачем, ймовірно, для вирішення цієї проблеми може бути розроблена програма ES. З іншого боку, якщо користувач не може описати проблему усно, або якщо експерт не може на основі телефонного інтерв'ю послідовно прийти до розумного рішення, тоді розробка ES, швидше за все, буде незадовільною. Телефонний тест гарантує, що експерт не отримує додаткову інформацію про проблему з інших органів почуттів, і гарантує, що користувач може адекватно описати проблему словами (що важливо, оскільки від користувача ES потрібно адекватно описати проблему) [4].

2.1.1 Компоненти експертної системи

Експертна система в AI складається з таких компонентів (рис. 2.2):

1) *Інтерфейс користувача* – найважливіша частина програмного забезпечення експертної системи. Цей компонент приймає запит користувача у зручній для читання формі та передає його механізму виведення. Після цього він відображає результати для користувача. Іншими словами, це інтерфейс, який допомагає користувачеві спілкуватися з експертною системою.

2) *Механізм умовиводу* – це мозок експертної системи. Механізм умовиводу містить правила для вирішення конкретної проблеми. Це посилення на знання з бази знань. Він вибирає факти та правила, які слід застосовувати при спробі відповісти на запит користувача. Він містить міркування щодо інформації в базі знань. Це також допомагає у виведенні проблеми з пошуку рішення. Цей компонент також корисний для формулювання висновків.

3) *База знань* – це сховище фактів. Він зберігає всі знання про проблемну область. Це як великий контейнер знань, який отримують різні експерти певної галузі.

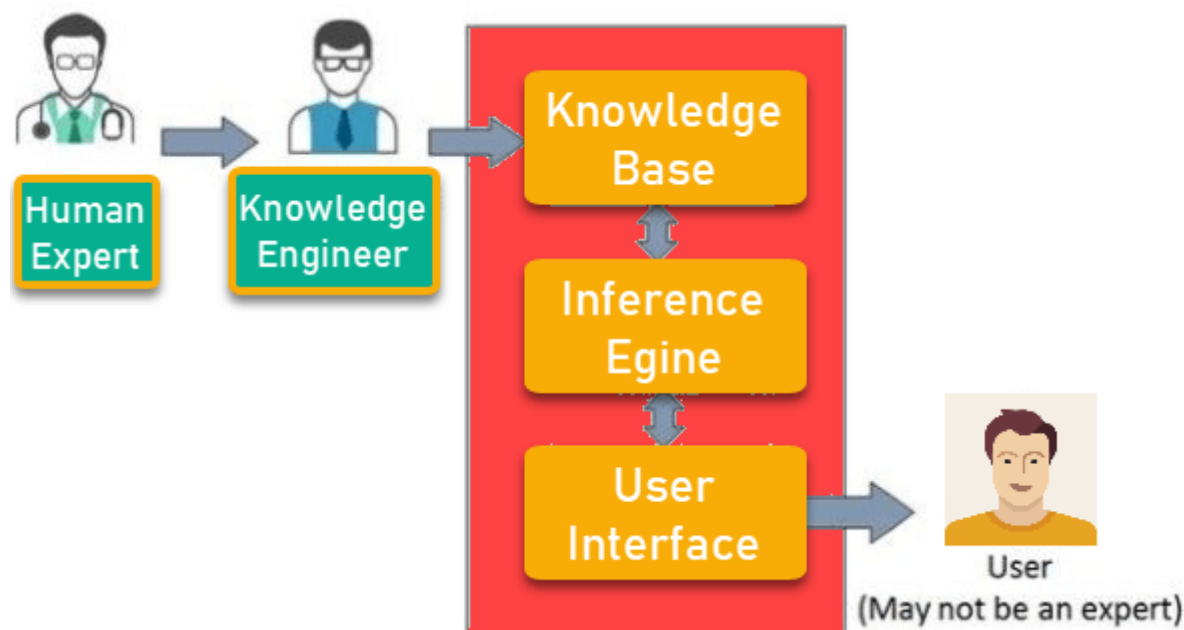


Рис. 2.2. Компоненти експертної системи

Таким чином, можна сказати, що успіх програмного забезпечення Expert System в основному залежить від високоточних та точних знань.

2.1.2. База знань

Основою експертної системи є сукупність знань (бази знань), структурованих в цілях формалізації процесу прийняття рішень.

База знань – це сукупність моделей, правил і чинників (даних), що породжують аналіз та висновки для знаходження рішень складних завдань в деякій предметній області.

Виділені і організовані у вигляді окремих, цілісних структур інформаційного забезпечення знання про предметну область стають явними і відокремлюються від інших типів знань, наприклад загальних знань. Бази знань дозволяють виконувати міркування не тільки і не стільки на основі формальної (математичної) логіки, а й на основі досвіду, фактів, евристик, тобто вони наближені до людської логіки.

Розробки в області штучного інтелекту мають на меті використання великих обсягів високоякісних спеціальних знань про деяку вузької предметної області для вирішення складних, неординарних завдань.



Рис. 2.3. Основні властивості бази знань

База знань, що обумовлює компетентність експертної системи, втілює в собі знання фахівців установи, відділу, досвід групи фахівців і являє собою інституційні знання (звід кваліфікованих, оновлюються стратегій, методів, рішень).

Знання і правила роботи можна розглядати в різних аспектах:

- глибинні і поверхневі;
- якісні і кількісні;
- наближені (невизначені) і точні (певні);
- конкретні і загальні;
- описові та розпорядчі.

Зміст бази знань може бути застосоване користувачем для отримання ефективних управлінських рішень.



Рис. 2.4. Технологія використання бази знань

Експерт – це фахівець, що вміє знаходити ефективні рішення в конкретній предметній області.

Блок придбання знань відображає накопичення бази знань, етап модифікацій знань і даних. База знань відображає можливість використання високоякісного досвіду на рівні мислення кваліфікованих фахівців, що робить експертну систему рентабельною відповідно до потреб бізнесу і замовника.

Блок логічних висновків, здійснюючи зіставлення правил з фактами, породжує ланцюжка висновків. При роботі з ненадійними даними формуються нечітка логіка, слабкі коефіцієнти впевненості, низький ступінь заходи довіри і т.д.

Блок пояснень відображає в технології використання бази знань користувачем послідовність кроків, які призвели до того чи іншого висновку з можливістю відповіді на питання «чому».

До теперішнього часу поширення баз знань в значній мірі визначається темпом накопичення професійних знань.

Та область професійної людської діяльності, яка поки піддається формалізації, а значить, і автоматизації на базі ЕОМ, – це невелика частина накопичених людиною знань.

У складі накопичених знань величезний шар складають індивідуально накопичуються невідчужувані знання. Менший обсяг складають знання, які доступні для традиційної передачі. І, нарешті, ледь помітні в загальному обсязі всіх інших знань – це формалізовані знання.

Структуризація або формалізація знань заснована на різних способах представлення знань. У сучасних системах найпопулярніший спосіб використовує факти і правила. Вони забезпечують природний спосіб опису процесів в деякій предметній області.

Правила забезпечують формальний спосіб представлення рекомендацій, вказівок, стратегій. Вони підходять в тих випадках, коли предметні знання виникають з досвідчених (емпіричних) асоціацій, накопичених за роки роботи з вирішення завдань в даній області. Правила найчастіше виражаються у вигляді тверджень типу: «якщо – то».

2.1.3 Учасники розробки експертних систем

Табл. 2.1

Учасники розробки експертних систем та їх ролі

Учасник	Роль
Експерт предметної області	Це людина чи група, досвід і знання якої використовуються для розробки експертної системи.

Інженер	Інженер - це технічна особа, яка інтегрує знання в комп'ютерні системи.
Користувач	Це людина чи група людей, які використовують експертну систему, щоб отримати консультацію, яку експерт не надасть.

2.1.4. Процес побудови експертних систем

Процес побудови експертних систем складається з:

- Визначення характеристик проблеми
- Інженер знань та експерт предметної області узгоджено працюють над визначенням проблеми
 - Інженер перекладає знання на зрозумілу для комп'ютера мову. Він розробляє механізм висновків, структуру міркувань, яка може використовувати знання за потреби.
 - Експерт предметної області також визначає, як інтегрувати використання невизначених знань у процесі міркувань та який тип пояснення буде корисним.

Якщо рішення ES є гідною кандидатурою, слід підходити до розробки систематично, так само як кроки системної методології та кроки розробки моделі, розглянуті раніше в семестрі. Цей процес в основному полягає в доопрацюванні та розширенні прототипу. База знань збільшується як в глибину, так і вшир за рахунок організаційних і репрезентативних поліпшень, допомагаючи направляти послідовні етапи розвитку. Прототип стає основою для подальшої розробки, незалежно від того, допрацьовується він або відкидається, а процес перезапускається. Це допомагає визначити підходи, які мають найбільші гідності, та інші, від яких слід відмовитися. Ці рішення можна приймати заздалегідь, що зводить до мінімуму витрати на розробку.

Швидке прототипування дає уявлення про те, яким буде готовий продукт. Важливо повідомляти про хід виконання будь-якого проекту і стежити за його

ходом не тільки для фінансують агентств і керівників, а й для експертів в предметній області, які зацікавлені в тому, щоб найкращим чином використовувати дорогоцінний час. Прототипи повинні бути документально оформлених індикаторами прогресу. Це основна сила ES в порівнянні з традиційними підходами до програмування.

Було запропоновано кілька загальних підходів до розробки ЕС. Уотерман (1986) представив найбільш широко поширений підхід:

- ідентифікація
- концептуалізація
- формалізація
- Реалізація
- тестування

Ці етапи дуже взаємопов'язані і взаємозалежні. Ітераційний процес продовжується до тих пір, поки програмне забезпечення не буде стабільно працювати на прийнятному рівні. Зверніть увагу, що описані вище кроки по суті відносяться до розробки моделі і методології системного аналізу.

2.1.4.1. Ідентифікація

Ідентифікація – це етап аналізу вимог, що виконується при традиційній розробці програмного забезпечення. Він включає в себе формальний аналіз завдання для визначення зовнішніх вимог, форми введення і виведення, установки того, де програма буде використовуватися, і визначення користувача (дуже важливо!). На цьому етапі необхідно чітко визначити учасників, проблеми, мети, ресурси, витрати і тимчасові рамки.

Учасники – група, що спонсорує роботу, експерт в предметної області і інженер по знаннях. Вибір відповідного експерта в предметній області має важливе значення для успіху проекту. Експерт в предметної області повинен бути законним авторитетом в предметної області, так як програмне забезпечення повинно мати високоякісними знаннями, і у цієї людини повинно бути час і інтерес для участі в

проекті. Потрібно не тільки особисту участь експерта, а й адміністративна підтримка роботодавця, щоб звільнити людину від деяких існуючих обов'язків. Розвиток - нетривіальне завдання, особливо коли ви намагаєтеся виконати вже зайняту на повну ставку роботу.

Хоча використання експертів в людській області є типовим методом розробки, слід зазначити, що кілька успішних програм були створені з використанням тільки довідкових матеріалів або з мінімальною участю фахівця в області людської діяльності. Може здатися, що це суперечить більш-менш прийнятим визначенням ES, однак ці програми дійсно використовують методи програмування, такі як зворотна ланцюжок для пошуку значень параметрів програми, пояснення логіки програми і т.д. сам по собі повинен існувати, або може бути досить тлумача знання. Програми, в яких не бере участь експерт, зазвичай називають системами, заснованими на знаннях, системами знань або системами, заснованими на правилах.

Більшість цих інтерпретацій є запити до бази даних, що використовуються для поліпшення пошуку відповідної інформації в товстому довідковому керівництві, наприклад, інформації про бур'янах або хімічних речовинах, або, можливо, для пошуку конкретної інформації у великому керівництві з діагностики. Можна стверджувати, що був задіяний великий людський досвід не тільки для підготовки вихідного довідкового матеріалу, але і з боку програміста, який знає предметну область, для перетворення інформації в формат і послідовність, необхідні для вирішення проблеми або відповіді на питання. Це розмиття визначення традиційного процесу розробки ES триватиме, оскільки методи інженерії знань, вперше розроблені дослідниками II, будуть включені поряд з традиційними мовами програмування в програми і програмне забезпечення для управління базами даних.

Щоб виправдати час і вартість розробки, проблема повинна бути важливою для організації, яка фінансує і бути чітко визначена. Хоча розробник не може ігнорувати взаємодії між проблемою і іншою частиною предметної області, слід докласти зусиль, щоб обмежити проблемну область, щоб рекомендації програми були конкретними і цінними, а не загальноосвітніми. Вибір глибини замість широти не тільки робить програму більш потужною і корисною, але і більш ефективною,

оскільки зводить до мінімуму обсяг інформації, який повинен бути отриманий від користувача перед тим, як дати рекомендацію. Наприклад, для користувача, у якого є проблема з шкідниками сої, було б більш ефективно запуснути програму, що має справу з шкідниками соєвих бобів, а не програму, що має відношення до виробництва сої в цілому,

Конкретні цілі або завдання програмного забезпечення повинні бути прийняті всіма сторонами. Цілі повинні бути чимось більшим, ніж просто рішення проблем. Важливо ретельно враховувати передісторію і потреби кінцевого користувача.

Яким би важливим і очевидним не здавався правильно розроблений для користувача інтерфейс, їм часто нехтують. Часто боротьба за заповнення бази знань настільки важка і вимагає багато часу, що у розробників залишається мало енергії для призначеного для користувача інтерфейсу.

Фінансування і час – основні ресурси, які слід враховувати. Додаткові ресурси, які необхідно визначити, включають джерела знань, комп'ютерне обладнання та програмне забезпечення для розробки. Як і у випадку всіх програмних проєктів, ці оцінки складні, але вони повинні бути реалістичними. Бюджетні витрати повинні включати вартість втрати продуктивності експерта і програміста, які будуть присвячувати час цим зусиллям, і поточні витрати на підтримку бази знань. Точно так же очікувані вигоди повинні включати оцінку економії дорогоцінного часу в майбутні роки.

Слід зробити деяку оцінку строку корисного використання програми. Додаткові питання включають в себе, як часто потрібно досвід, вартість і наявність альтернативних методів вирішення проблеми і ймовірна прийнятність на робочому місці. Реалістична оцінка витрат і вигод може допомогти встановити рівень деталізації програми, який може бути виправданий.

Доступне для поставки обладнання може сильно вплинути на вибір комп'ютера, використововуваного для розробки, оскільки розробник повинен визначити обсяг довідкових повідомлень, графіку, форму задається питання, обсяг і формат виведення, а також необхідність взаємодії з іншими програмами та бази даних. Багато проблем усунення неполадок і класифікації вимагають введення

даних на основі результатів сенсорного дослідження (візуального, нюху, дотику і т. Д.) Навколишнього середовища.

Кольорова графіка з високою роздільною здатністю повинна бути особливо корисна при пошуку і усунення несправностей в сільському господарстві або при класифікації. Там, де це вигідно, слід використовувати високоякісну недорогу графіку для ПК, а також кольорові сканери з високою роздільною здатністю і пристрої відеозахоплення, щоб зменшити потенційну плутанину з боку користувача при відповідях на питання, що задаються програмою, або при інтерпретації результатів програми. Чим менш досвідчений кінцевий користувач з комп'ютерним обладнанням і програмним забезпеченням, тим більше зусиль необхідно докласти для створення інтерфейсу користувача з машиною. ЕС мають додаткову перевагу в тому, що вони більш прозорі (потік програми може бути представлений користувачеві за запитом), ніж звичайні програми, і цю здатність слід використовувати, якщо користувач, ймовірно, скептично ставиться до висновків комп'ютера «чорного ящика».

2.1.4.2. Концептуалізація

Другий етап розробки ЕС, концептуалізація, включає розробку пропонованої програми, щоб гарантувати розуміння і визначення конкретних взаємодій і відносин в проблемній області. Визначено ключові поняття, взаємозв'язки між об'єктами і процесами і механізми управління. Це початковий етап придбання знань. Він включає в себе конкретну характеристику ситуації і визначає досвід, необхідний для вирішення проблеми.

Інженер по знаннях може використовувати такі питання, щоб допомогти зрозуміти, що робить експерт:

- Які саме рішення приймає експерт?
- Які результати рішення?
- Які результати вимагають більшого роздуми, дослідження або взаємодії?
- Які ресурси або вклад необхідні для прийняття рішення?

- Які умови присутні, коли вирішується конкретний результат?
- Наскільки послідовно ці умови пророкують даний результат?
- В який момент після отримання впливової інформації приймається рішення?

– З урахуванням деталей конкретного випадку, чи будуть прогнози команди інженерів знань узгоджуватися з прогнозами експерта?

Використовується один з декількох або комбінації декількох методів придбання знань. Додаткові відомості представлені в модулі «Отримання знань».

Типовий підхід полягає в тому, щоб охарактеризувати питання, які кінцевий користувач може задати експерту в предметній області, і діапазон можливих рішень. Один із способів почати роботу - це почати з ряду остаточних рекомендацій, а потім прокладати шляхи до них. Наприклад, при розробці ЕС для усунення екологічних проблем на підприємствах тваринництва (спрощено для прикладу) верхній рівень програмування може включати такі типові симптоми та рекомендації:

- тварини дуже холодні ==> додайте ізоляцію і / або обігрівач
- висока вологість ==> додати обігрівач і / або збільшити інтенсивність вентиляції
- тваринам занадто жарко ==> збільште вентиляцію і / або додайте ізоляцію і / або зменшіть щільність тварин

Процес розробки за межами цієї точки в основному складається з уточнення і додавання деталей після того, як цей верхній рівень буде встановлено. Наприклад, в номер один вище може бути додана додаткова інформація, щоб допомогти визначити, чи вірна гіпотеза «тварини дуже холодні». Це не так просто, як може здатися на перший погляд, оскільки температура в приміщенні сама по собі не є точним показником комфорту тварин. Інші міркування включають в себе, сухий чи підлогу і чи добре він укладений, використовуваний матеріал для підлоги, протяг в будівлі, де в загоні зазвичай зупиняються тварини, чи є у всіх тварин в будівлі схожі симптоми або проблема ізольована. виникає, якщо тварини витягнуті або притиснуті

один до одного, якщо їх волосся відкинуті назад або стає дибки, або якщо вони помітно тремтять.

Крім того, для визначення конкретних ліків потрібна більш детальна інформація. Остаточна рекомендація в пункті номер один буде залежати від відповідей на питання, які підтверджують або спростовують гіпотезу про те, що тварини дуже холодні, і якщо вони замерзли, то в чому причина. Наприклад, якщо встановлено, що в приміщенні низький рівень ізоляції, остаточні рекомендації будуть залежати від типу і віку тварин, що утримуються, кліматичних умов влітку і взимку для розташування будівлі, чи будуть тварини в фізичному контакті з стіною, яка містить ізоляційний матеріал, а також державні і місцеві будівельні норми і правила пожежної безпеки. Точно так же рекомендований тип обігрівача залежить від типу і віку тварин, що утримуються, типу і стану будівлі, місцевих правил, типу і вартості доступного палива, кліматичних умов,

Робота інженера по знаннях полягає в тому, щоб визначити джерела знань, які потрібні фахівцеві в предметній області при винесенні конкретної рекомендації, т. Е. Визначити довідники, до яких слід звертатися, обчислення, які необхідно виконати (або інші комп'ютерні програми), і які практичні правила (евристика) вступають в гру. Інформація, яку користувач, швидше за все, не впізнає, повинна бути визначена і представлена додатковими правилами або іншими структурами знань. Додаткова інформація, необхідна для застосування цих правил, може бути отримана від користувача або створені додаткові правила. Ця структура зазвичай створюється шляхом частих і інтенсивних інтерв'ю з експертом в предметній області.

Слід шукати можливості угруповання, ранжування та впорядкування знань. Наприклад, в задачі вентиляції, як тільки експерт дізнається, що тварина, що міститься в приміщенні, пороситься або годує грудьми, він автоматично відкидає великі частини бази знань, що відносяться до більш великим тваринам, тим самим звужуючи простір пошуку. Часто експерту представляють 3-5 сценаріїв потенційних проблем на кожному сеансі співбесіди з інженером по знаннях, який видає себе за кінцевого користувача, можливо, як допитливого користувача, який постійно

запитує експерта про мету його питання і докладного обґрунтування його відповідей. . Це дещо схоже на наполегливий дитина, запитувач: «Чому?»

Зібрана і проаналізована інформація формує основу сценаріїв, які будуть представлені на наступній сесії з експертом. Правильно і повністю описати логіку вирішення проблем експертом складно, тому що справжні експерти зазвичай не знають точно, як вони приходять до рішення, і тому часто не можуть ефективно висловити словами свій власний процес вирішення проблеми. Ретельне вивчення докладних випадків часто виявляє закономірності в процесі рішення, які все ще залишаються неясними. Необхідні уточнення концепцій і взаємозв'язків стануть очевидними в ході поглибленого аналізу. Крім того, записи інтерв'ю між експертом і клієнтами можуть бути корисні, якщо всі сторони згодні на запис. Це може визначити моменти, які зазвичай можуть бути втрачені під час контрольованого сеансу між експертом і інженером, який виконує роль користувача. Це також може допомогти запобігти перетворенню процесу в академічне вправа і забезпечити задоволення потреб кінцевого користувача.

Легко задокументувати всі, що відомо про предмет, і в процесі випустити з уваги початковий задум проблеми. Наприклад, для розробки системи рекомендацій по боротьбі з бур'янами виникає спокуса створити програму, яка конкретно визначає рід всіх можливих бур'янів, виявлених в регіоні. Це зажадає від користувача великої кількості даних, які можуть бути зайвими. Можливо, єдиною важливою інформацією є те, чи є бур'ян широколистих або травою, для якої буде рекомендований один з двох типів гербіцидів, затверджених для конкретної культури.

Деякі інструменти розробки ES мають індуктивні функції, які дозволяють створювати правила на основі прикладів, створених експертом. Такі підходи до розробки часто корисні для завдань класифікації. Нейронні мережі також функціонують приблизно так само і будуть вивчені в майбутніх завданнях.

2.1.4.3. Формалізація

Формалізація включає в себе організацію ключових понять, підзадач і інформаційного потоку в формальні уявлення. Фактично, на цьому етапі розробляється логіка програми. Часто буває корисно згрупувати або розбити на модулі зібрані знання, можливо, навіть спробувати відобразити кроки вирішення проблеми графічно.

По суті, завдання інженерів по знаннях - побудувати набір взаємозв'язаних деревовидних структур для подання бази знань. Вони повинні вирішити, які атрибути необхідно визначити для вирішення проблеми, а потім, які з цих атрибутів слід запитати у користувача або уявити внутрішнім набором дерев рішень. Хоча дерева рішень привабливі своєю простотою і є хорошим способом почати формалізувати знання у вигляді схеми представлення знань, яку можна візуалізувати, на практиці рідко буває так просто, і суворе дотримання деревовидної структури рідко буває задовільним.

Подання знань важливо для довіри і прийняття користувачем. Питання, що ставляться і перевіряються правила повинні бути в тій же послідовності, що і експерт-людина. Питання і їх порядок визначаються шляхом надання експерту кількох докладних сценаріїв. При виробленні рекомендацій координуються деталізація і структура концепцій, в тому числі те, як концепції пов'язані в логічний потік і як пов'язані невизначеності.

Проблемна область аналізується для виявлення незрозумілих поведінкових і математичних моделей, які можуть існувати в процесі прийняття рішень. Зізнаються характеристики необхідної інформації. Звідси випливає, що в міру визначення і пояснення невизначеностей відповідні взаємозв'язки стають більш зрозумілими і, в кінцевому підсумку, можуть бути пояснені з використанням традиційних методів програмування більш доцільним чином. Відповідно, процес розробки програми функціонує як збирач знань, який можна використовувати для дослідження погано понятих відносин.

Важко відокремити етап концептуалізації від етапу формалізації, і, в дійсності, проектування бази знань відбувається майже паралельно з придбанням знань. На етапі формалізації найбільш важливими є два елементи: (1) уточнення частин знань до їх конкретних відносин і ієрархії і (2) більш точне визначення очікуваного взаємодії користувача з системою.

2.1.4.4. Реалізація

На наступному етапі реалізації формалізовані знання відображаються або кодуються в рамках інструменту розробки для створення робочого прототипу. Зміст структур знань, правил виводу і стратегій управління, встановлених на попередніх етапах, організовано в потрібний формат. Часто інженери по знаннях використовували інструмент розробки програм для створення робочого прототипу для документування та систематизації інформації, зібраної на етапі формалізації, щоб реалізація на цьому етапі завершилася. Якщо немає, то в цей час кодуються примітки з попередніх етапів.

Слід враховувати довгострокове обслуговування. Згодом слід очікувати змін в базі знань. База знань повинна бути детально документована в міру її написання. По можливості слід мінімізувати ймовірність непорозуміння і плутанини в майбутньому. Крім того, повинні бути включені докладні обґрунтування і пояснення, щоб допомогти кінцевому користувачеві повністю зрозуміти питання, поставлені перед ним програмою, щоб користувач міг ефективно використовувати вихідні дані програми і показати користувачеві, за запитом, як рекомендація була виконана. логічно виведений.

Обсяг допомоги, яку потрібно включити, буде залежати від здібностей передбачуваного користувача. Хоча консультант може бути зацікавлений в швидкому отриманні відповіді на питання, ЕП, призначений для використання тими, хто повинен виконати рекомендацію, відрізняється. Зазвичай, щоб повірити в рекомендацію, кінцевому користувачеві необхідний доступ до припущень, які

лежать в основі рекомендації, і йому потрібне надійне обґрунтування програмних рекомендацій.

Це також момент, коли розробник повинен вирішити, як програма буде взаємодіяти з іншими комп'ютерними програмами і базами даних. Перше покоління ES були автономними програмами. У багатьох не було засобів зв'язку з операційною системою, читання або запису в бази даних.

2.1.4.5. Тестування

Останній етап, тестування, включає в себе набагато більше, ніж пошук і виправлення синтаксичних помилок. Він охоплює перевірку індивідуальних взаємин, валідацію роботи програми і оцінку корисності програмного пакета. Тестування направляє переформулювання концепцій, зміна уявлень і інші уточнення. Верифікація та валідація повинні відбуватися протягом всього процесу розробки. Перевірка доводить, що моделі в програмі є істинними відносинами. Це гарантує, що знання точно імітуються, оскільки експерт в предметній області керує програмою для всіх можливих непередбачених обставин.

Можливо, найскладніший аспект тестування - це точне управління невизначеністю, яка так чи інакше закладена в більшості ЕС. Фактори достовірності - один з найбільш поширених методів обробки невизначеності. Перевірка факторів достовірності, привласнених базі знань, в основному представляє собою процес проб і помилок, з уточненням первинних оцінок експертом в предметній області до тих пір, поки програма послідовно не надасть рекомендації на рівні достовірності, що задовольняє експерта. Щоб забезпечити точність програми, необхідно ретельно оцінити всі можливі шляхи вирішення.

Ефективна процедура валідації має вирішальне значення для успіху і прийняття програми. Під час валідації викликають заклопотаність наступні області: (1) правильність, послідовність і повнота правил; (2) здатність стратегії управління розглядати інформацію в порядку, відповідному процесу вирішення проблеми; (3) доречність інформації про те, як робляться висновки і чому потрібна певна

інформація; і, що найбільш важливо, (4) узгодження виведення комп'ютерної програми з відповідними рішеннями експерта в предметній області.

Те, як послідовність питань і висновків видається кінцевому користувачеві, може мати таке ж відношення до прийняття і використання, як і точність рекомендацій. Уроки, витягнуті з людської інженерії, не можна ігнорувати, якщо ми хочемо, щоб програма була успішною.

Перевірка - це безперервний процес, що вимагає, щоб рекомендації по вихідним даним були точними для конкретного випадку користувача. Перевірка поліпшується, дозволяючи іншим критично переглядати і рекомендувати поліпшення. Формальна оцінка проекту допомагає встановити, чи відповідає система наміченій первісній меті. Процес оцінки спрямований на виявлення проблем з достовірністю, прийемлемістю і корисністю. Це можна визначити по точності програми, яка визначається порівнянням з реальним середовищем. Включені розуміння і гнучкість програми, простота використання, адаптованість дизайну і правильність рішень.[4]

2.1.5. Переваги та обмеження експертних систем

Табл. 2.1

Звичайна система проти експертної системи

Звичайна система	Експертна система
Знання та обробка поєднані в одній одиниці.	База знань та механізм обробки - це два окремі компоненти.
Програма не робить помилок (окрім випадків, коли відбувається програмування).	Експертна система може помилитися.
Система працює лише в повному обсязі.	Експертна система оптимізується на постійній основі і може бути запущена з невеликою кількістю

	правил.
Потрібно поетапне виконання за фіксованими алгоритмами.	Виконання виконується логічно та евристично.
Потрібна повна інформація.	Він може бути функціональним, маючи достатню або недостатню інформацію.

Переваги:

- Це покращує якість рішення
- Скорочує витрати на консультації експертів для вирішення проблем
- Він забезпечує швидкі та ефективні рішення проблем у вузькій галузі спеціалізації.
- Він може зібрати дефіцитний досвід та ефективно використовувати його.
- Пропонує послідовну відповідь на повторювану задачу
- Підтримує значний рівень інформації
- Допомогає отримати швидкі та точні відповіді
- Правильне пояснення прийняття рішення
- Здатність вирішувати складні та складні питання
- Експертні системи штучного інтелекту можуть стабільно працювати, не стаючи емоційними, напруженими чи втомленими.

Обмеження експертної системи:

- Неможливо зробити творчий відгук у надзвичайній ситуації
- Помилки в базі знань можуть призвести до неправильного рішення
- Вартість обслуговування експертної системи є занадто дорогою
- Кожна проблема різна, тому рішення від експерта-людини також може бути різним та більш креативним.[3]

2.2. Приклади експертної системи

Приклади експертної системи:

- 1) MYCIN: Він базувався на зворотному ланцюжку та міг ідентифікувати різні бактерії, які можуть викликати гострі інфекції. Він також може рекомендувати препарати на основі ваги пацієнта. Це один із найкращих прикладів експертної системи.
- 2) ДЕНДРАЛ: Експертна система, що використовується для хімічного аналізу для прогнозування молекулярної структури.
- 3) PXDES: Приклад експертної системи, що використовується для прогнозування ступеня та типу раку легенів.
- 4) CaDet: Один із найкращих прикладів експертної системи, який дозволяє виявити рак на ранніх стадіях [3].

2.2.1. MYCIN

MYCIN, рання експертна система або програма штучного інтелекту (ШІ) для лікування заражень крові. У 1972 році розпочалася робота над MYCIN в Стенфордському університеті в Каліфорнії. MYCIN намагатиметься діагностувати пацієнтів на основі повідомлених симптомів та результатів медичних тестів. Програма могла б вимагати додаткової інформації щодо пацієнта, а також запропонувати додаткові лабораторні дослідження для встановлення ймовірного діагнозу, після чого вона рекомендувала би курс лікування. За запитом MYCIN пояснить міркування, що призвели до його діагностики та рекомендацій. Використовуючи близько 500 виробничих правил, MYCIN працював приблизно на тому самому рівні компетентності, що і спеціалісти-люди з питань зараження крові, і швидше краще, ніж лікарі загальної практики [5].

Піонер ШІ Аллен Ньюел назвав його "дідусем" усіх експертних систем, "тією, що запустила поле". У першій заявці на грант на проект, поданій у жовтні 1973 р., Зазначалося, що "Протягом останніх півтора року відділи клінічної фармакології та

інфекційних хвороб, а також члени кафедри комп'ютерних наук співпрацювали над початковою розробкою комп'ютерної системи (що називається MYCIN), який зможе використовувати як клінічні дані, так і рішення, що стосуються терапії інфекційних хвороб".

Інтерактивна програма, що консультує лікарів «щодо належного вибору антимікробної терапії, як визначено на основі даних мікробіологічних та клінічних хімічних лабораторій та прямих клінічних спостережень, які лікар вводив у відповідь на комп'ютерні запитання»; компонент, що пояснює знання програми з терапії інфекційних хвороб та обґрунтовує її рекомендації; і третій компонент, що дозволяє експертам у галузі терапії інфекційних хвороб "навчати систему MYCIN терапевтичним правилам прийняття рішень, які вони вважають корисними у своїй клінічній практиці".

Експертні системи представляли новий етап еволюції ШІ, переходячи від первісного акценту до загальних вирішувачів проблем, зосереджених на вираженні в коді міркувань людини, тобто висновків і висновків. Новий акцент був зроблений на знаннях, зокрема знаннях спеціалізованих (вузьких) експертів, а особливо їх евристичних знаннях.

Едвард Фейгенбаум, ключова фігура переходу до експертних систем з середини 1960-х, пояснив евристичні знання (у своїй доповіді «Інженерія знань: прикладна сторона штучного інтелекту» 1983 року) як «знання, яке становить правила експертизи, правила належної практики, засуджувальні правила галузі, правила правдоподібних міркувань ... На відміну від фактів у цій галузі, її правила експертизи, правила належного здогадування рідко записуються".

Фейгенбаум використовував MYCIN для ілюстрації різних аспектів інженерії знань і заявив, що експертні системи повинні пояснити користувачеві, як вони дійшли до своїх рекомендацій, "інакше системи не будуть надійними для своїх професійних користувачів".

Іншим важливим фактором переконання професійних користувачів прийняти експертну систему є її ефективність або обґрунтованість її рекомендацій. MYCIN показав себе досить добре, або, як Бьюкенен та Шортліфф висловилися в резюме

своїх тестів 1984 року: «MYCIN рекомендує терапевтичні дії, які є настільки ж доречними, як і дії експертів на факультеті інфекційних хвороб Стенфорда, - як оцінюють експерти, що не працюють у Стенфорді. (Точніше, сторонні експерти не погоджувались з рекомендацією MYCIN не частіше, ніж з рекомендаціями експертів Стенфорда.) "

MYCIN міг пояснити свої рекомендації своїм передбачуваним користувачам і добре працював. І все-таки, як повідомляли Бьюкенен та Шортліфф у 1984 р., «Хоча явно було нашим початковим наміром впровадити та протестувати MYCIN у лікарняних відділеннях, цей експеримент так і не був проведений. Натомість база знань про інфекційні хвороби була започаткована у 1978 році "

Чому? «Що ми не змогли побачити, так це те, що прийняття нового інструменту базується не лише на продемонстрованій потребі у поєднанні з продемонстрованою високою продуктивністю інструменту. В ретроспективі це було наївно. Прийнятність відрізняється від високої продуктивності "

Прийнятність - або її відсутність - пов'язана з багатьма проблемами: Питання про юридичну відповідальність - хто несе відповідальність за участь рекомендацій ШІ? Питання економічної ефективності - витрати на розробку та впровадження (як з точки зору системи підтримки прийняття рішень, так і підтримуваних фахівців) порівняно зі значенням продемонстрованих переваг.

Іншим питанням було те, що ми сьогодні називали б "інтеграцією робочого процесу", або, за висловом Бьюкенена та Шортліффа, "наша нездатність природним чином інтегрувати MYCIN у повсякденну діяльність практикуючих". Плавна інтеграція наприкінці 1970-х мала багато спільного із станом обчислювальної техніки на той час (програма працювала на Стенфордському мейнфреймі, до якого лікар отримував доступ за допомогою комп'ютерного терміналу). Але як і сьогодні, це також мало багато спільного з тим, щоб зробити «консультацію» невід'ємною частиною діяльності лікаря.

Нарешті, інженерія знань та експертні системи повинні були йти в ногу з постійно розширюваними та змінюваними знаннями. База знань MYCIN, як писали Бьюкенен та Шортліфф, "швидко застаріла, особливо щодо антимікробних засобів.

Цефалоспорини третього покоління були впроваджені в ці роки і мали глибокий вплив на підбір антибіотиків для ряду загальних проблем при інфекційних захворюваннях ".

Експертні системи були «ною новинкою» у 1980-х роках, і, за підрахунками, дві третини компаній із списку Fortune 500 застосовували цю технологію у повсякденній комерційній діяльності, лише закінчившись «Зимою ШІ» кінця 1980-х. Вже в 1983 році Фейгенбаум визначив «ключове вузьке місце», що призвело до їх остаточної загибелі: «Наразі знання набуваються дуже копітким чином, що нагадує одну з котеджних галузей, в якій окремі комп'ютерні вчені працюють з окремими експертами в дисциплінах, копітко роз'яснюючи евристика. У найближчі десятиліття ми повинні мати більше автоматичних засобів для заміни того, що в даний час є дуже втомливою, трудомісткою та дорогою процедурою. Проблема набуття знань є ключовою проблемою вузького місця у штучному інтелекті ".

Зрештою сталася автоматизація набуття знань, але не за допомогою передбачених на той час методів. У 1988 році члени ІВМ Т.І. Дослідницький центр Уотсона опублікував "Статистичний підхід до перекладу мови", провіщаючи перехід від заснованих на правилах до ймовірнісних методів машинного перекладу та відображаючи черговий зсув в еволюції ШІ до "машинного навчання" на основі статистичного аналізу відомих прикладів, розуміння та "розуміння" поставленого завдання.

І хоча знання для Фейгенбаума були евристичними знаннями експертів у дуже конкретних областях, знання стали, особливо після появи Мережі, кожною оцифрованою сутністю, доступною через Інтернет (і не тільки), яку можна видобувати та аналізувати за допомогою машинного навчання, останнє десятиліття, в його більш досконалії версії, "глибоке навчання".

У 2007 році, за п'ять років до приходу нашої нинішньої «Шляхи штучного інтелекту», де глибоке навчання вперше продемонструвало успіх у виявленні прихованих закономірностей у великих обсягах даних, Едвард Шортліфф досліджував стан ШІ в медицині через 16 років після того, як він це зробив у ключова промова на конференції "Штучний інтелект у Європі медицини" (АІМЕ),

що проводиться раз на два роки. Відзначаючи прогрес з 1991 року в ряді областей, наприклад, у „національних та міжнародних інфраструктурах біомедичних мереж для зв'язку, обміну даними та пошуку інформації”, Шортліфф ще раз заявив у 2007 році про необхідність плавної інтеграції засобів штучного інтелекту в середовище охорони здоров'я:

«Ми повинні усвідомити, що практичний вплив штучного інтелекту в медицині в реальних умовах залежатиме від розробки інтегрованих середовищ, що дозволяють поєднувати інструменти, засновані на знаннях, з іншими програмами ... ми повинні шукати шляхи поєднання «бекенд» поняття ІІІ з такими всюдисущими системами, як електронні медичні записи, системи введення замовлення постачальників, системи звітування про результати, системи електронного призначення або (з біологічної сторони) інструменти для управління та аналізу геномних / протеомічних даних ».

На початку 2020 року новий "вік ІІІ" та його нові програми глибокого навчання в галузі охорони здоров'я зіткнулися з піднесенням нового "віку коронавірусу". ІІІ в медицині сьогодні означає застосовувати дуже складний статистичний аналіз до дуже великих баз даних, щоб допомогти розробляти методи лікування та вакцин, захищати медичних працівників та автоматизувати догляд (див. Тут, тут і тут). На відміну від MYCIN, сьогодні в центрі уваги ІІ є вірус, а не бактерії, як причина інфекційних захворювань. Як і у випадку з MYCIN (і майже п'ятдесятирічним ІІІ в медицині), ключовим питанням є інтеграція з існуючим середовищем охорони здоров'я та робота медичних працівників, перехід від досліджень до „впровадження у лікарняні палати”.

Ще на початку цього місяця Ерік Тополь, лікар, який минулого року опублікував вичерпне резюме поглибленого навчання в медицині, сказав Fortune: "Я ще не бачив нічого, у чому б нам ІІІ допоміг клінічно". Пояснена Фортуна: «Тополь є великим віруючим у все більшу роль ІІІ у медичній галузі. Використання ІІІ для видобування баз даних про наркотики та виявлення ефективних методів лікування коронавірусу (та інших захворювань) може бути корисним, але він ще не побачив жодної конкретної техніки, яка перейшла від досліджень до клінічних умов " [6].

2.2.2. DENDRAL

DENDRAL, рання експертна система, розроблена з 1965 року дослідником штучного інтелекту (EI) Едвардом Фейгенбаумом та генетиком Джошуа Ледербергом, обидва з Стенфордського університету в Каліфорнії. Евристична DENDRAL (згодом скорочена до DENDRAL) була експертною системою хімічного аналізу. Речовина, що підлягає аналізу, може, наприклад, бути складною сполукою вуглецю, водню та азоту. Починаючи із спектрографічних даних, отриманих від речовини, DENDRAL висуває гіпотезу про молекулярну структуру речовини. Ефективність DENDRAL конкурувала з хіміком-експертом у цій справі, і програма використовувалась у промисловості та наукових колах [5].

Проект DENDRAL розпочався в 1965 році. Фейгенбаум шукав середовище завдань, в якому можна було б досліджувати процеси емпіричної індукції (моделей та теорій на основі даних), і зосередив своє мислення на пошуку такого середовища завдань серед видів діяльності, якими займаються вчені. Ледерберг, генетик, робота якого в 1965 р. Щодо екзобіології включала мас-спектри амінокислот, запропонував завдання аналізу мас-спектрів - формування гіпотез про органічну молекулярну структуру з даних мас-спектра. Бьюкенен долучився до цих зусиль незабаром після цього; його орієнтацією була філософія науки, змішана з ШІ, турбота про природу наукових відкриттів та інформаційні процеси, що лежать в їх основі. Робота DENDRAL була в основному експериментальною роботою. Один із ранніх результатів експерименту був, мабуть, і найважливішим. Це була гіпотеза знання-сила-сила, яка стала гаслом, за яким багато хто пам'ятає проєкт DENDRAL. Коли ми розширювали межі здібностей DENDRAL, ми виявили, що нам більше за все потрібно більше спеціальних знань з хімії та мас-спектрометрії (наявність більш потужних методів вирішення проблем AI було корисним, але не вирішальним для нашого успіху; більше знань було вирішальним). З цією метою ми найняли співпрацю Джерасі, фахівця світового класу в галузі мас-спектрометрії, та Джерасі, його команди дослідників, відвідувачів та докторів.

Одне з наших важливих ранніх спонукань, дослідження процесів формування теорії (в науковій роботі та в інших місцях) відкладалося на кілька років. Ми прийняли рішення (у правильній ретроспективі) досягти експериментальних результатів та отримати досвід побудови системи спочатку більш конкретної проблеми: проблеми формування гіпотези - виведення з одного набору спектральних даних однієї (або кількох) кандидатних молекулярних структур . Успіх з DENDRAL повернув нас до вихідної проблеми формування теорії, яка тепер постала у цілком конкретній і конкретній формі, яка була "мета" DENDRAL (звідси і назва проекту, Meta-DENDRAL). Якщо знання мас-спектрометрії мали вирішальне значення для прогресу DENDRAL, то ми повинні його кодифікувати, і ми знали лише два шляхи. Будь-яка робота кропітко, один на один, у нашій взаємодії з хіміками Джерассі (яка з тих пір стала відомою як частина набуття знань в інженерії знань). Або зробити висновок про знання безпосередньо з електронних бібліотек мас-спектральних даних та відомих структур, що породили дані. Останнє було завданням навчальної програми Meta-DENDRAL.

Як DENDRAL вплинув на ШІ? У науці про штучний інтелект 1965 року переважно домінувала тема "загального виміру" вирішення проблем. На центральному етапі була програма GPS, але модель GPS була оскаржена вирішувачами проблем на основі теореми, що підтверджує використання нещодавно відкритого та запрограмованого методу роздільної здатності. В обговоренні навряд чи згадувалася роль знань у вирішенні проблем. Наприклад, у дуже важливій ранній статті Ньюелла, Саймона та Шоу [11], яка описує їх програму гри в шахи, слово "знання" вживається лише один раз, і це використання є випадковим для основної статті.

До 1967 року проект DENDRAL зіткнувся з кризою представлення знань. Кількість нових знань (представлених як код LISP), що надходили через взаємодію з хіміками щодо набуття знань, створило складність бази знань, яку не могли ні управляти, ні підтримувати. Спочатку натхненні використанням виробництва Ньюеллом Саймоном виробництва як архітектури для вирішення проблем, концептуалізували постановки як модульні "правила" ситуації та дії, з точки зору

яких могли б представити всю базу знань DENDRAL, забезпечили чітке представлення решти проекту DENDRAL і дали собі правильний репрезентативний важіль для роботи, що відбудеться незабаром MYCIN. Цей внесок DENDRAL у ШІ був одним із найпотужніших. Оскільки експертні системи перейшли у промислове та комерційне використання, більшість реалізацій були заснованими на правилах. [7]

Meta-Dendral - це система машинного навчання, яка приймає набір можливих хімічних структур та відповідні мас-спектри як вхідні дані та пропонує набір правил мас-спектрометрії, які співвідносять структурні особливості з процесами, що виробляють спектр мас. [8] Ці правила будуть передані Евристичному Дендралу (у програмах планування та випробувань, описаних нижче), щоб перевірити їх придатність. [9] Таким чином, "Евристичний Дендрал - це система ефективності, а Мета-Дендрал - система навчання". [8] Програма базується на двох важливих особливостях: парадигма генерації-тестування плану та інженерія знань. [8]

Парадигма генерації-тестування плану є основною організацією методу вирішення проблем і є загальною парадигмою, що використовується як евристичною дендральною, так і мета-дендральною системами. [8] Генератор (згодом названий CONGEN) генерує потенційні рішення для певної проблеми, які потім виражаються у вигляді хімічних графіків у Дендралі. [8] Однак це можливо лише тоді, коли кількість рішень-кандидатів є мінімальною. Коли існує велика кількість можливих рішень, Дендрал повинен знайти спосіб встановити обмеження, що виключають великі набори рішень-кандидатів. [8] Це головна мета планувальника Дендрала, що є програмою «формування гіпотез», яка використовує «знання, специфічні для завдання, для пошуку обмежень для генератора» [8]. І останнє, але не менш важливе: тестувальник аналізує кожне запропоноване рішення кандидата та відкидає ті, які не відповідають певним критеріям. [8] Цей механізм парадигми генерування тестування плану є тим, що утримує Дендрала разом. [8]

Основною метою інженерії знань є досягнення продуктивної взаємодії між наявною базою знань та методами вирішення проблем. [8] Це можливо завдяки розробці процедури, в якій велика кількість інформації про конкретну задачу

кодується в евристичні програми. [8] Отже, першим важливим компонентом інженерії знань є велика „база знань”. Дендрал володіє конкретними знаннями про техніку мас-спектрометрії, великою кількістю інформації, яка лежить в основі хімії та теорії графів, та інформацією, яка може бути корисною у пошуку рішення конкретної проблеми з з'ясуванням хімічної структури. [8] Ця "база знань" використовується як для пошуку можливих хімічних структур, що відповідають вихідним даним, так і для вивчення нових "загальних правил", які допомагають проводити пошук. Перевага, яку Dendral надає кінцевому користувачеві, навіть не експерту, - це мінімізований набір можливих рішень для перевірки вручну.

2.2.3. PXDES

Експертна система рентгенівської діагностики пневмоконіозу (PXDES) - це комп'ютерна програма, яка була розроблена на мікрокомп'ютері IBM-PC з мовою *micro-PROLOG* для надання допомоги лікарям в області пневмоконіоза. Ця експертна система включає в себе механізм виведення для вивчення тіней на рентгенівських променях.

Система PXDES складається з чотирьох модулів: бази знань, інтерфейсу пояснення, придбання знань і модуля механізмів виведення. *Vanguard* поєднує в собі технологію прийняття рішень експертними системами з кількісними методами для створення додатків експертних систем, які здаються користувачам дуже простими інструментами, подібними майстрам. Конкретним прикладом експертної системи є PXDES, яка представляє собою пневмоконіоз, захворювання легенів, рентгенівську діагностику. Вважається, що це найвищий рівень людського розуміння і здібностей. Експертна система здатна консультивати студентів, використовуючи яка дозволяє модель консультивання і модель розвиваючого консультивання.

Експертна система характеризується як інтуїтивно зрозуміла і надійна базова структура лідерства на базі ПК, яка використовує дві впевненості і евристику для вирішення складних основних питань лідерства. [10]

2.2.4. CaDet

На основі аналізу епідеміологічних та клінічних характеристик окремих пацієнтів комп'ютерна програма CaDet представляє лікарям схеми даних, які можуть вимагати клінічної уваги щодо раннього виявлення раку. У цьому дослідженні оцінювали ефективність програми як можливого інструменту допомоги клініцистам в останніх. Дані про пацієнтів були отримані за допомогою опитування від 160 здорових добровольців. Підраховані попередження про рак, сформовані комп'ютерною програмою у відповідь на ці дані, були оцінені в порівнянні з аналогічними попередженнями, наданими п'ятьма експертами-інтерністами, які розглядали ту саму інформацію сліпо. Профілі попереджень, сформовані комп'ютером для кожного з обстежених пацієнтів, сильно корелювали з тими, що надавались клініцистами. Частота попереджень комп'ютера зростала із збільшенням кількості лікарів, які подавали відповідні сповіщення (20%, 32%, 54%, 73% та 91%, відповідно, для 1, 2, 3, 4 та 5 лікарів, $p < 0,0001$) та з загальною вагою, яку лікарі призначають цим попередженням (22%, 56%, 84% та 90% для сукупних балів 1-4, 5-8, 9-12 та 13 або більше балів, відповідно, $p < 0,0001$). Зроблено висновок, що комп'ютерна програма CaDet може відігравати роль у покращенні раннього виявлення до отримання результатів подальших клінічних досліджень.[11]

2.3. Приклад створення експертної системи CLIPS

Розширена кардіоміопатія або ДКМ - це стан, при якому серце стає ослабленим і збільшеним і не може ефективно перекачувати кров. Зниження функції серця може вплинути на інші системи організму. Розширена кардіоміопатія є найпоширенішою формою неішемічної кардіоміопатії. Пацієнти можуть відчувати значні симптоми. Вони можуть включати:

- задишка
- набряки
- стенокардія

Генетичне тестування може бути важливим. Ген TTN (який кодує білок, який називається титином) відповідає за "приблизно 25% сімейних випадків ідіопатичної розширеної кардіоміопатії. Загальне збільшення серця спостерігається при нормальній рентгенографії грудної клітки. Також може бути помічений плевральний випіт, що обумовлено до легеневої венозної гіпертензії. Ехокардіограма показує дилатацію лівого шлуночка з нормальними або витонченими стінками. [16]

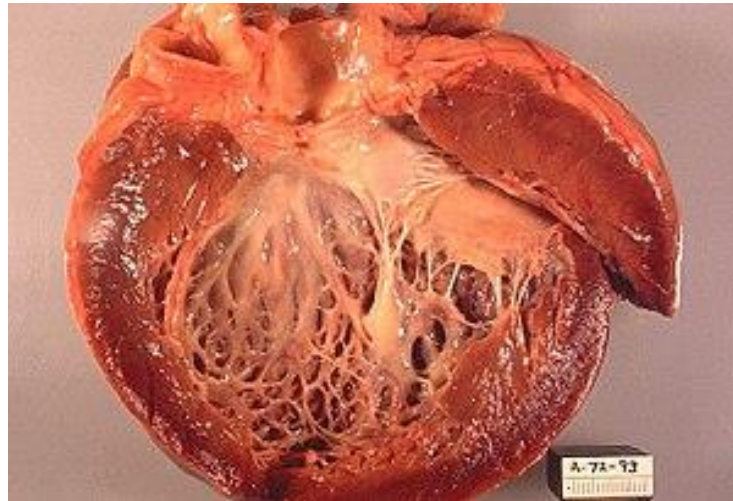


Рис.2.5. Розширена кардіоміопатія, загальний та мікроскопічний вигляд [15]

CLIPS - це програмний інструмент загального користування для побудови систем. CLIPS - це системна оболонка, що базується на знаннях та відповідає шаблонам. CLIPS розшифровується як система впровадження мови С. Мова процедурного програмування, що надається CLIPS, має особливості, подібні до мов, таких як С. Перші версії CLIPS були розроблені, починаючи з 1985 року в Космічному центрі NASA-Джонсон до середини 1990-х. Первісною назвою проекту була мова ШІ НАСА (NAIL). MYCIN - це рання експертна система, яка використовувала штучний інтелект для виявлення бактерій, що викликають важкі інфекції, такі як бактеріємія та менінгіт

- Пояснювальна система
- Система придбання правил

Система Mycin також використовувалася для діагностики захворювань на згортання крові. Він містить 500 правил і використовує певну форму невизначених

міркувань. [17] Конкретним прикладом експертної системи є PXDES - пневмоконіоз, захворювання легенів, рентгенодіагностика. Ця експертна система включає механізм висновку для дослідження тіней на рентгені. Тіні використовують для визначення типу та ступеня пневмоконіозу. Ця система також включає три інші режими: базу знань, інтерфейс пояснення та режими набуття знань. Система болю в животі DeDombal's Leeds була експертною системою для гострого болю в животі. Система Де Домбала, розроблена в Університеті Лідса, була розроблена для підтримки діагностики гострого болю в животі та, на основі аналізу, необхідності хірургічного втручання. CASNET (Causal Associational NETworks), розроблена в 1960-х роках, була загальним інструментом для побудова експертної системи для діагностики та лікування захворювань. Найбільш значним додатком експертної системи на основі CASNET була CASNET / глаукома для діагностики та лікування глаукоми.

Експертні клінічні знання були представлені в моделі причинно-асоціативної мережі (CASNET) для опису процесів захворювання. CASNET / глаукома була розроблена в Університеті Рутгерса і впроваджена у FORTRAN. Медична експертна система управління онкологічними протоколами, заснована на правилах, розроблена в Стенфордському університеті. ONCOCIN був розроблений для допомоги лікарям у лікуванні онкологічних хворих, які отримують хіміотерапію. ONCOCIN був одним із перших DSS, який намагався моделювати рішення та послідовності дій з часом, використовуючи спеціальну мову блок-схем. Це поширило техніку скелетного планування на область застосування, де важливі історія минулих подій та тривалість дій.

У статті [19] проблема розширеної хвороби кардіоміопатії реалізована за методологією систем, заснованих на правилах. Наша експертна система містить 46 правил CLIPS. CF використовується як числове значення, яке виражає ступінь переконання в тому, що певний факт чи правило відповідає дійсності.[19]

Аналіз дерева рішень давно використовується, коли бере участь багатоступеневий процес прийняття рішень.

Запитання, які може задати система, такі:

- 1) У вас ангіна? (Y / n)

- 2) Чи є у вас задишка? (Y / n)
- 3) Чи є у вас набряки? (T / H)
- 4) Введіть результат тесту BNP (стор. / Мл)
- 5) Перевірте результат ЕКГ, чи є відхилення в хвилях T і S? (Y / n)
- 6) Чи позитивний тест на ген TTN? (T / H)
- 7) Перевірте результат Ехо, чи показує він шлуночкові аритмії, збільшення лівого передсердя? (T / H)
- 8) Перевірте рентгенологічний результат, чи показує він плевральний випіт? (T / H)

```

(deftemplate person
  (slot sex)
  (slot age)
  (slot ECG)
  (slot Echo)
  (slot Xray)
  (slot Certainty(default 100.0)))

(deffacts initial
  (pahse start))

(defrule prepare-question1
  (pahse start)
=>
  (printout t "-----"crLf )
  (printout t "Do you have Angina (y/n) : " )
  (assert(angina (read))))

```

Рис.2.6. Типи правил у нашій експертній системі

```

(defrule r19
  (Sex female)
  (Age? age)
  (BNP? Bnp)
  (ECG? Ecg)
  (Test (<= ?age 46))
  (Test (> ?bnp 140))
  (ECG y)
  (Echo y)
  (Xray y)
=>
  (printout t " ======" crlf)
  (printout t "          Dilated cardiomyopathy          " crlf)
  (printout t " ======" crlf)
  (halt))

```

Рис.2.7. Типи результатів у нашій експертній системі

Результати роботи системи наведені на рисунках 2.8 та 2.9.

Female	Yes	Yes	Yes	Yes	yes	Yes
BNP-test Abnormalit y	Yes	Yes	Yes	Yes	No	Yes
ECG-test Abnormalit y	Yes	yes	Yes	Yes	-	-
ECHO Abnormalit y	Yes	No	Yes	Yes	-	-
X-Ray Abnormalit y	Yes	-	No	No	-	-
TTN-gene Test result	-	-	Yes	No	-	-
System result	Dilated Cardio myopa thy	Check TTN- Gene	Dilated cardio myopa thy	Suspected to Hypertrophic or Restrictive cardiomyopat hy	No Cardio myopa thy	Suspected to Renal failure(kidney test)

Рис.2.8. Результати нашої експертної системи (жінки)

Male	Yes	Yes	Yes	Yes	Yes	yes
BNP-test Abnormality	Yes	Yes	Yes	Yes	Yes	No
ECG-test Abnormality	Yes	yes	Yes	Yes	-	-
ECHO Abnormality	Yes	No	Yes	Yes	-	-
X-Ray Abnormality	Yes	-	No	No	-	-
TTN-gene Test result	-	-	Yes	No	-	-
System result	Dilated Cardio myopa thy	Check TTN- Gene	Dilated cardio myopa thy	Suspected to Hypertrophic or Restrictive cardiomyopat hy	Suspected to Renal failure(kidney test)	No Cardiomyo pathy

Рис.2.9. Результати нашої експертної системи (Чоловік)

Для того, щоб оцінити нашу систему, її вихідні рецепти були порівняні з двома іншими лікарями (оцінювачами). Порівняння між нашою системою та реальними лікарями довело точність нашої системи. Вони показані на рисунку 2.10.

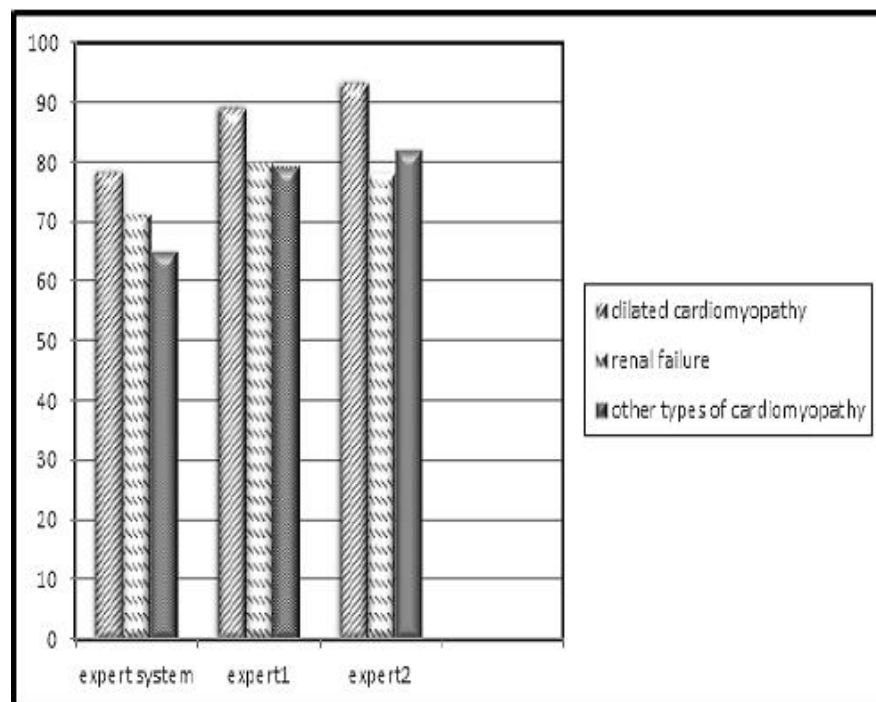


Рис.2.10. Порівняння результатів з думкою оцінювачів

ВИСНОВОК ДО РОЗДІЛУ 2

Експертна система – це інтерактивна та надійна комп'ютерна система прийняття рішень, яка використовує як факти, так і евристику для вирішення складної проблеми прийняття рішень

Основними компонентами експертної системи є 1) Інтерфейс користувача, 2) Механізм висновків, 3) База знань

Ключовими учасниками розробки експертних систем штучного інтелекту є 1) експерт доменів 2) інженер знань 3) кінцевий користувач

Покращена якість прийняття рішень, зниження вартості, послідовність, надійність, швидкість - це ключові переваги експертної системи

Експертна система не може дати творчих рішень і може коштувати дорого в обслуговуванні.

Експертна система може бути використана для широких програм, таких як фондовий ринок, склад, HR та ін

РОЗДІЛ 3

ВИКОРИСТАНІ ТЕХНОЛОГІЇ РОЗРОБКИ

Для реалізації серверної частини проекту використана технологія розробки веб-проектів ASP.NET MVC5 та мова програмування C#. Для розробки логіки для взаємодії з базою даних у проекті використана технологія Entity Framework. Для клієнтської частини використана мова розмітки Razor, а також мова розмітки веб-документу(HTML), каскадна таблиця стилів(CSS), мова програмування JavaScript, з використанням бібліотеки JQuery. Середовищем розробки додатку обрана Visual Studio 2017.

3.1. Мова програмування C#

C # - це сучасна мова програмування загального призначення, яка може бути використана для виконання широкого кола завдань та завдань, що охоплюють різні професії. C # в основному використовується на платформі Windows .NET, хоча його можна застосувати до платформи з відкритим кодом.

C # розроблений для Common Language Infrastructure (CLI), яка складається з виконуваного коду та середовища виконання, що дозволяє використовувати різні мови високого рівня на різних комп'ютерних платформах та архітектурах.

Порівняно з давніми мовами, такими як Python та PHP, C # є молодим доповненням до сімейства програмування майже двадцяти років. Мова була розроблена в 2000 році датським інженером-програмістом з історії популярних творінь Anders Hejlsberg від Microsoft. Андерс взяв участь у створенні декількох надійних інструментів програмування та мов, включаючи Microsoft TypeScript та Delphi, що є відповідною заміною Turbo Pascal.

Кафедра КІТ (47)				НАУ 20 11 93 000 ПЗ			
Виконав	Костянян П.К.			Використані технології розробки	Літ.	Арк.	Аркушів
Керівник	Воронін А.М.					75	12
Консульт.					УС-211М 122		
Н-контроль	Райчев І.Е.						

Станом на лютий 2019 року C # посідає четверте місце за рейтингом популярності програмного забезпечення PYPL, безпосередньо за Java та JavaScript. Дані, що використовуються для складання цього індексу, базуються на тому, я часто люди шукають підручник з різних мов програмування в Google.

Протягом останніх років C # також постійно виступав у першій десятці мов програмування в Індексі ТЮВЕ - звіті, який отримує дані зі збірки популярних пошукових систем, включаючи Google, YouTube та Bing.

Серед початківців та досвідчених розробників C # можуть використовувати люди з широким і різноманітним рівнем кваліфікації.

C # - чудовий вибір для розробників із середнім і просунутим досвідом написання коду. Незважаючи на те, що експерти визнають, що мова є однією з помірних складностей, вони погоджуються з тим, що її досить просто зрозуміти та досягти успіху.

Це пов'язано з тим, що C # - мова високого рівня, а це означає, що її порівняно легко читати та писати, що робить її надійним вибором для початківців та зручним варіантом для експертів. Окрім читабельності, C # може також використовуватися для автоматизації складних завдань, що вимагають багато часу для досягнення незначних результатів.

Ця мова програмування також має статистичний тип, що означає, що помилки виявляються до запуску програми. Це значно полегшує виявлення невеликих вад у вашому стеку, які в іншому випадку були б майже непомітними - не кажучи вже про неймовірно обтяжуючі.

Незважаючи на те, що C # може бути добре використаний в руках усіх видів програмістів, значну частину користувачької бази мови складають ті, хто прихильний до платформи Microsoft.

3.1.1 Сфери використання мови C#

Як і інші мови програмування загального призначення, C # може використовуватися для створення цілого ряду різних програм та додатків: мобільних

додатків, настільних програм, хмарних служб, веб-сайтів, корпоративного програмного забезпечення та ігор. Багато-багато ігор. Хоча C # надзвичайно універсальний, є три сфери, в яких він найчастіше використовується.

Розробка веб-сайтів

C # часто використовується для розробки професійних, динамічних веб-сайтів на платформі .NET або програмного забезпечення з відкритим кодом. Оскільки ця мова є об'єктно-орієнтованою, її часто використовують для розробки веб-сайтів, які є неймовірно ефективними, легко масштабованими та легко підтримуються.

Програми Windows

C # був створений Microsoft для Microsoft, тому легко зрозуміти, чому він найпопулярніший для розробки настільних додатків Windows. Додатки C # вимагають платформи Windows .NET для найкращого функціонування, тому найефективнішим варіантом використання цієї мови є розробка програм та програм, які специфічні для архітектури платформи Microsoft.

Ігри

C # може бути просто однією з найкращих мов програмування для ігор. Ця мова активно використовується для створення улюблених уболівальниками ігор, таких як Rimworld на Unity Game Engine Unity - це, безумовно, найпопулярніший ігровий движок, на якому побудовано більше третини найкращих і найпоширеніших ігор у цій галузі. C # легко інтегрується з двигуном Unity і може бути використаний практично на будь-якому сучасному мобільному пристрої або консолі завдяки крос-платформним технологіям, таким як Xamarin.

3.1.2 Переваги мови C#

Наступні причини роблять C # широко вживаною професійною мовою:

- Це сучасна мова програмування загального призначення;
- C# орієнтований на компоненти;
- Програмуванню на C# легко навчитися;
- Це структурована мова;

- Він виробляє ефективні програми;
- Він може бути складений на різних комп'ютерних платформах;
- Це частина .Net Framework.

С # може заощадити час

Мабуть, найбільшою перевагою є те, скільки часу ви можете заощадити, використовуючи С # замість іншої мови програмування. Оскільки С # набирається статично і легко читається, користувачі можуть витратити менше часу на пошук своїх скриптів на наявність крихітних помилок, які порушують роботу програми.

С # також підкреслює простоту та ефективність, тому програмісти можуть витратити менше часу на написання складних стеків коду, які неодноразово використовуються протягом проекту.

Це масштабована мова, яку легко підтримувати

С # - це мова програмування, яка надзвичайно масштабована і проста в обслуговуванні. Через сувору природу написання статичних кодів програми С # надійно узгоджуються, що робить їх набагато простішими в налаштуванні та обслуговуванні, ніж програми, написані іншими мовами. Якщо розробнику коли-небудь доведеться повернутися до старого проекту, написаного на С #, стек залишається незмінним на всьому рівні. Всьому є місце і все на своїх місцях.

Існує велика громада програмістів С#

У світі кодування та програмування важливість корисної спільноти, від якої людина, яка пише свої проекти на С# можете залежати, просто не можна переоцінити. Мови програмування - це не платформа чи послуга із спеціальною лінією довідки чи зручною ІТ-підтримкою. Програмісти повинні розраховувати на підтримку інших у тій же галузі, які зазнали однакових перешкод і розчарувань.

Одну таку спільноту корисних експертів з програмування можна знайти на StackOverflow. Оскільки цей сайт із запитаннями та відповідями був побудований на С #, не дивно, що розробники С # складають величезну частину спільноти, куди ви

можете звернутися, щоб запитати, відповісти, провести мозковий штурм чи випустити повітря.

С # є об'єктно-орієнтованим

С # повністю об'єктно-орієнтований, що є рідкісною характеристикою мови програмування. Багато найпоширеніших мов до певної міри включають об'єктну орієнтацію, але дуже мало хто досягнув величини С #, не втрачаючи прихильності людей. Об'єктно-орієнтоване програмування (або ООП) має багато різних переваг, таких як ефективність та гнучкість, щоб назвати декілька.[21]

3.2. Мова програмування JavaScript

JavaScript спочатку був відомий як LiveScript, але Netscape змінив свою назву на JavaScript, можливо, через хвилювання, яке створює Java. JavaScript вперше з'явився в Netscape 2.0 у 1995 році під назвою LiveScript. Ядро мови загального призначення було вбудовано в Netscape, Internet Explorer та інші веб-браузери.

Сторінки HTML чудово підходять для відображення статичного вмісту, напр. просте зображення або текст. Однак сьогодні більшість сторінок рідко бувають статичними. На багатьох сучасних сторінках є меню, форми, слайд-шоу та навіть зображення, що забезпечують взаємодію з користувачем. Javascript - це мова, яку веб-розробники використовують для забезпечення такої взаємодії. Оскільки JavaScript працює зі сторінками HTML, розробник повинен знати HTML, щоб використати весь потенціал цієї мови сценаріїв. Хоча існують інші мови, які можна використовувати для створення сценаріїв в Інтернеті, на практиці це, по суті, усі Javascript.

Існує два способи використання JavaScript у файлі HTML. Перший передбачає вбудовування всього коду JavaScript у HTML-код, тоді як другий метод використовує окремий файл JavaScript, який викликається з елемента Script, тобто, укладеного тегам Script. Файли JavaScript ідентифікуються за розширенням .js. Хоча JavaScript в основному використовується для взаємодії з об'єктами HTML, він

також може бути створений для взаємодії з іншими об'єктами, що не є HTML, такими як плагіни браузера, властивості CSS (каскадні таблиці стилів), поточна дата або сам браузер.

Загальні властивості мови JavaScript:

- JavaScript - це легка, інтерпретована мова програмування.
- Призначений для створення мережево орієнтованих додатків.
- Доповнює та інтегрує Java.
- Доповнює та інтегрує HTML.
- Відкритий і крос-платформний

Клієнтський JavaScript

Клієнтський JavaScript - найпоширеніша форма мови. Сценарій повинен бути включений у документ HTML або посилатися на нього, щоб код інтерпретувався браузером. Це означає, що веб-сторінка не повинна бути статичним HTML, але може включати програми, які взаємодіють з користувачем, керують браузером та динамічно створюють вміст HTML.

Клієнтський механізм JavaScript надає багато переваг перед традиційними серверними сценаріями CGI. Наприклад, ви можете використовувати JavaScript, щоб перевірити, чи ввів користувач правильну адресу електронної пошти у поле форми.

Код JavaScript виконується, коли користувач подає форму, і лише якщо всі записи є дійсними, вони будуть надіслані на веб-сервер.

JavaScript можна використовувати для захоплення ініційованих користувачем подій, таких як клацання кнопок, навігація посиланнями та інші дії, які користувач ініціює явно або неявно.

3.2.1 Переваги та недоліки мови JavaScript

Переваги JavaScript

- Менше взаємодії з сервером – дає можливість перевірити введення даних користувача, перш ніж відправляти сторінку на сервер. Це економить серверний трафік, а це означає менше навантаження на ваш сервер.
- Негайний відгук відвідувачам - їм не потрібно чекати перезавантаження сторінки, щоб побачити, чи не забули вони щось ввести.
- Підвищена інтерактивність – є змога створювати інтерфейси, які реагують, коли користувач наводить на них курсор миші або активує їх за допомогою клавіатури.
- Багатіші інтерфейси - Ви можете використовувати JavaScript, щоб включити такі елементи, як компоненти перетягування та повзунки, щоб надати розширений інтерфейс відвідувачам вашого сайту.
- Однією з головних сильних сторін JavaScript є те, що він не вимагає дорогих засобів розробки. Ви можете почати з простого текстового редактора, такого як Блокнот. Оскільки мова інтерпретується в контексті веб-браузера, вам навіть не потрібно купувати компілятор.

Обмеження JavaScript

JavaScript не можна розглядати як повноцінну мову програмування. У ньому відсутні такі важливі особливості -

- Клієнтський JavaScript не дозволяє читати або писати файли. Це було збережено з міркувань безпеки.
- JavaScript не можна використовувати для мережевих додатків, оскільки така підтримка відсутня.
- JavaScript не має багатопотокових або багатопроцесорних можливостей.

Знову ж таки, JavaScript - це легка, інтерпретована мова програмування, яка дозволяє вбудувати інтерактивність у інші статичні HTML-сторінки.[22]

3.3. Архітектурний патерн MVC

ASP.NET MVC - це платформа для розробки веб-додатків від Microsoft, яка поєднує у собі ефективність архітектури MVC(Model-View-Controller), тобто «модель-представлення-контроллер», новітні ідеї і прилади, що розвиваються, а також все найкраще з існуючих платформ ASP.NET. Вона представляє собою повну альтернативну традиційній технології ASP.NET Web Forms, надаючи істотні переваги для всіх проектів веб-розробників. Нова платформа ASP.NET MVC забезпечила радикальний вплив на розробку веб-сайтів - на платформі Microsoft. Її перевагою є ясна архітектура, шаблони проектування та тестування, а також не робиться спроб приховування того, як працює веб-середовище.

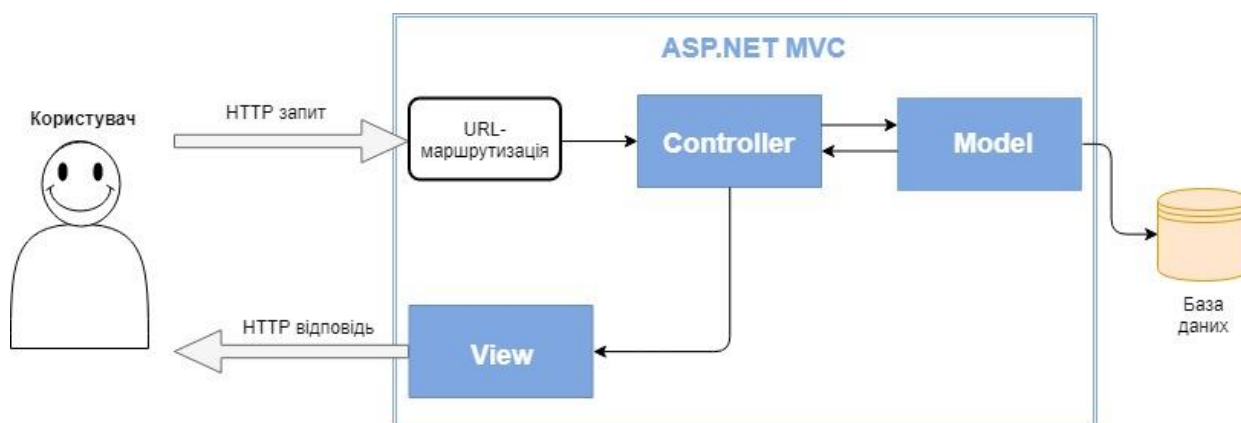


Рис. 3.3. Архітектура ASP.NET MVC

Model

Компонент Model додатка MVC представляє стан конкретної частини даних або частини даних програми і зазвичай взаємодіє з файлом, базою даних і веб-службою і т.д. Ця частина програми MVC обробляє логіку програми для даних програми. Зазвичай вона зберігає та отримує дані з бази даних. Модель містить всю бізнес-логіку програми, логіку перевірки та логіку доступу до бази даних. Якщо ми хочемо отримати доступ до бази даних, ми можемо створити класи баз даних у папці Model програми.

View

Компонент View програми MVC представляє інтерфейс користувача і містить всі взаємодії з користувачем. Компонент View приймає інформацію з компонента Controller і надає користувальницький інтерфейс для відображення цієї інформації. Компонент View програми MVC обробляє операції відображення, такі як відображення даних і інформації. Вона надає візуальну форму моделі, наприклад, коли модель витягує дані з бази даних, перегляд надає візуальну форму цим даним і відповідям назад до браузера. Компонент View у MVC надає користувачеві графічний користувальницький інтерфейс (GUI) та спеціальні форми додатків або інтерактивні вікна для користувачів, які надсилають дані користувачеві та отримують вхідні дані від користувача за допомогою графічного інтерфейсу, наприклад, Web Forms і об'єктів керування, наприклад текстових , прапорці, перемикачі, комбіновані поля та командні кнопки тощо. Оскільки ця частина програми безпосередньо пов'язана з користувачем, тому, якщо вона добре розроблена, зручна для користувача, перевірена на введення даних, є всеохоплюючою та реагуючою, вся програма буде привабливою для користувачів і буде більш вигідною та ефективною. Якщо ця частина програми розроблена погано, то все програмне забезпечення буде непрофесійним і не привабливим для користувачів.

Controller

Компонент контролера MVC працює між компонентами Model і View і відповідає за управління логікою і потоком виконання програми. Контролер обробляє будь-який вхідний запит URL-адреси. Коли ми надсилаємо запит до програми MVC, контролер обробляє наш запит і повертає відповідь на цей запит. Контролер приймає вхідні дані від користувачів і перетворює їх у команди, після чого надсилає компонент View, де він відображається в інтерфейсі користувача.

Контролер також посилає команди моделі(Model), щоб оновити її стан, наприклад, редагування документа і т.д. Контролер приймає вхід від користувача, використовуючи компонент View у MVC, і відправляє до Модального компонента

для обробки і логічних операцій, потім відправляє відповідь назад запитуваному користувачеві за допомогою View.

Контролер використовує базову логіку програми MVC і обробляє команди, логічно приймає рішення і оцінює, наприклад, якщо користувач посилає дані, контролер вирішує, яку модель використовувати для обробки введеного користувача. Додаток MVC містить один або більше контролерів, і кожен контролер в MVC є класом, який містить різні методи. Методи контролера називаються методами дій. Кожен метод дії повертає результат дії.

Переваги MVC

1. Швидкий процес розвитку:

MVC дозволяє швидку і паралельну розробку. Якщо модель MVC використовується для розробки будь-якого конкретного веб-додатку, то можливо, що один програміст може працювати над представленням(View), а інший може працювати над контролером для створення бізнес-логіки веб-додатку. Отже, таким чином, додаток, розроблений з використанням моделі MVC, може бути завершено в три рази швидше, ніж додатки, розроблені з використанням іншого шаблону розробки;

2. Можливість надання кількох представлень:

У моделі MVC можна створити кілька View для моделі. Сьогодні існує зростаючий попит на нові способи доступу до вашої програми і для цього розробка MVC, безумовно, є чудовим рішенням. Більш того, у цьому методі дублювання коду дуже обмежена, оскільки вона відокремлює дані та бізнес-логіку від того, що відображається у браузері;

3. Підтримка асинхронної техніки:

Архітектура MVC також може інтегруватися з JavaScript Framework. Це означає, що програми MVC можуть працювати навіть з файлами PDF, веб-переглядачами, а також з віджетами робочого столу. MVC також підтримує асинхронну техніку, яка допомагає розробникам розробляти додаток, що завантажується дуже швидко;

4. Модифікація не впливає на всю модель:

Для будь-якого веб-додатка користувальницький інтерфейс має тенденцію змінюватися частіше, ніж навіть ділові правила компанії. Очевидно, що ви часто змінюєте свій веб-додаток, наприклад, змінюючи кольори, шрифти, макети екрана та додаючи нову підтримку пристроїв для мобільних телефонів або планшетів. Крім того, додавання нових типів представлень дуже просто в шаблоні MVC, оскільки частина моделі не залежить від частини переглядів. Тому будь-які зміни в моделі не вплинуть на всю архітектуру;

5. Модель MVC повертає дані без форматування:

Шаблон MVC повертає дані без застосування будь-якого форматування. Отже, однакові компоненти можна використовувати і викликати для використання з будь-яким інтерфейсом. Наприклад, будь-які дані можуть бути відформатовані за допомогою HTML, але вони також можуть бути відформатовані за допомогою Macromedia Flash або Dream viewer;

6. SEO дружні платформи розвитку:

Платформа MVC підтримує розробку SEO дружніх веб-сторінок або веб-додатків. Використовуючи цю платформу, дуже легко розробити SEO-дружні URL-адреси, щоб генерувати більше відвідувань з конкретної програми. Ця архітектура розробки зазвичай використовується в додатках для тестування. Крім того, мови сценаріїв, такі як JavaScript і jQuery можуть бути інтегровані з MVC для розробки багатофункціональних веб-додатків [20].

Таким чином, шаблон дизайну MVC є безсумнівно чудовим підходом до побудови програмних додатків. MVC легко реалізовується, оскільки він пропонує вищевказані численні переваги. Проекти, розроблені за допомогою моделі MVC, можуть бути легко розроблені з меншими витратами і за менший час. Перш за все, його влада для управління кількома представленнями робить MVC кращим шаблоном архітектури для розробки веб-додатків.

ВИСНОВОК ДО РОЗДІЛУ 3

Використані технології розробки є досить складними. По-перше, мова програмування C# є повноцінною об'єктно-орієнтованою мовою програмування, за допомогою якої можна розробляти безліч різних десктопних додатків, ігор та веб-додатків. По-друге, архітектура MVC(model-view-controller) має 3 основні компоненти, які потребують детального вивчення та розуміння. Інструмент для поєднання таблиць бази даних з програмним кодом Entity Framework значно полегшує реалізацію додатків, які працюють разом із базами даних. А мова розмітки Razor надає можливість поєднувати програмний код, написаний на мові C# з кодом мовою HTML та мовою JavaScript. Обрані мною інструменти розробки дозволяють створювати складні веб додатки, з великою кількістю функціоналу, що є найбільшою перевагою.

РОЗДІЛ 4

РОЗРОБКА ЕКСПЕРТНОЇ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ВІРОГІДНОСТІ ЗАХВОРЮВАННЯ ЛЕГЕНЬ

4.1 Постановка задачі

Головною метою експертної системи є обчислити у процентному співвідношенні вірогідність різних захворювань легень на основі симптомів, які обере користувач системи.

В основі експертної системи повинна бути база даних(база знань), яка містить назви хвороб та симптоми, що їм відповідають.

У готовому варіанті системи перед користувачем повинен бути зручний та зрозумілий інтерфейс. Користувачу надається вибір з переліку симптомів(тип кашлю, температуру тіла, колір мокроти(якщо є) та загальні симптоми хвороби). Далі на основі обраних симптомів система повинна обчислити найбільш вірогідні хвороби, які найбільше відповідають обраним симптомам. Далі повинна побудуватися стовпчикова діаграма, яка відобразатиме назви підходящих хвороб та їх вірогідність у порядку спадання.

Система повинна враховувати всі можливі розповсюджені хвороби легень, а також мати у своїй базі знань захворювання, які мають схожі симптоми із симптомами захворювання легень, але насправді не є такими хворобами. До таких хвороб слід віднести ОРВІ, Грип та COVID-19.

Система також повинна враховувати те, що більшість симптомів можуть бути спільними для декількох хвороб. Тому при повному збіганні для декількох хвороб обраних симптомів, система має відобразити однакове значення вірогідності для підходящих хвороб.

Кафедра КІТ (47)				НАУ 20 11 93 000 ПЗ			
Виконав	Костанян П.К.			Розробка експертної системи для визначення вірогідності захворювання легень	Літ.	Арк.	Аркушів
Керівник	Воронін А.М.					87	6
Консульт.					УС-211М 122		
Н-контроль	Райчев І.Е.						

4.2 Розробка бази даних, алгоритму обчислення, програмного коду та інтерфейсу експертної системи

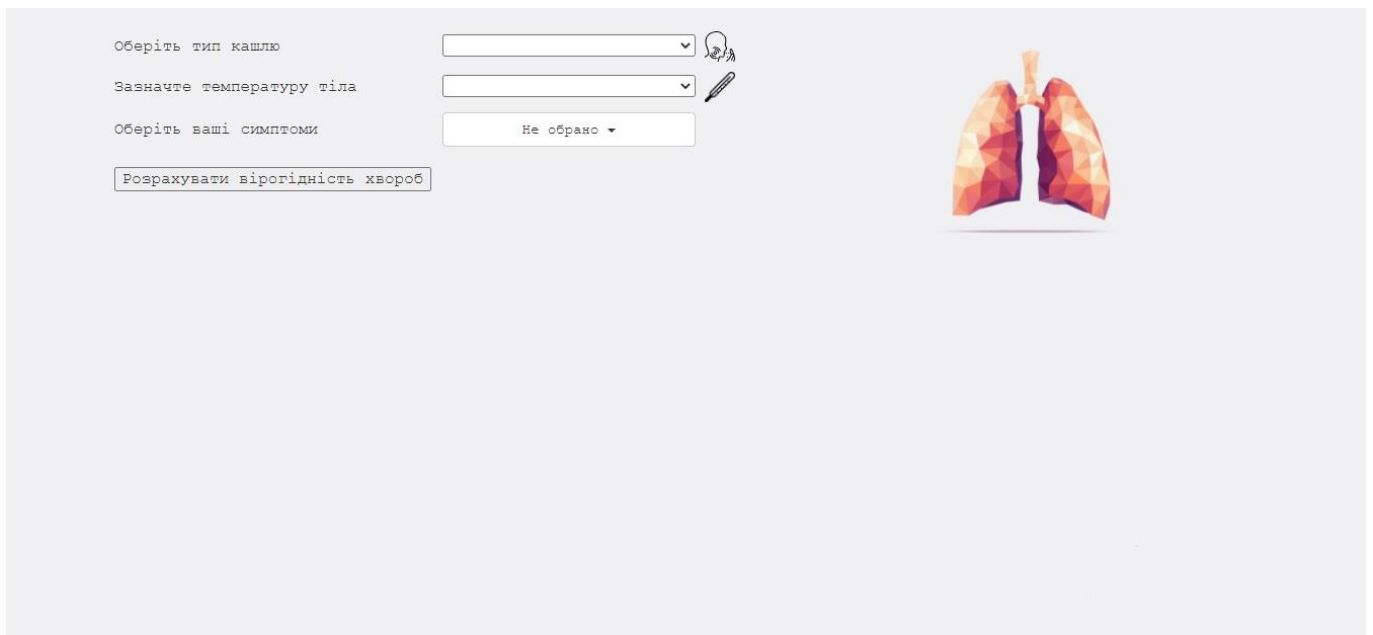
Першим етапом розробки експертної системи є розробка бази даних, що є базою знань для системи. База даних має 3 таблиці:

- Хвороби – таблиця, що містить дані хвороб;
- Симптоми – таблиця, що містить список усіх можливих симптомів;
- СимтомХвороби – таблиця, кожний запис якої представлений у вигляді Хвороба-Симptom-Вірогідність. Вірогідність – цифра від 0 до 100, що характеризує те, наскільки той чи інший симптом притаманний тій чи іншій хворобі. Ця вірогідність була взята на основі інформації з різних джерел, в яких описується симптоми хвороб та те, як часто вони зустрічаються на практиці.

База даних містить у собі 11 найрозповсюдженіших хвороб легенів, а також 3 додаткові хвороби, що не є хворобами легень, але можуть дати ускладнення і призвести до таких хвороб. Також база даних містить 43 різних симптома.

Другим етапом розробки була розробка алгоритмів обчислення вірогідностей. Для цього був побудований алгоритм, який додає значення вірогідностей симптомів між собою для тих хвороб, яким відповідають обрані користувачем симптоми. Далі отримуємо підходящі хвороби з підсумованою сумою вірогідностей. Наступним кроком усі вірогідності хвороб обчислюються як частка зі 100%. За 100% узят суму усіх вірогідностей усіх симптомів для усіх підходящих хвороб.

Третім етапом розробки була розробка інтерфейсу для експертної системи. Інтерфейс простий, зрозумілий та лаконічний. Представляє собою веб-сторінку, що містить поля з випадючими списками, у які завантажуються відповідна інформація з бази даних.



Оберіть тип кашлю

Зазначте температуру тіла

Оберіть ваші симптоми




Рис. 4.1. Головна сторінка експертної системи

Головна мета випадаючих списків – допомогти користувачу легко обрати підходящі симптоми. Перше поле – тип кашлю(оскільки кашель є основним симптомом для будь-якого захворювання легень), друге поле – температура тіла, третє поле може бути списком із кольорами мокроти(у випадку, якщо у першому полі було обране значення “Вологий кашель”, якщо було обране значення “Сухий кашель”, то поле з вибором кольору мокроти не відображається). Останнє поле – список загальних симптомів. Список організований таким чином, що можна обрати будь-яку кількість запропонованих симптомів.

4.3. Приклад роботи експертної системи

Приклад 1

В першу чергу обираємо у першому полі “Вологий кашель”, далі обираємо температуру тіла 37-38°C. Далі у полі, що з’явилося обираємо колір мокроти(оскільки був обраний вологий кашель). Наприклад оберемо мокроту жовтого кольору.

Рис. 4.2. Приклад 1, крок 1

Далі серед списку загальних симптомів оберемо такі симптоми як “Нападоподібний кашель”, “Втома”, “Слабкість”, “Задишка”.

Рис. 4.3. Приклад 1, крок 2

Після того, як ми обрали усі симптоми, що нас цікавлять, натискаємо кнопку “Розрахувати вірогідність хвороб”. Надалі ми отримаємо результат обчислень у вигляді горизонтальної стовпчикової діаграми.

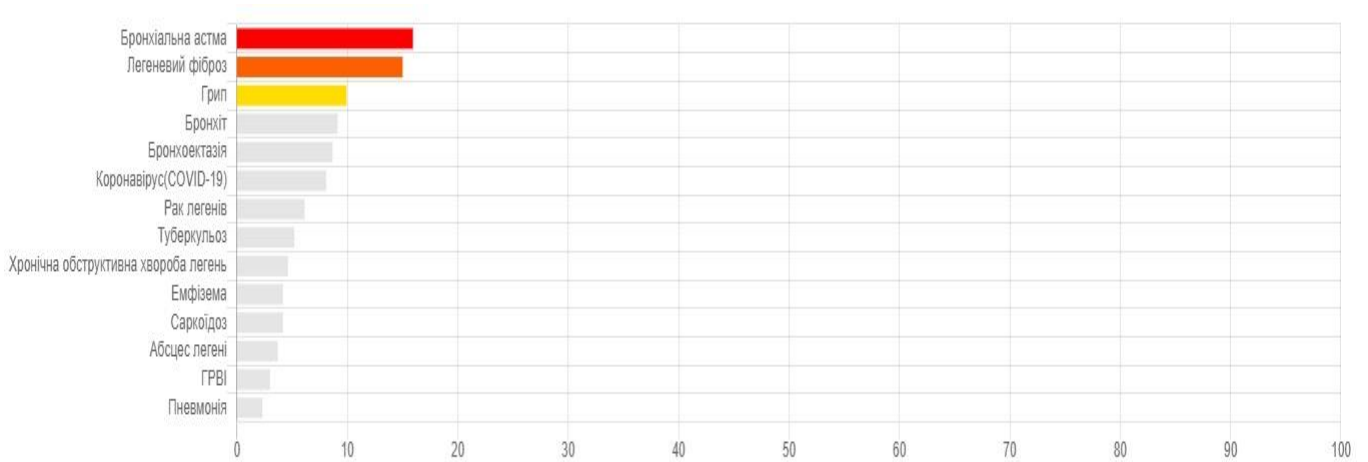


Рис. 4.4. Результат обчислень прикладу 1

На результаті бачимо, що обрані нами симптоми найбільше відповідаються таким хворобам як бронхіальна астма та легеневий фіброз.

Приклад 2

У другому прикладі спробуємо обрати симптоми, які найбільш притаманні для COVID-19. Для цього оберемо такі симптоми як сухий кашель, температура тіла 37-38°C, втома, слабкість, втрата смаку та втрата запаху.

Оберіть тип кашлю: Сухий кашель

Зазначте температуру тіла: Температура тіла 37-38°C

Оберіть ваші симптоми:

Втрата смаку

Втрата запаху

Невелика кількість мокроти

Нежить

Чхання

Віль у горлі

Віль у м'язах

Першіння у горлі

Віль в очних яблуках

Втрата смаку

Втрата нюху

Рис. 4.5. Обрані симптоми для прикладу 2

Результат буде очікуваним, оскільки найбільш вірогідною хворобою, що відповідає обраним симптомам є коронавірус.

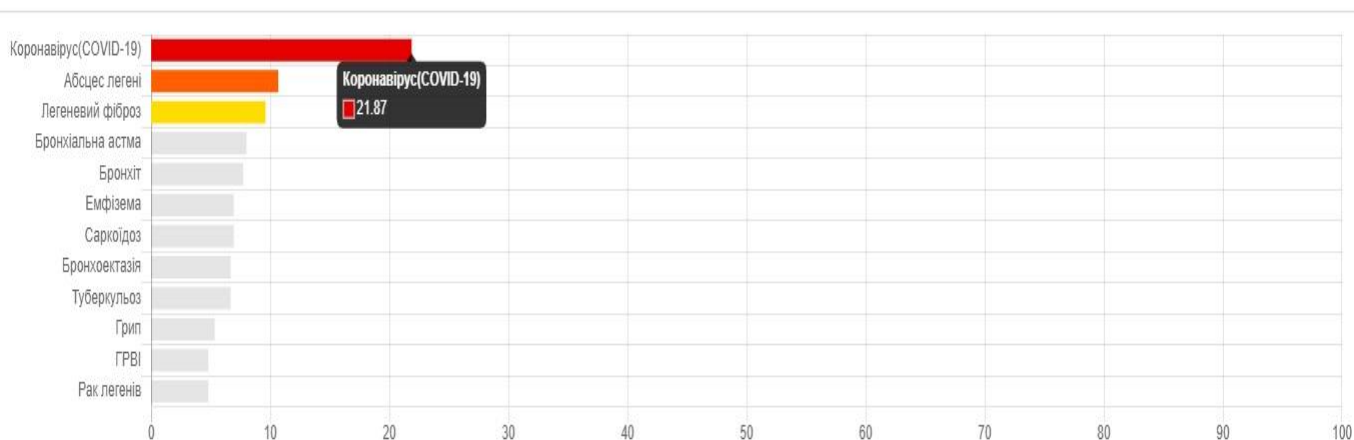


Рис. 4.6. Результат обчислень прикладу 2

ВИСНОВОК ДО РОЗДІЛУ 4

У результаті розробки була побудована експертна система, що надає можливість користувачу обрати з достатньо великого списку симптомів ті симптоми, які його цікавлять. На основі вибору користувача система обчислює вірогідність тієї чи іншої хвороби легень та будує горизонтальну стовпчикову діаграму, яка відображає вірогідності хвороб у відсотках.

ВИСНОВКИ

Проблема захворювання легень людини завжди була серйозною та масовою. Такою вона залишається і сьогодні. Щорічно мільйони людей в усьому світі помирають через хвороби, що пов'язані з легенями. Медицина, на жаль, не досягла такого розвитку, щоб лікувати усіх хворих. Але з розвитком комп'ютерних технологій розвивається і медицина, створюються нові апарати та додатки, які покращують медицину та роблять її більш ефективною.

У першому розділі були розглянуті наймасовіші хвороби легень, а також способи їх діагностики. Найбільш розповсюдженими хворобами легень є астма, рак легенів, пневмонія, ХОЗЛ(хронічна обструктивна хвороба людини), туберкульоз. Головними причинами захворювання органів дихання є паління(у тому числі і пасивне), забруднене повітря та шкідливі випаровування, які пов'язані з деякими професіями. Найбільш розповсюдженими симптомами хвороб легень є кашель(частіше вологий), задишка, утруднене дихання, хрипи при диханні, біль у грудях. При чому у деяких хвороб симптоми можуть бути майже ідентичними, а от серйозність захворювання – зовсім різною. Для точного визначення хвороби застосовують такі процедури як рентген(є найменш точним приладом), бронхоскопія(фото дихальних шляхів з середини, а також збір тканин легень), процедура грудної трубки(використовується для відводу рідини або повітря з грудної клітини), комп'ютерна томографія(серію комп'ютеризованих оглядів, зроблених під різними кутами, для створення детальних внутрішніх зображень) тощо.

У другому розділі було розглянуте поняття експертних систем, їх призначення, будова та процес розробки. Були розглянуті переваги та недоліки експертних систем. Також були розглянуті приклади медических експертних систем. Завдяки прикладам було продемонстроване те, що застосування експертних систем у медицині створило значне значення систем діагностики.

Третій розділ був присвячений розгляду та опису технологій розробки, що були засновані для розробки експертної системи. До цих технологій розробки

увійшли 2 мови програмування(C# та JavaScript), а також патерн проектування MVC. Були розглянуті сфери використання обох мов програмування, описані переваги та недоліки мов. Також були описані основи архітектурного патерну проектування додатків MVC. Розглянуті його компоненти, будова та переваги.

У заключному четвертому розділі був наведений приклад роботи розробленої експертної системи, яка і була головною метою дипломного проекту. На знімках екрану було наведено 2 різні сценарії та 2 різні результати обчислень.

Список літератури

1. Lung disease. [Electronic resource] – 2019. – Access mode: <https://www.womenshealth.gov/a-z-topics/lung-disease>. – Title from the screen.
2. Lung Procedures, Tests & Treatments. [Electronic resource] – 2020. – Access mode: <https://www.lung.org/lung-health-diseases/lung-procedures-and-tests>. – Title from the screen.
3. Expert System in AI: What is, Applications & Example. [Electronic resource] – 2020. – Access mode: <https://www.guru99.com/expert-systems-with-applications.html>. – Title from the screen.
4. Building Expert Systems. [Electronic resource] – 1986. – Access mode: <https://engineering.purdue.edu/~engelb/abe565/es.htm>. – Title from the screen.
5. MYCIN. [Electronic resource] – 2018. – Access mode: <https://www.britannica.com/technology/MYCIN>. – Title from the screen.
6. 12 AI Milestones: 4. MYCIN, An Expert System For Infectious Disease Therapy. [Electronic resource] – 2020. – Access mode: <https://www.forbes.com/sites/gilpress/2020/04/27/12-ai-milestones-4-mycin-an-expert-system-for-infectious-disease-therapy/?sh=6e5b47bd76e5>. – Title from the screen.
7. DENDRAL and Meta-DENDRAL roots of knowledge systems and expert system applications. [Electronic resource] – 1993. – Access mode: <https://stacks.stanford.edu/file/druid:pj337tr4694/pj337tr4694.pdf>. – Title from the screen.
8. Lindsay, Robert K., Bruce G. Buchanan, Edward A. Feigenbaum, and Joshua Lederberg. Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project / Lindsay, Robert K., Bruce G. Buchanan, Edward A. Feigenbaum, and Joshua Lederberg. – McGraw-Hill Book Company, 1980. – 194pp.
9. Berk, A A. LISP: the Language of Artificial Intelligence – New York: Van Nostrand Reinhold Company, 1985. – 160pp.

10. Pxdes expert systems. [Electronic resource] – 2020 – Access mode: <http://farsmarterbids.com/faq/journal/c7c02b-pxdes-expert-system> – Title from the screen.
11. Evaluation of CaDet. [Electronic resource] – 2004 – Access mode: <https://pubmed.ncbi.nlm.nih.gov/15542260/> – Title from the screen.
12. S. Jafari, S. Abdollahzade, M. Shirzad, M. Eshghanmalek. Designing an Expert System for heart disease detection using CLIPS, National Conference on Application of intelligent system in science and technology – Islamic Azad University, GuchanBranch, Iran, 2013. – 150pp.
13. Rashid J. Q. and, Syed A. H. Design an Expert System for Diagnosis of Coronary Artery Disease Using Myocardial Perfusion Imaging – National Conference on Emerging Technologies, 2004. – 105pp.
14. Hatzilygeroudis P., Vassilakos J., and Tsakalidis A. An Intelligent Medical System for Diagnosis of Bone Diseases, Proceedings of the International Conference on Medical Physics and Biomedical Engineering (MPBE'94) – Nicosia, Cyprus, 1994. – 152pp.
15. C. L. Chi, W. N. Street and D. A. Katz. A decision support system for cost-effective diagnosis – Artificial Intelligence in Medicine, 2010. – 13pp.
16. Bonow R., Mann D., Zipes D., Libby P. Braunwald's heart disease Ninth Edition – Saunders, 2012. – 2048pp.
17. Bruce G. Buchanan. Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley series in artificial intelligence) – Addison-Wesley, Reading, MA, 1984. – 702pp.
18. International Journal of Engineering Science Invention ISSN (Online) – ISSN (Print), 2004. – 42pp.
19. Six benefits of using MVC [Электронный ресурс] – 2017г. – режим доступа: <https://www.brainvire.com/six-benefits-of-using-mvc-model-for-effective-web-application-development/> – Title from the screen.

20. Everything you need to know about C [Электронный ресурс] – 2019г. – режим доступа: <https://www.pluralsight.com/blog/software-development/everything-you-need-to-know-about-c> – Title from the screen.
21. Javascript overview [Электронный ресурс] – 2018г. – режим доступа: https://www.tutorialspoint.com/javascript/javascript_overview.htm - Назва з екрану.

ДОДАТОК А

```
using Lung_disease.Models;
using Lung_disease.Services;
using Lung_disease.Data;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Lung_disease.Repository;
using Lung_disease.Domain;

namespace Lung_disease.Controllers
{
    public class SymptomController : Controller
    {
        MyContext _context = new MyContext();

        private ISymptomService symptomService;
        private ISymptomRepository symptomRepository;

        public SymptomController()
        {
            symptomService = new SymptomService();
            symptomRepository = new SymptomRepository();
        }

        [HttpGet]
        public ActionResult Symptoms()
        {
            IEnumerable<SymptomViewModel> symptomViewModel = symptomService.GetAllSymptoms();
            IEnumerable<SymptomViewModel> diseasesymptomViewModel =
            symptomService.GetAllDiseaseSymptoms();

            SelectList symptoms_cough = new SelectList(symptomViewModel.Where(_ => _.Symptom_name
            == "Вологий кашель" | _.Symptom_name == "Сухий кашель").ToList(), "Symptom_name",
            "Symptom_name");
            SelectList symptoms_temp = new SelectList(symptomViewModel.Where(_ =>
            _.Symptom_name.Contains("Температура")).ToList(), "Symptom_name", "Symptom_name");
            SelectList symptoms_sputum = new SelectList(symptomViewModel.Where(_ =>
            _.Symptom_name.Contains("Мокрота")).ToList(), "Symptom_name", "Symptom_name");

            using (var db = new MyContext())
            {
                var structureQry = db.Symptoms.Select(
                    c => new {
                        Text = c.Symptom_name,
                        Value = c.id
                    }
                ).ToList();

                for (int i = 0; i < structureQry.Count; i++)
                {
                    if (structureQry[i].Text.Contains("Температура тіла ") ||
                    structureQry[i].Text.Contains("Мокрота"))
                    {
                        structureQry.Remove(structureQry[i]);
                    }
                    if (structureQry[i].Text == "Сухий кашель")
                    {
                        structureQry.Remove(structureQry[i]);
                    }
                }
            }
        }
    }
}
```

ПРОДОВЖЕННЯ ДОДАТКУ А

```
if (structureQry[i].Text == "Вологий кашель")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Температура тіла 36-37°C")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Температура тіла 37-38°C")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Температура тіла 38-39°C")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Температура тіла 39-40°C")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Мокрота жовтого кольору")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Мокрота зеленого кольору")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Мокрота блідо-сірого кольору")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Мокрота жовто-зеленого кольору")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Мокрота коричневого кольору")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Мокрота блідо-сірого кольору")
{
    structureQry.Remove(structureQry[i]);
}
if (structureQry[i].Text == "Мокрота коричневого кольору")
{
    structureQry.Remove(structureQry[i]);
}
}

ViewBag.structure = new MultiSelectList(structureQry, "Value", "Text");
}

ViewBag.Symptoms_cough = symptoms_cough;
ViewBag.Symptoms_temp = symptoms_temp;
ViewBag.Symptoms_sputum = symptoms_sputum;

return View();
}

[HttpPost]
public ActionResult Diagram(string name, string temp, string sputum)
{
```

```

IEnumerable<SymptomViewModel> symptomViewModel = symptomService.GetAllSymptoms();
IEnumerable<SymptomViewModel> diseasesymptomViewModel1 =
symptomService.GetAllDiseaseSymptoms();
List<string> SelectedSymptoms = new List<string>();

var value = Request.Form["structure"];
var db = new MyContext();
List<Symptom> result = (from r in db.Symptoms
                       where value.Contains(r.id.ToString())
                       select r).ToList();

SelectedSymptoms.Add(name);
SelectedSymptoms.Add(temp);
SelectedSymptoms.Add(sputum);
for (int i = 0; i < result.Count; i++)
{
    SelectedSymptoms.Add(result[i].Symptom_name);
}

List<int> probabilities = new List<int>();
List<string> names = new List<string>();
for (int y = 0; y < SelectedSymptoms.Count; y++)
{
    foreach (var model in diseasesymptomViewModel1)
    {
        if (model.Symptom.Symptom_name == SelectedSymptoms[y])
        {
            names.Add(model.Disease_name);
            probabilities.Add(model.Probability);
        }
    }
}

int count = names.Count();

for (int i = 0; i < (names.Count); i++)
{
    for (int y = 0; y < (names.Count); y++)
    {
        if (names[i] == names[y] & i != y)
        {
            probabilities[i] += probabilities[y];
        }
    }
}

names = names.Distinct().ToList();
int dist = names.Count();
int range = count - dist;
probabilities.RemoveRange(dist, range);

var sum = 0;
foreach (var i in probabilities) sum += i;

List<double> probabilities_double = probabilities.Select<int, double>(i =>
i).ToList();
for (int i = 0; i < probabilities_double.Count; i++)
{
    probabilities_double[i] = Math.Round(((100 * probabilities_double[i]) / sum),2);
}

double tem;

```

```
string tem1;
for (int i = 0; i < probabilities_double.Count - 1; i++)
{
    for (int j = 0; j < probabilities_double.Count - i - 1; j++)
    {
        if (probabilities_double[j + 1] > probabilities_double[j])
        {
            tem = probabilities_double[j + 1];
            probabilities_double[j + 1] = probabilities_double[j];
            probabilities_double[j] = tem;

            tem1 = names[j + 1];
            names[j + 1] = names[j];
            names[j] = tem1;
        }
    }
}

ViewBag.Names = Newtonsoft.Json.JsonConvert.SerializeObject(names);

ViewBag.Probabilities =
Newtonsoft.Json.JsonConvert.SerializeObject(probabilities_double);

return View();
}
}
```