

Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота

за освітнім ступенем «магістр»

**на тему: Комп'ютерне імітаційне моделювання фізіологічних процесів
у системах та органах людини**

Виконала: магістрантка 2 курсу
групи М-КН-21
спеціальності 122 «Комп'ютерні науки»
Чепчур Мар'яна Миколаївна
Керівник:
к.т.н., доцент Сяський В.А.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СКЛАДНИХ СИСТЕМ	6
РОЗДІЛ 2. ВІДОМОСТІ З АНАТОМІЇ ТА ФІЗІОЛОГІЇ ДИХАЛЬНОЇ СИСТЕМИ ЛЮДИНИ	11
2.1. Будова легені	11
2.2. Процес газообміну у легенях	16
2.3. Вентиляція альвеол	19
2.4. Хімічний склад повітря	20
2.5. Вплив фізичних навантажень на процес дихання	23
РОЗДІЛ 3. МОДЕЛЬ ЛЕГЕНІ	25
3.1. Структурно-функціональна модель легені	25
3.2. Імітаційна модель легені на основі штучних нейронних мереж	26
РОЗДІЛ 4. ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІМІТАЦІЙНОЇ МОДЕЛІ ЛЕГЕНІ	29
4.1. Етапи програмної реалізації	29
4.2. Опис програмної системи	32
4.3. Симуляція роботи легені	34
4.4. Вимоги до апаратно-програмних ресурсів ЕОМ, що забезпечують ефективне функціонування програмної системи	40
ВИСНОВКИ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43
ДОДАТКИ	45

ВСТУП

Актуальність роботи.

Людський організм – складна система, яка виконує велику кількість функцій. У свою чергу кожен орган в організмі також є системою. У таких системах можуть виникати та розвиватися різні патології, які потребують складного лікування. Основним шляхом вирішення цієї проблеми без хірургічного втручання є моделювання фізіологічних процесів. Це дозволяє дослідити людський організм безкровно.

Сучасні технології та математичні методи дають можливість із високою точністю моделювати цілі системи та фізіологічні процеси у них. При моделюванні можна враховувати вплив різних факторів та патологій.

Більш перспективним підходом в дослідженнях механізмів виникнення, розвитку та лікування патологій різних органів і систем є використання динамічних моделей, що базуються на відтворенні реальних біологічних процесів – функціональних динамічних моделей. В їх основі часто використовується особлива абстракція – «чорний ящик» – деяка функція, що реалізує відображення у багатовимірних просторах. Розробка таких моделей є актуальною та важливою для дослідження динамічного зв'язку між окремими структурними елементами складних живих систем та елементарними процесами в них.

Математичне моделювання широко використовується для вирішення багатьох прикладних задач дослідження фізіологічного стану людини при екстремальних факторах впливу. Не є виключенням і дослідження процесів збагачення крові киснем та газообміну в легенях.

Імітаційне (програмне) моделювання передбачає побудову структурно-функціональної моделі об'єкта дослідження у вигляді певної послідовності дій (алгоритму), що реалізована як комп'ютерна програма та відтворює функціонування досліджуваної системи шляхом послідовного виконання великої кількості елементарних операцій [12]. Зазвичай такий вид моделювання

використовують, якщо певні дії відбуваються багаторазово і мають відомі статистичні характеристики. Імітаційний підхід в моделюванні поведінки складних систем дозволяє з достатньо високим ступенем вірогідності прогнозувати динаміку розвитку захворювання органу та процесу його лікування. Тому розробка ефективної моделі, яка б дозволила досліджувати складні динамічні внутрішні процеси в легенях, є безсумнівно **актуальною** задачею.

Метою роботи є: побудова і програмна реалізація імітаційної моделі легені на основі багатопрошаркових штучних нейронних мереж.

Для досягнення сформульованої мети були поставлені та вирішені наступні **завдання**:

- аналіз джерельної бази з анатомії та фізіології дихальної системи людини;
- на основі визначальних принципів імітаційного моделювання поведінки складних систем побудувати базову модель легені;
- засобами штучних нейронних мереж реалізувати імітаційну модель процесів газообміну в легені.

Об'єкт дослідження: імітаційне моделювання динамічних процесів у складних фізіологічних системах людського організму.

Предмет дослідження: імітаційне моделювання засобами штучних нейронних мереж процесів газообміну в легенях людини.

Для реалізації поставлених завдань було проведено аналіз можливостей, ефективності та областей застосування об'єктно-орієнтованих мов програмування та технологій, що на них базуються [2, 3]. Для цього використані наступні мови та засоби програмування: C#, WPF, XAML, JSON [1, 13].

Структура роботи. Дипломна робота складається зі вступу, чотирьох розділів, висновків та переліку використаних джерел.

У першому розділі викладені особливості імітаційного моделювання складних систем та сучасна класифікація його видів та методів.

Другий розділ містить стислий виклад відомостей з анатомії та фізіології

людини про будову та функціонування легень.

Третій розділ висвітлює особливості структурно-функціональної та імітаційної моделі легені, процесів збагачення крові киснем та газообміну.

Четвертий розділ містить інформацію про програмний продукт «Lung Simulator», особливості програмної реалізації побудованої моделі та результати модельних експериментів.

Загальний обсяг роботи становить 65 сторінок. Вона містить 13 рисунків та 1 таблицю. Список використаних джерел включає 17 найменувань. Обсяг додатків – 22 сторінки.

РОЗДІЛ 1. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СКЛАДНИХ СИСТЕМ

Методи комп'ютерного моделювання широко використовують у всіх сферах діяльності людини – від конструювання моделей технічних, технологічних й організаційних систем до вирішення проблем розвитку людства і навколишнього світу.

При вивченні складних процесів, що відбуваються у людському організмі, природі чи під час наукових експериментів, не завжди враховуються усі наявні фактори: одні є досить вагомими, а іншими можна проігнорувати. При цьому з'являються моделі таких явищ, процесів, які спроможні їх замінити повністю, а вивчаючи їх, можна отримати нову інформацію про них.

Моделювання – це спосіб дослідження будь-яких явищ, процесів або об'єктів шляхом побудови й аналізу їх моделей. У широкому розумінні моделювання є однією з основних категорій теорії пізнання і мало не єдиним науково обґрунтованим методом наукових досліджень систем і процесів будь-якої природи в багатьох сферах людської діяльності.

Модель – це об'єкт довільної природи, котрий створено людиною штучно. Він відтворює або замінює об'єкт, котрий розглядається, так, що при детальному вивченні такої моделі можемо отримати нову інформацію про цей об'єкт. Модель бідніша за об'єкт, котрий вона відтворює або замінює. Вона відображає лише деякі його риси.

У загальному випадку під час побудови моделі потрібно враховувати такі вимоги:

- незалежність результатів розв'язання задач від конкретної фізичної інтерпретації елементів моделі;
- змістовність, тобто здатність моделі відображати важливі риси і властивості реального процесу, який вивчається і моделюється;
- дедуктивність, тобто можливість конструктивного використання моделі для отримання результату (управління, прогнозування);

- індуктивність – вивчення причин і наслідків, від окремого до загального, з метою накопичення необхідних знань.

Ступінь деталізації моделі потрібно вибирати з урахуванням цілей моделювання, можливості отримання необхідних вхідних даних для моделі і враховуючи наявні ресурси для її створення. З іншого боку, чим детальніше розроблена модель, тим вона стійкіша до вхідних впливів, які не були передбачені під час проектування, і на більшу кількість питань може дати правильні відповіді.

Методи математичного моделювання широко використовується при дослідженнях фізіологічного стану організму людини в цілому та окремих його органів і систем. Процес моделювання органічних систем фактично відбувається на тлі суперечностей між вимогами спрощення складного біологічного явища й збереження його специфіки, його найбільш сутнісних рис.

Сьогодні моделювання складних систем найчастіше реалізують за допомогою сучасних комп'ютерних технологій. Такий підхід передбачає необхідність попередньої формалізації концептуальної моделі об'єкта дослідження та її подання у вигляді, придатному для реалізації тих чи інших алгоритмів чисельного аналізу або комп'ютерної імітації. Обидва підходи передбачають необхідність застосування сучасних методів, що використовуються при конструюванні алгоритмів моделювання. Навіть при застосуванні спеціалізованих пакетів програмного забезпечення досліднику необхідно володіти базовими принципами відповідних методів, оскільки користування такими пакетами зазвичай передбачає необхідність вибору оптимального алгоритму й певних параметрів його реалізації, іноді з декількох десятків можливих варіантів.

Відповідно до класифікації моделей виділяють такі основні різновиди моделювання складних систем [11]:

- *Концептуальне моделювання* за допомогою природних або штучних умов визначає принципові аспекти побудови або функціонування об'єкта дослідження.

- *Інтуїтивне моделювання* передбачає прогнозування поведінки досліджуваної системи на основі уявних експериментів і врахування практичного досвіду дослідника.

- *Фізичне (натурне) моделювання* здійснюють шляхом дослідження в якості моделі реального фізичного об'єкта, між яким і об'єктом-оригіналом встановлені відношення подібності, що ґрунтуються на подібності фізичних процесів, що відбуваються в них.

- *Структурно-функціональне моделювання* передбачає побудову і використання як моделей графіків, таблиць, блок-схем тощо, для яких встановлено певні правила їх об'єднання й перетворення.

- *Математичне (логіко-математичне) моделювання* передбачає побудову й дослідження моделей математичними та логічними засобами. Прикладами математичних моделей є: рівняння коливань як модель циклічного процесу; логістична крива як модель кривої зростання; рівняння рівномірного та рівноприскореного руху тощо.

- *Імітаційне (програмне)* – це метод дослідження, при якому досліджувана система замінюється моделлю, яка з достатньою точністю описує реальну систему, і з нею проводяться експерименти з метою отримання інформації про цю систему. Імітаційне статистичне моделювання являє собою чисельний метод проведення на ЕОМ обчислювальних експериментів з математичними моделями, що імітують поведінку реальних об'єктів, процесів і систем у часі протягом заданого періоду.

З усіх видів моделювання основна увага приділяється імітаційному моделюванню. Огляд науково-дослідницьких робіт показує, що імітаційне моделювання є чи не найпопулярнішим, за використанням на практиці його випереджають лише методи математичного програмування. Головна цінність імітаційного моделювання полягає в тому, що в основу його покладена методологія системного аналізу. Вона дозволяє досліджувати проектувану або аналізовану систему методами операційного аналізу, який включає такі взаємопов'язані етапи: змістовна постановка задачі, розробка концептуальної

моделі, розробка та програмна реалізація імітаційної моделі, перевірка адекватності моделі та оцінка точності результатів моделювання, планування і проведення експериментів та прийняття рішень.

Імітаційне моделювання - це окремий випадок математичного моделювання. Існує клас об'єктів, для яких з різних причин не розроблені аналітичні моделі, або не розроблені методи розрахунку отриманої моделі. В цьому випадку математична модель замінюється імітатором або імітаційною моделлю.

Відмінність імітаційної та математичної моделей полягає в тому, що в імітаційній моделі замість математичного опису відношень між вхідними та вихідними даними система розбивається на ряд досить малих підсистем(елементів або модулів). Поведінка такої вихідної системи відтворюється як поведінка сукупності підсистем(елементів або модулів), котрі взаємопов'язані в єдине ціле.

Ефективність імітаційного моделювання фізіологічних систем і процесів є предметом багатьох досліджень і дискусій, які дають право сформулювати наступні положення [13]:

- імітаційне моделювання дає змогу досліджувати внутрішні взаємодії у складних системах або підсистемах у межах складної системи, а також експериментувати з ними;
- на основі знань, отриманих під час проектування імітаційної моделі, можна визначити способи вдосконалення системи, що моделюється;
- змінюючи вхідні дані під час моделювання і спостерігаючи за вихідними даними, можна виявити, які змінні найбільш важливі та як вони взаємодіють.

Необхідність у використанні імітаційного моделювання може виникнути з наступних причин:

1) методи аналітичного моделювання не можуть бути використані через складність моделі;

2) застосування методів аналітичного моделювання дасть результати, які не задовольняють меті дослідження системи;

3) застосування методів імітаційного моделювання дозволяє зробити рішення множини задач, що розглядається універсальним.

Імітаційне моделювання системи передбачає, що процес функціонування системи відтворюється за допомогою алгоритму, який реалізується за допомогою комп'ютера.

РОЗДІЛ 2. ВІДОМОСТІ З АНАТОМІЇ ТА ФІЗІОЛОГІЇ ДИХАЛЬНОЇ СИСТЕМИ ЛЮДИНИ

Дихальна система (або органи дихання) — система органів, яка бере участь у транспортуванні повітря через дихальні шляхи, газообміні між організмом і навколишнім середовищем.

Дихальна система з'єднує органи газообміну між кров'ю та зовнішнім середовищем. «Без середовища, яке підтримує його існування, організм не може вижити», і між організмом і середовищем відбувається постійний обмін речовин. Організм поглинає кисень і поживні речовини з навколишнього середовища. Завдяки їм будуються нові клітини, функціонують і працюють органи, наприклад м'язи. При цьому в навколишнє середовище виділяються шкідливі продукти розпаду, надлишки речовин і тепло.

2.1. Будова легень

Легені — це пара великих конусоподібних губчастих органів, які видаляють відходи вуглекислого газу з організму й замінюють його свіжим киснем. Під час розширення грудної клітини повітря втягується в легені, а потім залишає тіло, оскільки грудна порожнина стискається і виштовхує повітря з тіла.

Легені займають більшу частину грудної порожнини. Верхня частина западини обмежена ребрами і міжреберними м'язами. Дно порожнини обмежене перегородкою - плоскою смугою тканини, яка утворює стінку між грудною кліткою та животом.

Легені є частиною дихальної системи.

Вентиляція легень є першим етапом процесу дихання та становить собою обмін газів між навколишнім середовищем та легенями, який здійснюється ритмічними дихальними рухами грудної клітки. При вдиху об'єм грудної клітки та легенів, які знаходяться в ній, збільшується, а тиск у них зменшується та повітря входить в альвеоли по повітроносних шляхах.

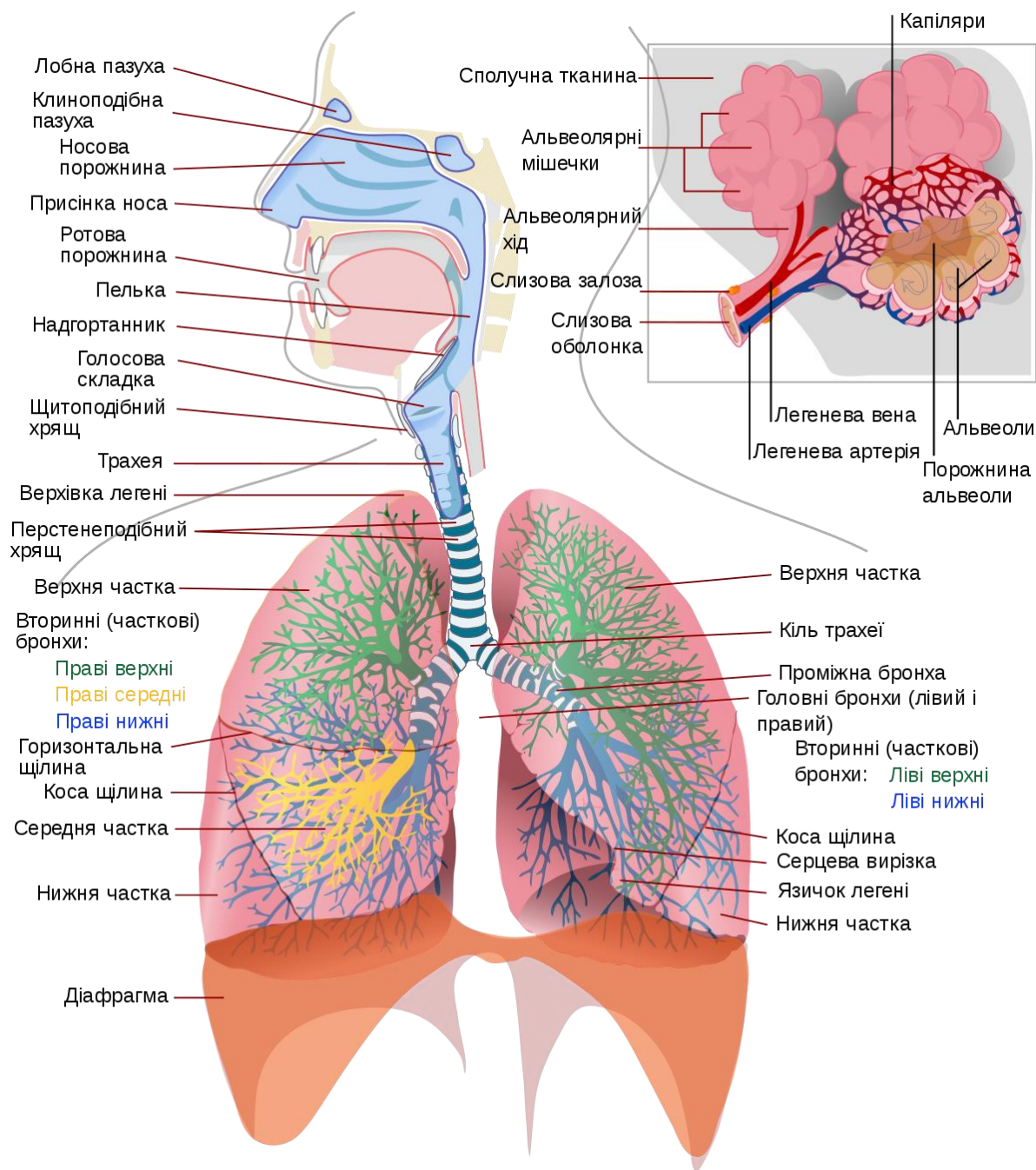


Рис. 2.1 Будова дихальної системи

Легені - складний орган, який за будовою є трубчасто-паренхіматозним. Їх трубчасту частину утворює внутрішньолегеневе розгалуження дихальних шляхів, тобто *бронхіальне дерево*. Головний бронх входить у ворота легень і дає початок бронхіальному дереву. *Головний бронх* (бронх I порядку) в воротах поділяється на *часткові бронхи* (бронхи II порядку) у відповідності до кількості часток в кожній легені. Частковий бронх поділяється на *сегментарні бронхи* (бронхи III порядку) у відповідності до кількості сегментів в кожній частці. Кожний сегментарний бронх *дихотомічно* поділяється до *бронхів VII-XII порядків*. Бронх VII-XII порядку діаметром 1 мм має назву *часточкового бронха*. В легені визначається до 20000 часточок. Часточковий бронх в середині легеневої часточки поділяється на 18-20 *кінцевих бронхіол*, якими закінчується бронхіальне дерево (Рис. 2.2).

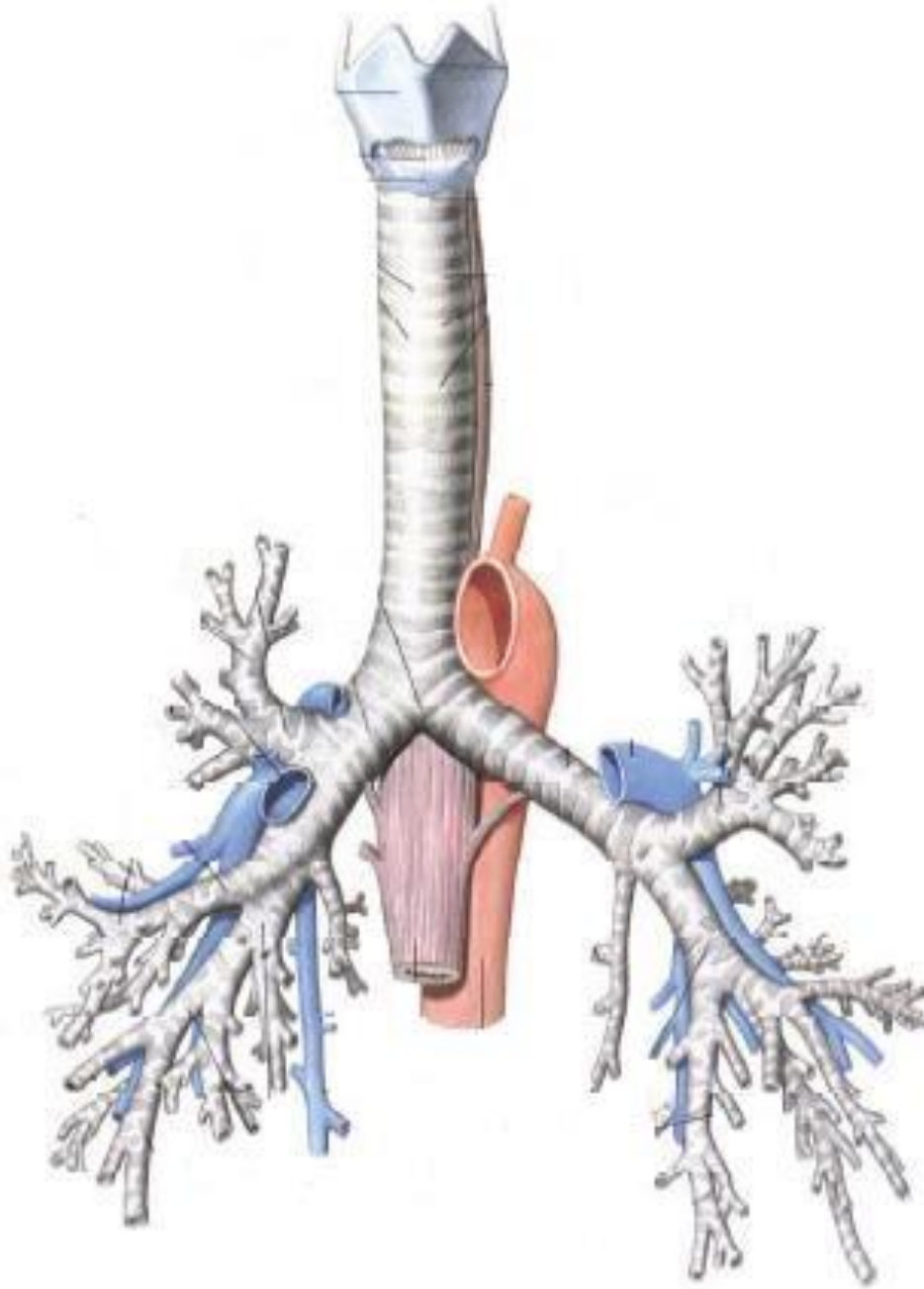


Рис. 2.2 Бронхіальне дерево.

Кожен порядок бронх може містити у собі декілька генерацій. Всього нараховується близько 20-25 генерацій бронхів

Дихальні шляхи ділять на кондуктивну (провідну), транзиторну (перехідну) і дихальну зони. До першої зони належать від 1-ї до 16-ї генерації бронхів, до другої – наступні. Із загального об'єму легень провідна зона займає 3 % (близько 150 мл), транзиторна – близько 30 % (1500 мл).

Починаючи з 17-го поділу, на стінках бронхів з'являються поодинокі альвеоли. Далі, аж до 23-ї генерації, кількість альвеол збільшується. В зв'язку з цим 17–23-тя генерації називаються дихальними бронхіолами; 23-тя генерація бронхіол переходить альвеолярні мішечки.

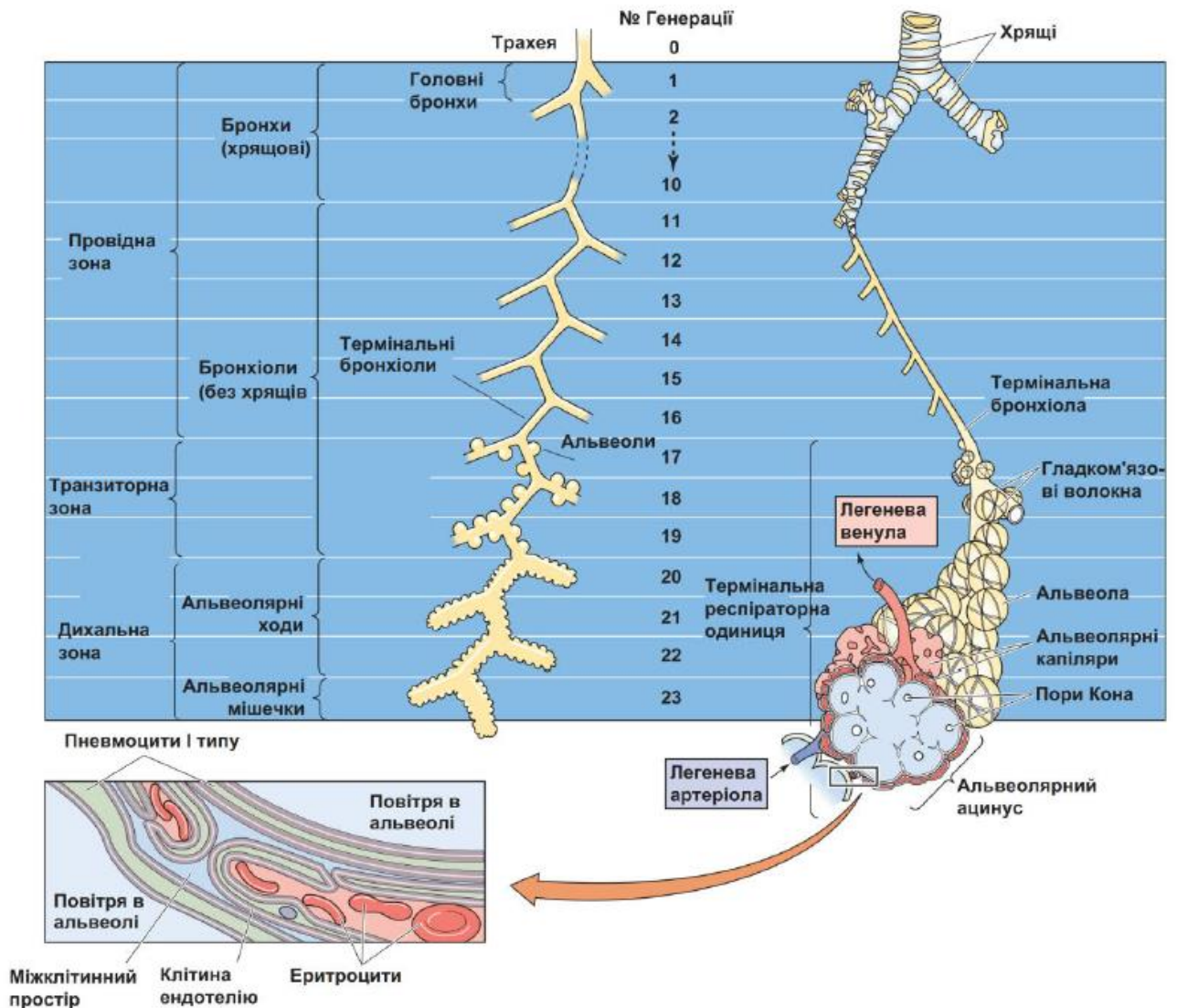


Рис. 2.3 Будова бронхіального дерева легень

Паренхіму легені утворює *альвеолярне дерево*. Кожна кінцева бронхіола дихотомічно поділяється на дихальні або респіраторні бронхіоли, в стінках яких вже визначаються окремі альвеоли. Від кожної дихальної бронхіоли відходять альвеолярні ходи, які закінчуються альвеолярними мішечками, що утворені альвеолами. Дихальні бронхіоли, альвеолярні ходи і альвеолярні

мішечки утворюють альвеолярне дерево. В альвеолярному дереві 57% складають альвеоли, а 27% складають альвеолярні ходи. Структурно-функціональною одиницею паренхіми легень, на рівні якої відбувається газообмін, є *ацинус*.

Ацинус - це розгалуження однієї дихальної бронхіоли, яке оточено кровоносними капілярами. В кожній легені визначається 150000 ацинусів, кількість альвеол дорівнює 300-500 млн., в кожній часточці виділяють 16-20 ацинусів.

2.2. Процес газообміну у легенях

Газообмін – це процес обміну газом між організмом і зовнішнім середовищем, а також між кров'ю і тканинами організму.

Сутність процесу газообміну полягає в переході кисню з альвеолярного повітря у венозну кров, яка циркулює по легеневих капілярах (поглинання кисню), і в переході вуглекислого газу з венозної крові в альвеолярне повітря (виділення вуглекислого газу).

Газообмін відбувається через аеро - гематичний бар'єр, до складу якого входить стінка альвеоли (сурфактант, епітелій і базальна мембрана) і стінка кровоносного капіляра (ендотелій і базальна мембрана).

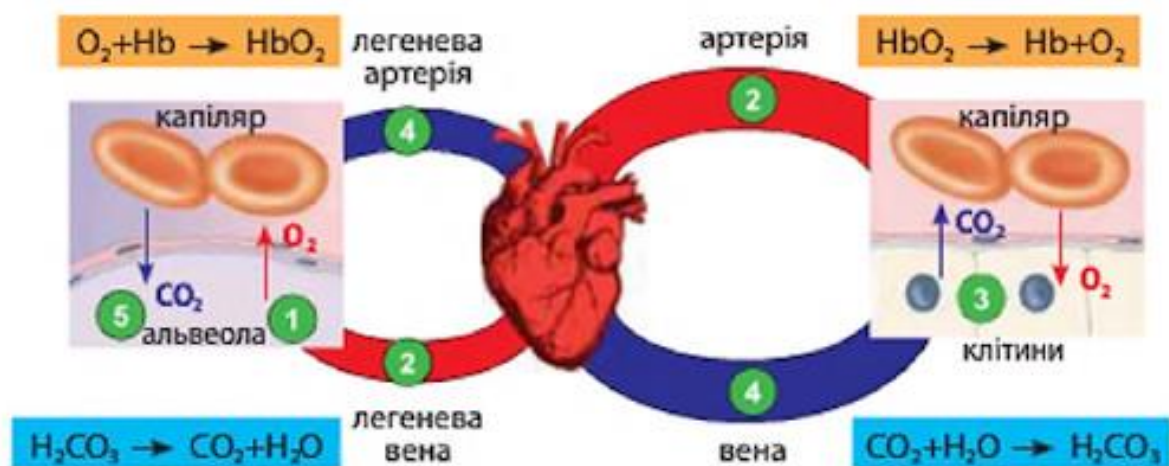


Рис. 2.4 Процес газообміну

Газообмін між атмосферним і альвеолярним повітрям здійснюється, як за рахунок конвекційного переносу, так і за рахунок дифузії. Оскільки із розгалуженням бронхів зростає площа сумарного поперечного їх перерізу, то відповідно зменшується лінійна швидкість потоку повітря в них. Розрахунки показують, що під час спокійного дихання перші порції повітря досягають 17-ї генерації бронхів за 0,87 с, 20-ї - за 2,04 с, а 23-ї – лише за 6,7 с. Але вдих триває не більше, ніж 1,5-2 с. Це означає, що якби повітря доходило до альвеол тільки шляхом конвекції, то вони б не вентильовалися. Тому вважається, що, починаючи із 17-ї генерації бронхіол, до процесу конвекції приєднується і дифузійний обмін O_2 і CO_2 . При форсованому диханні транспорт шляхом конвекції продовжується до 19-ї генерації бронхіол. Таким чином, транзиторна частина бронхіального дерева (17-19 генерації), є своєрідною буферною зоною між атмосферним та альвеолярним повітрям. Верхні її ділянки обмінюються із вдихуваним повітрям, а прилеглі до альвеол - із альвеолярним повітрям. При цьому склад альвеолярного повітря сталий і не залежить від фаз дихання.

У людини процес дихання та газообміну відбувається у декілька етапів (Рис. 2.5)[16]:

- 1) зовнішнє дихання – обмін повітря між атмосферою та легневими пухирцями. До цього етапу належать вдих та видих;
- 2) перехід кисню з легенів у кров та вуглекислого газу у зворотному напрямі;
- 3) транспортування газів кров'ю;
- 4) газообмін між кров'ю та клітинами;
- 5) клітинне дихання.

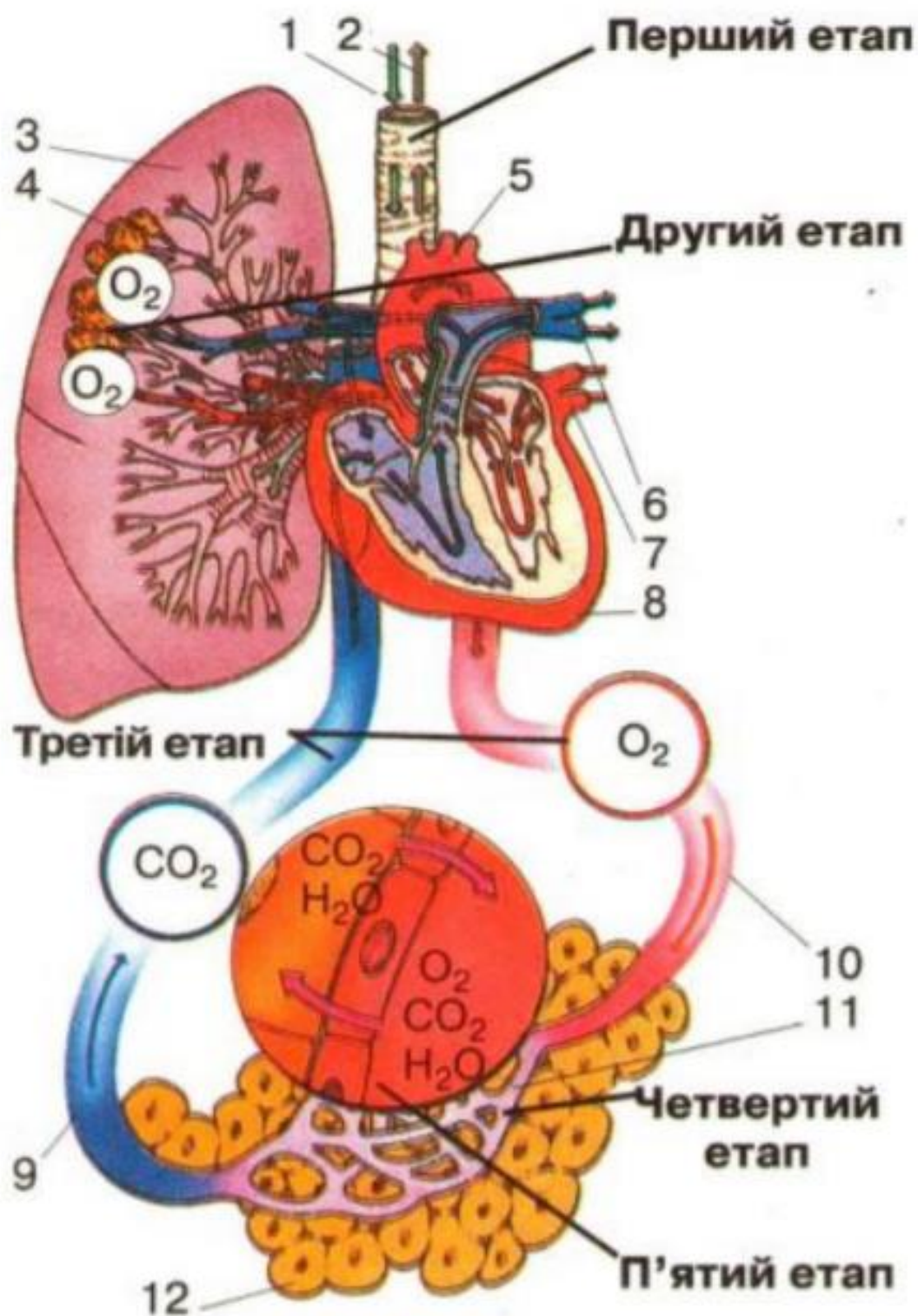


Рис. 2.5 Етапи дихання

На Рис. 2.5 зображено: 1 – кисень, 2 – вуглекислий газ, 3 – легеня, 4 – альвеоли, 5 – аорта, 6 – легеневі артерії, 7 – легеневі вени, 8 – серце, 9 – вена, 10 – артерія, 11 – капіляри, 12 – клітини тіла.

2.3. Вентиляція альвеол

Газообмін між атмосферою і внутрішнім середовищем організму відбувається за рахунок постійного оновлення складу повітря в альвеолах, які оточені легневими капілярами. В організмі людини кількість альвеол складає близько 300 мільйонів.

Альвеолярної вентиляцією називають об'єм нової порції повітря, що надходить в альвеоли за 1 хв. Основним завданням такої вентиляції є постійний обмін O_2 та CO_2 в газообмінних зонах легень. До цих зон належать альвеоли, альвеолярні мішечки, протоки та бронхіоли. Альвеолярна вентиляція є частиною легеневої.

Повітря, що вдихається, потрапляє до альвеол. Там кисень з повітря розчиняється в рідині, що вкриває їхню поверхню. Кисень переміщується через стінки альвеол і капілярів у кров. Такий процес перетворює венозну кров на артеріальну. Це відбувається, тому що концентрація кисню в альвеолярному повітрі вища, ніж у венозній крові.

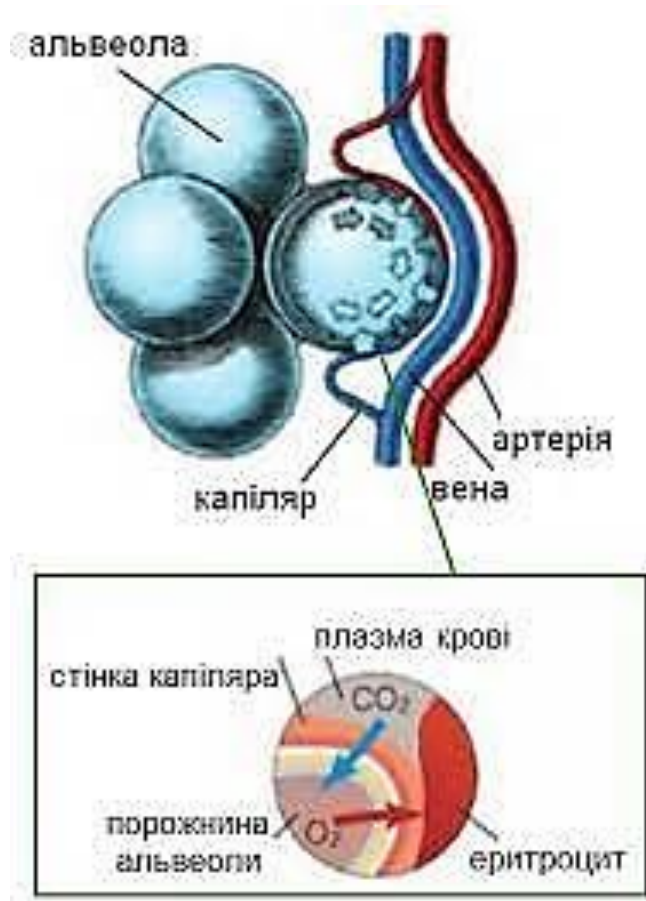


Рис. 2.6 Газообмін у легенях з перетворенням венозної крові на артеріальну

Концентрація вуглекислого газу в альвеолярному повітрі менша, тому що CO_2 з капілярів венозної крові переходить в альвеоли та під час видиху виходить назовні.

2.4. Хімічний склад повітря

Атмосфера являє собою суміш різних газів. Містить постійні компоненти атмосфери (кисень, азот, вуглекислий газ), інертні гази (аргон, гелій, неон, криптон, водень, ксенон, радон), невеликі кількості озону, закису азоту, метану, йоду, водяної пари тощо. Різні кількості різноманітних домішок і забруднень природного походження, що утворюються в результаті виробничої діяльності людини.

Альвеолярне повітря - це повітря, що міститься в альвеолах. *Атмосферне повітря* при вдиху змішується з повітрям, що залишився в альвеолах після видиху. Вентиляція легенів (вдих і видих) підтримує відносно стабільний склад газу в альвеолах.

Вдихуване (атмосферне) повітря – це повітря, яким ми вдихаємо.

Видихуване повітря – це повітря, яке ми видихаємо (в якому збільшується вміст вуглекислого газу і водяної пари, а зменшується вміст кисню).

Хімічний склад приземного повітря може незначно коливатися.

Таблиця 2.1

Хімічний склад атмосферного, видихуваного та альвеолярного повітря

Повітря	O₂, %	CO₂, %	N₂, %
Атмосферне	20,93	0,03	79,04
Видихуване	16,0	4,5	79,5
Альвеолярне	14,0	5,5	80,5

Із наведеної таблиці видно, що між атмосферним і альвеолярним повітрям існує градієнт концентрацій дихальних газів, який і є рушійною силою дифузії. Аналіз видихуваного повітря свідчить, що його різні порції істотно відрізняються щодо процентного співвідношення O₂ і CO₂. За складом перші порції повітря ближчі до атмосферного, а останні - ближчі за складом до альвеолярного.

Кисень (O₂) є найважливішою частиною повітря для людини. Він необхідний для здійснення окисних процесів в організмі. Вміст кисню в атмосфері становить 20,95%. Зменшення вмісту атмосферного повітря до 13-15% порушує фізіологічну функцію, до 7-8% з летальним наслідком. У процесі дихання людина вдихає близько 21% кисню (O₂), адже він необхідний для здійснення окиснювальних процесів в організмі.

Азот (N_2) – є основним компонентом атмосфери. Повітря, яке вдихає і видихає людина, містить приблизно однакову кількість азоту - 78,97-79,2%. Біологічна роль азоту полягає насамперед як розчинник для кисню. Це тому, що життя не може існувати на чистому кисні. Смерть настає при підвищенні вмісту азоту до 93%.

Вуглекислий газ (вуглекислий газ), CO_2 - є фізіологічним регулятором дихання. У чистому повітрі його вміст становить 0,03%, у видихуваному – 3%.

Зменшення концентрації CO_2 у вдихуваному повітрі не представляє небезпеки, оскільки необхідний рівень CO_2 в крові підтримується регуляторними механізмами, вивільняючись під час метаболічних процесів.

Підвищення вмісту вуглекислого газу у вдихуваному повітрі до 0,2 % викликає у людини нудоту, 3-4 % — збудження, головний біль, шум у вухах, серцебиття, уповільнення серцебиття, 8 % — важку інтоксикацію, втрату свідомості. приходиться втрата і смерть

Вуглекислий газ (CO_2) людина теж вдихає. До організму потрапляє 0,03%, а виходить 4%. Це пояснюється тим, що вуглекислий газ виділяють органи і тканини людського організму, внаслідок біологічних реакцій.

Вміст CO_2 в повітрі є важливим гігієнічним показником, який визначає чистоту повітря промислових, житлових і громадських будівель.

Збільшення вмісту CO_2 вище встановлених норм свідчить про загальне погіршення гігієни повітря. Це пояснюється тим, що інші токсичні речовини можуть накопичуватися разом з вуглекислим газом, погіршуючи режим іонізації та збільшуючи пилове та мікробне забруднення.

Вміст інших газів не змінюється, адже вони ніяк не взаємодіють з людським організмом, тобто не засвоюються і не виділяються. Лише при деяких захворюваннях людина може видихувати різні леткі речовини.

2.5. Вплив фізичних навантажень на процес дихання

М'язова діяльність служить найпотужнішим природним стимулом для дихання, оскільки енергія до виконання фізичної роботи виробляється в окислювальному процесі. Легенева вентиляція при м'язовій роботі знаходиться у прямій залежності від сили роботи (споживання кисню). Тому при напруженій м'язовій роботі вентиляція легень у тренованої людини збільшується більш ніж до 100 літрів за хвилину. До певного субмаксимального рівня ця залежність має лінійний характер, а при більш тяжкій роботі вентиляція легень збільшується швидше, ніж збільшується споживання кисню.

Незабаром після початку фізичного навантаження спостерігається збільшення вентиляції легень за рахунок збільшення частоти та глибини дихання. У той же час початкове швидке зростання дихання обумовлене нервовими факторами - складними рефlekсами на ситуацію (підготовка до дії), сильною імпульсацією до працюючих м'язів (ірадіація), аферентною імпульсацією від пропріорецепторів у м'язах, що працюють.

Подальша регуляція дихання тісно пов'язана з хеморецепторами, що реєструють зміни рівня кисню, вуглекислого газу та рН крові. Зокрема, молочна кислота, накопичена в тканинах під час роботи, надходить у кров і витісняє вуглекислоту із зв'язку з К і Na, збуджує рефлекси та одночасно збільшує вентиляцію легень. Цьому сприяє зниження тиску O_2 та підвищення парціального тиску CO_2 під впливом окисних процесів у працюючих м'язах.

Отже, підвищена вентиляція легень у стані м'язової роботи зумовлена хімічними змінами, що відбуваються в організмі (накопиченням вуглекислого газу, недоокислених метаболітів). Крім того, вентиляція легень посилюється за рахунок рефлєкторних ефектів.

Дихання при м'язовій роботі тісно пов'язане з біомеханічними особливостями положення тіла та самим механізмом руху. Різні положення на тілі може призвести до менш рівномірної вентиляції легень та зменшення загальної вентиляваної альвеолярної площі («гімнастичний місток», їзда велосипедом та ін.).

Таким чином, легенева вентиляція залежить від багатьох особливостей руху (циклічних, ациклічних, силових, тренувальних та інших особливостей, потужності, темпу та ін.). Тому вивчення рухового акта поєднують із дихальними вправами. Чим сильніший зв'язок між диханням та рухом, тим легше його виконувати.

У стані спокою людина споживає 250-350 мл кисню за хвилину, під час роботи - до 4500-5000 мл. При цьому максимальне споживання кисню (МПК) визначається здатністю дихальної та серцево-судинної систем транспортувати кисень до тканин. Оскільки величина МПК визначається насамперед кисневою ємністю крові, робота серця в цих умовах збільшується (у 2-3 рази збільшується частота серцевих скорочень та систолічний об'єм, УОК – до 35 л/хв). Під час роботи збільшується киснева ємність крові (за рахунок виходу еритроцитів з депо), а також збільшується швидкість поглинання кисню (від 60-80 мл у спокої до 120 мл під час роботи). Підвищене постачання тканин киснем у робочих умовах обумовлено концентрацією H^+ , що призводить до зниження парціального тиску кисню, підвищення парціального тиску вуглекислого газу та посилення дисоціації оксигемоглобіну.

РОЗДІЛ 3. МОДЕЛЬ ЛЕГЕНІ

3.1. Структурно-функціональна модель легені

Структурно-функціональні моделі легені та процесів газообміну в ньому вперше були побудовані відповідно до визначальних принципів імітаційного моделювання [12]. Кожному конструктивному елементу моделі приділяється певна функція. Це реалізація поведінки максимально наближеного до функціоналу біологічного прототипу.

Фактично легені є серію частинок, з'єднаних деревоподібним чином. Альвеоли розташовуються на кінцях гілок частинок. Всі альвеоли виконують ту саму функцію - забезпечують процес газообміну. Це забезпечує збагачення крові киснем та видалення вуглекислого газу. Саме альвеоли – це особливі «чорні ящики», які перетворюють вхідні інформаційні потоки *моделі вдихуваного повітря та моделі венозної крові* на вихідні інформаційні потоки *моделі видихуваного повітря та моделі артеріальної крові*. Хоча список компонентів повітряного потоку однаковий, їх кількісний склад сильно відрізняється. Подібно до складу крові людини, склад повітря містить безліч різних компонентів, тому на основі принципу абстракції буде обрано кінцеве число N основних компонентів, найбільш важливих з точки зору нормального функціонування людського тіла.

Як вже відомо, функціонування легені охоплює три етапи: біфуркація, біфуркація з частковим газообміном та повний газообмін на останньому рівні біфуркації. Газообмін у легенях відбувається між повітрям альвеол і кров'ю, яка їх омиває. У свою чергу при диханні повітря альвеол повинне обмінюватися із зовнішнім повітрям. Однак, як би не були глибокі дихальні рухи, повного обміну альвеолярного повітря з атмосферним ніколи не відбувається. Альвеолярна вентиляція визначається глибиною та частотою дихальних рухів та ставленням провідних шляхів до альвеолярного об'єму.

3.2. Імітаційна модель легені на основі штучних нейронних мереж

«Великий чорний ящик» моделі легенів перетворює два вхідних потоки моделі неочищеної крові, деоксигенацію та декарбонізацію, а також модель вдихуваного атмосферного повітря у два вихідні потоки моделі очищеної крові, оксигенацію та часткову декарбонізацію, та модель повітря, що видихається. «Великий чорний ящик», у свою чергу, утворює деревоподібну структуру «середнього чорного ящика», імітуючи розгалуження бронхів від рівня I до XII до термінальних бронхіол і ацинусів. Починаючи з VII порядку бронха, в структуру вводяться часточкові бронхи, модель термінальних бронхіол і ацинусів, де відбуваються процеси газообміну. Вони також підпадають під категорію «середніх чорних ящиків». Враховуючи специфіку анатомічної будови та функції легені, як базовий структурний елемент її моделі використовуються штучні альвеоли – «маленькі чорні скриньки», які трансформують вхідний інформаційний потік крові та потік альвеолярного повітря на вихідні інформаційні потоки повітря, що видихається, та очищеної крові.

Функціонування «малих чорних ящиків» реалізовано за допомогою моделей штучних нейронів [9]. Кожен окремий штучний нейрон визначає кількісний вміст однієї окремої компоненти в інформаційних потоках моделей крові та повітря у відповідних моделях. Кожна фаза функціонування штучної альвеоли, тобто окремий «малий чорний ящик», моделюється прошарком штучних нейронів – «атомарних чорних ящиків». Функціонування імітаційних моделей нирки та легені передбачає комп'ютерну підтримку складної багатопрощаркової штучної нейронної мережі зі зворотними зв'язками.

Кожен прошарок мережі має P^N основних штучних нейронів, де P – ступінь галуження на рівнях без газообміну. Число N - кількість ступенів поділу.

Окрім цих основних штучних нейронів у прошарках до $N+k$ також є пари

додаткових штучних нейронів, які відповідають за забезпечення часткового газообміну.

В межах кожного прошарку всі штучні нейрони сполучаються паралельно, тобто інформаційний потік рівномірно розподіляється між усіма «атомарними чорними ящиками».

Імітаційна модель функціонування легені є *відкритою складною штучною нейронною мережею із неповними глобальними зворотними зв'язками*. Відкритість мережі означає, що до інформаційного потоку моделі збагаченого киснем повітря щоразу додаються окремі компоненти і формується модель потоку повітря, що насичене вуглекислим газом.

Структурна схема описаної моделі зображена на рис. 3.1, де позначено наступні елементи:

- малі зелені прямокутники з номерами від 1 до N – «атомарні чорні ящики», що відповідають за кількісне входження потоку повітря по дереву галуження;
- малі зелені прямокутники з номерами від N до N+k – «атомарні чорні ящики», що відповідають за подальше кількісне входження потоку повітря, з того рівня, на якому розпочинається частковий процес газообміну в альвеолах;
- овали з номерами 1 до A-1 – «атомарні чорні ящики», що відповідають за забезпечення газообміну на його початкових рівнях;
- овали з номерами A – «атомарні чорні ящики», що відповідають за забезпечення газообміну на його останньому рівні;
- суцільні жирні стрілки – інформаційні потоки повітря та крові;
- пунктирні лінії і стрілки – зв'язки, які моделюють процеси поширення інформаційних потоків

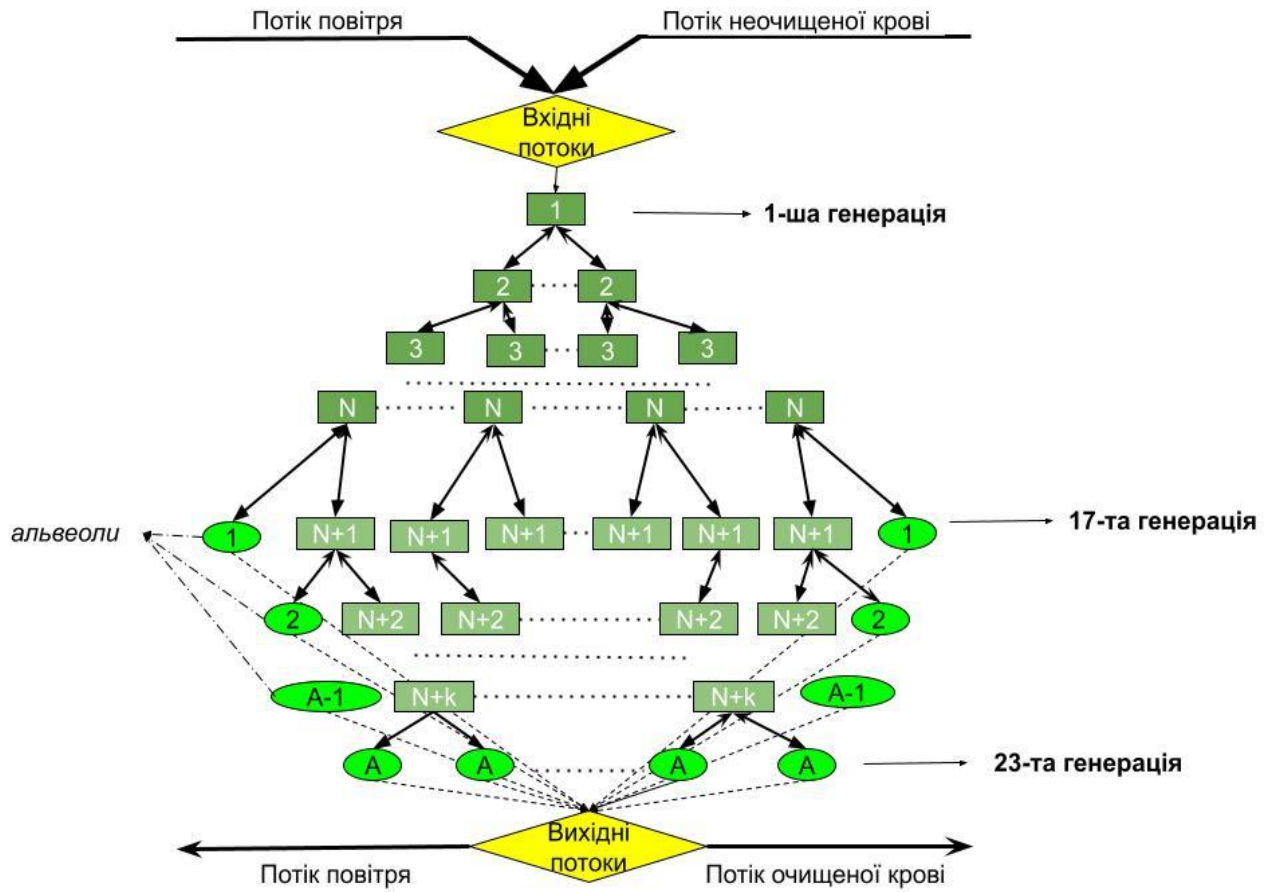


Рис. 3.1. Структурна схема моделі легені

РОЗДІЛ 4. ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІМІТАЦІЙНОЇ МОДЕЛІ ЛЕГЕНІ

Імітаційна модель легені побудована на базі багатопрошаркової штучної нейронної мережі. Програмна реалізація здійснена засобами об'єктно-орієнтовного програмування на мові C# із застосуванням апарату наслідування, віртуальності та поліморфізму при побудові ієрархій класів, та використанням презентаційної підсистеми WPF (Windows Presentation Foundation) та розширюваної мови розмітки застосунків XAML (Extensible Application Markup Language) при розробці програмного інтерфейсу [1, 2, 3, 13]. Компоненти XAML дозволяють сконструювати інтерфейс користувача WPF, вони визначають розміщення кнопок, панелей та інших елементів керування у вікні застосунку WPF. Для зберігання структури нейронної мережі на диску використано технологію JSON.

4.1. Етапи програмної реалізації

Програмна реалізація застосунку досить громіздка та здійснена у декілька етапів.

Першим важливим етапом стало вивчення функціонування системи дихання як одного цілого та її компонентів як окремих одиниць. Це дало змогу побачити загальну картину процесу дихання в організмі людини.

Наступним етапом була побудова моделі легені, що є багатопрошарковою штучною мережею із неповними зворотними зв'язками. Спочатку ця модель була досить проста. Але глибше дослідження окремих складових системи дихання дозволило її значно деталізувати. Цим самим доволі ускладнило саму модель, оскільки було враховано ряд особливостей та деталей процесу дихання.

Третім етапом створення програми став аналіз програмних застосунків, які можна було б використати у своїй роботі. Їх кількість досить велика, але потрібно було обрати лише ті, які дадуть змогу якомога точніше побудувати

модель процесу дихання та дозволять зобразити якомога зрозумілий для користувача інтерфейс.

Вибір зупинився на використанні презентаційної підсистеми WPF (Windows Presentation Foundation) та XAML (Extensible Application Markup Language).

Після підбору програмних продуктів було розпочато програмну реалізацію симулятора роботи легені. Він отримав назву «Lung Simulator».

Для початку було створено зображення головного вікна програми, розміщено поля для вводу та виводу даних, розміщено усі потрібні для роботи кнопки (рис. 4.1.).

Згодом додане ще одне робоче вікно, за допомогою якого можна задавати зміни у відсотковому складі повітря та передбачати вплив на процес дихання (рис. 4.2.).

Коли необхідні для роботи застосунку вікна створені, наступним етапом стала програмна обробка усіх компонентів обох вікон.

Для початку створено усі необхідні для роботи програми елементи:

```
double CO2VenKrov, O2VenKrov, N2VenKrov, H2OVenKrov; //
складові венозної крові
double CO2ArtKrov, O2ArtKrov, N2ArtKrov, H2OArtKrov; //
складові артеріальної крові
double azot;
double kysen;
double para;
double CO2;
double azotN, kysenN, paraN, CO2N; // додаткові
компоненти повітря
```

Згодом усім елементам надано початкові значення, де це потрібно. Деякі елементи отримують свої значення, зчитуючи їх з вікон робочого вікна. Для цього у програмі також прописано усі необхідні обробники.

```
azot = Convert.ToDouble(win1.TBAzot.Text);
kysen = Convert.ToDouble(win1.TBKysen.Text);
para = Convert.ToDouble(win1.Para.Text);
```

```
CO2 = Convert.ToDouble(win1.TBVuhlek.Text);
```

Наступним кроком написання програми стало визначення кількості рівнів перетворення, кількість стадій поділу без газообміну, ступенів галуження та відсоток газообміну на рівнях часткового газообміну.

```
if (rivniPeretv.Text == "") rivniPeretvor = 12;
else rivniPeretvor = int.Parse(rivniPeretv.Text); //
кількість рівнів перетворення(усіх разом)

if (Podily.Text == "") Npodilu = 6;
else Npodilu = int.Parse(Podily.Text); //кількість стадій
поділу без газообміну

if (StupinPodily.Text == "") Spodilu = 3;
else Spodilu = int.Parse(StupinPodily.Text); // ступінь
галуження

if (Gazoobmin.Text == "") gazoobmin = 3;
else gazoobmin = int.Parse(Gazoobmin.Text); // газообмін
```

Маючи усі необхідні дані створюємо масиви необхідних вхідних та вихідних потоків, з якими будемо працювати.

```
double[] potikVidsoTk = { kysen, CO2, para, azot};
double[] potikNorm = { kysenN, CO2N, paraN, azotN};
double[] potikKroviVen = { O2VenKrov, CO2VenKrov,
H2OVenKrov, N2VenKrov };
double[] potikKroviArt = { O2ArtKrov, CO2ArtKrov,
H2OArtKrov, N2ArtKrov };
double[] potikPovitrjaVdych = { kysen, CO2, para, azot };
double[] potikPovitrjaVydych = { kysenV, CO2V, paraV,
azotV};
double[] PovitrjaValveolach = { kysenAlv, CO2Alv,
paraAlv, azotAlv};
```

Після створення масивів нам потрібно звести усі величини до єдиної міри, тобто дізнатися абсолютну величину газів у потоках повітря.

```

    for (int i = 0; i <
potikPovitrjaVdych.Length;i++)//визначення абсолютної величини
газів у повітрі
    {
        potikPovitrjaVdych[i] = potikPovitrjaVdych[i] *
masapovitrja / 100 ;
        potikPovitrjaVydych[i] = potikPovitrjaVydych[i] *
masapovitrja1 / 100 ;
        PovitrjaValveolach[i] = PovitrjaValveolach[i] *
masapovitrja / 100;
    }

```

Наступним кроком стало програмування необхідного алгоритму роботи за допомогою ієрархії класів. Спочатку описуємо рух потоку повітря на рівнях галуження, далі вводимо у алгоритм частковий газообмін. Завершаючим етапом став повний газообмін на останніх рівнях галуження, та повернення вихідних потоків. По завершенню роботи цього алгоритму отримуємо кінцеві дані, які як результат виводимо у раніше створене для цього вікно.

4.2. Опис програмної системи

Програма «Lung Simulator» представляє собою симулятор роботи легені, за допомогою якого користувач має можливість відстежувати біологічні показники легені в залежності від параметрів побудованої моделі (Рис. 4.1).

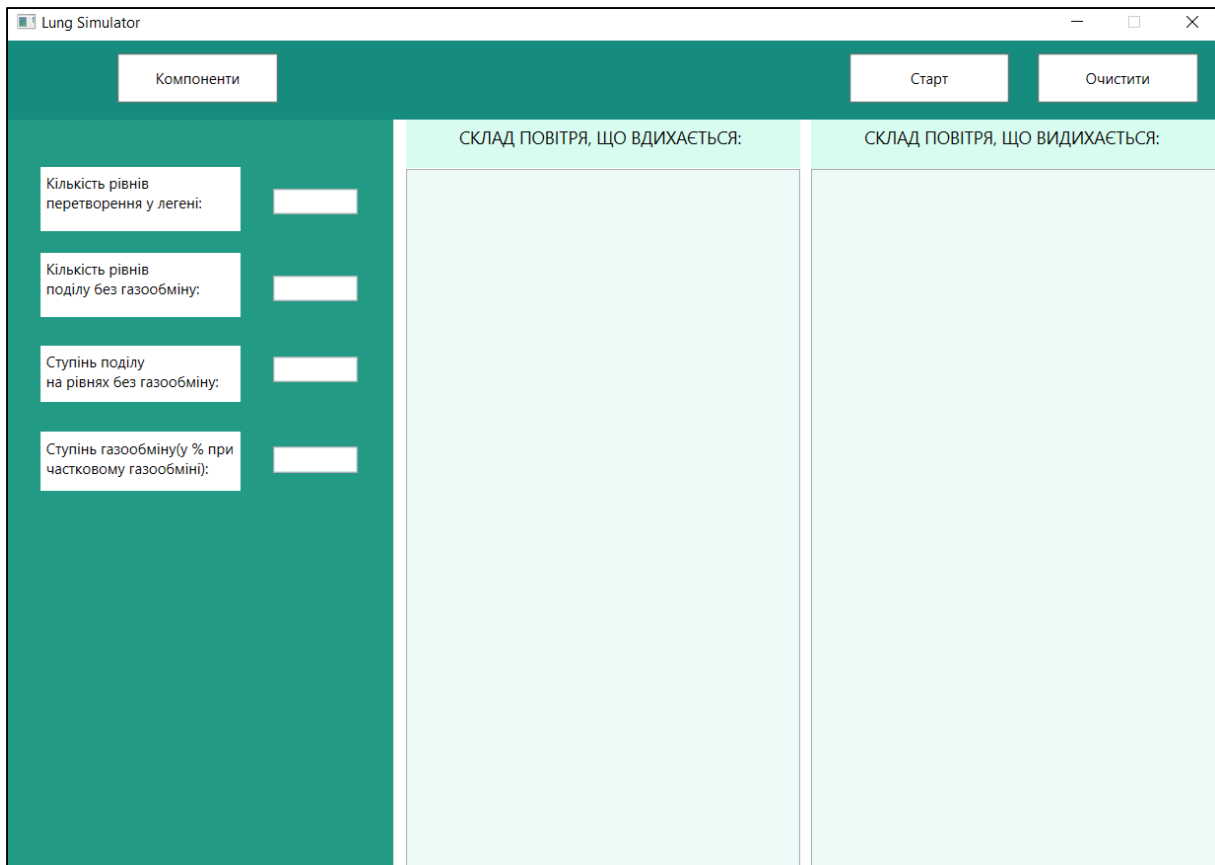


Рис. 4.1. Головне вікно програми.

В режимі інтерпретації моделі відбувається відтворення процесу газообміну. При цьому виводяться числові дані про склад потоку повітря, що вдихається та потоку повітря, що видихається.

Перед тим як приступити до симуляції роботи легені, користувачу потрібно задати відповідні параметри моделі в програмі. А саме: вказати кількість рівнів перетворення (саме кількість рівнів поділу, а не кількість генерацій), кількість рівнів поділу без газообміну, ступінь поділу на рівнях без газообміну та ступінь газообміну при частковому газообміні, тобто, на рівнях, де продовжується галуження, але вже частково відбувається зміна потоків моделей у зв'язку з газообміном.

Якщо дані параметри не будуть вказані, то програма запуститься із стандартними даними. А саме: кількість рівнів перетворення – 12, кількість рівнів поділу без газообміну – 6, ступінь поділу на рівнях без газообміну – 3, ступінь газообміну при частковому газообміні – 3.

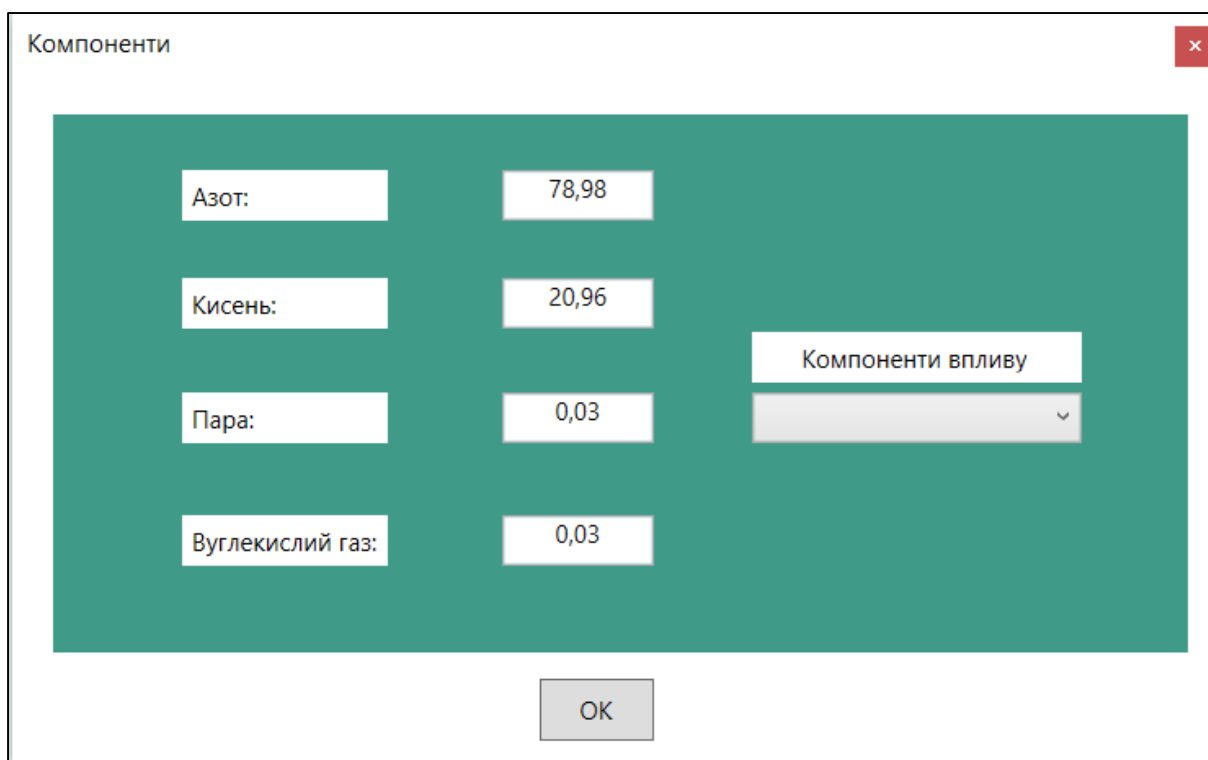
4.3. Симуляція роботи легені

Коли всі системні параметри задані, а саме: кількість рівнів перетворення без газообміну, кількість рівнів перетворення з газообміном, ступінь поділу без газообміну та відсоток газообміну, можна приступати до запуску симуляції роботи легені.

Дані про відсотковий склад повітря можна переглянути в інформаційній моделі натиснувши кнопку «Компоненти» (Рис. 4.2). На екрані буде відображено вікно, у якому вказаний склад вхідного потоку повітря у відсотках.

Хімічний склад сухого чистого повітря такий: азот — 78,98 %, кисень — 20,96 %, вуглекислий газ — 0,03 %, водень, гелій, неон, криптон, ксенон та ін. — 0,03 %. В залежності від різних умов навколишнього середовища хімічний склад повітря може змінюватися. Наприклад, кількість компонентів у повітрі може змінюватися у зв'язку з підніманням людини у гори. Тому програма “Lung Simulator” дає можливість змінювати ці складові для моделювання роботи легені при різних умовах.

Відсоткові дані про склад повітря можна змінювати. Проте, сумарна відсоткова кількість не повинна перевищувати 100%.



The screenshot shows a dialog box titled "Компоненти" with a close button (X) in the top right corner. The dialog has a green background and contains four input fields for component percentages, each with a corresponding value displayed in a white box to its right. To the right of these fields is a dropdown menu labeled "Компоненти впливу". At the bottom center is an "OK" button.

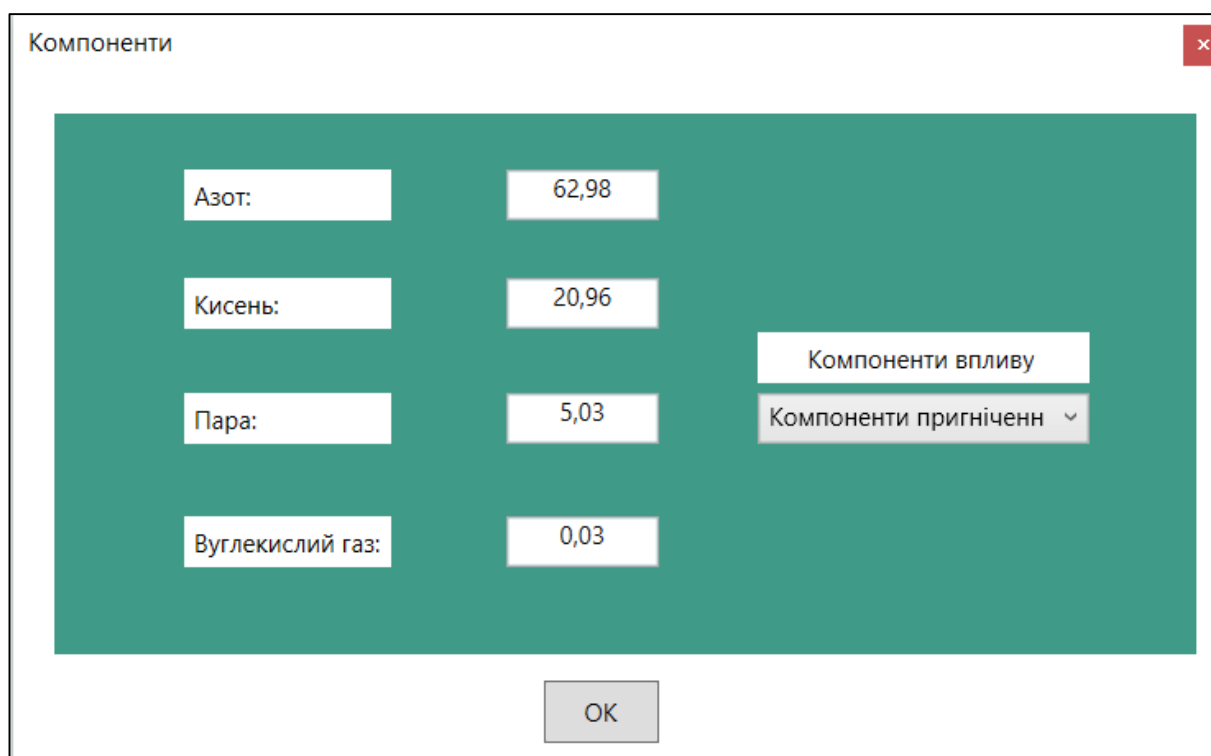
Компонент	Відсоток
Азот:	78,98
Кисень:	20,96
Пара:	0,03
Вуглекислий газ:	0,03

Компоненти впливу

OK

Рис. 4.2. Складові потоку повітря(у відсотках)

Вкажемо некоректні значення параметрів (Рис. 4.3), щоб перевірити, чи програма видасть нам помилку.



The screenshot shows the same "Компоненти" dialog box as in Figure 4.2, but with incorrect values entered in the input fields. The dropdown menu is now set to "Компоненти пригніченн".

Компонент	Відсоток
Азот:	62,98
Кисень:	20,96
Пара:	5,03
Вуглекислий газ:	0,03

Компоненти впливу

Компоненти пригніченн

OK

Рис. 4.3. Некоректні значення параметрів

Як бачимо (Рис. 4.4), програма видала повідомлення «Некоректні значення». Для продовження роботи з програмою потрібно закрити вікно з повідомленням, вказати коректні значення та натиснути «ОК».

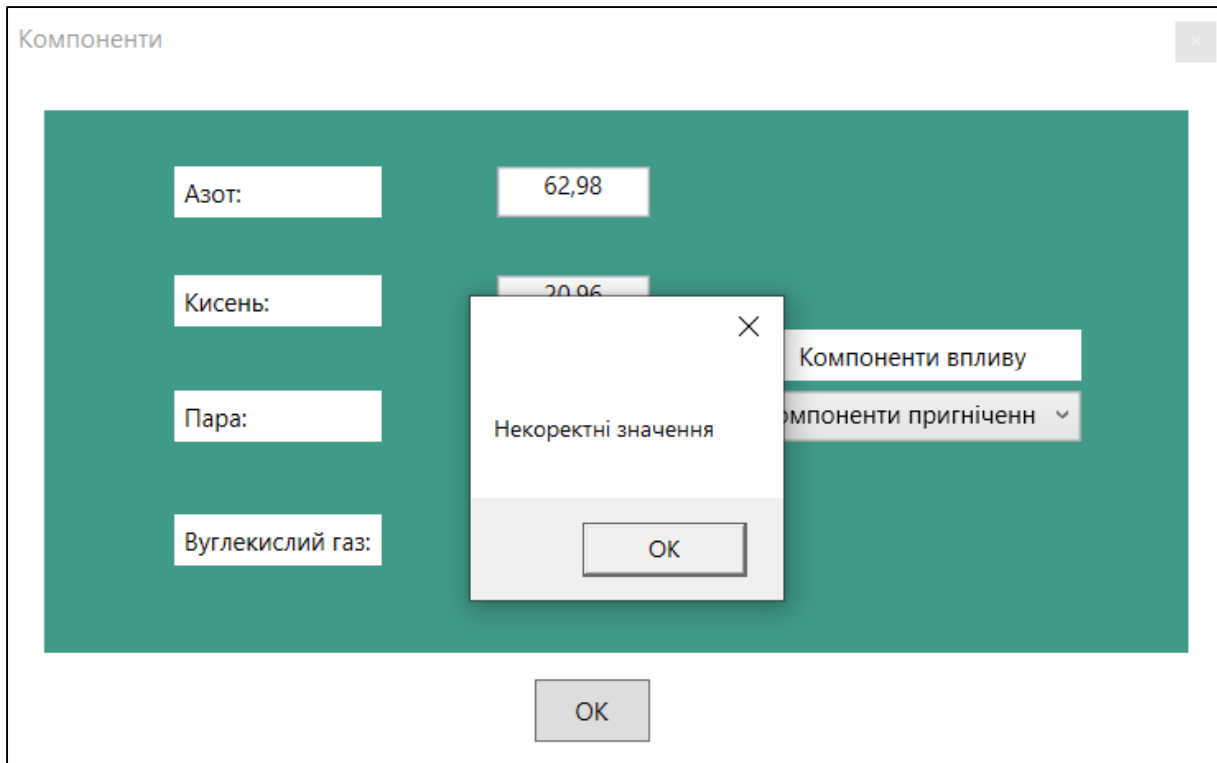


Рис. 4.4 Повідомлення про помилку

У даному вікні також можна задавати додатковий вплив на процес дихання, а саме вказувати, які компоненти впливають на процес газообміну: компоненти, що пришвидшують процес газообміну та компоненти, котрі навпаки, сповільнюють цей процес (Рис. 4.5).

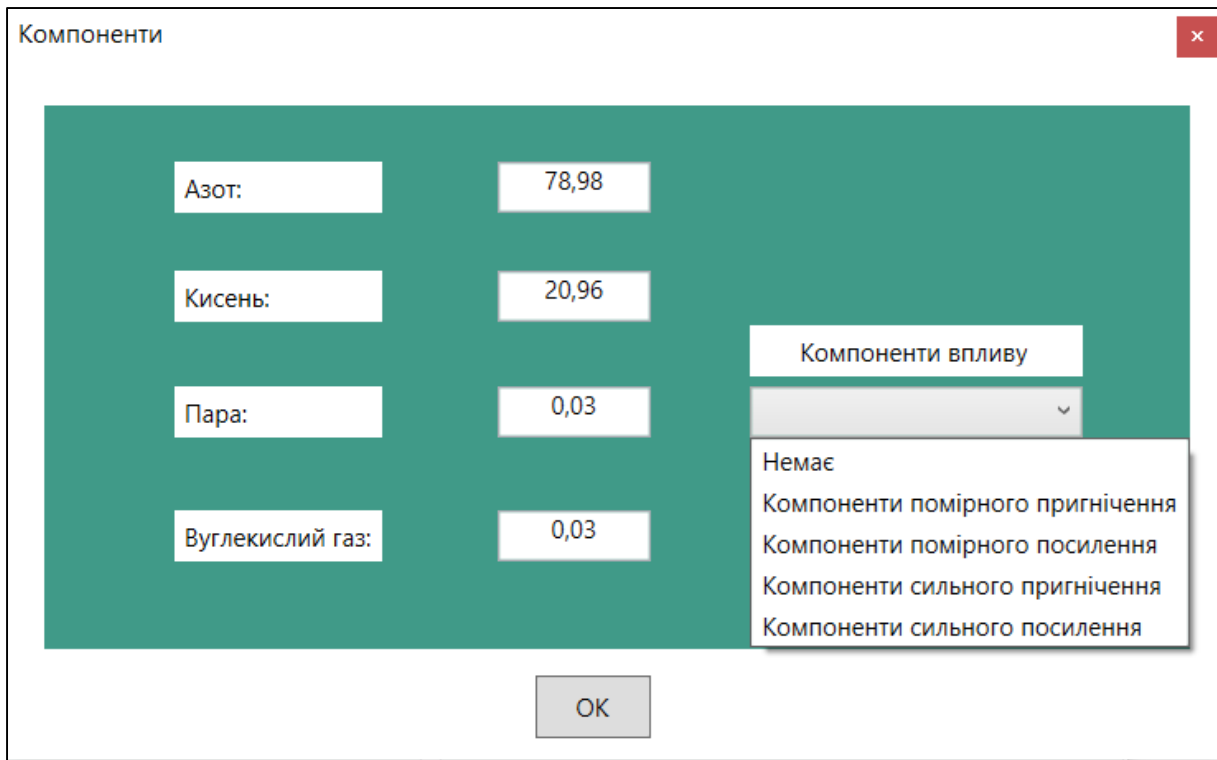


Рис. 4.5 Компоненти впливу на процес газообміну

Компонентами впливу на процес дихання та газообміну можуть бути як дія від прийому лікарських препаратів, так і хвороба. Розглянемо детальніше випадки, при яких процес газообміну відбувається з порушенням[17]:

а) прискорене дихання (тахіпное) — причини: емоції, фізичне навантаження, підвищена температура тіла;

б) сповільнене дихання (брадипное) — причини: захворювання ЦНС (також із підвищенням внутрішньочерепного тиску), отруєння;

в) дихання Чейна-Стокса — нерегулярне дихання, що полягає у поступовому прискоренні та поглибленні дихання, а потім сповільненні та поверхневому диханні з періодами апное (з періодичними перервами в диханні); причини: інсульт, метаболічна або постмедикаментозна енцефалопатія, серцева недостатність;

г) дихання Біота — швидке і поверхнєве нерегулярне дихання з довшими періодами апное (10–30 с); причини: підвищення внутрішньочерепного тиску, пошкодження ЦНС, постмедикаментозна кома.

Коли всі системні параметри задані, можна приступати до запуску симуляції роботи нирки. Для запуску програми активується кнопка «Старт» в правій верхній частині вікна. Система розпочне процес газообміну у легені.

Зазначимо, що у випадку незадання параметрів, програма розпочне роботу із стандартними значеннями.

За одну ітерацію інформаційний потік моделі легені проходить через усі альвеоли. При цьому відбувається обмін між потоком кисню та потоками венозної та артеріальної крові.

При нормальному функціонуванні легені у стані спокою доросла людина здійснює за хвилину 12-16 дихальних циклів, а при хвилюванні, фізичному навантаженні, хворобі – до 30-35 таких циклів. Під час спокійного вдиху до легень потрапляє 500 мл (300-800) повітря. Це називають *дихальним об'ємом*. Глибоко вдихнувши людина може набрати ще 1300-1500 мл повітря. Після глибокого видиху в повітроносних шляхах та альвеолах залишається 1000-1500 мл повітря – *залишковий об'єм*. Таким чином в межах доби моделюється в середньому 20160 ітерацій.

Зупинка симуляції роботи легені відбувається автоматично. При завершенні роботи програми буде відображено результат даної ітерації з певними параметрами (Рис. 4.6).

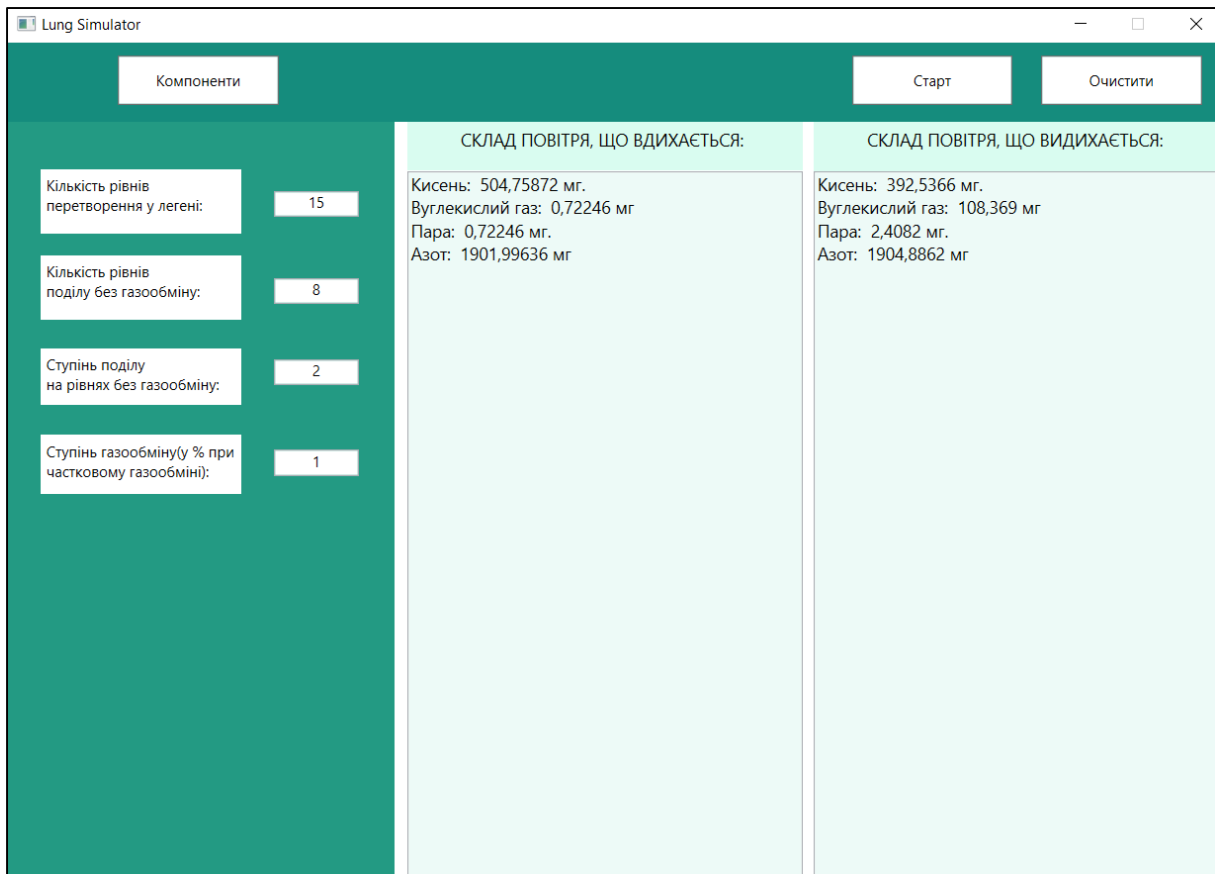


Рис. 4.6. Результат роботи програми

Коли програма завершить процес, інформаційні панелі не буде очищено. Для очищення панелей потрібно натиснути кнопку «Очистити». Тоді можна розпочати симуляцію спочатку, задавши потрібно параметри та знову натиснувши кнопку «Старт». Якщо ж панелі не очищено самостійно, а програму запущено знову, то результат роботи програми буде виведено на місці попереднього результату. При натисканні кнопки «Старт» програма автоматично очистить дані після попереднього запуску.

Дані про складові компоненти повітря в процесі моделювання роботи легені є результатом роботи програми. Їх можна переглянути в інформаційній панелі (Рис. 4.6): в правій частині вікна відображається склад потоку повітря у мг, що надходить до легені при вдиханні, а зліва відображається поточний склад повітря, котрий є результуючим при процесі газообміну у легені. Дані після кожної ітерації оновлюються, проте, якщо ніяких змін не зроблено, то дані будуть такі самі.

4.4. Вимоги до апаратно-програмних ресурсів ЕОМ, що забезпечують ефективне функціонування програмної системи

Програмна реалізація застосунку «Lung Simulator» здійснена засобами об'єктно-орієнтовного програмування на мові С#, із використанням презентаційної підсистеми WPF (Windows Presentation Foundation) та розширюваної мови розмітки застосунків XAML (Extensible Application Markup Language).

WPF дозволяє розробляти додатки, використовуючи як розмітку, так і код програмної частини, що звично для розробників на ASP.NET. Розмітка XAML зазвичай використовується для визначення зовнішнього вигляду програми, а керувані мови програмування (код програмної частини) - для реалізації його поведінки.

XAML - це мова розмітки на основі XML, який служить для визначення зовнішнього вигляду додатка в декларативній формі. Зазвичай він використовується для визначення вікон, сторінок і призначених для користувача елементів управління, а також їх заповнення елементами управління, фігурами і графічними елементами.

Необхідними умовами для роботи програми «Lung Simulator» є використання її в середовищах операційних систем Windows 7 та вище, та наявність встановленого .NET Framework версії 4.5.

Для встановлення .NET Framework версії 4.5 є певні вимоги до обладнання: процесор частотою 1 ГГц; ОЗУ - 512 Мб; дисковий простір (мінімум) для 32-розрядної версії - 4,5 Гб, для 64-розрядної версії - 4,5 Гб

Для установки платформи .NET Framework потрібні права адміністратора. Якщо у вас відсутні права адміністратора на комп'ютері, на якому потрібно встановити платформу .NET Framework, зверніться до адміністратора мережі.

ВИСНОВКИ

У ході дослідження була створена імітаційна модель легенів людини, яка являє собою складну багат шарову штучну нейронну мережу з неповним зворотним зв'язком.

При розробці моделі враховувалися основні особливості будови легень та функції окремих альвеол. Адекватність моделі на прийнятному рівні абстракції забезпечується правильним підбором основних компонентів (параметрів) інформаційних потоків моделей повітря і крові, які безпосередньо беруть участь у процесі газообміну в альвеолярній частині легені.

Комп'ютерна реалізація даної моделі базується на визначених принципах і прийомах об'єктно-орієнтованого програмування з використанням мови C#. Використання підсистеми представлення WPF і Extensible Application Markup Language XAML дає можливість розробляти інтерфейси програмування. Це дозволило одночасно забезпечити надійність, функціональну гнучкість, простоту налагодження програмної системи та адекватність імітаційної моделі в цілому.

Будова моделі легені досить складна і громіздка. Однак це не можна спростити, виключивши деякі рівні газообміну в легенях. Адже альвеоли на різних рівнях різного типу, по-різному функціонують і мають різний ступінь газообміну.

Створена програма є досить універсальною. За допомогою її функцій ви можете впливати на частоту і глибину дихання. Процедура передбачає введення в інформаційний потік моделі крові інших компонентів: ліків, токсинів, гормонів тощо. У свою чергу, вони впливають на рівень газообміну і, таким чином, на частоту і глибину дихання.

Також у моделі легені передбачено додатковий вплив на процес дихання. Наприклад, це вплив фізичних навантажень. Відсотково газообмін у легенях при бігу чи заняттю спортом не змінюється. Але здійснюється вплив на частоту дихання. Цим самим організм краще збагачується киснем.

Розробка таких моделей і програмного забезпечення, що їх підтримує, є особливо важливою з огляду на поширення пандемії COVID-19. Адже найбільше коронавірус вражає легені, сильно позначаючись на їх роботі та всьому процесі газообміну. Тому імітаційне моделювання їх функції дозволить ефективно досліджувати динаміку процесів, що відбуваються в них. А також такого типу моделі дозволять передбачити вплив фізичного навантаження на процес дихання під впливом інших чинників.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Албахари Дж. С# 6.0. Справочник. Полное описание языка / Джозеф Албахари, Бен Албахари. – С.-Пб.: Вильямс, 2016. – 1040 с.
2. Вайсфельд М. Объектно-ориентированное мышление / Мэтт Вайсфельд. – С.-Пб.: Питер, 2014. – 304 с.
3. Гамма Е. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Е. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссилес. – С.-Пб.: Питер, 2001. – 368 с.
4. Ганонг В.Ф. Фізіологія людини: Підручник / Вільям Ф. Ганонг. – Львів: БАК, 2002. – 784 с.
5. Григорян Р.Д. Программно-моделирующий комплекс для теоретических исследований взаимодействия физиологических систем организма / Р.Д. Григорян, К.Л. Атоев, П.Н. Лисов, А.А. Томин // Проблемы програмування. – 2006. – № 1. – С. 79-92.
6. Імітаційне моделювання систем та процесів: Електронне навчальне видання. Конспект лекцій / В. Б. Неруш, В. В. Курдеча. – К.: НН ІТС НТУУ «КПІ», 2012. – 115 с.
7. Інформаційні технології в професійній діяльності : матеріали XIII Всеукраїнської науково-практичної конференції / Рівне : РВВ РДГУ. 2020. 203 с.
8. Круглов В.В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. – М.: Горячая Линия – Телеком, 2002. – 382 с.
9. Руденко О.Г. Штучні нейронні мережі / О.Г. Руденко, Є.В. Бодянський – Харків: СМІТ, 2006. – 404 с.
10. Самарский А.А. Математическое моделирование: Идеи. Методы. Примеры / А.А. Самарский, А.П. Михайлов. – М.: Физматлит, 2005. – 320 с.
11. Сяський В.А. Імітаційна модель нирки людини на основі штучних нейронних мереж / В.А. Сяський, І.О. Сяська // «Сучасні проблеми математичного моделювання, обчислювальних методів та інформаційних

- технологій»: Матеріали Міжнародної наукової конференції. – Рівне: НУВГП, РДГУ, 2018. – С. 252-254.
12. Сяський В.А. Нейронно-мережне імітаційне моделювання фізіологічних процесів у нирці людини / Сяський В.А., Сяська І.О., Сяська І.В. // «Теоретичні та прикладні аспекти розвитку біологічних наук»: Матеріали III Всеукраїнської науково-практичної конференції. – Рівне: О. Зень, 2019. – С. 211-217.
 13. Томашевський В.М. Моделювання систем / В.М. Томашевський. – К.: Видавнича група ВНУ, 2005. – 352 с.
 14. Шеннон Р. Имитационное моделирование систем – искусство и наука / Роберт Ю. Шеннон. – М.: Мир, 1978. – 420 с.
 15. Microsoft Developer Network [Електронний ресурс] : Using WPF Controls
Режим доступу: [https://msdn.microsoft.com/en-us/library/bb384295\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb384295(v=vs.110).aspx). – Назва з екрану.
 16. Біологія: підруч. для 8 кл. загальноосвіт. навч. закл. / С. В. Страшко, Л.Г. Горяна, В. С. Білик, С. А. Ігнатенко. – К.:Грамота, 2016. – 288 с.: іл.
 17. Ependium - портал для лікарів
Режим доступу: <https://empendium.com/ua/chapter/B27.I.1.34>. – Назва з екрану.

ДОДАТКИ

Додаток А

Лістинг фрагменту коду реалізації головного вікна програмної системи

«Lung Simulator»

```
<Window x:Name="Wind" x:Class="Kursova.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:local="clr-namespace:Kursova"
  mc:Ignorable="d"
  Title="Lung Simulator"      ResizeMode="CanMinimize"      Height="712"
  Width="1000" >
  <Grid Margin="0,0,-0.4,-19.6" >
    <Grid.ColumnDefinitions>
      <ColumnDefinition/>
    </Grid.ColumnDefinitions>
    <Menu HorizontalAlignment="Left" Height="615" Margin="0,64,0,0"
  VerticalAlignment="Top" Width="313" Background="#FF239A83"/>
    <Label Content="Кількість рівнів &#xA;перетворення у легені:"
  HorizontalAlignment="Left" Margin="26,102,0,0" VerticalAlignment="Top"
  Width="163" Height="52" Background="White"/>
    <TextBox x:Name="rivniPeretv" HorizontalAlignment="Left" Height="21"
  Margin="215,120,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
  Width="69" TextAlignment="Center" RenderTransformOrigin="1.118,1.407"
  TextChanged="RivniPeretv_TextChanged" Text="12"/>
    <Label Content="Ступінь поділу &#xD;&#xA;на рівнях без газообміну:"
  HorizontalAlignment="Left" Margin="26,247,0,0" VerticalAlignment="Top"
  Width="163" Height="46" RenderTransformOrigin="0.502,0.71"
  Background="White"/>
```

```
<TextBox x:Name="Podily" HorizontalAlignment="Left" Height="21"
Margin="215,190,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
Width="69" TextAlignment="Center" TextChanged="Alveoly_TextChanged"
Text="6"/>
```

```
<Label Content="Ступінь газообміну(у % при &#xD;&#xA;частковому
газообміні):" HorizontalAlignment="Left" Margin="26,317,0,0"
VerticalAlignment="Top" Width="163" Height="48" Background="White"/>
```

```
<TextBox x:Name="Gazoobmin" HorizontalAlignment="Left" Height="21"
Margin="215,329,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
Width="69" TextAlignment="Center" RenderTransformOrigin="0.47,-0.292"
TextChanged="RivniPeretv_Copy1_TextChanged" Text="3"/>
```

```
<Menu HorizontalAlignment="Left" Height="64" VerticalAlignment="Top"
Width="994" Margin="0,0,-0.4,0" Background="#FF158C7D"/>
```

```
<Button x:Name="butKomp" Content="КОМПОНЕНТИ"
HorizontalAlignment="Left" Height="40" VerticalAlignment="Top" Width="130"
Background="White" Margin="89,10,0,0" Click="ButKomp_Click"/>
```

```
<Button x:Name="start" Content="Старт" HorizontalAlignment="Left"
Height="40" Margin="683,10,0,0" VerticalAlignment="Top" Width="130"
Background="White" Click="Start_Click"/>
```

```
<Label Content="Кількість рівнів &#xA;поділу без газообміну:"
HorizontalAlignment="Left" Margin="26,172,0,0" VerticalAlignment="Top"
Width="163" Height="52" Background="White"/>
```

```
<TextBox x:Name="StupinPodily" HorizontalAlignment="Left" Height="21"
Margin="215,256,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
Width="69" TextAlignment="Center" TextChanged="Alveoly_TextChanged"
Text="3"/>
```

```
<Canvas x:Name="canDrawing" HorizontalAlignment="Left" Height="615"
Margin="313,64,-0.4,0" VerticalAlignment="Top" Width="681"/>
```

```

    <Label Content="                СКЛАД ПОВІТРЯ, ЩОВДИХАЄТЬСЯ:"
Height="39"      Width="320"      FontSize="14"      Background="#FFD9FCF0"
Canvas.Left="10"/>

```

```

    <Label Content="                СКЛАД ПОВІТРЯ, ЩОВИДИХАЄТЬСЯ:"
Height="40"      Canvas.Left="339"      Width="332"      FontSize="14"
Background="#FFD9FCF0"/>

```

```

    <TextBox  x:Name="Vdych"      Height="575"      TextWrapping="Wrap"
Canvas.Top="40"  Width="320"      Background="#FFEDFAF7"  Canvas.Left="10"
TextChanged="TextBox_TextChanged_1"  FontSize="14"      IsEnabled="False"
FontWeight="Bold"/>

```

```

    <TextBox  x:Name="Vydych"      Height="575"      TextWrapping="Wrap"
Canvas.Top="40"      Width="332"      Canvas.Left="339"
TextChanged="TextBox_TextChanged"  Background="#FFEDFAF7"  FontSize="14"
IsEnabled="False"  FontWeight="Bold"/>

```

```

</Canvas>

```

```

    <Button  x:Name="Clear"  Content="Очистити"  HorizontalAlignment="Left"
Height="40"  Margin="836,10,0,0"  VerticalAlignment="Top"  Width="130"
Background="White"  Click="Clear_Click"/>

```

```

</Grid>

```

```

</Window>

```

Лістинг фрагменту коду реалізації логіки взаємодії з головним вікном
програмної системи «Lung Simulator»

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Diploma

{

    /// <summary>

    /// Логика взаимодействия для MainWindow.xaml

    /// </summary>
```



```
public partial class MainWindow : Window
{
    Window1 win1 = new Window1();

    public MainWindow()
    {
        InitializeComponent();
    }

    private void Button_Click(object sender, RoutedEventArgs e)
    {
    }

    private void Button_Click_1(object sender, RoutedEventArgs e)
    {
    }

    private void Window_Loaded(object sender, RoutedEventArgs e)
    {
        win1.Owner = this;
    }

    private void ButKomp_Click(object sender, RoutedEventArgs e)
    {
        win1.ShowDialog();
    }
}
```

```
}  
  
protected override void OnClosed(EventArgs e)  
{  
    base.OnClosed(e);  
  
    Application.Current.Shutdown();  
}  
  
private void RivniPeretv_TextChanged(object sender, TextChangedEventArgs  
e)  
{  
    //double rivni = Convert.ToDouble(rivniPeretv.Text);  
  
    //if (rivni > 23)  
  
    //    MessageBox.Show("Помилкове значення");  
}  
  
private void RivniPeretv_Copy1_TextChanged(object sender,  
TextChangedEventArgs e)  
{  
    //double gazoob = Convert.ToDouble(Gazoobmin.Text);  
  
    //if (gazoob > 100.0)  
  
    //    MessageBox.Show("Помилкове значення");  
}
```

```

private void Start_Click(object sender, RoutedEventArgs e)
{
    double masapovitrja; // маса повітря

    double masapovitrja1; // маса повітря, яка наявна у легенях

    double CO2VenKrov, O2VenKrov, N2VenKrov, H2OVenKrov; // складові
веннозної крові

    double CO2ArtKrov, O2ArtKrov, N2ArtKrov, H2OArtKrov; // складові
артеріальної крові

    double azot;

    double kysen;

    double para;

    double CO2;

    double azotN, kysenN, paraN, CO2N; // додаткові компоненти повітря

    masapovitrja = 2408.2;

    masapovitrja1 = 100.2;

    // const double SchilnistPovitrja = 1.2041; // щільність повітря для
перетворення л в мг

    CO2VenKrov = 46.0;

    O2VenKrov = 40.0;

    N2VenKrov = 573.0;

    H2OVenKrov = 47.0;

    CO2ArtKrov = 40.0;

```

```
O2ArtKrov = 95.0;
```

```
N2ArtKrov = 573.0;
```

```
H2OArtKrov = 47.0;
```

```
azot = Convert.ToDouble(win1.TBAzot.Text);
```

```
kysen = Convert.ToDouble(win1.TBKysen.Text);
```

```
para = Convert.ToDouble(win1.Para.Text);
```

```
CO2 = Convert.ToDouble(win1.TBVuhlek.Text);
```

```
azotN = 78.98;
```

```
kysenN = 20.96;
```

```
paraN = 0.03;
```

```
CO2N = 0.03;
```

```
int rivniPeretvor, Npodilu, Spodilu;
```

```
double kompVplyvu = 0, gazoobmin;
```

```
if (rivniPeretv.Text == "") rivniPeretvor = 12;
```

```
else    rivniPeretvor = int.Parse(rivniPeretv.Text); // кількість рівнів  
перетворення(усіх разом)
```

```
if (Podily.Text == "") Npodilu = 6;
```

```
else    Npodilu = int.Parse(Podily.Text); //кількість стадій поділу без  
газообміну
```

```

if (StupinPodily.Text == "") Spodilu = 3;

else Spodilu = int.Parse(StupinPodily.Text);// ступінь галуження

if (Gazoobmin.Text == "")gazoobmin = 3;

else gazoobmin = int.Parse(Gazoobmin.Text);// газообмін

double azotV = 79.1, kysenV = 16.3, paraV = 0.1, CO2V = 4.5; // склад
видихуваного повітря у %

double azotAlv = 80.6, kysenAlv = 14.2, paraAlv = 0.01, CO2Alv = 5.19; //
склад альвеолярного повітря у %

//double PazotVdych = 596.0, PkysenVdych = 158.0, PparaVdych = 5.7,
PCO2Vdych = 0.3; // склад вдихуваного повітря у мм рт ст

//double PazotVyd = 565.0, PkysenVyd = 116.0, PparaVyd = 47.0, PCO2Vyd
= 32.0; // склад видихуваного повітря у мм рт ст

//double PazotAlveol = 573.0, PkysenAlveol = 100.0, PparaAlveol = 47.0,
PCO2Alveol = 40.0; // склад альвеолярного повітря у мм рт ст

double[] potikVidsotk = { kysen, CO2, para, azot};

double[] potikNorm = { kysenN, CO2N, paraN, azotN};

double[] potikKroviVen = { O2VenKrov,CO2VenKrov, H2OVenKrov,
N2VenKrov };

double[] potikKroviArt = { O2ArtKrov,CO2ArtKrov, H2OArtKrov,
N2ArtKrov };

double[] potikPovitrjaVdych = { kysen, CO2, para, azot };

double[] potikPovitrjaVydych = { kysenV, CO2V,paraV, azotV};

```

```

double[] PovitrjaValveolach = { kysenAlv,CO2Alv, paraAlv,azotAlv };

double countVitokDo6 = Math.Pow(Spodilu,Npodilu-1); //кількість віток
при поділі

double countVitokVid6 = 0;

double    A1    =    Convert.ToDouble(win1.A1.Text),    A2    =
Convert.ToDouble(win1.A2.Text), A3 = Convert.ToDouble(win1.A3.Text), A4 =
Convert.ToDouble(win1.A4.Text);

kompVplyvu = A1 + A2 + A3 + A4;

gazoobmin = gazoobmin + kompVplyvu; // відсоток газообміну з
КОМПОНЕНТАМИ ВПЛИВУ

Vdych.Text += "Кисень: " + kysen * masapovitrja / 100 + " мг.";

Vdych.Text += "\r\n";

Vdych.Text += "Вуглекислий газ: " + CO2 * masapovitrja / 100 + " мг";

Vdych.Text += "\r\n";

Vdych.Text += "Пара: " + para * masapovitrja / 100 + " мг.";

Vdych.Text += "\r\n";

Vdych.Text += "Азот: " + azot * masapovitrja / 100 + " мг";

Vdych.Text += "\r\n";

Vdych.Text += "\r\n";

```

```

for (int i = 0; i < potikPovitrjaVdych.Length; i++)//визначення абсолютної
величини газів у повітрі

```

```

{

```

```

    potikPovitrjaVdych[i] = potikPovitrjaVdych[i] * masapovitrja / 100 ;

```

```

    potikPovitrjaVydych[i] = potikPovitrjaVydych[i] * masapovitrja1 / 100 ;

```

```

    PovitrjaValveolach[i] = PovitrjaValveolach[i] * masapovitrja / 100;

```

```

}

```

```

//поділ інформаційного потоку на вузли (кількість комп повітря на
кожну вітку)

```

```

for (int i = 0; i < Npodilu; i++)

```

```

{

```

```

    for (int j = 0; j < potikPovitrjaVdych.Length; j++)

```

```

    {

```

```

        potikPovitrjaVdych[j] = potikPovitrjaVdych[j] / Spodilu;

```

```

    }

```

```

}

```

```

for (int i = 1; i < rivniPeretvor - Npodilu; i++)// поділ(кількість комп
повітря на кожну вітку) з частковим газообміном(у крайніх вітках)

```

```

{

```

```

    for (int j = 0; j < potikPovitrjaVdych.Length; j++)//подальше галуження

```

```

    {

```

```

        potikPovitrjaVdych[j] = potikPovitrjaVdych[j] / 2;

```

```

    }

    countVitokVid6 = countVitokDo6 * Math.Pow(2, i) - 2; // кількість віток
на кожному рівні при галуженні і газообміні

    }

    for (int j = 0; j < Math.Pow(2,(rivniPeretvor-Npodilu-1)); j++)//частковий
газообмін повітря у крайніх вітках
    {
        for (int k = 0; k < PovitrjaValveolach.Length; k++)//повітря в альвеолах
        {
            PovitrjaValveolach[k] = potikPovitrjaVdych[k] * gazoobmin /
100;//обмін вдихуваного і альвеолярного

            potikPovitrjaVdych[k] -= potikPovitrjaVdych[k] * gazoobmin /
100;//зміна складу вдихуваного

            potikPovitrjaVydych[k] += PovitrjaValveolach[k] * gazoobmin / 100; //
наповнення потоку на видих
        }
    }

    countVitokVid6 = countVitokDo6 * Math.Pow(2, (rivniPeretvor - Npodilu));
// кількість віток на останньому рівні при газообміні

```



```

for (int i = 0; i < countVitokVid6; i++)//повний газообмін повітря
{
    for (int k = 0; k < PovitrjaValveolach.Length; k++)//повітря в альвеолах
    {
        PovitrjaValveolach[k] = potikPovitrjaVdych[k] * gazoobmin /
100;//обмін вдихуваного і альвеолярного

        potikPovitrjaVdych[k] -= potikPovitrjaVdych[k] * gazoobmin /
100;//зміна складу вдихуваного

        potikPovitrjaVydych[k] += PovitrjaValveolach[k] * gazoobmin / 100; //
наповнення потоку на видих
    }

    if (i == countVitokVid6-1)
    {
        Vydych.Text += "Кисень: " + potikPovitrjaVydych[0] * masapovitrja /
100 + " мг.";

        Vydych.Text += "\r\n";

        Vydych.Text += "Вуглекислий газ: " + potikPovitrjaVydych[1] *
masapovitrja / 100 + " мг";

        Vydych.Text += "\r\n";
    }
}

```

```
Vydych.Text += "Паpa: " + potikPovitrjaVydych[2] * masapovitrja /  
100 + " мг.";
```

```
Vydych.Text += "\r\n";
```

```
Vydych.Text += "Азот: " + potikPovitrjaVydych[3] * masapovitrja /  
100 + " мг";
```

```
Vydych.Text += "\r\n";
```

```
Vydych.Text += "\r\n";
```

```
break;
```

```
}
```

```
}
```

```
}
```

```
private void Alveoly_TextChanged(object sender, TextChangedEventArgs e)
```

```
{
```

```
}
```

```
private void TextBox_TextChanged(object sender, TextChangedEventArgs e)
```

```
{
```

```
}
```

```
private void TextBox_TextChanged_1(object sender, TextChangedEventArgs e)
```

```
{
```

```
}
```

```
private void Clear_Click(object sender, RoutedEventArgs e)
{
    Vdych.Clear();
    Vydych.Clear();
    //System.Environment.Exit(0);
}
}
```

}

Лістинг фрагменту коду реалізації панелі складу потоку повітря, яке надходить у легені

```

<Window x:Name="win1" x:Class="Kursova.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:local="clr-namespace:Kursova"
  mc:Ignorable="d"
  Title="Компоненти"           Height="374.4"           Width="604.8"
  WindowStartupLocation="CenterScreen"
  ResizeMode="NoResize"  ShowInTaskbar="False"  WindowStyle="ToolWindow"
  AutomationProperties.Name="Компоненти" Closing="Window_Closing">
  <Grid Margin="0,0,-6.6,1.6">
    <Menu HorizontalAlignment="Left" Height="263" Margin="20,20,0,0"
  VerticalAlignment="Top" Width="556" Background="#FF409A88"/>
    <Label Content="Азот:" HorizontalAlignment="Left" Height="25"
  Margin="83,47,0,0" VerticalAlignment="Top" Width="101" Background="White"
  />
    <Label Content="Вуглекислий газ:" HorizontalAlignment="Left" Height="25"
  Margin="83,216,0,0" VerticalAlignment="Top" Width="101"
  Background="White"/>
    <Label Content="Пара:" HorizontalAlignment="Left" Height="25"
  Margin="83,156,0,0" VerticalAlignment="Top" Width="101"
  Background="White"/>
    <Label Content="Кисень:" HorizontalAlignment="Left" Height="25"
  Margin="83,100,0,0" VerticalAlignment="Top" Width="101"
  Background="White"/>

```

```
<TextBox x:Name="TBAzot" HorizontalAlignment="Left" Height="25"
Margin="240,47,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="74"
Text="78,98" TextAlignment="Center" TextChanged="TextBox_TextChanged"
AcceptsReturn="True"/>
```

```
<TextBox x:Name="TBVuhlek" HorizontalAlignment="Left" Height="25"
Margin="240,216,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
Width="74" Text="0,03" TextAlignment="Center"
TextChanged="TBVuhlek_TextChanged"/>
```

```
<TextBox x:Name="Para" HorizontalAlignment="Left" Height="25"
Margin="240,156,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
Width="74" Text="0,03" TextAlignment="Center"
TextChanged="TBInshi_TextChanged" AcceptsReturn="True"/>
```

```
<TextBox x:Name="TBKysen" HorizontalAlignment="Left" Height="25"
Margin="240,100,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
Width="74" Text="20,96" TextAlignment="Center"
TextChanged="TBKysen_TextChanged" AcceptsReturn="True"/>
```

```
<Button Content="OK" HorizontalAlignment="Left" Height="30"
Margin="258,296,0,0" VerticalAlignment="Top" Width="56"
Click="Button_Click"/>
```

```
<Label Content="Компоненти впливу" HorizontalAlignment="Left"
Height="25" Margin="362,126,0,0" VerticalAlignment="Top" Width="162"
Background="White"/>
```

```
<ComboBox x:Name="CBKomp" HorizontalAlignment="Left" Height="25"
Margin="362,156,0,0" VerticalAlignment="Top" Width="162"
SelectionChanged="ComboBox_SelectionChanged">
```

```
<TextBlock>Немає</TextBlock>
```

```
<TextBlock>Компоненти помірного пригнічення</TextBlock>
```

```
<TextBlock>Компоненти помірного посилення</TextBlock>
```

```
<TextBlock>Компоненти сильного пригнічення</TextBlock>
```

```
<TextBlock>Компоненти сильного посилення</TextBlock>
```

</ComboBox>

</Grid>

</Window>

Лістинг фрагменту коду реалізації логіки взаємодії
з панеллю складу потоку повітря, яке надходить у легені

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Kursova
{
    /// <summary>
    /// Логика взаимодействия для Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {
        public Window1()
        {
            InitializeComponent();
        }
    }
}
```

```

private void Window_Closing(object sender,
System.ComponentModel.CancelEventArgs e)
{
    e.Cancel = true;
    Hide();
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    double povitrja;
    povitrja = Convert.ToDouble(TBAzot.Text) +
Convert.ToDouble(TBKysen.Text) + Convert.ToDouble(Para.Text) +
Convert.ToDouble(TBVuhlek.Text);
    if(povitrja != 100.0)
        MessageBox.Show("Некоректні значення");
    else win1.Close();
}

private void TextBox_TextChanged(object sender, TextChangedEventArgs e)
{
    double azot = Convert.ToDouble(TBAzot.Text);
    if (azot > 100.0)
        MessageBox.Show("Помилкове значення");
}

private void TBKysen_TextChanged(object sender, TextChangedEventArgs e)
{
    double kysen = Convert.ToDouble(TBKysen.Text);
    if (kysen > 100.0)

```



```
        MessageBox.Show("Помилкове значення");
    }

private void TBInshi_TextChanged(object sender, TextChangedEventArgs e)
{
    double para = Convert.ToDouble(Para.Text);
    if (para > 100.0)
        MessageBox.Show("Помилкове значення");
}

private void TBVuhlek_TextChanged(object sender, TextChangedEventArgs e)
{
    double CO2 = Convert.ToDouble(TBVuhlek.Text);
    if (CO2 > 100.0)
        MessageBox.Show("Помилкове значення");
}
}
}
```