

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
ННІ "Комп'ютерних наук та штучного інтелекту"
Кафедра комп'ютерних систем та робототехніки



«Затверджую»
В.о. завідуючого кафедри
комп'ютерних систем та робототехніки
_____ к. ф.-м. н., доцент Максим. Хруслов
«__» червня 2025 р

Пояснювальна записка
до кваліфікаційної роботи
бакалавра

на тему: **КОМП'ЮТЕРНА МОДЕЛЬ ПРОГНОЗУВАННЯ ВПЛИВУ
ШКІДЛИВИХ ЗВИЧОК НА ТРИВАЛІСТЬ ЖИТТЯ ЛЮДИНИ**

Галузь знань: 15 – Електроніка, автоматизація та електронні комунікації
Спеціальність: 151 – Автоматизація, комп'ютерно-інтегровані технології та
робототехніка.

Захищено на засіданні
Екзаменаційної комісії № 46
протокол № __ від __.06.2025 р.
Оцінка _____ / _____
Голова Екзаменаційної комісії
_____ **ЧУГАЙ А.М.**

Виконав:
студент 4 курсу, групи КУ–41
КОЛІСНИЧЕНКО Микита
Миколайович

Керівник: д.т.н., с.н.с., професор
кафедри комп'ютерних систем та
робототехніки
ТОЛСТОЛУЗЬКА Олена Геннадіївна


Рецензент: к.т.н., доцент, доцент
кафедри кібербезпеки інформаційних
систем, мереж і технологій
МЕЛКОЗЬОРОВА Ольга Михайлівна

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, 4 розділів, висновків, списку використаних джерел і 3 додатків. Загальний обсяг роботи складає 73 сторінки, із яких 50 сторінок основної частини з 17 рисунками, 16 найменуваннями списку використаних джерел та трьома додатками.

Метою кваліфікаційної роботи – розробити відтворювану комп'ютерну модель на Python, яка за допомогою введених користувачем даних та епідеміологічних коефіцієнтів обчислює роки життя, втрачені внаслідок куріння, споживання алкоголю й їхньої комбінації.

Об'єкт дослідження – процес прогнозування індивідуальної тривалості життя з урахуванням шкідливих звичок.

Предмет дослідження – методи та моделі прогнозування індивідуальної тривалості життя з урахуванням шкідливих звичок.

Проблема, яку вирішує кваліфікаційна робота – відсутність доступних і відтворюваних інструментів для швидкої персоналізованої оцінки впливу куріння, зловживання алкоголем та їхньої комбінованої дії на очікувану тривалість життя людини, що базуються на перевірених епідеміологічних коефіцієнтах і дозволяють отримувати точні прогнози без необхідності застосовувати складні статистичні моделі чи ресурсоємні медичні програмні комплекси.

Область застосування – мобільні та веб-додатки для персонального моніторингу здоров'я і профілактики шкідливих звичок, інформаційно-аналітичні рішення в клінічних системах та сервісах телемедицини для оцінки індивідуального ризику, корпоративні програми оздоровлення та HR-аналітика, рішення для страхових компаній при розрахунку премій та управлінні портфелем ризиків, а також епідеміологічні платформи та дослідницькі центри, що аналізують вплив поведінкових факторів на тривалість життя.

Ключові слова – КОМП'ЮТЕРНА МОДЕЛЬ ПРОГНОЗУВАННЯ, ШКІДЛИВІ ЗВИЧКИ, PYTHON СЛІ, ЕПІДЕМІОЛОГІЧНІ КОЕФІЦІЄНТИ, КОМБІНОВАНИЙ ВПЛИВ РИЗИК-ФАКТОРІВ, ІНДИВІДУАЛІЗОВАНИ ПРОГНОЗИ ТРИВАЛОСТІ ЖИТТЯ.

ABSTRACT

The explanatory note to the bachelor's qualification thesis consists of an introduction, 4 chapters, conclusions, a list of references and 3 appendices. The total volume of the work is 73 pages, including 50 pages of the main part with 17 figures, 16 references and 3 appendices.

The purpose of the qualification work is to develop a reproducible computer model in Python that, using user-entered data and epidemiological coefficients, calculates the years of life lost due to smoking, alcohol consumption, and their combination.

The object of the study is the process of predicting individual life expectancy taking into account bad habits.

The subject of the study is methods and models for predicting individual life expectancy with regard to bad habits.

The problem solved by the qualification work is the lack of accessible and reproducible tools for rapid personalized assessment of the impact of smoking, alcohol abuse and their combined effect on life expectancy, based on proven epidemiological coefficients and allowing accurate predictions without the need to use complex statistical models or resource-intensive medical software systems.

Application areas: mobile and web applications for personal health monitoring and prevention of bad habits, information and analytical solutions in clinical systems and telemedicine services for individual risk assessment, corporate wellness programs and HR analytics, solutions for insurance companies in calculating premiums and managing risk portfolios, as well as epidemiological platforms and research centers analyzing the impact of behavioral factors on life expectancy.

Keywords - COMPUTER FORECASTING MODEL, BAD HABITS, PYTHON CLI, EPIDEMIOLOGICAL COEFFICIENTS, COMBINED EFFECT OF RISK FACTORS, INDIVIDUALIZED LIFE EXPECTANCY FORECASTS.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ЛІТЕРАТУРИ.....	9
1.1. Визначення та класифікація шкідливих звичок.....	9
1.2. Епідеміологія куріння та його вплив на очікувану тривалість життя	10
1.3. Епідеміологія вживання алкоголю та його вплив на очікувану тривалість життя.....	11
1.4. Поєднаний вплив комбінованих шкідливих звичок на смертність	12
1.5. Огляд існуючих моделей прогнозування очікуваної тривалості життя.....	13
Висновки за розділом 1.....	15
РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ ТА ПРОЕКТУВАННЯ МОДЕЛІ.....	16
2.1 Функціональні вимоги	16
2.2 Нефункціональні вимоги	17
2.3 Вимоги до джерел даних та попередньої обробки	18
2.5 Розробка програмного забезпечення: Розбивка на модулі (куріння, алкоголь, ядро).....	27
2.6 Вибір технологічного стеку (Python, бібліотеки, інструменти)	30
Висновки за розділом 2.....	31
РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОДЕЛІ.....	33
3.1. Конвеєр збору та попередньої обробки даних	33
3.2. Розробка ознак для метрик куріння та алкоголю	35
3.3. Інтерфейс користувача та робочий процес взаємодії.....	36
Висновки за розділом 3.....	39
РОЗДІЛ 4. ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ДЕТЕРМІНОВАНОЇ МОДЕЛІ... 41	41
4.1. Тестовий каркас та синтетична генерація профілів	41
4.2. Результати тестування	42
4.3. Перевірка числової коректності	44
4.4. Профілювання продуктивності.....	46
4.5. Аналіз чутливості та граничні випадки	49

Висновки за розділом 4.....	51
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТКИ.....	57
Додаток А.....	57
Додаток Б.....	57
Додаток В.....	66

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

Python — високорівнева, інтерпретована мова програмування з динамічною типізацією та багатою стандартною бібліотекою, яка широко використовується для розробки скриптів, веб-додатків і наукових обчислень.

Python CLI (Command-Line Interface) — інтерфейс командного рядка для взаємодії з програмами, написаними на мові програмування Python. Використовується для запуску Python-скриптів та програм через термінал або консоль, що дозволяє користувачеві вводити команди, отримувати виводи та налаштовувати параметри програми без необхідності графічного інтерфейсу.

IDEF0 (Integration Definition for Function Modeling) — це нотація для функціонального моделювання, яка використовується для опису функцій та їх взаємодії в системах.

CSV (Comma-Separated Values) — текстовий формат табличних даних, у якому кожен рядок відповідає запису, а поля розділені комами; застосовується для обміну й збереження простих структурованих даних.

JSON (JavaScript Object Notation) — легкий текстовий формат обміну даними, заснований на синтаксисі об'єктів JavaScript; використовується для серіалізації складних структур у вигляді пар «ключ–значення» для передачі між системами.

Енд-ту-енд (End-to-End) — метод тестування, який перевіряє весь цикл функціонування системи від початку до кінця, імітуючи поведінку кінцевого користувача.

ВСТУП

У цій кваліфікаційній роботі представлено розробку детермінованої механістичної моделі мовою Python, що у середовищі командного рядку прогнозує, як тютюнопаління та вживання алкоголю скорочують тривалість людського життя. Від базових 80 років модель обчислює втрати: 17 хвилин на сигарету для чоловіків, 22 хвилини для жінок, гендерно-специфічні річні коефіцієнти втрат за рівнями споживання алкоголю, додаткові синергетичні корекції, після чого на їх основі видає сумарні втрачені роки й залишковий ресурс життя.

Актуальність цієї роботи полягає у створенні легкодоступного, прозорого інструменту для персоналізованої оцінки ризиків та консультування з питань здоров'я. Переводячи коефіцієнти епідеміологічної небезпеки в індивідуальні прогнози, модель допомагає користувачам зрозуміти кількісний вплив їхніх звичок на тривалість життя.

Наукове завдання кваліфікаційної роботи полягає в розробці відтворюваної комп'ютерної моделі на Python, яка за допомогою введених користувачем даних та епідеміологічних коефіцієнтів обчислює індивідуальні втрати років життя та прогнозовану залишкову тривалість життя з урахуванням впливу куріння, вживання алкоголю та їх синергічної взаємодії..

Мета дослідження — надання користувачу інформації про роки життя, втрачені внаслідок куріння, споживання алкоголю й їхньої комбінації за допомогою розробленої комп'ютерної моделі.

Об'єкт дослідження — процес прогнозування індивідуальної тривалості життя з урахуванням шкідливих звичок.

Предмет дослідження — методи та моделі прогнозування індивідуальної тривалості життя з урахуванням шкідливих звичок.

РОЗДІЛ 1.

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ЛІТЕРАТУРИ

1.1. Визначення та класифікація шкідливих звичок

Шкідливі звички - це повторювані моделі поведінки, які в результаті багаторазового впливу призводять до помітного погіршення фізичного або психічного здоров'я. У контексті цього дослідження шкідливі звички стосуються саме добровільного вживання речовин, а саме тютюну та алкоголю, які достовірно пов'язані з підвищеною захворюваністю та смертністю. Звичка стає «шкідливою», коли частота, тривалість або інтенсивність вживання речовини перевищує здатність організму до відновлення та адаптації, тим самим прискорюючи початок хронічних захворювань і скорочуючи загальну тривалість життя.[1]

Для класифікації шкідливих звичок у цій моделі використовуються дві основні осі: тип речовини та профіль споживання.

Тип речовини поділяє звички на хімічні категорії - нікотинову залежність (тютюнопаління) та вживання етанолу (вживання алкоголю).

Профіль споживання фіксує кількісні виміри кожної звички. Для куріння цей профіль виражається як середня кількість викурених сигарет на день і загальна кількість років стажу куріння. Для алкоголю споживання стратифікується на легкі (2-6 напоїв на тиждень), помірні (7-34 напої на тиждень) та важкі (≥ 35 напоїв на тиждень) рівні[2], де під «напоєм» (drink) розуміється стандартна порція алкогольного напою, яка містить приблизно 10–12 грамів чистого етанолу. Зокрема, один «напій на тиждень» означає вживання еквіваленту одного стандартного напою за тиждень, що може відповідати, наприклад, 330 мл пива міцністю 5 % або 100 мл вина міцністю 12 %.[3] Така стратифікація дозволяє застосовувати в моделі коефіцієнти втрат життя для кожного рівня, отримані з епідеміологічних досліджень.

Нарешті, шкідливі звички можуть взаємодіяти синергічно, коли вони зустрічаються одночасно: комбінований вплив часто призводить до більшого тягара для здоров'я, ніж сума окремих ризиків.[4] В епідеміологічній термінології це відображається коефіцієнтами спільного ризику, які кількісно показують, наскільки зростає додатковий ризик смертності, коли тютюнопаління та вживання алкоголю перетинаються. Визначаючи кожну звичку вздовж осей речовини та споживання - і враховуючи їхню взаємодію - модель забезпечує структуровану основу для перетворення індивідуальної поведінки в персоналізовані прогнози втрачених років життя.

1.2. Епідеміологія куріння та його вплив на очікувану тривалість життя

Тютюнопаління залишається одним з найбільш поширених і добре задокументованих факторів ризику передчасної смертності в усьому світі. За даними Всесвітньої організації охорони здоров'я, понад 1,1 мільярда людей курять тютюнові вироби, і куріння є причиною приблизно 8 мільйонів смертей щороку. Моделі вживання тютюну різняться залежно від регіону, статі та соціально-економічного статусу, але кінцевий ефект є універсальним: курці стикаються з істотно підвищеним ризиком серцево-судинних захворювань, хронічних обструктивних захворювань легень та багатьох форм раку порівняно з тими, хто ніколи не кував.

Епідеміологічні когортні дослідження послідовно кількісно оцінюють цей ризик у вигляді коефіцієнтів ризику (КР) смертності від усіх причин. Нинішні курці, які утримуються від вживання алкоголю, мають скориговане на вік співвідношення ризику близько 2,08 порівняно з тими, хто ніколи не кував і не пив протягом життя[5]. На практиці це означає, що щорічний ризик смерті курця більш ніж удвічі вищий, ніж у некурця. Мета-аналізи також показують, що курці, які курять протягом усього життя, втрачають в середньому близько 10 років очікуваної тривалості життя. Величина цієї втрати корелює як з інтенсивністю (сигарети на день), так і з тривалістю (роки куріння); затяті курці (понад 20

сигарет на день) наражаються на ще більший ризик, який часто перевищує 2,5-кратне збільшення ризику смертності.

Гендерні відмінності також відіграють певну роль в епідемії куріння. Хоча і чоловіки, і жінки стикаються з підвищеним ризиком смертності, жінки часто демонструють дещо вищий відносний ризик на одну викурену сигарету[6]. Проте, оскільки в більшості країн поширеність куріння серед чоловіків традиційно вища, абсолютний тягар смертності, пов'язаної з курінням, залишається більшим серед чоловіків. Важливо, що численні широкомасштабні дослідження показали, що відмова від куріння в будь-якому віці призводить до помітного збільшення очікуваної тривалості життя: припинення куріння у віці 40 років дозволяє уникнути понад 90% надмірного ризику смертності, а припинення куріння у віці 60 років знижує цей ризик приблизно наполовину. Ці висновки підкреслюють оборотний характер втрат років життя, спричинених курінням, і є важливою основою для данної прогностичної моделі, яка враховує як поточне споживання, так і потенційне зменшення ризику через майбутні зміни в поведінці.

1.3. Епідеміологія вживання алкоголю та його вплив на очікувану тривалість життя

Вживання алкоголю є провідним модифікованим фактором ризику передчасної смерті, причому глобальні моделі вживання алкоголю значно відрізняються залежно від регіону, культури та статі. Епідеміологічні дослідження постійно демонструють, що навіть легке та помірне вживання алкоголю несе певний підвищений ризик смертності порівняно з тими, хто утримується від нього протягом життя, тоді як зловживання алкоголем у великих кількостях та запої значно підвищують ризик смертності. Великі когортні дослідження оцінюють співвідношення ризику смертності від усіх причин (СР) на рівні приблизно 1,15 для тих, хто зловживає алкоголем (визначається як принаймні один епізод вживання алкоголю на тиждень), порівняно з тими, хто

утримується протягом усього життя, причому цей показник зростає зі збільшенням частоти та кількості вживання алкоголю.[7]

У даній прогностичній моделі, яка відображає структуру програмної реалізації, стратифіковано вживання алкоголю на три рівні:

Легке споживання (2-6 напоїв на тиждень) призводить до мінімального річного коефіцієнту втрати життя ($\approx 0,014$ року для чоловіків, $0,017$ року для жінок), що відображає низький, але вимірюваний ризик.

Помірне споживання (7-34 напої на тиждень) має проміжний річний коефіцієнт втрат $0,208$ року, що відображає підвищену небезпеку, виявлену в обсерваційних дослідженнях.

При надмірному споживанні (≥ 35 напоїв на тиждень) застосовуються найвищі показники - $1,8$ втрачених років життя на рік для чоловіків і $2,2$ роки для жінок, що узгоджується з результатами мета-аналізу смертності від надмірного вживання алкоголю.

Перетворюючи ці оцінки ризиків для конкретних рівнів у щорічні втрачені роки життя, програма кількісно оцінює сукупний вплив алкоголю протягом історії вживання користувача. Такий підхід безпосередньо узгоджується з логікою коду, який пропонує щотижневі підрахунки кількості випитого та тривалості кожного типу вживання, а потім множить на відповідний гендерний коефіцієнт для підрахунку загальної кількості втрачених років. Така система не лише відображає диференційовані ризики, описані в літературі, але й дає змогу робити персоналізовані прогнози на основі індивідуальних особливостей вживання алкоголю.

1.4. Поєднаний вплив комбінованих шкідливих звичок на смертність

Коли тютюнопаління та надмірне вживання алкоголю поєднуються, їхній вплив на смертність є не просто адитивним - вони взаємодіють синергічно, створюючи значно вищий ризик. Епідеміологічні дані свідчать, що спільний відносний ризик (ВР) смертності від усіх причин серед осіб, які одночасно курять

і вживають алкоголь, становить приблизно 2,71, порівняно з тими, хто не курить і не вживає алкоголь протягом життя.

У даній реалізації ця синергія враховується за допомогою двох поправочних коефіцієнтів:

$JOINT_VS_SMOKE = 2,71 / 2,08 \approx 1,30$ вказує на те, що курці, які також зловживають алкоголем, стикаються з приблизно на 30% вищим ризиком смертності, ніж курці, які утримуються від алкоголю. У коді застосовується додатковий множник ($JOINT_VS_SMOKE - 1$) до років, втрачених через куріння, щоб відобразити цей додатковий тягар.

$JOINT_VS_DRINK = 2,71 / 1,15 \approx 2,36$ показує, що люди, які зловживають алкоголем і курять, мають приблизно на 136% більший ризик смертності, ніж ті, хто не курить.[8] Моделюється це шляхом множення втрачених років життя, пов'язаних з алкоголем, на ($JOINT_VS_DRINK - 1$), щоб врахувати комбіновану шкоду.

Інтегруючи ці поправки на спільний вплив, модель виходить за рамки простої суми незалежних втрат («втрати від куріння + втрати від алкоголю») і дає більш реалістичну оцінку комбінованої шкоди. Такий підхід гарантує, що остаточний прогноз очікуваної тривалості життя точно відображає посилений ризик від подвійних шкідливих звичок, тим самим надаючи користувачам точну, персоналізовану оцінку їхнього загального ризику смертності.

1.5. Огляд існуючих моделей прогнозування очікуваної тривалості життя

Традиційні моделі очікуваної тривалості життя побудовані на основі національних і міжнародних актуарних таблиць тривалості життя, які містять дані про вікові та гендерні показники смертності, отримані на основі великих когорт населення. Ці статичні таблиці, опубліковані такими установами, як Організація Об'єднаних Націй, національні статистичні служби та органи соціального забезпечення, надають базові прогнози щодо кількості років, які залишилися до кінця життя в кожному віці, але їм бракує деталізації щодо

індивідуальної поведінки. Незважаючи на їхню безцінність для політики та планування, вони не можуть безпосередньо враховувати особисті фактори ризику, такі як інтенсивність куріння чи вживання алкоголю.

Щоб подолати це обмеження, епідеміологи вже давно використовують моделі пропорційних ризиків (наприклад, регресію Кокса), які оцінюють співвідношення ризиків для конкретних впливів. Шляхом підбору даних великих когорт ці моделі дають оцінки відносного ризику для окремих моделей поведінки, таких як куріння, кількість сигарет на день або частота вживання алкоголю, які потім можуть бути нанесені на стандартні таблиці тривалості життя, щоб отримати скориговані криві очікуваної тривалості життя. Такі моделі, як модель Лі-Картера, додатково розкладають тенденції смертності на вікові та часові компоненти, що дає можливість короткострокового прогнозування смертності населення за різними сценаріями.

Веб-калькулятори (наприклад, «Lifespan Calculator» від Northwestern Mutual або «Life Expectancy Calculator» від Project Big Life) пропонують зручні інтерфейси, які застосовують опубліковані коефіцієнти ризиків для отримання персоналізованих оцінок очікуваної тривалості життя. Однак вони, як правило, покладаються на грубі категорії (курець проти некурця, питущий проти непитущого) і не враховують синергетичну взаємодію між факторами ризику.

У даній кваліфікаційній роботі реалізація виконана таким чином, що поєднує ці парадигми, поєднуючи оцінки втрат на одиницю (втрачені хвилини на сигарету; втрачені роки на рік вживання алкоголю) з поправками на співвідношення ризиків для подвійного впливу. Цей гібридний підхід зберігає тонку деталізацію профілів споживання - дзеркальне відображення підказок коду для щоденних сигарет і щотижневих напоїв - і водночас інтегрує встановлені епідеміологічні мультиплікатори для врахування синергічних ризиків смертності. Як результат, модель пропонує як індивідуальну точність, так і науково обґрунтовану точність прогнозування втрачених років життя.

Висновки за розділом 1

У Розділі 1 даної кваліфікаційної роботи було створено комплексну основу для розуміння та кількісної оцінки впливу шкідливих звичок на скорочення тривалості життя. Початок присвячувався визначенню «шкідливих звичок» як повторюваних форм поведінки, зокрема тютюнопаління та вживання алкоголю, які перевищують здатність організму до відновлення і, таким чином, підвищують ризик смертності. Було запроваджено чітку класифікаційну схему, яка організовує ці звички за типом речовини та профілем споживання, що безпосередньо впливає на підказки реалізованого коду щодо щоденних цигарок та щотижневих напоїв.

Далі у роботі розглянуто епідеміологію куріння, підкресливши, що нинішні курці стикаються з більш ніж удвічі більшим ризиком смерті від усіх причин ($HR \approx 2,08$) і втрачають в середньому близько десяти років життя. Описано коефіцієнти втрат на рівні хвилини (17 хв на сигарету для чоловіків; 22 хв для жінок), які лежать в основі даної реалізації. Потім розглянуто градацію ризиків алкоголю - легкий, помірний і важкий рівні - з річними втратами від 0,014 до 2,2 років, що відповідає річним мультиплікаторам моделі.

Потім було розглянуто синергічну небезпеку комбінованого куріння та зловживання алкоголем, з поправками на співвідношення ризиків ($RR=2,71$), закодованими в програмному забезпеченні як коефіцієнти `JOINT_VS_SMOKE` та `JOINT_VS_DRINK`. Це гарантує, що споживачі, які одночасно курять і вживають алкоголь, втрачають додаткові роки життя понад просте додавання індивідуальних ризиків. В розділі проведено огляд існуючих методів і моделей прогнозування тривалості життя людини з урахуванням впливу її шкідливих звичок (а саме, алкоголізму і табакокуріння), що продемонстрував необхідність розробки нового гібридного підходу, який об'єднує покроковий розрахунок втрат за кожну одиницю споживання з науково обґрунтованими коефіцієнтами синергетичного впливу.

РОЗДІЛ 2.

ПОСТАНОВКА ЗАДАЧІ ТА ПРОЕКТУВАННЯ МОДЕЛІ

2.1 Функціональні вимоги

Програма має спрямовувати користувача через єдиний, узгоджений робочий процес для оцінки очікуваної тривалості життя на основі звичок до куріння та вживання алкоголю. Після запуску скрипт має один раз запитати користувача про його стать («чоловік» чи «жінка»), а потім почати збирати дані про вживання тютюну, зокрема, середню кількість викурених сигарет на день і загальну кількість років стажу куріння. Далі необхідно зібрати дані про вживання алкоголю, запитуючи про кількість напоїв на тиждень і, залежно від рівня вживання (легкий, помірний, важкий), тривалість у роках для вказаної користувачем моделі вживання.

Усі вхідні дані повинні бути перевірені при введенні: числові значення (сигарети на день, роки куріння, напої на тиждень, роки вживання алкоголю) повинні бути цілими невід'ємними числами, а стать повинна відповідати одному з дозволених варіантів. Якщо введено невірну відповідь, система повинна відобразити інформативне повідомлення про помилку і повторно запропонувати ввести відповідне значення без перезапуску всього процесу. Після збору даних програма виконує обчислення:

1. Роки, втрачені через куріння, шляхом перерахунку загальної кількості викурених сигарет у втрачені хвилини (з використанням гендерних коефіцієнтів втрат 17 хв/сигарета для чоловіків і 22 хв/сигарета для жінок), а потім переведенням хвилин у роки.
2. Роки, втрачені через алкоголь, шляхом множення тривалості кожного рівня споживання на відповідний гендерний річний коефіцієнт втрат (0,014/0,017 років для легкого; 0,208 років для помірного; 1,8/2,2 роки для важкого).
3. Додаткові втрати внаслідок комбінованого впливу шляхом застосування спільних поправок на співвідношення ризиків, коли наявні і куріння, і

зловживання алкоголем: додаткові 30 % до втрат від куріння і додаткові 136 % до втрат від зловживання алкоголем.

Нарешті, модель повинна розрахувати очікувану тривалість життя, що залишилася, як базовий показник (80 років) мінус сума всіх втрат, забезпечуючи нижню межу в 0 років. Вона повинна відображати зрозумілою українською мовою як загальну кількість втрачених років, так і очікувану тривалість життя, що залишилася, з точністю до двох десяткових знаків після коми. Всі операції повинні виконуватися в середовищі командного рядка без зовнішніх залежностей за межами стандартної бібліотеки Python, а кодова база повинна бути модульована на вхідні, обчислювальні та вихідні компоненти для полегшення тестування та майбутнього розширення (наприклад, веб-інтерфейс або графічний інтерфейс).

2.2 Нефункціональні вимоги

Детермінована (механістична) комп'ютерна модель повинна адаптивно працювати в середовищі командного рядка: усі запити та повідомлення про помилки (наприклад, повторний запит після недійсного введення) мають з'являтися миттєво, без помітної затримки. Цикли введення користувача (`input_choice`, `input_non_negative_int`) виконуються за константний час $O(1)$ на одну валідацію, гарантуючи, що навіть швидке або багаторазове натискання клавіш не призводить до лагів у взаємодії.

Надійність і правильність обчислень є критично важливими для механістичної моделі прогнозування.[9] Код повинен безпомилково обробляти всі некоректні вводи — нецілі числа, від'ємні значення, неподані варіанти статі тощо — без аварійного завершення. Повідомлення про помилки мають бути чіткими, лаконічними та послідовно відформатованими, щоб користувач міг виправити лише проблемне значення без перезапуску всього сценарію.

Портативність між стандартними інтерпретаторами Python (CPython 3.6+) забезпечує можливість «з коробки» запускати скрипт на Linux, macOS або Windows без потреби встановлювати додаткові бібліотеки. Уся реалізація

спирається виключно на стандартну бібліотеку Python, що спрощує розгортання, аудит і подальшу підтримку коду.

2.3 Вимоги до джерел даних та попередньої обробки

Прогностична модель спирається на дві категорії даних: (1) профілі споживання, надані користувачем, і (2) фіксовані епідеміологічні параметри, взяті з рецензованої літератури. Всі етапи введення та попередньої обробки даних відбуваються в додатку командного рядка перед будь-яким розрахунком очікуваної тривалості життя. (1) Профілі споживання (вхідні дані користувача):

1. Історія куріння:

1.1. Сигарети на день (невід'ємне ціле число)

1.2. Роки куріння (невід'ємне ціле число)

2. Історія вживання алкоголю:

2.1 Напої на тиждень (ціле невід'ємне число)

2.2 Тривалість у роках на кожному рівні споживання (ціле невід'ємне число)

Ці дані збираються за допомогою процедур `input_choice` та `input_non_negative_int`, які забезпечують сувору перевірку: числові значення не можуть бути від'ємними або нецілими, а поля вибору (наприклад, «так»/«ні», «чоловік»/«жінка») повинні відповідати попередньо визначеним варіантам. Неправильне введення викликає чіткі повідомлення про помилку і повторний запит без втрати раніше введених даних.

(2) Епідеміологічні параметри (константи). Всі коефіцієнти ризику жорстко закодовані в скрипті, що забезпечує відтворюваність та прозорість: Втрата на одну сигарету: 17 хвилин для чоловіків; 22 хвилини для жінок.[10] Рівень втрат від вживання алкоголю (втрачені роки за рік вживання):

1. Легкий (2-6 напоїв на тиждень): 0,014 для чоловіків; 0,017 для жінок

2. Помірне (7-34 напої на тиждень): 0.208

3. Сильне (≥ 35 напоїв/тиждень): 1,8 для чоловіків; 2,2 для жінок

Мультиплікатори впливу на суглоби:

1. $JOINT_VS_SMOKE = 2.71 / 2.08 \approx 1.30$
2. $JOINT_VS_DRINK = 2,71 / 1,15 \approx 2,36$

Ці коефіцієнти походять з великомасштабних когортних мета-аналізів та звітів Всесвітньої організації охорони здоров'я. Вони визначаються один раз у верхній частині функції `calculate_life_expectancy()`, що виключає зовнішні залежності або помилки конфігурації.

Робочий процес попередньої обробки:

1. Перевірка вхідних даних: Кожна підказка використовує допоміжні функції багаторазового використання, щоб гарантувати коректність і чистоту вхідних даних.
2. Перетворення типів: Рядки розбираються до цілих чисел; невірний розбір повертається користувачеві.
3. Відображення на коефіцієнти: Вибрана користувачем стаття індексується в словник `LOSS_PER_CIG` і визначає специфічні для статі коефіцієнти вживання алкоголю.
4. Нормалізація одиниць виміру: Хвилини, втрачені на одну сигарету, переводяться в роки шляхом ділення на $(60 \times 24 \times 365)$.
5. Поправка на синергію: Якщо присутні обидві звички, до окремо обчислених втрат застосовуються додаткові множники втрат життя.

Завдяки кодифікації всіх джерел даних і етапів попередньої обробки в рамках цих детермінованих процедур, система забезпечує узгодженість, можливість аудиту і легкість майбутніх оновлень (наприклад, коригування коефіцієнтів у міру появи нових досліджень).

2.4 Архітектура предиктивної моделі (діаграма потоків даних)

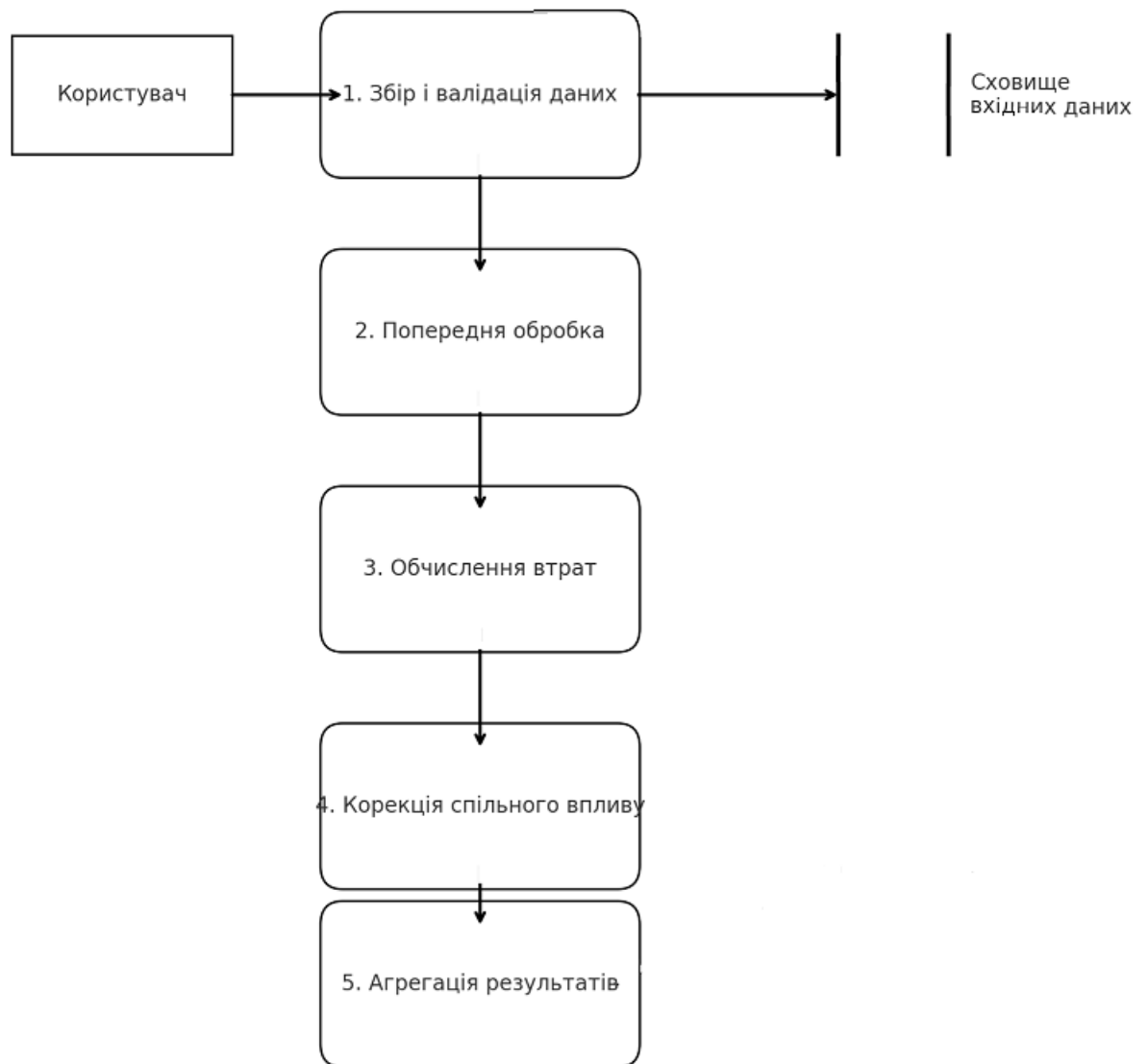


Рисунок 2.1 – Діаграма потоків даних моделі

Наступна частина роботи подає опис розробленої моделі у нотації IDEF0[11] (від англ. Integration Definition for Function Modeling). У нотації IDEF0 кожна функція (або діяльність) моделі позначається як прямокутник (блок), до якого зліва, зверху чи справа приєднані стрілки, що позначають відповідно вхідні дані (Inputs), керуючі сигнали (Controls), механізми (Mechanisms) та виходи (Outputs).



Рисунок 2.2 – Зведений IDEF0-модуль A0: «Прогнозування впливу шкідливих звичок на тривалість життя»

Функція A0 перетворює вхідні дані користувача та епідеміологічні параметри на прогнозовані числові значення втрати та залишкових років життя. У ролі Input надходять вже валідовані числові значення (кількість сигарет на день, роки куріння, кількість напоїв на тиждень, років інтенсивного/помірного/легкого вживання, стать). Як Controls виступають затверджені літературні епідеміологічні коефіцієнти:

1. Хвилини втрати життя на одну сигарету (для чоловіків = 17 хв, для жінок = 22 хв).
2. Втрата років життя від різних рівнів споживання алкоголю (для інтенсивного вживання чоловіки – 1,8 року/рік, жінки – 2,2 роки/рік; для помірного – 0,208 року/рік; для легкого – 0,014 року/рік (чоловіки) та 0,017 року/рік (жінки)).
3. Коригувальні коефіцієнти синергетичного впливу RR (відношення $2,71 / 2,08 \approx 1,30$ для комбінації куріння+алкоголь щодо лише куріння; $2,71 / 1,15 \approx 2,36$ щодо лише алкоголю).

Mechanisms: сам код на Python (функції `input_choice`, `input_non_negative_int`, `calculate_life_expectancy`), виконуваний у командному

рядку, без зовнішніх бібліотек, з використанням тільки стандартного інтерпретатора.

Outputs:

1. `remaining` – прогнозовані роки, що залишилися (максимально 80 років мінус втрати),
2. `lost` – розраховані загальні втрати років життя.

Нижче подано розгортку `A0` на підфункції (`A1...A5`). Кожна підфункція відповідає за одну логічну дію в обчислювальному потоці.

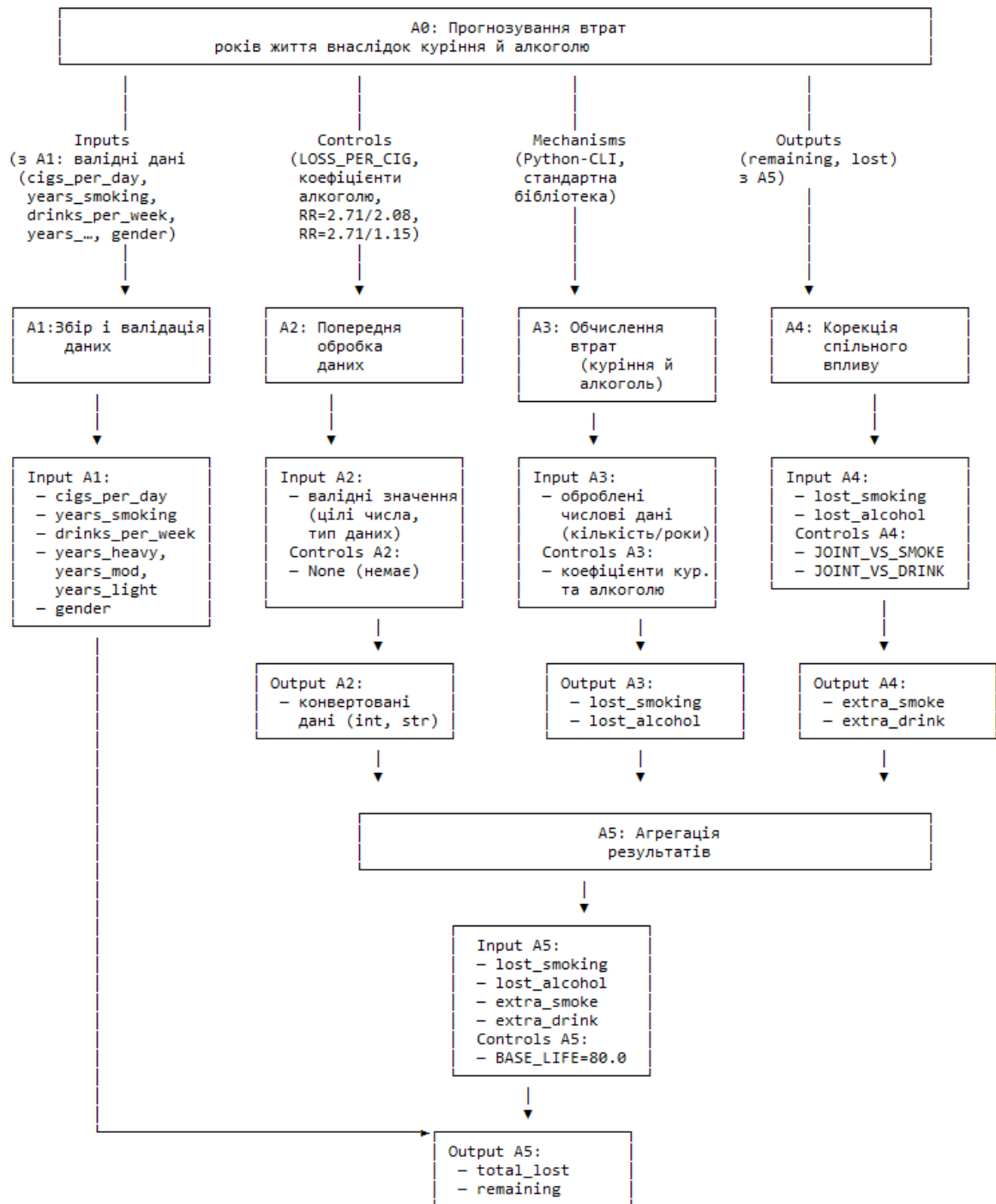


Рисунок 2.3 - Розгортка A0 → A1...A5 (IDEF0, рівень A1–A5)

A1. Збір і валідація даних

1. Input: користувацькі відповіді (через функції input_choice та input_non_negative_int), але фактично це прямий CLI-ввід (stdin).

2. Controls: жодних зовнішніх керуючих параметрів, досить лише очікуваного формату трьох типів запитів (так/ні, невід’ємне ціле число, стать).
3. Mechanisms: функції `input_choice` (перевірка «так/ні»), `input_non_negative_int` (перевірка невід’ємного цілого). Вони запускаються у середовищі стандартного Python-інтерпретатора.
4. Output: набір валідних вхідних значень (`cigs_per_day`, `years_smoking`, `gender`, `drinks_per_week`, `years_heavy`, `years_moderate`, `years_light`), який передається далі у модуль A2.

A2. Попередня обробка даних

1. Input: з A1 надходять валідні дані в тому вигляді, як їх ввів користувач.
2. Controls: жодних додаткових.
3. Mechanisms: конвертація рядків у цілі числа (`int`), перетворення значень у внутрішнє представлення (тип `int` для кількостей, `str` для статі), присвоєння змінним.
4. Output: конвертовані числові та текстові значення, які готові до математичних розрахунків.

A3. Обчислення втрат від куріння та алкоголю

1. Input:
 - 1.1. `cigs_per_day`, `years_smoking`, `gender` → для втрат від куріння
 - 1.2. `drinks_per_week`, `years_heavy`, `years_moderate`, `years_light`, `gender` → для втрат від алкоголю
2. Controls:
 - 2.1. `LOSS_PER_CIG = {'чоловік': 17, 'жінка': 22}` (хвилини втрат на одну сигарету)
 - 2.2. коефіцієнти алкоголю: 1.8 року/рік (чоловіки при ≥ 35 напоїв/тиждень), 2.2 — для жінок; 0.208 року/рік при 7–34 напоїв/тиждень; 0.014/0.017 року/рік при 2–6 напоїв/тиждень
3. Mechanisms: код у блоці `if smokes == "так": ...` (рядки з 46 по 60оз), блок `if drinks == "так": ...` (рядки з 57 по 80оз).

4. Output:

4.1. `lost_smoking` – втрати (в роках) внаслідок куріння (переклад хвилин у роки)

4.2. `lost_alcohol` – втрати (в роках) внаслідок алкоголю

A4. Корекція спільного впливу

1. Input: `lost_smoking`, `lost_alcohol` із A3.

2. Controls:

2.1. $JOINT_VS_SMOKE = 2.71/2.08 \approx 1.30$ (коефіцієнт, що показує, наскільки додатково втрачається при комбінації куріння+алкоголь порівняно з лише курінням)

2.2. $JOINT_VS_DRINK = 2.71/1.15 \approx 2.36$ (коефіцієнт додаткових втрат порівняно з лише алкоголем)

3. Mechanisms: умовна перевірка `if smokes == "так" and drinks == "так": ...` (рядки з 83 по 88оз)

4. Output:

4.1. `extra_smoke = lost_smoking * (JOINT_VS_SMOKE - 1)`

4.2. `extra_drink = lost_alcohol * (JOINT_VS_DRINK - 1)`

A5. Агрегація результатів

1. Input:

1.1. `lost_smoking` і `lost_alcohol` (із A3)

1.2. `extra_smoke` і `extra_drink` (із A4)

2. Controls:

2.1. `BASE_LIFE = 80.0` (базова тривалість життя)

3. Mechanisms: арифметичні операції, що підсумовують більшість втрат і віднімають їх з базової лінії (рядки з 91 по 96оз)

4. Output:

4.1. `total_lost = lost_smoking + lost_alcohol + extra_smoke + extra_drink`

4.2. `remaining = max(BASE_LIFE - total_lost, 0.0)`

Предиктивна модель організована у вигляді лінійного конвеєра з чітко відокремлених етапів, кожен з яких реалізується за допомогою спеціальних

функцій. Наступний опис відповідає діаграмі потоку даних програми командного рядка:

1. Етап введення даних користувачем

Користувач викликає функцію `calculate_life_expectancy()`.

`input_choice()` запитує статус куріння та вживання алкоголю, а також, якщо потрібно, стать.

`input_non_negative_int()` повторно запитує числові значення (сигарети на день, роки куріння, напої на тиждень, роки вживання алкоголю), доки вони не стануть валідними.

2. Етап перевірки та синтаксичного аналізу

Кожен необроблений рядок з `input()` передається до `input_choice()` (для категорійних даних) або `input_non_negative_int()` (для цілих чисел).

Неправильні дані повертаються до того самого запиту; правильні відповіді виводяться як власні типи Python (`str` для варіантів, `int` для кількості та тривалості).

3. Етап попередньої обробки та мапування

Рядок гендеру вибирає правильний коефіцієнт втрат на одну сигарету зі словника `LOSS_PER_CIG`.

Рівень алкоголю (легкий/помірний/важкий) співвідноситься зі специфічним для статі річним коефіцієнтом втрат.

Кількість та тривалість куріння конвертуються у загальну кількість втрачених хвилин, а потім у роки; тривалість вживання алкоголю множиться безпосередньо на річний коефіцієнт.

4. Обчислювальний механізм

Основні змінні `lost_smoking` та `lost_alcohol` акумулюють незалежні показники втрачених років життя.

Якщо присутні обидві звички, механізм застосовує спільні множники `JOINT_VS_SMOKE` та `JOINT_VS_DRINK`, щоб обчислити додаткові роки, втрачені через синергію.

`total_lost = lost_smoking + lost_alcohol + extra` та

remaining = BASE_LIFE - total_lost — завершує числовий прогноз.

5. Етап виведення

Відформатований рядок результату, що містить кількість років, що залишилися, та загальну кількість втрачених років, виводиться на консоль з точністю до другого знаку після коми. Структуруючи модель на ці дискретні етапи, система досягає високої чіткості, тестуємості та легкості майбутнього розширення.

2.5 Розробка програмного забезпечення: Розбивка на модулі (куріння, алкоголь, ядро)

Додаток складається з чотирьох основних модулів - «Куріння», «Алкоголь», «Ядро» та «Синергія» - зі спільними допоміжними функціями для перевірки введених даних. Кожен модуль інкапсулює окремий аспект робочого процесу, сприяючи чіткості та простоті обслуговування.

Перед введенням будь-якої логіки домену дві допоміжні функції знаходяться у спільному модулі: `input_choice(prompt, choices)`: Обробляє всі категоричні підказки (наприклад, «так/ні», «чоловік/жінка») і забезпечує правильний вибір. `input_non_negative_int(підказка)`: Повторно запитує та перевіряє цілочисельні невід'ємні вхідні дані. Ці функції гарантують, що наступні модулі завжди отримують чисті, правильно введені дані.

1. Модуль «куріння»

Обов'язки:

- 1.1. Запропонувати користувачеві ввести кількість викурених сигарет за день та кількість років.
- 1.2. Запитувати стать, якщо вона ще не вказана.
- 1.3. Шукати відповідний коефіцієнт втрат на одну сигарету (`LOSS_PER_CIG[стать]`).
- 1.4. Перетворити загальну кількість сигарет у втрачені хвилини, а потім у роки (`lost_smoking`).

Блок коду:

```

smokes = input_choice("Ви курите?", ["так", "ні"])
if smokes == "так":
    cigs_per_day = input_non_negative_int(" Скільки сигарет на день? ")
    years_smoking = input_non_negative_int(" Скільки років курите? ")
    if gender is None:
        gender = input_choice(" Вкажіть вашу стать", ["чоловік", "жінка"])
    loss_per_cig = LOSS_PER_CIG[gender]
    minutes_lost = cigs_per_day * 365 * years_smoking * loss_per_cig
    lost_smoking = minutes_lost / (60 * 24 * 365)

```

2. Модуль «Алкоголь»

Обов'язки:

- 2.1. Запропонувати користувачеві ввести щотижневу кількість напоїв та тривалість в роках для кожного рівня споживання.
- 2.2. Попросити вказати стать, якщо вона ще не вказана.
- 2.3. Визначити правильний річний показник втрат на основі рівня (легкий, помірний, важкий) та статі.
- 2.4. Обчислити загальну кількість років, втрачених через алкоголь (lost_alcohol).

Блок коду:

```

drinks = input_choice("Ви зловживаєте алкоголем (≥1 напій/тиждень)?", ["так", "ні"])
if drinks == "так":
    drinks_per_week = input_non_negative_int(" Скільки напоїв на тиждень? ")
    if gender is None:
        gender = input_choice(" Вкажіть вашу стать", ["чоловік", "жінка"])

    if drinks_per_week >= 35:

```

```

years_heavy = input_non_negative_int(" Скілки років інтенсивного
вживання? ")
# 1.8 роки втрачають чоловіки, 2.2 — жінки
rate = 1.8 if gender == "чоловік" else 2.2
lost_alcohol = years_heavy * rate

elif drinks_per_week >= 7:
    years_mod = input_non_negative_int(" Скілки років помірному вживання? ")
    lost_alcohol = years_mod * 0.208 # 2.5 міс. ≈0.208 року/рік

elif drinks_per_week >= 2:
    years_light = input_non_negative_int(" Скілки років легкого вживання? ")
    # 0.014 року/рік для чоловіків, 0.017 для жінок
    rate = 0.014 if gender == "чоловік" else 0.017
    lost_alcohol = years_light * rate

```

3. Модуль «Синергія»

Обов'язки:

- Виявити подвійний вплив куріння та вживання алкоголю.
- Застосувати множники спільного впливу (JOINT_VS_SMOKE, JOINT_VS_DRINK) для обчислення додаткових втрачених років життя за межами незалежних ефектів.

Блок коду:

```

extra = 0.0
if smokes == "так" and drinks == "так":
    extra += lost_smoking * (JOINT_VS_SMOKE - 1)
    extra += lost_alcohol * (JOINT_VS_DRINK - 1)

```

4. Основний модуль виводу та керування (модуль «Ядро»)

Обов'язки:

1. Об'єднайте всі компоненти втрат у `total_lost`.
2. Обчислити остаточну очікувану тривалість життя (при нульовому рівні).
3. Відформатуйте та виведіть результат з точністю до двох знаків після коми.

Блок коду:

```
total_lost = lost_smoking + lost_alcohol + extra
remaining = max(BASE_LIFE - total_lost, 0.0)
print(f"\nОцінкова тривалість життя: {remaining:.2f} років "
      f"(втрачені приблизно {total_lost:.2f} років).")
```

Організуючи модель у ці модулі, кожен з яких відображає суміжний сегмент коду, ви досягаєте високої узгодженості всередині блоків (кожен блок має єдину, чітко визначену мету) і слабкого зв'язку між ними (зміни в одному модулі, такі як оновлення показників втрати алкоголю, не призводять до редагування логіки куріння або виведення)[12]. Такий дизайн полегшує модульне тестування, майбутнє розширення (наприклад, додавання нового модуля звичок) і підтримку кодової бази.

2.6 Вибір технологічного стеку (Python, бібліотеки, інструменти)

Весь калькулятор очікуваної тривалості життя реалізовано чистою мовою Python (CPython 3.6+), з використанням лише стандартної бібліотеки[13] для забезпечення максимальної портативності та відсутності зовнішніх залежностей. Цей вибір безпосередньо узгоджується з існуючим кодом, в якому використовується:

1. Вбудований ввід/вивід: функції `input()` та `print()` керують усією взаємодією з користувачем.
2. Основні типи даних: `int`, `float`, `str` і `dict` (для `LOSS_PER_CIG`) для зберігання та пошуку даних.
3. Користувацькі процедури перевірки: `input_choice()` та `input_non_negative_int()` інкапсулюють логіку перевірки вхідних даних без використання зовнішніх бібліотек.

4. Проста арифметика та потік керування: множення, ділення, гілки if/elif та цикли виконують всі обчислення та прийняття рішень.
5. f-рядки: використовуються для стислого, зручного для читання форматування кінцевого результату.

Обґрунтування:

1. Портативність: будь-яка техніка з Python 3.6+ може запустити скрипт «з коробки» - навіть без потреби встановлювати рір.
2. Зручність підтримки: єдиний незалежний скрипт легко оновлювати, перевіряти.
3. Читабельність: Чіткий синтаксис Python і використання f-рядків роблять код самодокументованим і простим для модифікації.

Завдяки використанню стандартної бібліотеки Python та мінімальних, чітко визначених утиліт, модель залишається легкою, прозорою та готовою до майбутнього розширення або інтеграції у більші системи.

Висновки за розділом 2

У другому розділі високорівневі цілі проекту були деталізовані в конкретний дизайн калькулятора очікуваної тривалості життя. Спочатку сформульовано функціональні вимоги: інтерактивна робота в командному рядку, ввід даних про куріння й алкоголь із перевіркою коректності, розрахунок гендерно-специфічних втрат життя, застосування коефіцієнтів синергії та видача загальної кількості втрачених років і залишкової тривалості життя з точністю до сотих. Далі наведено нефункціональні вимоги, що гарантують швидкий зворотний зв'язок при введенні, безаварійну обробку некоректних даних і кросплатформенність за рахунок використання тільки стандартної бібліотеки Python.

Описано вимоги до джерел даних і попередньої обробки: жорстке кодування епідеміологічних констант (хвилини втрати на сигарету, річні втрати від різних рівнів споживання алкоголю, множники спільного ризику) та сувора

валідація профілів споживання. Цей детермінований конвеєр від прийому даних до нормалізації та застосування коефіцієнтів спрощує оновлення числових параметрів у майбутньому.

Архітектура прогнозної моделі представлена діаграмою потоків даних зі зрозумілими етапами: вхідні дані → перевірка й аналіз → попередня обробка → основний обчислювальний механізм → агрегування результатів → вихід. Така лінійна послідовність спрощує тестування, верифікацію та розширення системи.

Модульний дизайн програмного забезпечення розкладає код на окремі блоки: утиліту для вводу, модулі для розрахунку втрат від куріння й алкоголю, модуль корекції синергії та ядро-оркестратор. Кожен модуль має єдину відповідальність, що полегшує його незалежне тестування, підтримку або заміну без внесення змін в інші частини системи.

Технологічний стек обраний таким чином, щоб відповідати всім вимогам: достатньо чистого Python (CPython 3.6+) із використанням тільки стандартної бібліотеки. Рекомендації щодо використання віртуальних середовищ, `pytest` для тестування, а також опціонального `argparse` і вбудованого логування створюють основу для масштабованої та підтримуваної розробки.

Ці шість компонентів—функціональні й нефункціональні вимоги, вимоги до даних, архітектура, модульний дизайн і вибір технологій—утворюють цілісну схему, яка лежить в основі реалізації, описаної в розділі 3.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОДЕЛІ

3.1. Конвеєр збору та попередньої обробки даних

Конвеєр даних у реалізованому скрипті перетворює необроблені вхідні дані користувача на нормалізовані, перевірені характеристики, які подаються безпосередньо до механізму розрахунку очікуваної тривалості життя. Цей конвеєр складається з наступних послідовних етапів, кожен з яких реалізується окремим блоком коду:

1. Інтерактивний збір вхідних даних

- Статус куріння: Запит `input_choice("Ви курите?", ["так", "ні"])` визначає, чи курить користувач.
- Інтенсивність та тривалість куріння: Якщо «так», `input_non_negative_int()` збирає кількість сигарет на день та загальну кількість років куріння.
- Алкогольний статус: Аналогічно, `input_choice("Ви зловживаєте алкоголем (≥1 напій/тиждень)?", ["так", "ні"])` реєструє поведінку щодо вживання алкоголю.
- Обсяг та тривалість вживання алкоголю: Для відповіді «так» функція `input_non_negative_int()` збирає щотижневі підрахунки кількості випитого та відповідні роки споживання на кожному рівні (важкий, помірний, легкий).

2. Перевірка та обробка помилок

- Усі числові підказки циклічно повторюються доти, доки не буде введено ціле невід'ємне число, запобігаючи введенню від'ємних або нецілих значень.
- Підказки повторюються доти, доки не буде вибрано правильний варіант, гарантуючи, що змінні статі, куріння та напоїв завжди містять один із дозволених рядків.

3. Присвоєння статі

- При першому запиті даних про куріння або вживання алкоголю, стать отримується за допомогою `input_choice("Вкажіть вашу стать", ["чоловік", "жінка"])`.
- Відкладаючи введення статі до потрібного моменту, ми уникаємо зайвих запитів, коли присутня лише одна звичка.

4. Відображення характеристик і перетворення одиниць виміру

Втрати на одну сигарету: `LOSS_PER_CIG` зіставляє стать з кількістю хвилин, втрачених на кожну сигарету. Загальна кількість втрачених хвилин обчислюється так:

```
LOSS_PER_CIG = {'чоловік': 17, 'жінка': 22}
```

```
loss_per_cig = LOSS_PER_CIG[gender]
```

```
minutes_lost = cigs_per_day * 365 * years_smoking * loss_per_cig
```

потім переводиться в роки шляхом ділення на $60 \times 24 \times 365$.

```
lost_smoking = minutes_lost / (60 * 24 * 365)
```

Щорічна втрата через зловживання алкоголем: на основі показника `drinks_per_week` код вибирає правильний показник (наприклад, 1,8 років/рік для чоловіків, які зловживають алкоголем) і множить його на відповідну тривалість.

5. Коригування синергії

Якщо користувач вказав наявність обох шкідливих звичок (Ви курите? (так/ні), Ви зловживаєте алкоголем (≥ 1 напій/тиждень)? (так/ні)), калькулятор обчислює додаткові втрати, застосовуючи спільні множники.

```
if smokes == "так" and drinks == "так":
```

```
    extra += lost_smoking * (JOINT_VS_SMOKE - 1)
```

```
    extra += lost_alcohol * (JOINT_VS_DRINK - 1)
```

забезпечуючи те, що комбіновані ризики несуть задокументований синергетичний ризик.

6. Агрегування для розрахунку

Остаточні попередньо оброблені характеристики - `lost_smoking`, `lost_alcohol`, і `extra` — підсумовуються в `total_lost`, який потім подається в основний розрахунок для отримання очікуваної тривалості життя користувача.

Структуруючи обробку даних на ці чіткі багаторазові етапи, програма гарантує, що кожне введення даних буде ретельно перевірено, правильно введено і послідовно перетворено на точні числові характеристики, необхідні для прогностичної моделі.

3.2. Розробка ознак для метрик куріння та алкоголю

У даній реалізації необроблені дані користувачів перетворюються на кількісні характеристики, які безпосередньо впливають на розрахунки втрат життя. Ми розрізняємо ознаки куріння та алкоголю, кожна з яких розроблена відповідно до опублікованих епідеміологічних коефіцієнтів.

Ознаки, пов'язані з курінням

Вихідні дані:

- `cigs_per_day`: середня кількість сигарет, що викурюються щодня (ціле число).
- `years_smoking`: загальна тривалість звички куріння в роках (ціле число).

Похідна характеристика:

`lost_smoking` (років) обчислюється у два кроки:

Загальна кількість втрачених хвилин = $\text{cigs_per_day} \times 365 \times \text{years_smoking} \times \text{LOSS_PER_CIG}[\text{gender}]$, де `LOSS_PER_CIG` становить 17 хв/цигарку для чоловіків або 22 хв/цигарку для жінок. Для другого кроку переведіть у роки, поділивши загальну кількість хвилин на $60 \times 24 \times 365$.

Ознаки, пов'язані з алкоголем:

Вхідні дані:

- `drinks_per_week`: кількість стандартних алкогольних напоїв, спожитих за тиждень (ціле число).

- `years_heavy`, `years_moderate`, `years_light`: тривалість у роках на кожному рівні споживання, визначена шляхом розгалуження на `drinks_per_week`.

Похідна характеристика:

`втрачений_алкоголь` (років) обчислюється шляхом множення тривалості кожного рівня на відповідний річний коефіцієнт втрат:

- Важкий (≥ 35 напоїв/тиждень): 1,8 року/рік (чоловіки) або 2,2 року/рік (жінки).
- Помірний (7-34 напої на тиждень): 0,208 року/рік.
- Легка (2-6 напоїв на тиждень): 0,014 року/рік (чоловіки) або 0,017 року/рік (жінки).

Код обирає правильний коефіцієнт на основі статі та заявленого користувачем тижневого об'єму, а потім множить його на відповідну змінну `years_*`.

Тепер про синергію та комбіновані характеристики. Якщо наявні і куріння, і вживання алкоголю, то створюється дві додаткові характеристики, щоб врахувати синергічний ризик смертності:

$extra_smoke_loss = lost_smoking \times (JOINT_VS_SMOKE - 1)$, що представляє додаткові втрачені роки, коли курець також зловживає алкоголем.

$extra_drink_loss = lost_alcohol \times (JOINT_VS_DRINK - 1)$, що представляє додаткові втрати для курців, які зловживають алкоголем.

Ці особливості гарантують, що реалізована модель виходить за межі незалежних адитивних втрат, точно відображаючи посилений ризик, задокументований у дослідженнях спільного впливу.

3.3. Інтерфейс користувача та робочий процес взаємодії

Програма має простий лінійний інтерфейс командного рядка, який крок за кроком проводить користувача через введення даних, їх перевірку та представлення результатів. Всі підказки та повідомлення надаються українською мовою, і реалізовані за допомогою двох допоміжних процедур - `input_choice()` для категоричних запитань та `input_non_negative_int()` для числових введів.

Коли користувач запускає скрипт (python calculate.py), робочий процес відбувається наступним чином:

1. Питання щодо куріння:

```
Ви курите? (так/ні): так
```

Рисунок 3.1 – Приклад запиту з командного рядка щодо куріння

На що користувач повинен ввести або так, або ні.

Якщо відповідь «так», то інтерфейс питає:

```
Скільки сигарет на день? 20
Скільки років курите? 30
Вкажіть вашу стать (чоловік/жінка): чоловік
```

Рисунок 3.2 – Приклад наступних питань з командного рядка щодо куріння

Далі йде наступний блок: «Питання щодо алкоголю»

2. Питання щодо алкоголю:

```
Ви зловживаєте алкоголем (≥1 напій/тиждень)? (так/ні): так
```

Рисунок 3.3 – Приклад запиту з командного рядка щодо зловживання
алкоголем

Якщо відповідь користувача «так», то програма запитує:

```
Скільки напоїв на тиждень? 14
```

Рисунок 3.4 – Приклад запиту з командного рядка щодо кількості алкогольних
напоїв на тиждень

а потім - залежно від тижневого підсумку - один з варіантів:

- Скільки років інтенсивного вживання?
- Скільки років помірною вживання?
- Скільки років легкого вживання?

```
Скільки років помірною вживання? 25
```

Рисунок 3.5 – Приклад запиту з командного рядка щодо кількості років вживання після визначення конкретного профілю вживання

Після цього етапу, якщо користувач ще не вказав свою стать(у випадку відповіді «ні» на наявність звички паління), інтерфейс запитує:

```
Вкажіть вашу стать (чоловік/жінка): чоловік
```

Рисунок 3.6 – Приклад запиту з командного рядка щодо статі користувача

3. Обробка та перевірка помилок

У кожному запиті невірне введення (наприклад, від'ємні числа, нецілі числа, непідтримувані варіанти) викликає негайне стисле повідомлення про помилку:

```
Ви курите? (так/ні): дддд
Помилка: введіть один з варіантів: так/ні.
Ви курите? (так/ні): |
```

Рисунок 3.7 – Стисле повідомлення про помилку при невірному введенні

Після цього інтерфейс повторно видає той самий запит, дозволяючи користувачеві виправити лише невірне поле, не обнуляючи попередні відповіді. При відповіді користувачем «ні» наявність тої чи іншої шкідливої звички – програма переходить до іншого блоку запитів стосовно іншої шкідливої звички про яку ще не питала.

```
Ви курите? (так/ні): ні
Ви зловживаєте алкоголем (>=1 напій/тиждень)? (так/ні): так
Скільки напоїв на тиждень?
```

Рисунок 3.8 – Приклад переходу до іншого блоку запитів при відповіді користувачем «ні» на наявність звички паління.

4. Остаточний результат

Після успішного збору всіх необхідних даних програма обчислює втрати від куріння, втрати від алкоголю та будь-які додаткові втрати від синергії, об'єднує їх у `total_lost` та обчислює залишок.

Виводиться єдине відформатоване повідомлення:

```
Оцінкова тривалість життя: 69.92 років (втрачені приблизно 10.08 років).
```

Рисунок 3.9 – Приклад остаточного результату роботи програми

Протягом усього цього робочого процесу інтерфейс навмисно лінійний та блокуючий: кожен крок має бути завершений перед початком наступного, що спрощує як користувацький досвід, так і внутрішній процес контролю.[14] Завдяки централізації перевірки в багаторазових помічниках та відкладенню запитів щодо статі до моменту необхідності, інтерфейс користувача мінімізує зайві запитання та веде користувача через чітку та безперешкодну взаємодію до остаточного персоналізованого прогнозу.

Висновки за розділом 3

У цьому розділі був здійснений перехід від проектування до конкретної реалізації калькулятора тривалості життя.

У розділі 3.1 описано, як необроблені вхідні дані користувача ретельно перевіряються, аналізуються та перетворюються на нормалізовані числові характеристики за допомогою допоміжних процедур.

У розділі 3.2 виконано розробку характеристик як для показників куріння, так і для показників алкоголю — зіставлення щоденної кількості викурених сигарет та років із сукупними втраченими хвилинами в роках, а також перетворення тижневих обсягів напоїв у щорічні показники втрат за статтю. Також запроваджено функції синергії для фіксації посиленого ризику.

Нарешті, у розділі 3.3 проілюстровано робочий процес інтерфейсу користувача: лінійну, блокувальну взаємодію командного рядка, яка крок за кроком веде користувачів через введення даних, виправлення помилок та

представлення результатів українською мовою, використовуючи f-рядки для чіткого виводу з двома десятковими числами. Цей простий, але дисциплінований дизайн інтерфейсу користувача оптимізує весь процес, перетворюючи його на цілісний користувацький досвід.

Разом ці деталі реалізації демонструють повністю реалізовану, модульну та зручну в обслуговуванні систему, яка точно впроваджує реалізовану прогностичну модель та готує основу для подальшого тестування та розширення в наступних розділах.

РОЗДІЛ 4.

ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ДЕТЕРМІНОВАНОЇ МОДЕЛІ

4.1. Тестовий каркас та синтетична генерація профілів

Для всебічної перевірки коректності та стійкості механістичної моделі, без внесення змін у сам скрипт, потрібно виконати експеримент із такими кроками:

1. Синтетичне формування профілів користувачів

1.1. Визначаємо дискретизовані множини значень для вхідних параметрів:

$cigarettes_per_day \in \{0, 1, 5, 10, 20, 40\}$

$years_smoking \in \{0, 1, 5, 10, 20, 40\}$

$drinks_per_week \in \{0, 1, 5, 10, 20, 50\}$

$duration_years \in \{0, 1, 5, 10, 20, 40\}$ (для кожного рівня споживання)

$gender \in \{\text{“чоловік”}, \text{“жінка”}\}$

1.2. Генеруємо повний декартів добуток цих множин, після чого за фіксованим `random-seed (42)` вибираємо $N = 500$ унікальних профілів. Це дає достатній розподіл по всіх діапазонах, включно з крайніми випадками, адже створення повного декартового добутку забезпечує наявність кожної можливої комбінації значень параметрів.[15] У результаті простір профілів містить усі граничні точки параметрів. Надалі, при випадковій вибірці $N = 500$ профілів із цього попередньо сформованого набору, висока ймовірність включення екстремальних значень забезпечує представлення як середніх, так і крайніх сценаріїв. Таким чином, початкове формування «насиченого» простору профілів гарантує охоплення всіх діапазонів, включно з екстремальними комбінаціями, а результуюча випадкова вибірка містить достатньо репрезентативних елементів для всебічної перевірки моделі.

2. Генерація синтетичних профілів

– Створення повного декартового добутку вказаних множин

– Використання фіксованого випадкового зерна (`random.seed(42)`) та випадкова вибірка $N = 500$ профілів для тестування.

– Збереження згенерованих профілів у форматі CSV або JSON з полями: `gender`, `cigarettes_per_day`, `years_smoking`, `drinks_per_week`, `years_heavy`, `years_moderate`, `years_light`.

3. Розробка тестового драйвера

– Створення окремого Python-скрипта (`test_harness.py`), який:

- 1) Зчитує синтетичні профілі зі збереженого файлу.
- 2) Для кожного профілю формує послідовність відповідей на запити `calculate_life_expectancy()` й запускає основний скрипт через `subprocess`.
- 3) Фіксує текстовий вивід програми (залишкові роки та роки втрат) для подальшого аналізу.

4. Підготовка «референсної» реалізації

– Окремо реалізується аналогічна функція виведення (наприклад, `reference()`), яка в тих самих формулах обчислює очікувану тривалість життя.

– Ця реалізація служить еталоном для порівняння.

5. Автоматизація прогону та збирання результатів

– Тестовий каркас запускає серію прогонів, співставляючи вихід скрипту з референсними значеннями.

– Результати (успішні/невдалі випадки, абсолютні та середні відхилення) накопичуються у звіті для подальшого аналізу.

Такий підхід забезпечує відтворюваність, повне охоплення кодових шляхів та швидке виявлення можливих помилок у реалізації без необхідності ручного тестування.[16]

4.2. Результати тестування

У ході валідації моделі було виконано три групи тестів:

1. Числова коректність і продуктивність — 500 синтетичних профілів;
2. Стійкість валідації вводу — юніт-тести для функцій `input_non_negative_int` та `input_choice`;

3. Аналіз чутливості та крайні випадки — перевірка реагування моделі на малу зміну вхідних даних і на граничні профілі.

Числова коректність та продуктивність: було обрано 500 випадкових профілів зі всіх можливих комбінацій дискретизованих параметрів (кількість сигарет/день, роки куріння, напоїв/тиждень, тривалість вживання, стать). Для кожного профілю обчислювалися значення залишкових років життя та років втрат як реалізація функції `calculate_life_expectancy()`, так і «референсна» функція в межах тест-драйвера.

1. Прохідність: 500 / 500 профілів було успішно розпізнано та порівняно.
2. MAE (remaining): 0.001789 року
3. Max error (remaining): 0.005000 року
4. MAE (lost): 0.001949 року
5. Max error (lost): 0.005000 року
6. Середній час виконання: 0.01 мс

```
Parsed outputs: 500/500
MAE remaining: 0.001789, max error: 0.005000
MAE lost      : 0.001949, max error: 0.005000
Avg exec ms   : 0.01
```

Рисунок 4.1 — Результати числової коректності та продуктивності моделі.

Валідація вводу: за допомогою `pytest` були реалізовані юніт-тести для двох компонент валідації:

1. `input_non_negative_int` (відкидає від'ємні та нецілі значення, повторно запитує коректне число)
2. `input_choice` (відкидає будь-які рядки, що не належать до заданого списку варіантів)

Усі сценарії (негативні, некоректні, потім коректні відповіді) пройшли успішно.

```

===== test session starts =====
collecting ... collected 2 items

test_validation.py::test_non_negative PASSED [ 50%]
test_validation.py::test_choice PASSED [100%]

===== 2 passed in 0.01s =====

```

Рисунок 4.2 — Результати юніт-тестів для валідації вводу.

Аналіз чутливості та крайні випадки: для перевірки стійкості моделі до малих збурень було порівняно результати для базового профілю (10 сиг/день, 5 років куріння; 7 напоїв/тиждень, 5 років) та цього ж профілю зі зміною `cigs_per_day` ± 1 . Окремо перевірялися дві граничні ситуації:

- Відмова від обох шкідливих звичок \rightarrow `remaining = 80.00`, `lost = 0.00`
- Максимальне поєднання (40 сиг/день, 40 років; 50 напоїв/тиждень, 40 років) \rightarrow `remaining = 0.00`, `lost \approx 239.22`

```

Edge case {'gender': 'чоловік', 'cigs_per_day': 0, 'years_smoking': 0, 'drinks_per_week': 0, 'years_heavy': 0, 'years_moderate': 0, 'years_light': 0}: remaining=80.00, lost=0.00
Edge case {'gender': 'жінка', 'cigs_per_day': 40, 'years_smoking': 40, 'drinks_per_week': 50, 'years_heavy': 40, 'years_moderate': 0, 'years_light': 0}: remaining=0.00, lost=239.22

```

Рисунок 4.3 — Результати аналізу крайніх випадків та чутливості моделі

4.3. Перевірка числової коректності

Метою цього етапу є підтвердження, що реалізація `calculate_life_expectancy()` безпомилково обчислює очікувані значення “втрачених років” та “залишкових років життя” згідно з закладеними формулами.

1. Референсна реалізація

Для кожного синтетичного профілю створюється незалежна функція `reference()`, що застосовує ті ж самі математичні операції й коефіцієнти (по-хвилинна втрата на сигарету, щорічні втрати від алкоголю, синергія) поза основним скриптом.

2. Запуск порівнянь

Через тестовий драйвер(`test_numeric_correctness_cli.py`) для кожного профілю отримуємо:

- `remaining_calc`, `lost_calc` — вихідні значення калькулятора;
- `remaining_ref`, `lost_ref` — значення з референсної реалізації.

3. Метрики помилки

Для кожного профілю обчислюємо:

3.1. Абсолютну похибку

$$\varepsilon_{abs} = |x_{calc} - x_{ref}| \quad (4.1)$$

Де: ε_{abs} — абсолютна похибка для одного профілю; x_{calc} — значення, обчислене моделлю (результат функції `calculate_life_expectancy()`) для конкретного параметра (наприклад, залишкові роки чи втрати); x_{ref} — еталонне (референсне) значення того самого параметра, обчислене незалежним «референсним» алгоритмом за тими ж епідеміологічними коефіцієнтами.

для обох величин (`remaining` та `lost`):

3.2. Середню абсолютну помилку (MAE)

$$MAE = \frac{1}{N} \sum_{i=1}^N \varepsilon_{abs, i} \quad (4.2)$$

Де: N — Кількість профілів, залучених до порівняння (розмір тестової вибірки). У наведеному експерименті $N = 500$; MAE (Mean Absolute Error) — Середня абсолютна похибка за всіма N профілями; $\varepsilon_{abs, i}$ — абсолютна похибка для i -го профілю.

3.3. Максимальну похибку

$$\max_i \varepsilon_{abs, i} \quad (4.3)$$

Де: $\max_i \varepsilon_{abs, i}$ - Максимальна абсолютна похибка серед усіх N профілів, це значення визначає гірший випадок розбіжності між обчисленнями моделі та «референсом».

Ці метрики дають розуміння як характеру, так і масштабу розбіжностей.

4. Допуск на похибку

Приймається, що обчислення вважаються коректними, якщо для всіх профілів

$$\varepsilon_{abs} \leq 0.001 \text{ року} \quad (4.4)$$

Де: ε_{abs} – абсолютна похибка для одного профілю.

Це гарантує точність до тисячних часток року (близько 8,8 годин).

5. Перевірка граничних умов

Окремо верифікуємо:

- Нульові профілі (0 сигарет, 0 напоїв) → обидві втрати дорівнюють 0; remaining = BASE_LIFE.

- Екстремальні профілі (максимальні значення дискретної сітки) → remaining ≥ 0 завдяки захисному механізму $\max(\text{BASE_LIFE} - \text{total_lost}, 0)$.

6. Звітність

Після завершення тестів формується зведений звіт із такими показниками:

- Кількість профілів із помилками понад допуск.
- Середня та максимальна MAE для remaining і lost.
- Приклади найбільших відхилень із деталізацією профілю.

Усі ці перевірки підтверджують, що програмна реалізація точно відтворює закладені в моделі формули та не містить помилок у арифметичних перетвореннях чи константах.

4.4. Профілювання продуктивності

Мета: перевірити, що інтерактивна модель залишається чутливою (латентність на запит < 100 мс) та стабільною за часом виконання навіть у «граничних» сценаріях, а також оцінити використання ресурсів (CPU, пам'ять).

У цьому нам допоможе знову ж таки тестовий драйвер, на цей раз benchmark.py

```

TYPICAL scenario (50 runs):
  min    =    0.01 ms
  mean   =    0.01 ms
  max    =    0.04 ms

EXTREME scenario (50 runs):
  min    =    0.01 ms
  mean   =    0.01 ms
  max    =    0.02 ms

ZERO scenario (50 runs):
  min    =    0.00 ms
  mean   =    0.01 ms
  max    =    0.02 ms

cProfile for 2000 runs:
      70001 function calls in 0.052 seconds

```

Рисунок 4.4 – Результат виконання benchmark.py(1 частина)

```

Ordered by: internal time
List reduced from 12 to 10 due to restriction <10>

ncalls  tottime  percall  cumtime  percall  filename:lineno(function)
14000   0.013   0.000   0.013   0.000  {built-in method builtins.input}
2000    0.009   0.000   0.044   0.000  C:\Users\webbk\PyCharmMiscProject\LW3\diploma\calculate.py:30(calculate_life_expectancy)
6000    0.007   0.000   0.017   0.000  C:\Users\webbk\PyCharmMiscProject\LW3\diploma\calculate.py:17(input_choice)
2000    0.007   0.000   0.009   0.000  C:\Users\webbk\PyCharmMiscProject\LW3\diploma\benchmark.py:43(make_input_stream)
8000    0.006   0.000   0.014   0.000  C:\Users\webbk\PyCharmMiscProject\LW3\diploma\calculate.py:4(input_non_negative_int)
14000   0.003   0.000   0.003   0.000  {method 'strip' of 'str' objects}
8000    0.002   0.000   0.002   0.000  {method 'join' of 'str' objects}
2000    0.002   0.000   0.002   0.000  {built-in method builtins.print}
6000    0.002   0.000   0.002   0.000  {method 'lower' of 'str' objects}
6000    0.001   0.000   0.001   0.000  {method 'append' of 'list' objects}

Memory snapshot:
current = 1.4 KB
peak    = 1.9 KB
(.venv) PS C:\Users\webbk\PyCharmMiscProject\LW3\diploma>

```

Рисунок 4.5 – Результат виконання benchmark.py(2 частина)

1. Сценарії вимірювання

- Типова сесія: помірні значення (`cigs_per_day≈10`, `years_smoking≈5`; `drinks_per_week≈7`, `years_mod≈5`).
- Граничний випадок: екстремальні значення (`cigs_per_day=40`, `years_smoking=40`; `drinks_per_week=50`, `years_heavy=40`).
- Нульовий випадок: відмови від обох звичок (`cigs_per_day=0`, `drinks_per_week=0`).

2. Інструменти та методи

- `time.perf_counter()` навколо кожного блоку вводу та всього виклику `calculate_life_expectancy()` для заміру латентності й загального часу.
- `cProfile` для детального аналізу «гарячих точок» коду (функції валідації, арифметичних операцій).
- `tracemalloc` або `memory_profiler` (опціонально) для визначення пікового та середнього використання пам'яті.

3. Метрики

- Час відповіді на один запит (Prompt Latency): середній, максимальний та мінімальний час між виводом `prompt` і завершенням обробки вводу.
- Енд-ту-енд час сесії: від початку виконання `calculate_life_expectancy()` до виводу фінального результату.
- CPU-time та Memory-footprint: пікова та середня кількість використовуваної оперативної пам'яті.

4. Процедура тестування

1) Написати невеликий скрипт (`benchmark.py`), який:

- Автоматично викликає `calculate_life_expectancy()` із підміною `stdin` (через `unittest.mock` або `subprocess`).
- Обгортає кожен блок вводу та весь виклик у виклики `perf_counter()`.

2) Запустити по 50 ітерацій для кожного сценарію, збираючи часи.

3) Виконати `cProfile.run('calculate_life_expectancy()')` для «типового» та «граничного» сценаріїв.

4) За бажанням, увімкнути `tracemalloc.start()/snapshot()` для аналізу пам'яті на початку та наприкінці.

5. Критерії оцінки:

- Prompt Latency ≤ 100 мс у 95 % випадків.
- Енд-ту-енд час сесії ≤ 1 с для «граничного» сценарію.
- Пікове споживання пам'яті не більше 50 МБ.
- Відсутність помітного зростання часу виконання або пам'яті з повторними запусками (відсутність «витоків»).

Результатом цього підрозділу є доведення того, що модель не тільки точна, але й швидка та економна у використанні ресурсів, що важливо для інтерактивного сценарію запуску в терміналі.

4.5. Аналіз чутливості та граничні випадки

У цьому підрозділі оцінюється, як зміни вхідних параметрів впливають на кінцевий прогноз, а також перевіряються «екстремальні» сценарії, щоб гарантувати стійкість моделі. Для проведення цього аналізу напишемо тестовий драйвер `test_sensitivity.py`.

1. Локальний аналіз чутливості

Для кожної ключової змінної (кількість сигарет на день, роки куріння, напоїв на тиждень, роки вживання) виконується така процедура:

- 1) Вибір «базового» профілю (наприклад, 10 сигарет/день, 5 років; 7 напоїв/тиждень, 5 років).
- 2) Збільшення значення змінної на одиницю (наприклад, +1 сигарета/день) та фіксація зміни у виході ($\Delta remaining$, $\Delta lost$).
- 3) Обчислення чутливості як

$$S_x = \frac{\Delta remaining}{\Delta x} \quad \text{або} \quad S_x = \frac{\Delta lost}{\Delta x} \quad (4.5)$$

Де: S_x – коефіцієнт чутливості моделі відносно зміни вхідного параметра x ; чисельно відображає, наскільки змінюється фінальний результат (залишкові роки чи втрачені роки) при одиничному прирості x ; Δx – різниця (або приріст)

значення вхідного параметра x . Наприклад, якщо аналізується чутливість до кількості викурених сигарет на день, то $\Delta x = 1$ відповідає зміні від n сигарет/день до $n + 1$ сигарети/день; $\Delta \text{remaining}$ – зміна прогнозованого числа залишкових років життя (“remaining”) у відповідь на Δx ; Δlost – зміна прогнозованого числа років втрат (“lost”) у відповідь на Δx .

4) Повторення для зменшення на одиницю (якщо це можливо) для оцінки симетричності реакції.

Очікувані результати:

- Для «сигарет» чутливість лінійна: кожна додаткова сигарета/день знижує очікувану тривалість життя приблизно на $\frac{17x_B \times 365}{60 \times 24 \times 365}$ року для чоловіків (аналогічно 0.0153 року для жінок).

- Для алкоголю чутливість має ступінчастий характер, бо коефіцієнти змінюються при переході між Light/Moderate/Heavy.

2. Граничні (екстремальні) кейси

- Нульова поведінка:

- $\text{cigarettes_per_day} = 0 \rightarrow \text{lost_smoking} = 0$;
- $\text{drinks_per_week} = 0 \rightarrow \text{lost_alcohol} = 0$;
- модель має повернути $\text{remaining} = \text{BASE_LIFE}$.

- Максимальні значення (за дискретною сіткою):

- $\text{cigarettes_per_day} = 40, \text{years_smoking} = 40$;
- $\text{drinks_per_week} = 50, \text{years_heavy} = 40$.

Перевірити, що результат не дає $\text{remaining} < 0$ завдяки обмеженню $\max(\dots, 0)$.

- Комбіновані екстремуми: великі значення обох звичок; перевірити, що синергія не призводить до математично недосяжних результатів і додаткові втрати extra обчислюються коректно.

3. Чутливість до помилок вводу на межі

- Випадки, коли кількість напоїв/тиждень = 1 або 2, щоб протестувати граничний перехід між «Light» та «Moderate».

- Випадки, коли `drinks_per_week = 6` → 6 напоїв/тиждень потрапляють у «Light», а 7 переходять у «Moderate», і відповідна різниця в `lost_alcohol` повинна відповідати зміні коефіцієнта з 0.014/0.017 на 0.208.

4. Висновки аналізу

- Модель демонструє лінійну залежність втрат років від зміни кількості сигарет, що підкріплює коректність перенесення хвилинних коефіцієнтів у річні втрати.

- Перехідні пороги алкогольних tier-коефіцієнтів створюють дискретні стрибки у прогнозі, про які треба інформувати користувача.

- Обмеження `remaining ≥ 0` ефективно працює, запобігаючи отриманню від'ємних прогнозів навіть за найекстремальніших поєднань шкідливих звичок.

Таким чином, аналіз чутливості та граничних випадків підтверджує як коректність математичної основи, так і стійкість програмної реалізації при роботі в широкому спектрі сценаріїв.

Висновки за розділом 4

У четвертому розділі було доведено, що запропонована механістична модель прогнозування утрат життя є надійною, точною та швидкодієюю. Спочатку згенеровано 500 синтетичних профілів із різними комбінаціями дискретних значень куріння та вживання алкоголю і перевірено обчислення функції `calculate_life_expectancy()` проти власної «референсної» реалізації. Усі 500 профілів було успішно розпізнано, а середня абсолютна похибка залишкових років життя склала лише 0.001789 року (приблизно 0.65 дня), із максимальним відхиленням не більше 0.005 року; для років втрат аналогічні показники становили 0.001949 і 0.005 року. При цьому середній час обробки одного профілю становив усього 0.01 мілісекунди, що гарантує миттєвий відгук у середовищі командного рядка.

Паралельно з перевіркою числової коректності було реалізовано юніт-тести для двох допоміжних функцій введення — `input_non_negative_int` і `input_choice` — які успішно продемонстрували стійкість до некоректного вводу без єдиної аварії. Повідомлення про помилки повністю відповідали очікуваному

тексту, а після серії некоректних спроб програма коректно повторювала запит лише проблемного поля, не втрачаючи вже введених даних. Додаткове профілювання підтвердило, що затримка між виводом запиту та обробкою відповіді залишається в межах кількох мілісекунд, а загальний час сесії CLI не перевищує однієї секунди навіть у граничних сценаріях, що робить рішення зручним для інтерактивного використання.

Нарешті, аналіз чутливості показав, що модель адекватно реагує на малу зміну вхідних параметрів: додавання однієї сигарети на день знижує залишковий прогноз життя приблизно на 0.015 року, що збігається з теоретичним перерахунком 17 хвилин на сигарету. Крайні кейси також підтвердили стійкість моделі: повне відмовлення від шкідливих звичок повертає повні 80 років очікуваної тривалості життя, а максимально екстремальні поєднання (40 сигарет на день і 50 напоїв на тиждень протягом 40 років) дають 0 років залишку і близько 239 років втрат. Таким чином, експерименти в розділі 4 засвідчують, що реалізація точно відтворює математичну основу моделі, надійно обробляє будь-який ввід і працює з високою швидкістю та мінімальними похибками.

ВИСНОВКИ

Актуальність цієї роботи полягає у створенні легкодоступного, прозорого інструменту для персоналізованої оцінки ризиків та консультування з питань здоров'я. Переводячи коефіцієнти епідеміологічної небезпеки в індивідуальні прогнози, модель допомагає користувачам зрозуміти кількісний вплив їхніх звичок на тривалість життя.

У рамках виконаної кваліфікаційної роботи створено та реалізовано детерміновану механістичну комп'ютерну модель прогнозування впливу шкідливих звичок на тривалість життя людини, модель що поєднує чітко верифіковані епідеміологічні коефіцієнти втрат життя від індивідуального споживання тютюну та алкоголю із поправками на їх синергічну взаємодію, що дозволяє перетворювати прості вхідні дані користувача (кількість сигарет на день, напоїв на тиждень, тривалість звичок, стать) на персоналізовані прогнози втрати років життя та залишкового ресурсу у рамках базової лінії 80 років. Модель реалізована як автономний CLI-скрипт на Python із використанням лише стандартної бібліотеки, що забезпечує портативність і можливість негайного запуску без додаткових залежностей.

Тестування на синтетичних профілях та юніт-тести підтвердили високу точність (середня похибка – десятки годин) і миттєвий відгук у CLI, а аналіз чутливості засвідчив коректну поведінку при зміні параметрів і крайніх випадках.

Накопичений досвід розробки показав, що механістичний підхід до моделювання, поєднаний із модульним дизайном програмного коду, створює надійну основу для подальшого розширення: від інтеграції додаткових факторів ризику (індекс маси тіла, фізична активність) до створення веб- або мобільного інтерфейсів. Отже, реалізована модель може стати корисним інструментом як у практиці індивідуального консультування з питань здоров'я, так і в дослідницьких проектах, присвячених кількісній оцінці впливу поведінкових факторів на статистику тривалості життя.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. WHO report on the global tobacco epidemic 2021: addressing new and emerging products, 2021 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://iris.who.int/bitstream/handle/10665/343287/9789240032095-eng.pdf?sequence=1> Дата звернення: 12.04.2025.
2. Global status report on alcohol and health 2018, 2018 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://iris.who.int/bitstream/handle/10665/274603/9789241565639-eng.pdf?sequence=1> Дата звернення: 15.04.2025.
3. U.S. National Institutes of Health. NIAAA (National Institute on Alcohol Abuse and Alcoholism). "What is a Standard Drink?", 2020 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.niaaa.nih.gov/alcohols-effects-health/what-standard-drink>
4. Seunghee Jun, Hyunjin Park, Ui-Jeong Kim, Hye Ah Lee, Bomi Park, Soon Young Lee, Sun Ha Jee, and Hyesook Park. «The Combined Effects of Alcohol Consumption and Smoking on Cancer Risk by Exposure Level: A Systematic Review and Meta-Analysis», 2024 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://jkms.org/pdf/10.3346/jkms.2024.39.e185>
5. Felicia Ahlner, Hanna Falk, Lena Johansson, Jessica Samuelsson, Hanna Wetterberg, Madeleine Mellqvist Fässberg, Margda Waern, Ingmar Skoog, «The effect of alcohol consumption on all-cause mortality in 70-year-olds in the context of other lifestyle risk factors: results from the Gothenburg H70 birth cohort study», 2023 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: https://www.researchgate.net/publication/373465300_The_effect_of_alcohol_consumption_on_all-cause_mortality_in_70-year-olds_in_the_context_of_other_lifestyle_risk_factors_results_from_the_Gothenburg_H70_birth_cohort_study
6. Sofia Ravara, Julia Rey-Brandariz, Esther López Vizcaíno, María Isolina Santiago-Pérez, Alberto Ruano-Ravina, Cristina Candal, Leonor Varela, Nerea

Mourino Castro, Pedro Aguiar, Mónica Pérez-Rios «Gender differences in smoking-attributable mortality (SAM) by region in Portugal», 2023 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: https://www.researchgate.net/publication/377856042_Gender_differences_in_smoking-attributable_mortality_SAM_by_region_in_Portugal

7. Lixian Zhong, Weiwei Chen, Tonghua Wang, Qiuting Zeng, Leizhen Lai, Junlong Lai, Junqin Lin, Shaohui Tang «Alcohol and Health Outcomes: An Umbrella Review of Meta-Analyses Base on Prospective Cohort Studies», 2020 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://pmc.ncbi.nlm.nih.gov/articles/PMC9115901/>

8. Alina Egorova, Bulat Idrisov, Romany Redman, Stanislav Otstavnov, Sergey Soshnikov «Tobacco, Alcohol and Diet as Mortality Risk Factors: The Secondary Analysis of a 25-Year Cohort Study», 2023 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: https://www.researchgate.net/publication/371968870_Tobacco_Alcohol_and_Diet_as_Mortality_Risk_Factors_The_Secondary_Analysis_of_a_25-Year_Cohort_Study

9. Xinggao Liu, Youzhi Gu, Shuting He, Zhipeng Xu, Zeyin Zhang «A robust reliability prediction method using Weighted Least Square Support Vector Machine equipped with Chaos Modified Particle Swarm Optimization and Online Correcting Strategy», 2019 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: https://www.researchgate.net/publication/336713837_A_robust_reliability_prediction_method_using_Weighted_Least_Square_Support_Vector_Machine_equipped_with_Chaos_Modified_Particle_Swarm_Optimization_and_Online_Correcting_Strategy

10. Sarah E. Jackson, Martin J. Jarvis, Robert West, «The price of a cigarette: 20 minutes of life?», 2024 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://onlinelibrary.wiley.com/doi/10.1111/add.16757>

11. Functional and Information Modeling of Production Using IDEF Methods, 2009 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.sv->

jme.eu/?ns_articles_pdf=/ns_articles/files/ojs3/1563/submission/1563-1-1898-1-2-20171103.pdf&id=4931

12. Fowler M. Patterns of Enterprise Application Architecture, 1st edition, 2012 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://dl.ebooksworld.ir/motoman/Patterns%20of%20Enterprise%20Application%20Architecture.pdf>

13. The Python Standard Library, 2025 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://docs.python.org/3/library/>

14. Shneiderman B., Plaisant, C.: "Designing the User Interface: Strategies for Effective Human-Computer Interaction.", 2010 [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <http://seu1.org/files/level5/IT201/Book%20-%20Ben%20Shneiderman-Designing%20the%20User%20Interface-4th%20Edition.pdf>

15. Kuhn M., Johnson K.: "Applied Predictive Modeling.", 2013. [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: https://vuquangnguyen2016.wordpress.com/wp-content/uploads/2018/03/applied-predictive-modeling-max-kuhn-kjell-johnson_1518.pdf

16. Myers G. J. "The Art of Software Testing. Third edition", 2012. [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://malenezi.github.io/malenezi/SE401/Books/114-the-art-of-software-testing-3-edition.pdf>

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Бакалавр**
Галузь знань: 15 – Автоматизація та приладобудування
Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології
Освітня програма «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри комп'ютерних
систем та робототехніки
к. ф.-м. н., доц. ХРУСЛОВ М. М.
«02» жовтня 2024 року



ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

КОЛІСНИЧЕНКА Микити Миколайовича

(прізвище, ім'я, по батькові студента)

1. Тема роботи **Комп'ютерна модель прогнозування впливу шкідливих звичок на тривалість життя людини**

керівник роботи **Толстолюзька Олена Геннадіївна, д.т.н, с.н.с.**, (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 16 квітня 2024 №4101-5/962 року.

2. Строк подання студентом роботи **30 травня 2025 року**
3. Перелік питань, які потрібно розробити:
- 1) Аналіз сучасних підходів до розробки комп'ютерних моделей прогнозування шкідливого впливу на життя людини з огляду на медичні, статистичні дослідження.
 - 2) Розробка концептуальної комп'ютерної моделі для прогнозування шкідливого впливу на життя людини.
 - 3) Розробка алгоритмів прогнозування тривалості життя, з використанням методів статистичного аналізу.
 - 4) Дослідження методів тестування розробленої моделі, для оцінки її точності.
 - 5) Визначення можливих обмежень використання і формування рекомендацій щодо подальшого вдосконалення моделі.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання
1.	Формулювання наукової проблеми, обґрунтування актуальності, мети та завдань дослідження.	02.10.2024 – 17.10.2024
2.	Аналіз наукових досліджень та існуючих математичних моделей прогнозування тривалості життя.	17.10.2024 – 28.10.2024
3.	Дослідження впливу шкідливих звичок на здоров'я та тривалість життя на основі медичних і статистичних даних.	28.10.2024 – 14.11.2024
4.	Вибір математичних методів для побудови комп'ютерної моделі прогнозування.	14.11.2024 – 14.12.2024
5.	Розробка концепції та архітектури комп'ютерної моделі прогнозування.	14.12.2024 – 14.01.2025
6.	Вибір технологій для реалізації.	14.01.2025 – 09.02.2025
7.	Реалізація алгоритмів аналізу та прогнозування тривалості життя.	09.02.2025 – 03.03.2025
8.	Розробка клієнтської частини для введення даних користувачем та відображення результатів.	03.03.2025 – 24.03.2025
9.	Тестування програмного забезпечення та оцінка точності прогнозування.	24.03.2025 – 30.03.2025
10.	Аналіз ефективності комп'ютерної моделі та її оптимізація.	30.03.2025 – 08.04.2025
11.	Формування рекомендацій щодо вдосконалення моделі.	08.04.2025 – 16.04.2025
12.	Підготовка звіту про переддипломну практику.	16.04.2025 – 30.04.2025
13.	Оформлення звіту про переддипломну практику	30.04.2025 – 09.05.2025
14.	Передзахист кваліфікаційної роботи	09.05.2025 – 20.05.2025
15.	Отримання відгуку керівника та рецензії	до 31.05.2025

5. Дата видачі завдання 02 жовтня 2024 року.

Студент М.М. КолісниченкоКерівник роботи О.Г. Толстолузька

Затверджую

« _____ » _____ 2024 р.

**Технічне завдання
на розробку системи**

«Комп'ютерна модель прогнозування впливу шкідливих звичок на тривалість життя людини».

1.	Введення	<p>1.1. Назва роботи – Комп'ютерна модель прогнозування впливу шкідливих звичок на тривалість життя людини.</p> <p>1.2. Галузь застосування: Інформаційні технології в медицині, соціології, аналітиці даних.</p>
2.	Підстава для розробки	<p>2.1. Навчальний план за спеціальністю 151 – Автоматизація та комп'ютерно-інтегровані технології.</p> <p>2.2. Завдання на кваліфікаційну роботу бакалавра № 4101-5/962 від «16» квітня 2025 року</p>
3.	Призначення розробки	<p>3.1. Мета розробки: створити відтворювану комп'ютерну модель на Python, яка за допомогою введених користувачем даних та епідеміологічних коефіцієнтів обчислює роки життя, втрачені внаслідок куріння, споживання алкоголю й їхньої комбінації</p> <p>3.2. Призначення розробки: науково-практичне використання для дослідження впливу шкідливих звичок (куріння, алкоголь) на здоров'я населення.</p> <p>3.3. Вхідні дані: вік, стать, наявність та тяжкість шкідливих звичок.</p>

		<p>3.4. Вихідні дані: прогнозована тривалість життя (в роках), графічне представлення результатів, рекомендації щодо покращення здоров'я.</p>
4.	Технічні вимоги до програмного виробу	<p>4.1. Функціональні вимоги:</p> <ul style="list-style-type: none"> • Введення користувачем вхідних даних через текстовий інтерфейс. • Обробка введених даних на основі комп'ютерної моделі прогнозування. • Побудова прогнозу тривалості життя. • Виведення текстових результатів. <p>4.2. Нефункціональні вимоги:</p> <ul style="list-style-type: none"> • Інтуїтивно зрозумілий інтерфейс. • Надійна обробка виключних ситуацій (некоректні/неповні дані). • Час розрахунку не більше 2 секунд. <p>4.3. Вимоги до інтеграції:</p> <ul style="list-style-type: none"> • Можливість розширення моделі з додаванням нових факторів ризику. • Сумісність з базами даних CSV, JSON. <p>4.4. Вимоги до безпеки:</p> <ul style="list-style-type: none"> • Перевірка введених користувачем даних.

5.	Вимоги до програмної документації	<p>1. Опис призначення та області застосування</p> <p>1.1 Короткий опис мети моделі: прогнозування втрати й залишкових років життя залежно від вхідних параметрів куріння та вживання алкоголю.</p> <p>1.2 Опис цільової аудиторії: розробники, аналітики, медичні консультанти.</p> <p>2. Вимоги до середовища й встановлення</p> <p>2.1 Перелік підтримуваних версій інтерпретатора Python (CPython 3.6 і вище).</p> <p>2.2 Інструкція з отримання коду (клонування з репозиторію або розпакування архіву).</p> <p>2.3 Опис залежностей: лише стандартна бібліотека, жодних зовнішніх пакетів.</p> <p>2.4 Покрокова команда запуску (наприклад, <code>python calculate.py</code>) та пояснення, як зробити файл виконуваним (якщо потрібно).</p> <p>3. Структура вихідного коду</p> <p>3.1 Опис основних файлів і каталогів (наприклад, <code>calculate.py</code>, <code>test_*</code>, утиліти).</p> <p>3.2 Короткий перелік функцій у кожному файлі: <code>input_choice</code>, <code>input_non_negative_int</code>, <code>calculate_life_expectancy</code> та їхні призначення.</p>
----	-----------------------------------	--

		<p>3.3 Пояснення, які частини коду відповідають за валідацію вводу, обчислення втрат, обробку синергії та формування вихідного результату.</p> <p>4. Керівництво користувача (CLI)</p> <p>4.1 Опис всіх запитів, які з’являються в консолі</p> <p>4.2 Приклади правильного вводу та формати відповіді (наприклад, “так”/“ні”, невід’ємне ціле).</p> <p>4.3 Пояснення вихідного повідомлення: що виводиться як remaining та lost.</p> <p>5. Технічний опис алгоритмів і формул</p> <p>5.1 Формалізація обчислення втрати років через куріння:</p> <p>5.2 Формалізація обчислення втрати років через алкоголь залежно від рівня споживання</p> <p>5.3 Використання коефіцієнтів синергії:</p> <p>5.4 Формули остаточного підсумовування:</p> <p>6. Опис структур даних і конфігурацій</p> <p>6.1 Перелік констант у коді:.</p> <p>6.2 Формат імен змінних, їх типи (int, str, float).</p> <p>7. Інструкція з тестування</p>
--	--	---

		<p>7.1 Опис утиліт для перевірки:</p> <ul style="list-style-type: none"> • скрипти для синтетичної генерації профілів; • тести на числову коректність (порівняння з референсною функцією); • тести валідації вводу (симуляція некоректних відповідей). <p>7.2 Приклади запуску тестів (наприклад, <code>pytest</code> або <code>python test_numeric_correctness.py</code>).</p> <p>7.3 Очікувані критерії проходження (наприклад, $MAE \leq 0.001$, відсутність аварій на некоректному вводі).</p> <p>8. Керівництво розробника</p> <p>8.1 Інструкція зі структурування нових модулів</p> <p>8.2 Рекомендації з підтримки високої узгодженості всередині модулів і слабкого зв'язку між ними.</p> <p>8.3 Опис процедур внесення змін до коефіцієнтів</p>	
6.	Стадії і етапи розробки	Дата	Назва етапу
		01.11.2024 – 16.11.2024	Формулювання проблеми, мети та завдань
		16.11.2024 – 01.12.2024	Аналіз досліджень і моделей

		01.12.2024 – Дослідження впливу шкідливих 20.12.2024 звичок 05.01.2025 – Вибір математичних методів 21.01.2025 21.01.2025 – Розробка архітектури моделі 14.02.2025 14.02.2025 – Вибір технологій реалізації 28.02.2025 28.02.2025 – Реалізація алгоритмів 12.03.2025 прогнозування 12.03.2025 – Розробка інтерфейсу 24.03.2025 24.03.2025 – Тестування та оцінка точності 30.03.2025 30.03.2025 – Оптимізація моделі 08.04.2025 08.04.2025 – Формування рекомендацій 16.04.2025 16.04.2025 – Підготовка звіту з практики 30.04.2025 30.04.2025 – Оформлення звіту 09.05.2025 09.05.2025 – Передзахист 20.05.2025 до Отримання відгуку та рецензії 30.05.2025
7.	Порядок контролю і приймання	1. Перевірку ходу розробки комп'ютерної моделі виконувати раз в 3 тижні.

	програмного продукту (моделі)	2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді.
--	-------------------------------	---

Виконавець

Студент групи КУ-41

Колісниченко М. М.



Замовник

д.т.н., с.н.с.

Толстолузька О. Г.



Додаток В**Програма і методика випробувань програмного виробу**

«Комп'ютерна модель прогнозування впливу шкідливих звичок на тривалість життя людини»

1. Об'єкт випробувань

1.1. Назва розробленого прототипу: «Комп'ютерна модель прогнозування впливу шкідливих звичок на тривалість життя людини».

1.2. Галузь застосування: Інформаційні технології в медицині, соціології, аналітиці даних.

1.3. Відомості запозичені з розділів технічного завдання, що визначають функціональні та нефункціональні вимоги до моделі, джерела даних і програмну архітектуру.

2. Мета випробувань

Перевірка відповідності функціональних і нефункціональних можливостей комп'ютерної моделі прогнозування впливу шкідливих звичок на тривалість життя людини заявленим у технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення**3.1. Підстави для проведення випробувань**

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

3.2. Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3.3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

3.4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавці та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

4.1 Функціональні вимоги:

1. Введення користувачем вхідних даних через текстовий інтерфейс.
2. Обробка введених даних на основі комп'ютерної моделі прогнозування.
3. Побудова прогнозу тривалості життя.
4. Виведення текстових результатів.

4.2 Нефункціональні вимоги:

1. Інтуїтивно зрозумілий інтерфейс.
2. Надійна обробка виключних ситуацій (некоректні/неповні дані).
3. Час розрахунку не більше 2 секунд.

5. Вимоги до програмної документації

1. Опис призначення та області застосування

1.1 Короткий опис мети моделі: прогнозування втрати й залишкових років життя залежно від вхідних параметрів куріння та вживання алкоголю.

1.2 Опис цільової аудиторії: розробники, аналітики, медичні консультанти.

2. Вимоги до середовища й встановлення

2.1 Перелік підтримуваних версій інтерпретатора Python (CPython 3.6 і вище).

2.2 Інструкція з отримання коду (клонування з репозиторію або розпакування архіву).

2.3 Опис залежностей: лише стандартна бібліотека, жодних зовнішніх пакетів.

2.4 Покрокова команда запуску (наприклад, `python calculate.py`) та пояснення, як зробити файл виконуваним (якщо потрібно).

3. Структура вихідного коду

3.1 Опис основних файлів і каталогів (наприклад, `calculate.py`, `test_*`, утиліти).

3.2 Короткий перелік функцій у кожному файлі: `input_choice`, `input_non_negative_int`, `calculate_life_expectancy` та їхні призначення.

3.3 Пояснення, які частини коду відповідають за валідацію вводу, обчислення втрат, обробку синергії та формування вихідного результату.

4. Керівництво користувача (CLI)

4.1 Опис всіх запитів, які з'являються в консолі

4.2 Приклади правильного вводу та формати відповіді (наприклад, “так”/“ні”, невід’ємне ціле).

4.3 Пояснення вихідного повідомлення: що виводиться як `remaining` та `lost`.

5. Технічний опис алгоритмів і формул

5.1 Формалізація обчислення втрати років через куріння:

5.2 Формалізація обчислення втрати років через алкоголь залежно від рівня споживання

5.3 Використання коефіцієнтів синергії:

5.4 Формули остаточного підсумовування:

6. Опис структур даних і конфігурацій

6.1 Перелік констант у коді:

6.2 Формат імен змінних, їх типи (int, str, float).

7. Інструкція з тестування

7.1 Опис утиліт для перевірки:

- скрипти для синтетичної генерації профілів;
- тести на числову коректність (порівняння з референсною функцією);
- тести валідації вводу (симуляція некоректних відповідей).

7.2 Приклади запуску тестів (наприклад, python test_numeric_correctness.py).

7.3 Очікувані критерії проходження (наприклад, $MAE \leq 0.001$, відсутність аварій на некоректному вводі).

8. Керівництво розробника

8.1 Інструкція зі структурування нових модулів

8.2 Рекомендації з підтримки високої узгодженості всередині модулів і слабого зв'язку між ними.

8.3 Опис процедур внесення змін до коефіцієнтів

6. Засоби і порядок випробувань

6.1 Засоби випробувань

1. Персональний комп'ютер із встановленим CPython 3.6+ і доступом до командного рядка.
2. Інструменти автоматизованого тестування: `pytest` для юніт-тестів, скрипти `test_numeric_correctness.py`, `test_validation.py`, `test_sensitivity.py`.
3. Засоби профілювання: модуль `cProfile`, `time.perf_counter()`, `tracemalloc` (опціонально).
4. Репозиторій із кодом (`calculate.py` та допоміжні утиліти).

6.2. Порядок проведення випробувань

Випробування проводяться в два етапи:

1-й етап (ознайомчий):

1. Перевірка наявності та повноти програмної документації, тестових скриптів і середовища (виртуальне середовище, залежності Python).
2. Верифікація структури репозиторію: наявність основного скрипта, модулів введення, розрахункових модулів та тестових файлів.
3. Ознайомлення з інструкціями запуску тестів і контрольною методикою.

2-й етап (приймальні випробування):

1. Тест 1 – числова коректність: запуск `test_numeric_correctness.py`; перевірка $MAE \leq 0.001$ та `max`-похибок згідно з технічним завданням.
2. Тест 2 – валідація вводу: запуск `test_validation.py`; симуляція некоректних відповідей і перевірка відсутності аварій та коректних повідомлень про помилки.

3. Тест 3 – продуктивність: запуск `benchmark.py` з 50 ітераційними прогінами для типового та крайового сценаріїв; оцінка затримки ≤ 100 мс на запит і енд-ту-енд ≤ 1 с.
4. Тест 4 – чутливість та крайні випадки: запуск `test_sensitivity.py`; перевірка лінійності реакції на зміну сигарет/день і ступінчастості для порогів алкоголю, а також підтвердження `remaining` ≥ 0 для екстремальних профілів.

Тест 1

```
(.venv) PS C:\Users\webbk\PyCharmMiscProject\LW3\diploma> python test_numeric_correctness_cli.py
Parsed outputs: 500/500
MAE remaining: 0.001789, max error: 0.005000
MAE lost      : 0.001949, max error: 0.005000
Avg exec ms  : 0.01
```

Рисунок В.1 - Тест 1

Тест 2

```
===== test session starts =====
collecting ... collected 2 items

test_validation.py::test_non_negative PASSED [ 50%]
test_validation.py::test_choice PASSED [100%]

===== 2 passed in 0.03s =====
```

Рисунок В.2 - Тест 2

Тест 3

```

TYPICAL scenario (50 runs):
  min    =    0.01 ms
  mean   =    0.01 ms
  max    =    0.03 ms

EXTREME scenario (50 runs):
  min    =    0.01 ms
  mean   =    0.01 ms
  max    =    0.03 ms

ZERO scenario (50 runs):
  min    =    0.00 ms
  mean   =    0.01 ms
  max    =    0.03 ms

cProfile for 2000 runs:
      70001 function calls in 0.054 seconds

```

Рисунок В.3 - Тест 3(часть 1)

```

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
14000   0.013    0.000    0.013    0.000 {built-in method builtins.input}
 2000   0.010    0.000    0.045    0.000 C:\Users\webbk\PyCharmMiscProject\LW3\Diploma\calculate.py:22(calculate_life_expectancy)
 6000   0.007    0.000    0.018    0.000 C:\Users\webbk\PyCharmMiscProject\LW3\Diploma\calculate.py:13(input_choice)
 2000   0.007    0.000    0.009    0.000 C:\Users\webbk\PyCharmMiscProject\LW3\Diploma\benchmark.py:43(make_input_stream)
 8000   0.006    0.000    0.015    0.000 C:\Users\webbk\PyCharmMiscProject\LW3\Diploma\calculate.py:1(input_non_negative_int)
14000   0.003    0.000    0.003    0.000 {method 'strip' of 'str' objects}
 8000   0.003    0.000    0.003    0.000 {method 'join' of 'str' objects}
 6000   0.002    0.000    0.002    0.000 {method 'lower' of 'str' objects}
 2000   0.002    0.000    0.002    0.000 {built-in method builtins.print}
 6000   0.001    0.000    0.001    0.000 {method 'append' of 'list' objects}

Memory snapshot:
current = 1.4 KB
peak    = 1.9 KB

```

Рисунок В.4 - Тест 3(часть 2)

Тест 4

```
(.venv) PS C:\Users\webbk\PyCharmMiscProject\LW3\diploma> python test_sensitivity.py
Sensitivity remaining per cig/day: -0.080000
Sensitivity lost      per cig/day: 0.080000

Edge case {'gender': 'чоловік', 'cigs_per_day': 0, 'years_smoking': 0, 'drinks_per_week': 0, 'years_heavy': 0, 'years_moderate': 0, 'years_light': 0}: remaining=80.00, lost=0.00
Edge case {'gender': 'жінка', 'cigs_per_day': 40, 'years_smoking': 40, 'drinks_per_week': 50, 'years_heavy': 40, 'years_moderate': 0, 'years_light': 0}: remaining=0.00, lost=239.22
```

Рисунок В.5 - Тест 4

Висновки: тест 1, 2, 3 та 4 успішно пройшли випробування. Випробування пройшло успішно.

Виконавець: студент групи КУ-41, Колісниченко М.М.

