

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

Бакалаврська кваліфікаційна робота на тему:
ПРОГРАМНИЙ МОДУЛЬ РОЗРАХУНКУ ВИТРАТ НА
ЗДІЙСНЕННЯ НАУКОВО-ДОСЛІДНОЇ РОБОТИ

Виконав: студент 4 курсу, групи 4КН - 216
спеціальності 122 – Комп'ютерні науки

(цифр і назва напрямку підготовки, спеціальності)

Казимірський Р.С.

(прізвище та ініціали)

Керівник: к.ф.м.н., доцент каф. КН

Шевчук О. Ф.

(прізвище та ініціали)

« 04 » Червень 2025 р.

Опонент: к.т.н., доцент каф. САІТ

Горячев Г.В.

(прізвище та ініціали)

« 05 » Червень 2025 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

« 06 » Червень 2025 р.

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо - професійна програма – « Комп'ютерні науки »

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

Д.т.н., проф. Яровий А. А.

« 05 » листопада 2025 р.

ЗАВДАННЯ

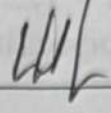
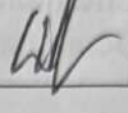
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Казимірському Руслану Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмний модуль розрахунку витрат на здійснення науково-дослідної роботи
керівник роботи к.ф.-м.н., доцент кафедри КН Шевчук О. Ф.
затверджені наказом ВНТУ від «20» серпня 2025 року № 34
2. Строк подання студентом роботи 30 травня 2025 року
3. Вихідні дані до роботи: мова програмування – об'єктно-орієнтована; мінімальна кількість статей розрахунку витрат – 14; мінімальна кількість функцій – 12; мінімальна кількість параметрів налаштування розрахунку витрат – 4; тип інтерфейсу користувача інтуїтивно-зрозумілий.
4. Зміст текстової частини:
Вступ, аналіз сучасних засобів розрахунку витрат на здійснення науково-дослідної роботи, моделювання програмного модуля витрат на здійснення науково-дослідної роботи, програмна реалізація модуля витрат на здійснення науково-дослідної роботи, тестування, висновки, перелік використаних джерел, додатки
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): алгоритму модуля витрат на здійснення науково-дослідної роботи, UML-діаграма послідовностей роботи та UML-діаграма класів модуля витрат на здійснення науково-дослідної роботи, результати тестування розробленого програмного модуля.

6. Консультанти розділів роботи

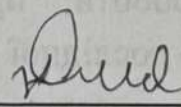
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1 - 3	Шевчук О.Ф., к.ф. - м.н, доцент каф. КН		

1. Дата видачі завдання 05 травня 2025 року

КАЛЕНДАРНИЙ ПЛАН

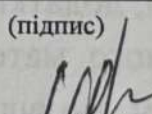
№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сучасних засобів розрахунку витрат на здійснення науково-дослідної роботи	05.05.25	07.05.25
2	Моделювання програмного модуля витрат на здійснення науково-дослідної роботи	08.05.25	11.05.25
3	Програмна реалізація модуля витрат на здійснення науково- дослідної роботи	12.05.25	15.05.25
4	Тестування програмного модуля та розробка інструкції користувача	16.05.25	26.05.25
6	Оформлення матеріалів до захисту БКР	02.06.25	09.06.25

Студент


(підпис)

Казимірський Р.С

Керівник роботи


(підпис)

Шевчук О.Ф.

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 92 сторінок формату А4 на яких є 37 рисунків, 4 таблиці, список використаних джерел містить 26 найменувань.

Бакалаврська кваліфікаційна робота присвячена розробці програмного модуля розрахунку витрат на здійснення науково-дослідної роботи. Проведено огляд сучасних засобів розрахунку витрат на здійснення науково-дослідної роботи. Розроблено та удосконалено алгоритм розрахунку витрат на здійснення науково-дослідної роботи. Розроблено структуру розрахунку витрат на здійснення науково-дослідної роботи. Реалізовано програмне забезпечення, обґрунтовано вибір мов та середовища розробки, створено UML-діаграми класів та послідовностей для ключових модулів системи. Тестування підтвердило коректність роботи модулів.

Ключові слова: автоматизований розрахунок, витрати, науково-дослідна робота, програмний модуль, збереження результатів.

ABSTRACT

The bachelor's qualification work consists of 92 A4-format pages, including 37 figures, 4 tables, and a list of references comprising 26 sources.

The bachelor's qualification work is dedicated to the development of a software module for calculating the costs of conducting research and development activities. An overview of modern tools for cost estimation in scientific and research activities has been conducted. An algorithm for calculating the costs of research and development was developed and improved. The structure of cost calculation for research and development was designed. The software was implemented, the choice of programming language and development environment was justified, and UML class and sequence diagrams were created for the key system modules. Testing confirmed the correctness of the module operations.

Keywords: automated calculation, costs, research and development, software module, result saving.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ СУЧАСНИХ ЗАСОБІВ РОЗРАХУНКУ ВИТРАТ НА ЗДІЙСНЕННЯ НАУКОВО-ДОСЛІДНОЇ РОБОТИ.....	8
1.1 Методичні підходи до засобів розрахунку витрат на здійснення науково-дослідної роботи.....	8
1.2 Аналіз сучасних програмних засобів розрахунку витрат на здійснення науково-дослідної роботи.....	19
1.3 Обґрунтування вибору методу розрахунку витрат на здійснення науково-дослідної роботи.....	23
1.4 Постановка задачі.....	25
1.5 Висновок до розділу 1.....	26
2 ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ РОЗРАХУНКУ ВИТРАТ НА ЗДІЙСНЕННЯ НАУКОВО-ДОСЛІДНОЇ РОБОТИ	27
2.1 Розробка алгоритмів роботи програмного модуля	27
2.2 Проєктування структури програмного модуля	30
2.3 Розробка UML діаграми класів.....	31
2.4 Висновок до розділу 2.....	33
3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ ВИТРАТ НА ЗДІЙСНЕННЯ НАУКОВО-ДОСЛІДНОЇ РОБОТИ.....	34
3.1 Обґрунтування вибору мови програмування та спеціалізованих бібліотек .	34
3.2 Обґрунтування вибору середовища програмування	37
3.3 Розробка функціоналу програмного модуля розрахунку витрат на здійснення науково-дослідної роботи.....	39
3.4 Тестування програмного модуля розрахунку витрат на здійснення науково-дослідної роботи.....	55
3.5 Висновок до розділу 3.....	65
ВИСНОВКИ.....	67
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
Додаток А (обов'язковий) ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ.....	71
Додаток Б (обов'язковий) ЛІСТИНГ ПРОГРАМИ.....	72

Додаток В (довідниковий) ІНСТРУКЦІЯ КОРИСТУВАЧА	81
Додаток Г (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА	86

ВСТУП

Актуальність досліджень. В умовах сучасної економіки питання ефективного планування, обліку та контролю витрат на здійснення науково-дослідної роботи (НДР) набуває особливої актуальності. Проведення досліджень, запуск інноваційних продуктів чи стартапів потребує не лише наукового обґрунтування, а й детального економічного аналізу доцільності таких ініціатив. Витрати на виконання НДР часто є складними для оцінювання, адже охоплюють широкий спектр ресурсів - від заробітної плати персоналу до специфічного програмного забезпечення, обладнання та супутніх послуг. Тому постає необхідність у створенні спеціалізованих інструментів для точного розрахунку таких витрат.

На сьогодні існують різні методичні підходи до оцінювання витрат: нормативний, факторний, аналоговий, експертний та інтегрований. Кожен з них має свої переваги, сфери застосування та обмеження. Аналіз цих підходів дозволяє визначити найбільш доцільний для розробки програмного модуля, що відповідає сучасним вимогам до автоматизації обліку витрат на НДР. Поряд із методичними аспектами, не менш важливим є вивчення вже наявних програмних рішень, які можуть бути використані або адаптовані для зазначених цілей.

Метою роботи є розширення функціональних можливостей програмного модуля, який забезпечує автоматизований розрахунок витрат на виконання науково-дослідної роботи.

Об'єктом дослідження є процес планування та розрахунку витрат на виконання науково-дослідної роботи.

Предметом дослідження є методи та програмні засоби обчислення витрат науково-дослідної роботи.

Задачі дослідження:

1. Аналіз предметної області та існуючих методів розрахунку витрат на НДР та програмних засобів, що їх реалізують.

2. Порівняльний аналіз програмних засобів розрахунку витрат на здійснення науково-дослідної роботи.
3. Проектування програмного модуля розрахунку витрат на здійснення науково-дослідної роботи.
4. Програмна реалізація розрахунку витрат на здійснення науково-дослідної роботи.
5. Тестування програмного модуля.
6. Розробка інструкції користувача.

1 АНАЛІЗ СУЧАСНИХ ЗАСОБІВ РОЗРАХУНКУ ВИТРАТ НА ЗДІЙСНЕННЯ НАУКОВО-ДОСЛІДНОЇ РОБОТИ

1.1 Методичні підходи до засобів розрахунку витрат на здійснення науково-дослідної роботи

Одним із найважливіших етапів організації та реалізації науково-дослідної роботи (НДР) є планування її витрат. Якісний і обґрунтований розрахунок кошторису дозволяє не лише ефективно розподілити ресурси, а й обґрунтувати фінансування проєкту, визначити його економічну доцільність та мінімізувати фінансові ризики. Саме тому особливого значення набувають методичні підходи, які застосовуються при розрахунку витрат [1].

Існує кілька основних методичних підходів до розрахунку витрат, кожен із яких має свої переваги та обмеження. До таких підходів належать нормативний, факторний, аналоговий, експертний та інтегрований. Розглянемо кожен з них більш детально [2].

Нормативний підхід полягає у використанні затверджених нормативів (тарифів, ставок, норм використання ресурсів тощо) для визначення обсягів витрат. Цей метод відзначається високою точністю, простотою реалізації в умовах стабільного нормативного середовища та можливістю автоматизації [3].

Більш детально розберемо нормативний підхід. Витрати, пов'язані з проведенням науково-дослідної, дослідно - конструкторської, конструкторсько - технологічної роботи, створенням дослідного зразка і здійсненням виробничих випробувань, під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуються за такими статтями [4]:

- витрати на оплату праці;
- відрахування на соціальні заходи;
- матеріали;
- паливо та енергія для науково - виробничих цілей;

- витрати на службові відрядження;
- спецустаткування для наукових (експериментальних) робіт;
- програмне забезпечення для наукових (експериментальних) робіт;
- витрати на роботи, які виконують сторонні підприємства та установи;
- інші витрати;
- накладні (загальновиробничі) витрати.

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно - технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь - які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці» [5].

Основна заробітна плата дослідників. Витрати на основну заробітну плату дослідників (Z_o) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (1.1)$$

де k – кількість посад дослідників, залучених до процесу дослідження; M_{ni} – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.; T_p – число робочих днів в місяці; приблизно $T_p = 22$ дні; t_i – число робочих днів роботи розробника (дослідника).

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт розраховують за формулою:

$$З_p = \sum_{i=1}^n C_i \cdot t_i, \quad (1.2)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год; t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду можна визначити за формулою:

$$C_i = \frac{M_m + K_i + K_c}{T_p \cdot t_{зм}}, \quad (1.3)$$

де M_m – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), грн; K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду; K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати; T_p – середня кількість робочих днів в місяці, приблизно $T_p = 21 \dots 23$ дні; $t_{зм}$ – тривалість зміни, год.

Додаткова заробітна плата дослідників та робітників. Додаткова заробітна плата розраховується як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_o + З_p) \cdot \frac{H_{дод}}{100\%}, \quad (1.4)$$

де $H_{дод}$ – норма нарахування додаткової заробітної плати.

Відрахування на соціальні заходи. До статті «Відрахування на соціальні заходи» належать відрахування внеску на загальнообов'язкове державне соціальне

страхування та для здійснення заходів щодо соціального захисту населення (ЄСВ – єдиний соціальний внесок).

Нарахування на заробітну плату дослідників та робітників розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{\text{дод}}) \cdot \frac{H_{\text{зп}}}{100\%}, \quad (1.5)$$

де $H_{\text{зп}}$ – норма нарахування на заробітну плату.

Сировина та матеріали. До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби й предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за прямим призначенням згідно з нормами їх витрачання, а також витрачені придбані напівфабрикати, що підлягають монтажу або виготовленню й додатковій обробці в цій організації, чи дослідні зразки, що виготовляються виробниками за документацією наукової організації. Витрати на матеріали (M) у вартісному вираженні розраховуються окремо для кожного виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot Ц_j \cdot K_j - \sum_{j=1}^n B_j \cdot Ц_{\text{в}j}, \quad (1.6)$$

де H_j – норма витрат матеріалу j - го найменування, кг.; n – кількість видів матеріалів; $Ц_j$ – вартість матеріалу j - го найменування, грн/кг; K_j – коефіцієнт транспортних витрат, $K_j = (1, 1 \dots 1, 15)$; B_j – маса відходів j - го найменування, кг; $Ц_{\text{в}j}$ – вартість відходів j - го найменування, грн/кг.

Витрати на комплектуючі вироби (K_v), які використовують при дослідженні нового технічного рішення, розраховуються, згідно з їхньою номенклатурою, за формулою:

$$K_v = \sum_{j=1}^n H_j \cdot C_j \cdot K_j, \quad (1.7)$$

де H_j – кількість комплектуючих j -го виду, шт.; C_j – покупна ціна комплектуючих j -го виду, грн; K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$).

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування, верстатів, пристроїв, інструментів, приладів, стендів, апаратів, механізмів, іншого спецобладнання, необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Вартість спецустаткування визначається за прейскурантом гуртових цін або за даними базових підприємств за відпускними і договірними цінами. До балансової вартості устаткування окрім прейскурантної вартості входять витрати на його транспортування і монтаж, тому ці витрати беруться додатково в розмірі 10...12% від вартості устаткування.

Балансову вартість спецустаткування розраховують за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.і}} \cdot K_i, \quad (1.8)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн; $C_{\text{пр.і}}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.; K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10 \dots 1,12$); k – кількість найменувань устаткування.

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення. До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10...12% від вартості програмного забезпечення.

Балансову вартість програмного забезпечення розраховують за формулою:

$$B_{\text{прг}} = \sum_{i=1}^k \text{Ц}_{\text{іпрг}} \cdot C_{\text{прг.і}} \cdot K_i, \quad (1.9)$$

де $\text{Ц}_{\text{прг.і}}$ – ціна придбання програмного забезпечення і-го виду, грн.; $C_{\text{прг.і}}$ – кількість одиниць програмного забезпечення відповідного виду, шт.; K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного забезпечення, $K_i = (1, 1.1 \dots 1.12)$; k – кількість видів програмного забезпечення.

До статті «Амортизація обладнання, програмних засобів та приміщень» відносять амортизаційні відрахування по кожному виду обладнання, устаткування та інших приладів і пристроїв, а також програмного забезпечення для проведення науково-дослідної роботи, за його наявності в дослідній організації або на підприємстві. В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{\text{Ц}_6}{T_v} \cdot \frac{t_{\text{вик}}}{12}, \quad (1.10)$$

де Ц_6 – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; t – термін

використання основного фонду, місяці; T_v – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

До статті «Паливо та енергія для науково - виробничих цілей» належать витрати на придбання у сторонніх підприємств, установ і організацій будь - якого палива, що витрачається з технологічною метою на проведення досліджень. Стаття формується у разі виконання енергоємних наукових досліджень за методом прямого внесення витрат і досягає значної питомої ваги у собівартості досліджень [6].

Витрати на силову електроенергію (B_e) розраховують за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i}, \quad (1.11)$$

де W_{yi} – встановлена потужність обладнання на певному етапі розробки, кВт; t_i – тривалість роботи обладнання на етапі досліджень, год; C_e – вартість 1 кВт - години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії); $K_{ен}$ – коефіцієнт, що враховує використання потужності, $K_{ен} < 1$; η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

До статті «Службові відрядження» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно - правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуються як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_o + З_p) \cdot \frac{H_{св}}{100\%}, \quad (1.12)$$

де $H_{\text{св}}$ – норма нарахування за статтею «Службові відрядження».

До статті «Витрати на роботи, які виконують сторонні підприємства, установи і організації» належать витрати на проведення досліджень, що не можуть бути виконані штатними працівниками або наявним обладнанням організацій, а виконуються на договірній основі іншими підприємствами, установами і організаціями незалежно від форм власності та позаштатними працівниками [7]. Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуються як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{\text{сп}} = (3_o + 3_p) \cdot \frac{H_{\text{сп}}}{100\%}, \quad (1.13)$$

де $H_{\text{сп}}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації».

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (3_o + 3_p) \cdot \frac{H_{\text{ів}}}{100\%}, \quad (1.14)$$

де $H_{\text{ів}}$ – норма нарахування за статтею «Інші витрати».

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з

освоєнням виробництва продукції; витрати на науково - технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{\text{нзв}} = (З_o + З_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (1.15)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = З_o + З_p + З_{\text{дод}} + M + K_v + B_{\text{спец}} + B_{\text{прг}} + A_{\text{обл}} + B_e + B_{\text{св}} + B_{\text{сн}} + I_v + B_{\text{нзв}}. \quad (1.16)$$

Факторний підхід базується на аналізі чинників, що впливають на обсяг витрат: тривалість виконання НДР, склад команди, складність дослідження, кількість обладнання, що використовується, та інші параметри. Цей підхід дозволяє побудувати модель витрат, яка враховує індивідуальні характеристики проекту. Його головною перевагою є гнучкість, проте реалізація вимагає глибокого аналізу та наявності якісних вхідних даних [8].

Аналоговий підхід ґрунтується на використанні інформації про витрати, понесені у попередніх подібних проєктах. Це дозволяє швидко сформувати орієнтовний бюджет на основі історичних даних. Основним обмеженням цього підходу є його слабка чутливість до унікальності нового дослідження [9].

Експертний підхід передбачає залучення фахівців, які на основі свого досвіду надають оцінку обсягу необхідних витрат. Цей метод є ефективним у випадках, коли відсутня можливість використати інші джерела інформації. Основним недоліком є суб'єктивність оцінок та неможливість автоматизації [10].

Інтегрований підхід поєднує кілька вищенаведених методів. Наприклад, базові витрати розраховуються за нормативами, а додаткові витрати - з урахуванням факторів чи експертних оцінок. Такий підхід дозволяє враховувати специфіку проєкту, забезпечити точність розрахунків та адаптивність моделі. Нижче подано порівняльну таблицю 1.1, що відображає основні характеристики кожного з підходів:

Таблиця 1.1 – Порівняння методичних підходів до розрахунку витрат

Підхід	Джерело даних	Складність впровадження	Точність	Придатність до автоматизації
Нормативний	Офіційні норми	Низька	Висока	Висока
Факторний	Аналітика	Середня	Висока	Середня
Аналоговий	Попередні проєкти	Низька	Середня	Висока
Експертний	Оцінки фахівців	Висока	Низька	Низька
Інтегрований	Комбінація	Середня	Висока	Середня

Підсумовуючи вищезазначене, можна зробити висновок, що вибір методичного підходу до розрахунку витрат має визначатись специфікою наукового проєкту, доступністю вхідних даних, вимогами до точності та можливістю автоматизації процесу.

У рамках цієї роботи за основу буде взято інтегрований підхід з домінуванням нормативно - факторного компонента. Це забезпечить можливість програмної реалізації розрахунків витрат, враховуючи як офіційні нормативи, так і гнучкі параметри, що задаються користувачем. Такий підхід відповідає сучасним вимогам до інформаційних систем у сфері управління НДР. Крім того, слід зазначити, що нормативний підхід є базовим інструментом в роботі бухгалтерів,

економістів та кошторисників. У контексті НДР цей підхід дає можливість не тільки визначити очікувану вартість дослідження, а й підготувати фінансову звітність, що відповідає вимогам контролюючих органів. Це особливо важливо при роботі з державним або міжнародним фінансуванням, де звітність має бути формалізованою та прозорою [11].

Факторний підхід, своєю чергою, дозволяє здійснювати сценарне планування, моделюючи витрати за різними умовами реалізації проєкту. Наприклад, можна оцінити, як зміниться загальна вартість дослідження за умови збільшення кількості учасників або тривалості етапу експериментів. Це створює підґрунтя для ухвалення обґрунтованих управлінських рішень щодо оптимізації бюджету.

Особливої актуальності набуває інтегрований підхід в умовах багатофакторного середовища. Поєднання нормативів із коригувальними коефіцієнтами, що враховують реальні умови реалізації дослідження, дозволяє досягти балансу між точністю та адаптивністю. Такий підхід легко масштабувати й адаптувати під конкретні організаційні та фінансові умови проєкту.

Усі вищезазначені підходи можуть бути використані як самостійно, так і в комбінації, залежно від рівня доступності вхідних даних, вимог замовника, особливостей нормативно - правової бази та типу фінансування. Тому вибір методу розрахунку витрат повинен бути обґрунтованим і враховувати не лише технічні аспекти, а й організаційні умови реалізації проєкту.

Із практичного погляду, кожен підхід до розрахунку витрат має свої оптимальні області застосування. Наприклад, нормативний підхід є ідеальним для проєктів, що фінансуються з бюджетних джерел і вимагають суворого дотримання державних норм. У той час як факторний краще підходить для досліджень, де параметри постійно змінюються та потребують регулярного коригування.

У сучасних умовах цифровізації та широкого використання інформаційних систем, важливим є також аспект інтеграції підходів до розрахунку витрат із програмними засобами. Це дозволяє автоматизувати обчислення, зменшити кількість помилок і прискорити підготовку фінансової документації. Саме такі

вимоги й лягли в основу обґрунтування необхідності створення програмного модуля, присвяченого розрахунку витрат на НДР.

Окремо слід відзначити роль вибору методичного підходу на стадії написання технічного завдання до програмного продукту. Чітке визначення методів розрахунку, типів вхідних даних і очікуваних результатів дозволяє значно полегшити подальше проектування та реалізацію інформаційної системи.

1.2 Аналіз сучасних програмних засобів розрахунку витрат на здійснення науково-дослідної роботи

У сучасних умовах цифровізації управлінських процесів розрахунок витрат на науково-дослідні роботи потребує автоматизації. Програмне забезпечення відіграє важливу роль у забезпеченні точності, швидкості та зручності планування кошторисів. У цьому підрозділі буде розглянуто кілька програмних засобів, які застосовуються для розрахунку витрат у різних сферах, включно з тими, що можуть бути адаптовані до потреб НДР.

COCOMO II (Constructive Cost Model) - це математична модель, призначена для оцінки витрат на розробку програмного забезпечення, яка може бути адаптована для розрахунку витрат на науково-дослідні роботи. Вона враховує складність проєкту, досвід команди, кількість рядків коду та інші фактори [12]. Модель дозволяє прогнозувати необхідні зусилля, тривалість та загальну вартість проєкту, що робить її ефективним інструментом для етапу планування НДР (рис. 1.1).

На рисунку 1.2 показано результати розрахунку моделі COCOMO II для оцінки трудомісткості та вартості розробки програмного забезпечення. В даному випадку - всі значення нулеві, тому що вхідні дані (розмір ПЗ, параметри проєкту) не були задані.

COCOMO II - Constructive Cost Model

Software Size Sizing Method

[SLOC](#) % Design Modified % Code Modified % Integration Required Assessment and Assimilation (0% - 8%) Software Understanding (0% - 50%) Unfamiliarity (0-1)

New

Reused 0 0

Modified

Software Scale Drivers

Precedentedness Architecture / Risk Resolution Process Maturity

Development Flexibility Team Cohesion

Software Cost Drivers

Product **Personnel** **Platform**

Required Software Reliability Analyst Capability Time Constraint

Data Base Size Programmer Capability Storage Constraint

Product Complexity Personnel Continuity Platform Volatility

Developed for Reusability Application Experience

Documentation Match to Lifecycle Needs Platform Experience

Language and Toolset Experience

Project

Use of Software Tools

Multisite Development

Required Development Schedule

Maintenance

Software Labor Rates

Cost per Person-Month (Dollars)

Рисунок 1.1 – Вебсторінка математичної моделі COCOMO II

Results

Software Development (Elaboration and Construction) Staffing Profile

Effort = 0.0 Person-months

Schedule = 0.0 Months

Cost = \$0

Total Equivalent Size = 0 SLOC

Effort Adjustment Factor (EAF) = 1.00

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.0	0.0	0.0	\$0
Elaboration	0.0	0.0	0.0	\$0
Construction	0.0	0.0	0.0	\$0
Transition	0.0	0.0	0.0	\$0

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.0	0.0	0.0
Environment/CM	0.0	0.0	0.0	0.0
Requirements	0.0	0.0	0.0	0.0
Design	0.0	0.0	0.0	0.0
Implementation	0.0	0.0	0.0	0.0
Assessment	0.0	0.0	0.0	0.0
Deployment	0.0	0.0	0.0	0.0

Рисунок 1.2 – Результати розрахунків математичної моделі COCOMO II

PLANTA Project - це комплексна система управління проєктами, яка включає модулі для планування, обліку, контролю витрат і ресурсів. Вона дозволяє вести облік витрат за центрами витрат, ресурсами та фазами проєкту. Підходить для науково - технічних організацій, які виконують кілька НДР одночасно. Система дозволяє створювати бюджети та аналізувати їх фактичне виконання, що сприяє ефективному управлінню витратами. PLANTA підтримує розрахунок планових і фактичних витрат у розрізі часових періодів та інтегрується з ERP - системами. Завдяки гнучкості налаштувань її можна адаптувати до специфіки різних дослідницьких проєктів. Інтерфейс програми надає інструменти для побудови звітів, діаграм Ганта, розрахунку навантаження ресурсів і оцінки економічної ефективності дослідження (рис. 1.3) [13].

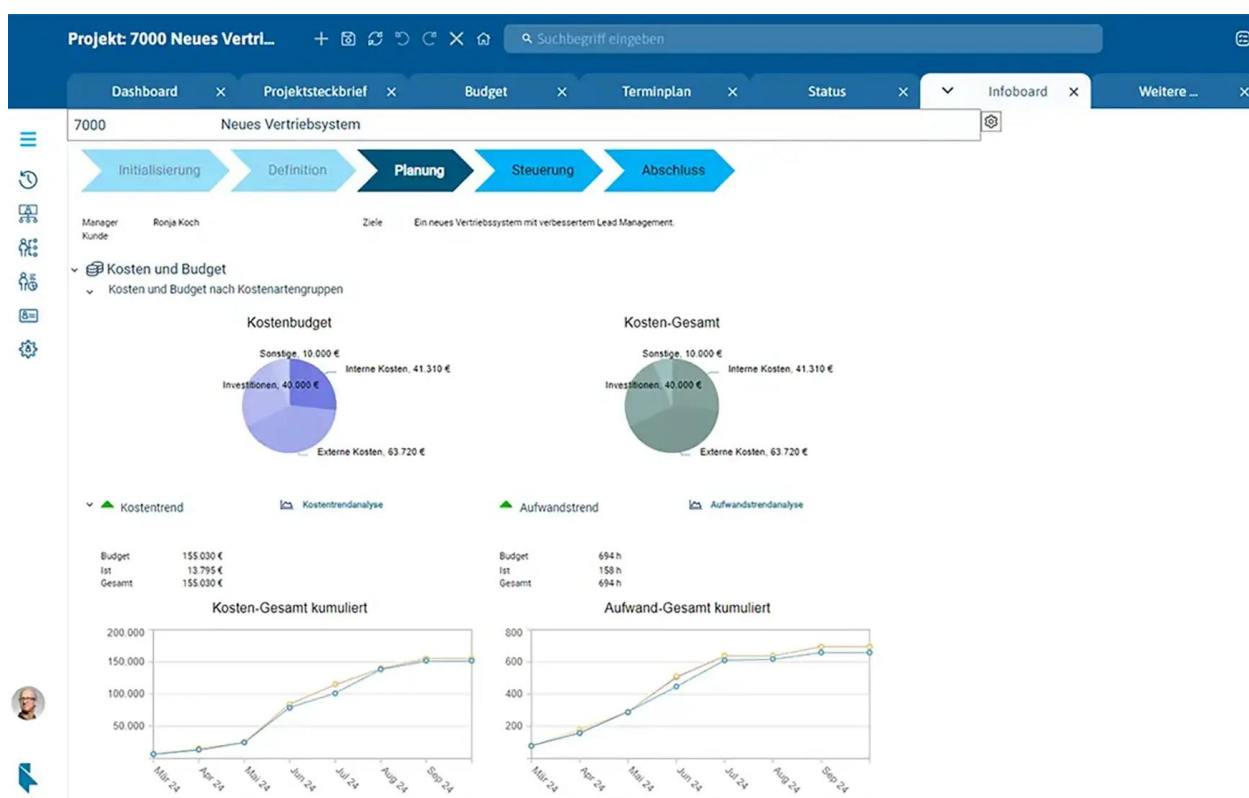


Рисунок 1.3 – Вебсторінка системи PLANTA Project

Microsoft Project - широко відома система управління проєктами, яка підтримує створення планів робіт, призначення ресурсів, облік витрат та контроль за виконанням. Вона дозволяє задавати бюджет, планувати трудовитрати,

моніторити відхилення та формувати звіти. Завдяки гнучкості підходить для науково-дослідних проєктів будь - якого масштабу [14].

OpenProject - це система з відкритим кодом для управління проєктами, яка містить функції для бюджетування, управління завданнями, відстеження часу та ресурсів. Її модуль бюджетування дозволяє задавати планові витрати на різні етапи виконання НДР та порівнювати їх з фактичними витратами. Підходить для команд, що прагнуть контролювати витрати з мінімальними фінансовими вкладеннями в інструменти. Система підтримує ролі й права доступу, що дозволяє організувати спільну роботу над проєктом у межах наукового колективу. OpenProject надає функціональність для формування детального бюджету з поділом за статтями витрат і автоматичного відслідковування перевищень. Вона також дозволяє інтегрувати дані про витрати з результатами робіт, що спрощує оцінку ефективності проведених досліджень (рис. 1.4) [15].

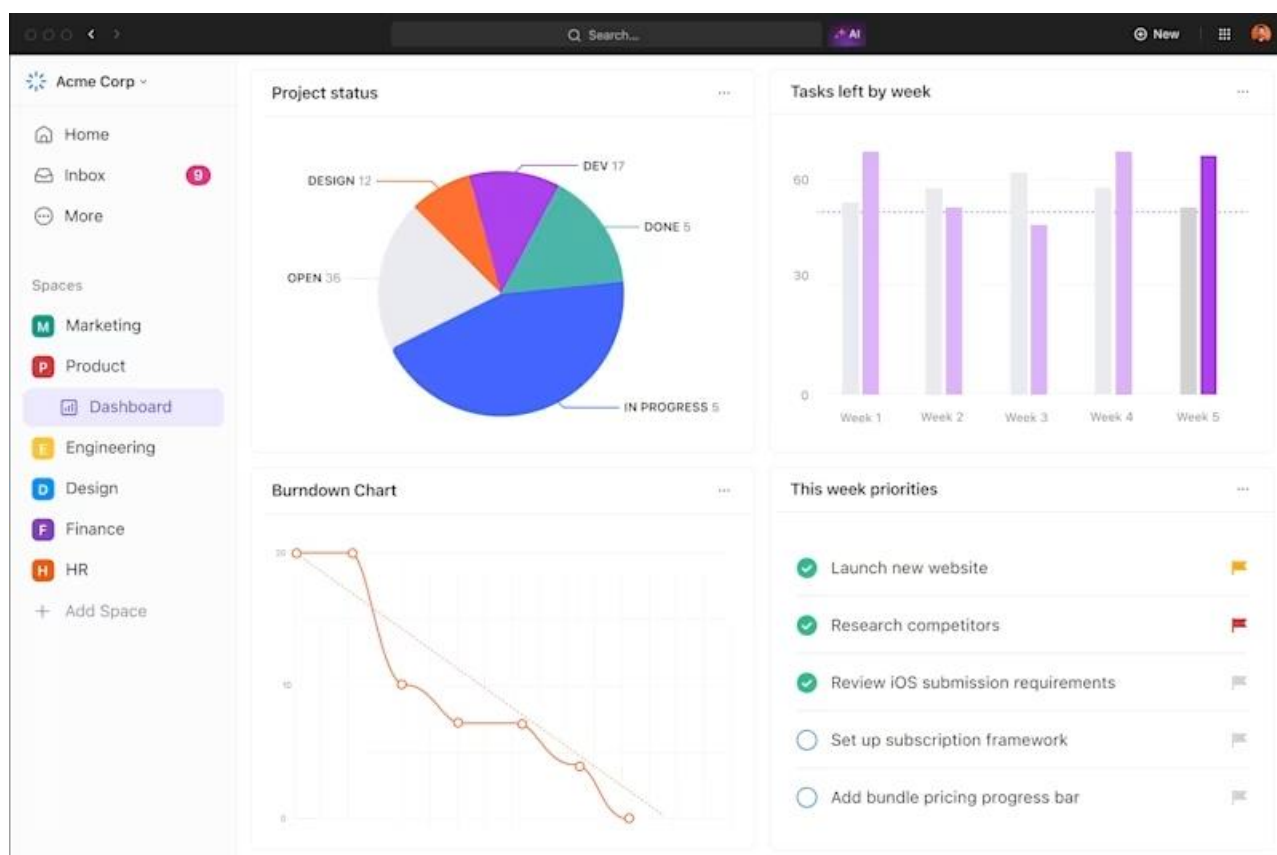


Рисунок 1.4 – Інтерфейсу інструменту керування проєктами OpenProject

Таким чином, аналіз сучасних програмних засобів свідчить про відсутність у широкому доступі універсального рішення, яке було б повністю адаптоване до потреб науково-дослідної діяльності. Більшість із доступних систем або надто загальні, або надто специфічні. Це створює передумови для розробки власного програмного модуля, що буде простим у використанні, адаптованим до методичних підходів, та орієнтованим саме на науково-дослідні проєкти [16].

1.3 Обґрунтування вибору методу розрахунку витрат на здійснення науково-дослідної роботи

На основі проведеного аналізу методичних підходів до розрахунку витрат та існуючих програмних засобів, було прийнято рішення використати інтегрований підхід до оцінки витрат для створення програмного модуля. Інтегрований підхід дозволяє поєднувати переваги кількох методів одночасно - зокрема, нормативного та факторного, що забезпечує баланс між точністю, простотою реалізації та гнучкістю в роботі з даними.

Нормативний компонент дає змогу застосовувати фіксовані значення, встановлені відповідними нормативними документами (наприклад, середні ставки оплати праці, нормативи використання матеріалів, коефіцієнти накладних витрат тощо). Такі дані легко організовуються у вигляді таблиць і зберігаються у базі або окремому конфігураційному файлі, що значно спрощує програмну реалізацію і забезпечує швидкий доступ до довідкової інформації.

Факторний компонент, у свою чергу, дозволяє враховувати змінні параметри конкретного проєкту: кількість виконавців, тривалість роботи, складність дослідження, тип обладнання тощо. Ці значення будуть вводитися користувачем через інтерфейс програми - з використанням простих форм із текстовими полями, випадаючими списками або перемикачами.

Поєднання нормативного і факторного методів створює універсальну модель, яка водночас точна й адаптивна. Такий підхід дає змогу будувати програмний модуль, який не потребує складної математики чи великої кількості

вкладених формул, і водночас дозволяє врахувати більшість необхідних аспектів розрахунку витрат.

Особливістю даної реалізації буде те, що програма матиме інтерфейс, орієнтований на простого користувача: лише найнеобхідніші поля для введення, логічне структурування розділів, підказки та автоматичне заповнення деяких значень на основі обраних параметрів. Це дозволить використовувати модуль не тільки фахівцям з фінансів чи управління проєктами, але й студентам, аспірантам та науковим співробітникам, які лише опановують основи економічного обґрунтування досліджень.

Крім того, інтегрований підхід дозволяє за необхідності легко масштабувати функціонал програми, додаючи нові параметри, види витрат або способи обліку. Наприклад, у подальших версіях модуля можна впровадити елементи експертного оцінювання або аналітичні звіти з графіками витрат на різні етапи проєкту. Таким чином, інтегрований підхід із домінуванням нормативно - факторного компонента є найбільш раціональним вибором для створення програмного модуля, що поєднує простоту використання, зручність розширення та відповідність актуальним вимогам щодо планування витрат на науково-дослідні роботи.

Ще однією перевагою обраного підходу є можливість легкого інтегрування з наявними інформаційними системами університетів або наукових установ. Це може бути реалізовано шляхом експорту результатів у формати CSV або Excel, що дозволить їх подальше використання в облікових системах, бухгалтерії або при складанні звітів для грантодавців.

Серед функціональних можливостей програмного модуля, що буде створений на основі цього підходу, слід зазначити: автоматичне обчислення загальної суми витрат, розрахунок витрат за етапами проєкту, формування структурованого кошторису з можливістю його редагування та експорту, а також опцію генерації звіту в зручному для друку вигляді. Все це сприятиме підвищенню ефективності планування та прозорості витрат науково-дослідної діяльності.

Варто підкреслити, що реалізація такого підходу також сприяє підвищенню рівня фінансової грамотності серед науковців, адже користувачі не лише

отримують готові цифри, а й можуть побачити, як саме формується структура витрат, які чинники мають на неї вплив, і як змінюється загальна вартість залежно від введених параметрів.

Таким чином, обраний інтегрований підхід із використанням нормативно - факторного механізму дозволяє створити ефективний, адаптивний і зручний у використанні інструмент для розрахунку витрат на науково-дослідні роботи, що відповідає сучасним вимогам та очікуванням користувачів.

1.4 Постановка задачі

На основі аналізу сучасних методичних підходів до розрахунку витрат на науково-дослідну діяльність та програмних засобів, що використовуються з цією метою, було сформульовано необхідність створення програмного модуля для спрощеного й водночас точного обчислення витрат на проведення науково-дослідної роботи.

Метою створення програмного модуля є розробка інструменту, що дозволяє здійснювати комплексний розрахунок витрат із використанням нормативно - факторного підходу:

$$Y = f(x), \quad (1.16)$$

де: Y – загальні витрати на здійснення науково дослідної роботи, які залежать від функцій витрат.

$X(Z_o, Z_p, Z_{\text{дод}}, Z_n, M, K_v, B_{\text{спец}}, B_{\text{прг}}, A_{\text{обл}}, B_e, B_{\text{св}}, B_{\text{сп}}, I_v, B_{\text{нзв}})$ – вхідні дані, де:

Z_o – основна заробітна плата дослідників;

Z_p – основна заробітна плата робітників;

$Z_{\text{дод}}$ – додаткова заробітна плата дослідників та робітників;

Z_n – відрахування на соціальні заходи;

M – матеріали;

K_v – витрат на комплектуючі;

$B_{\text{спец}}$ — спецустаткування для наукових (експериментальних) робіт;

$B_{\text{прг}}$ — програмне забезпечення для наукових (експериментальних) робіт;

$A_{\text{обл}}$ — амортизація обладнання, програмних засобів та приміщень;

B_e — паливо та енергія для науково - виробничих цілей;

$B_{\text{св}}$ — службові відрядження;

$B_{\text{сп}}$ — витрати на роботи, які виконують сторонні підприємства, установи і організації;

I_v — інші витрати;

$B_{\text{нзв}}$ — накладні (загальновиробничі) витрати.

Очікуваним результатом є створення програмного модуля з інтуїтивно зрозумілим інтерфейсом, що дозволяє швидко здійснити розрахунок загальної вартості науково-дослідного проекту і сформулювати кошторис з розподілом витрат за категоріями.

1.5 Висновок до розділу 1

У першому розділі здійснено аналіз основних методичних підходів до розрахунку витрат на здійснення науково-дослідної роботи. На основі порівняльної оцінки обґрунтовано вибір інтегрованого підходу з домінуванням нормативно - факторного компонента для програмної реалізації. Проведено аналіз сучасних програмних засобів, які можуть бути використані для розрахунку витрат у сфері НДР. З'ясовано, що жодна з існуючих систем не є універсальною для потреб науково-дослідної діяльності, що створює передумови для розробки програмного модуля з розширеними функціональними можливостями, адаптованого під специфіку таких проектів. Постановка задачі конкретизує мету, завдання та очікувані результати від створення програмного модуля для розрахунку витрат на НДР. Це стане основою для подальших етапів дослідження, зокрема для проектування архітектури програмного продукту та розробки його функціональних складових.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ РОЗРАХУНКУ ВИТРАТ НА ЗДІЙСНЕННЯ НАУКОВО-ДОСЛІДНОЇ РОБОТИ

2.1 Розробка алгоритмів роботи програмного модуля

Розробка алгоритмів є ключовим етапом побудови програмного модуля, що забезпечує обчислення загальних витрат на здійснення науково-дослідної роботи. Алгоритм розробляється відповідно до структури витрат, визначених у попередньому розділі роботи, та відображає логічну послідовність дій користувача та програмної системи [17].

На початковому етапі роботи користувач вводить вхідні дані через інтерфейс системи: це можуть бути посадові оклади, кількість днів або годин роботи, обсяги матеріалів, тарифні ставки, коефіцієнти, нормативи тощо. Далі програмний модуль послідовно виконує розрахунки кожної статті витрат відповідно до передбачених алгоритмів: спочатку розраховується основна та додаткова заробітна плата, потім здійснюється обчислення відрахувань, вартості матеріалів, енергії, послуг сторонніх організацій, амортизації, накладних витрат та інше. На завершальному етапі всі часткові результати підсумовуються для визначення загальної суми витрат на здійснення науково-дослідної роботи, після чого сформовані дані зберігаються у звіт. Такий підхід дозволяє забезпечити автоматизоване, точне та повторюване виконання обчислень. Таким чином, на основі вхідних параметрів користувач може отримати розрахунок повної вартості виконання науково-дослідної роботи. Кожен з наведених алгоритмів буде реалізовано в окремому функціональному модулі програмної системи.

На рисунках 2.1 та 2.2 наведено UML - діаграм діяльності, яка ілюструє логіку розрахунку загальних витрат на основі вхідних параметрів:

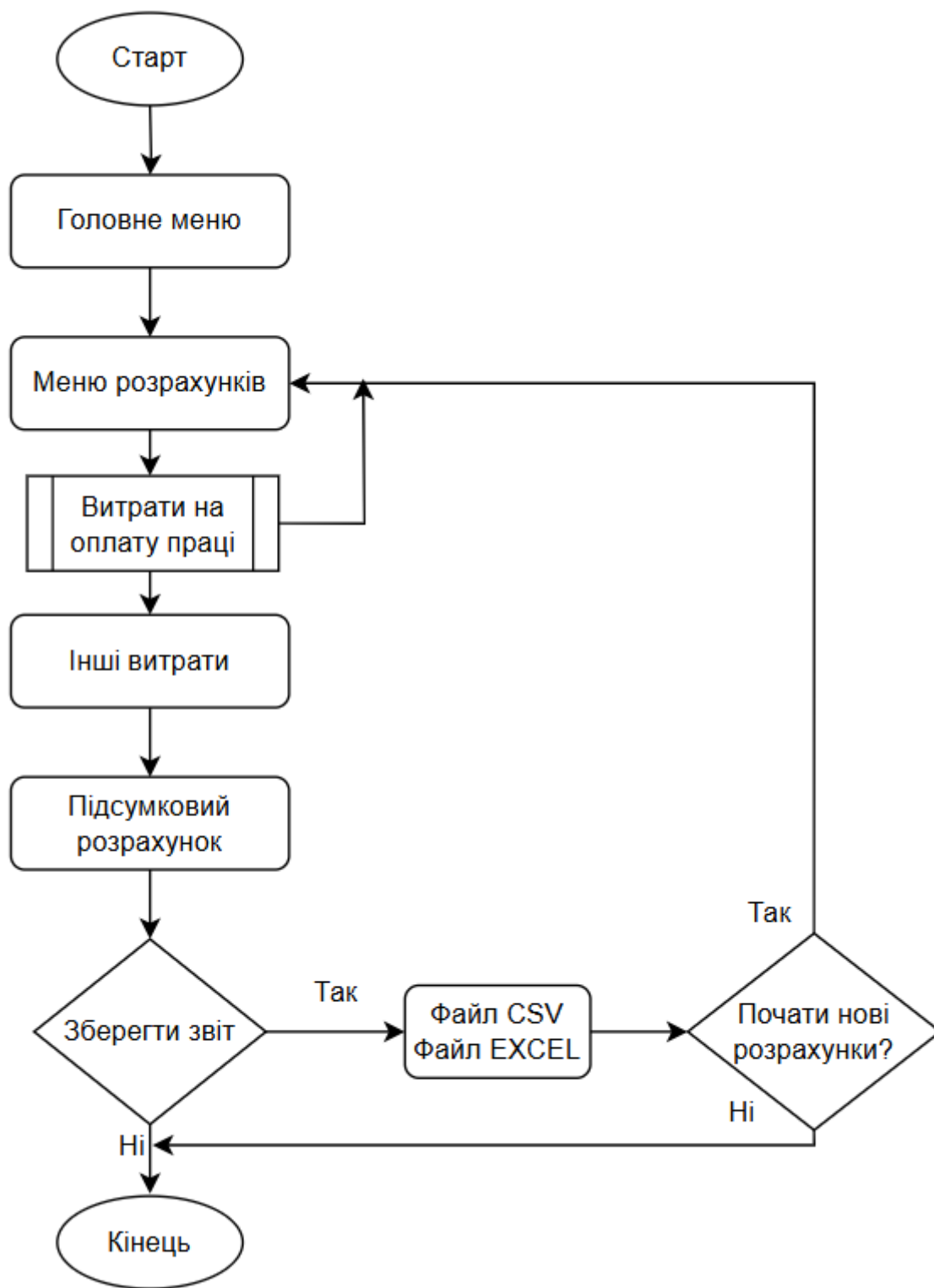


Рисунок 2.1 – UML - діаграма діяльності алгоритму розрахунку загальних витрат на здійснення НДР

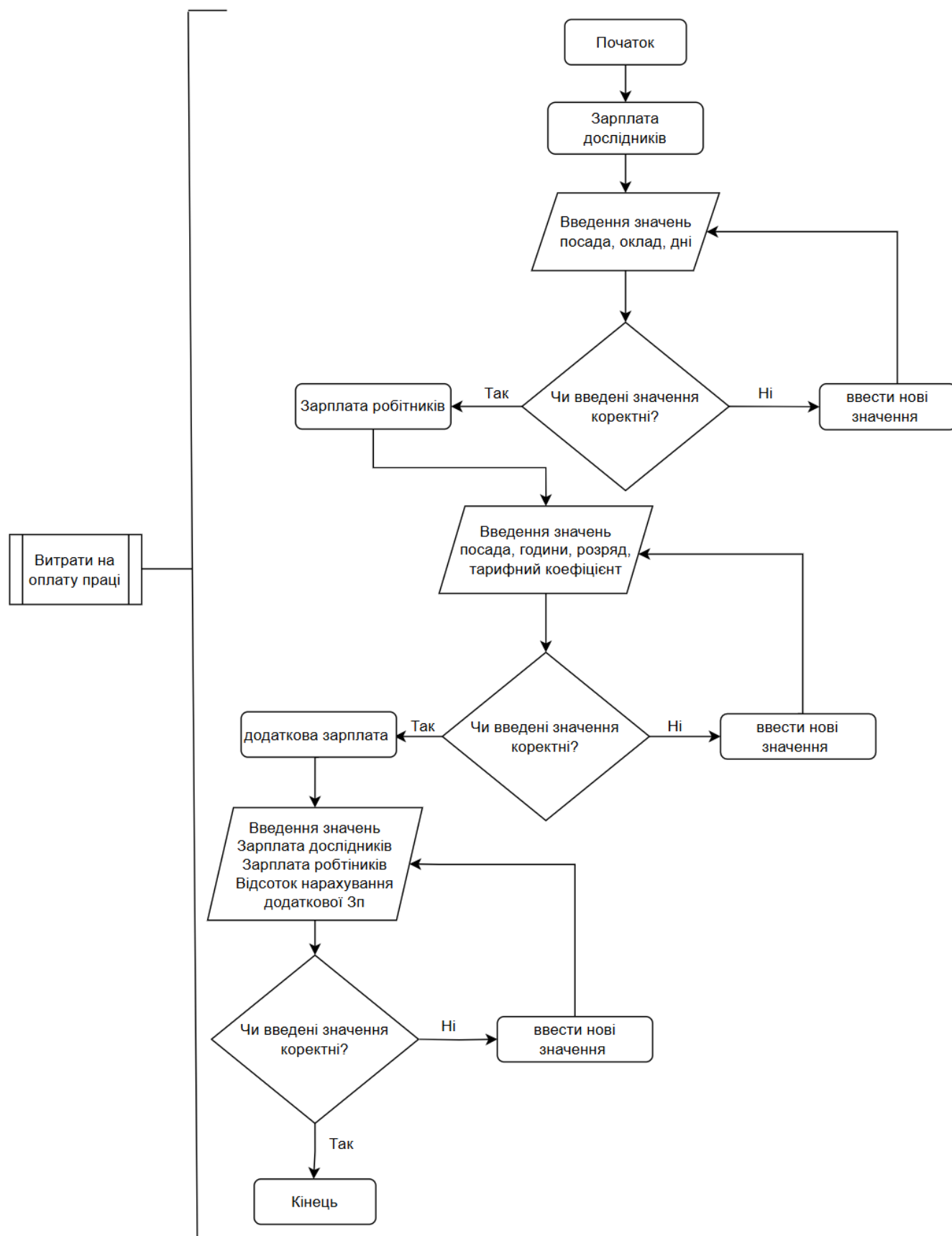


Рисунок 2.2 – UML - діаграма діяльності алгоритму розрахунку заробітної плати

2.2 Проєктування структури програмного модуля

Одним із ключових етапів розробки програмного забезпечення є побудова логічної архітектури, яка забезпечує ефективну організацію коду, спрощує подальший супровід системи та дозволяє легко вносити зміни у функціональність. Грамотно спроектована структура модуля дає змогу досягти високої гнучкості, повторного використання компонентів і забезпечення чіткої логіки виконання розрахунків [18].

Для забезпечення гнучкості, масштабованості та підтримуваності програмного модуля, його архітектура проєктується у вигляді окремих логічних компонентів, кожен з яких відповідає за реалізацію конкретної функціональності. Кожна стаття витрат, визначена в попередніх підрозділах, реалізується як окремий функціональний блок, який приймає на вхід необхідні параметри та повертає результат розрахунку [19].

Архітектура модуля поділена на такі основні компоненти:

1. Модуль введення даних – приймає інформацію від користувача (оклади, коефіцієнти, кількість днів, обсяги ресурсів тощо).
2. Модулі обчислення витрат – відповідають за проведення розрахунків окремих статей витрат відповідно до визначених економічних формул.
3. Модуль агрегації витрат – об'єднує всі результати обчислень у загальну суму витрат.
4. Модуль формування звітності – генерує фінальний звіт по категоріях витрат.
5. Модуль збереження – експортує дані у файл або базу даних.

На рисунку 2.1 подано спрощене графічне представлення структури програмного модуля розрахунку витрат:

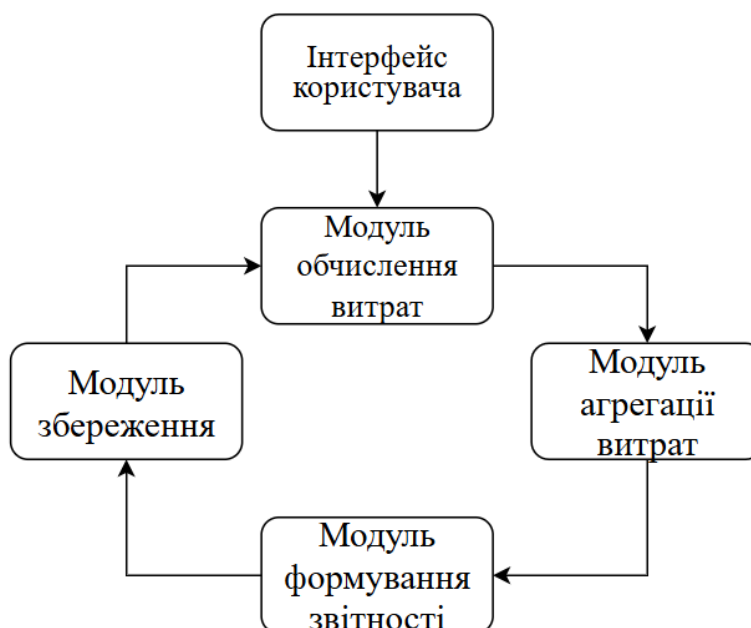


Рисунок 2.3 – Структура програмного модуля розрахунку витрат

2.3 Розробка UML діаграми класів

UML - діаграма класів є важливою частиною моделювання структури програмного модуля, що дає змогу формалізувати логіку його роботи та взаємодію між об'єктами. Вона наочно демонструє, як структуровано програмне забезпечення з точки зору об'єктно - орієнтованого підходу, відображаючи не лише перелік сутностей (класів), але й зв'язки між ними, їхні атрибути та методи [20].

У контексті розробки програмного модуля для розрахунку витрат на здійснення науково-дослідної роботи UML - діаграма класів слугує інструментом для візуалізації внутрішньої архітектури системи. Вона дозволяє зрозуміти, яким чином організовано зберігання, обробку та виведення інформації, а також яким чином окремі функціональні компоненти взаємодіють між собою.

На рисунку 2.4 зображено структуру класів, які реалізують ключові функціональні блоки модуля. Зокрема, тут представлено класи, відповідальні за взаємодію з користувачем, обчислення витрат за різними статтями, підсумовування результатів, формування звітів, а також збереження й завантаження даних. Кожен клас виконує окрему логічну функцію, що відповідає принципу єдиної

відповідальності, завдяки чому структура модуля залишається прозорою, гнучкою та масштабованою.

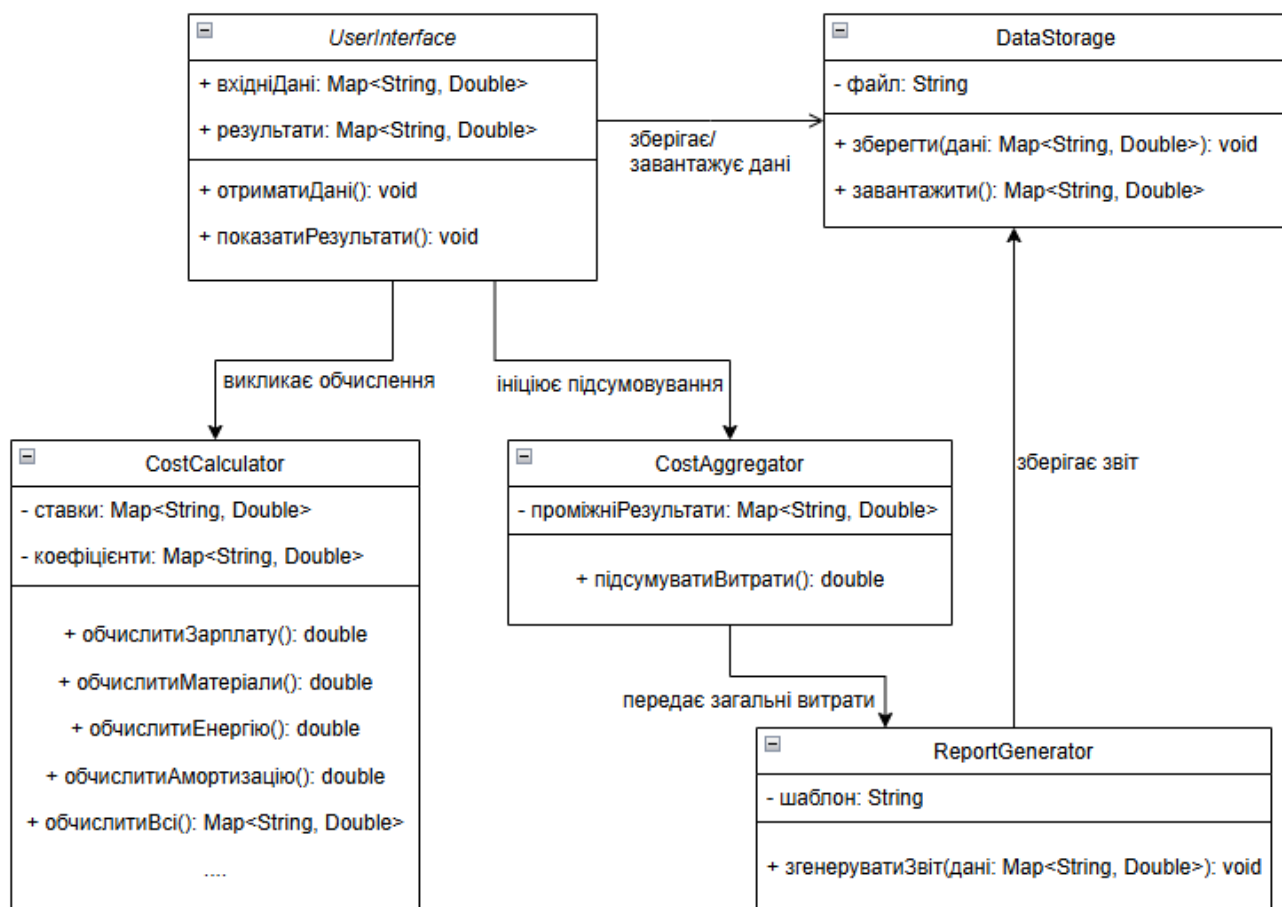


Рисунок 2.5 – UML - діаграма класів програмного модуля розрахунку витрат на НДР

Клас `UserInterface` – відповідає за взаємодію користувача із системою. Він приймає вхідні дані від користувача, ініціює процес розрахунку, а також відображає результати.

Клас `CostCalculator` – виконує обчислення витрат за кожною зі статей витрат (зарплата, матеріали, енергія, амортизація тощо).

Клас `CostAggregator` – підсумовує проміжні результати, отримані від `CostCalculator`, та обчислює загальні витрати.

Клас `ReportGenerator` – формує звіт за результатами розрахунків у зручному для користувача форматі.

Клас `DataStorage` – забезпечує збереження та завантаження даних. Використовується на етапі введення та виведення інформації.

Зв'язки між класами реалізовані таким чином: клас `UserInterface` ініціює взаємодію із `CostCalculator` та `CostAggregator`, а також виконує збереження і завантаження даних через `DataStorage`. `CostAggregator` передає підсумкові значення у `ReportGenerator`. `ReportGenerator`, у свою чергу, може також взаємодіяти з `DataStorage` для збереження згенерованого звіту.

Подібна структура дає змогу реалізувати програмний модуль, що відповідає принципам модульності та гнучкості, а також забезпечує простоту в обслуговуванні та майбутньому масштабуванні.

2.4 Висновок до розділу 2

У даному розділі здійснено проєктування програмного модуля розрахунку витрат на здійснення науково-дослідної роботи. Розроблено алгоритм функціонування системи, що включає етапи введення вихідних даних, обчислення окремих статей витрат, їх агрегування та формування звіту. В основу модуля покладено нормативну методику розрахунку витрат, згідно з якою враховуються основна та додаткова заробітна плата, матеріали, енергія, амортизація, інші витрати тощо. В межах проєктування побудовано основні UML - діаграми, що відображають логіку роботи системи.

Запропонована структура модуля відповідає принципам модульності, інкапсуляції та повторного використання. Такий підхід забезпечує можливість масштабування, підтримки та вдосконалення системи відповідно до змін у нормативно - правовій базі або внутрішніх вимог користувача.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ ВИТРАТ НА ЗДІЙСНЕННЯ НАУКОВО-ДОСЛІДНОЇ РОБОТИ

3.1 Обґрунтування вибору мови програмування та спеціалізованих бібліотек

На етапі планування програмного забезпечення для розрахунку витрат на виконання науково-дослідної роботи було розглянуто декілька мов програмування, кожна з яких має свої переваги та недоліки залежно від сфери застосування. Основна мета - обрати мову, яка забезпечить: швидкість розробки; простоту реалізації GUI; ефективну обробку табличних даних; зручне збереження результатів у поширених форматах (CSV, Excel); простоту розгортання для користувача.

Python - це інтерпретована мова програмування високого рівня, яка славиться своєю простотою та зручністю для розробників. Вона широко використовується в галузях наукових обчислень, машинного навчання, веб - розробки та автоматизації. Python має велику кількість бібліотек, що значно пришвидшує розробку складних додатків [21].

Java - об'єктно - орієнтована мова програмування, що забезпечує високу продуктивність, багатопоточність і платформну незалежність завдяки віртуальній машині Java (JVM). Java часто використовується для корпоративного програмного забезпечення, мобільних застосунків та великих клієнт - серверних систем [22].

C# (.NET) - це мова програмування від компанії Microsoft, що працює у середовищі .NET. Вона надає широкі можливості для створення настільних і веб - застосунків, але найбільш ефективна в середовищі Windows. Підтримує сучасні парадигми програмування та має розвинену систему розробки інтерфейсу [23].

JavaScript (Electron) - це мова сценаріїв, яка здебільшого використовується для веб - розробки. За допомогою фреймворку Electron можна створювати кросплатформенні настільні застосунки, але розмір таких додатків зазвичай є великим, а сам підхід - менш ефективним для простих задач [24].

Таблиця 3.1 – Порівняльний аналіз мов програмування

Критерій	Python	Java	C# (.NET)	JavaScript (Electron)
Простота синтаксису	Висока	Середня	Середня	Висока
Розробка GUI	Tkinter (проста реалізація)	JavaFX, Swing (складніші)	Windows Forms, WPF (залежність від ОС)	Electron (великий розмір додатка)
Обробка табличних даних	pandas, csv – висока гнучкість	Потребує додаткових бібліотек	Потребує бібліотек типу EPPlus	Обмежено, потрібні сторонні бібліотеки
Збереження в Excel / CSV	Пряма підтримка	Через Apache POI	Через сторонні .NET - бібліотеки	Через Node.js модулі
Швидкість розробки прототипу	Висока	Низька - середня	Середня	Висока
Кросплатформеність	Так	Так	Обмежена	Так
Складність розгортання	Низька	Середня	Висока	Висока
Популярність у науці / освіті	Дуже висока	Середня	Низька	Низька

На основі даних з таблиці 3.1 було обрано мову програмування Python як основну для реалізації програмного модуля. Python є одним із найпопулярніших інструментів сучасної розробки програмного забезпечення, і це обґрунтоване рішення з огляду на специфіку завдань, поставлених у межах цього проєкту.

Однією з головних переваг Python є простий та інтуїтивно зрозумілий синтаксис, що дозволяє швидко приступити до розробки навіть складних обчислювальних алгоритмів. Завдяки цьому мову часто використовують у наукових та технічних проєктах, де важливо мінімізувати час розробки без втрати якості реалізації.

Python також має вбудовану підтримку створення графічного інтерфейсу користувача (GUI) через стандартну бібліотеку tkinter. Це дає змогу створювати візуально зручні вікна та елементи управління без необхідності залучення зовнішніх фреймворків або бібліотек. Така функціональність особливо корисна при створенні прикладних програм з багатьма формами вводу та результатами обчислень.

Ще однією сильною стороною Python є ефективна обробка табличних даних завдяки бібліотеці pandas, яка дозволяє зручно структурувати, фільтрувати й обробляти інформацію. Це критично важливо для реалізації розрахунків витрат, оскільки дані можуть бути зчитані з таблиць, змінені за потребою користувача, а потім використані для обчислень або збереження.

Python також забезпечує прості механізми збереження результатів у різноманітних форматах, зокрема CSV та Excel (.xlsx). Це дає змогу користувачеві експортувати результати обчислень у формат, зручний для подальшого аналізу, друку або архівування.

Не менш важливою перевагою є платформна незалежність Python. Створена програма може бути запущена як на Windows, так і на Linux або macOS без необхідності значних змін у коді, що робить розробку більш гнучкою і масштабованою.

Крім того, велика спільнота користувачів Python забезпечує доступ до численних прикладів, документації та рішень типових проблем, що істотно прискорює процес розробки та налагодження.

Таким чином, Python повністю відповідає вимогам проєкту як у частині розрахункової логіки, так і з точки зору зручності взаємодії з користувачем та підтримки стандартів сучасного програмного забезпечення.

Одним із ключових факторів, що вплинув на вибір Python, стала наявність спеціалізованих бібліотек, які охоплюють увесь необхідний функціонал: від побудови графічного інтерфейсу до експорту звітів. Нижче наведено опис основних бібліотек, використаних у межах цього проєкту:

Бібліотека `tkinter` використовується для створення графічного інтерфейсу користувача (GUI). У межах цього проєкту задіяно компоненти `tk`, `ttk`, `Label`, `Entry`, `Button`, `StringVar`, `messagebox`, `filedialog`, `Menu`, `Frame`, які дозволяють реалізувати зручну і гнучку структуру вікон, полів для введення, кнопок та діалогових повідомлень. Завдяки цьому користувач отримує інтуїтивно зрозумілий інтерфейс для введення початкових даних, запуску розрахунків і збереження результатів.

Бібліотека `csv` - це стандартна бібліотека Python, що використовується для експорту даних у формат CSV. У даному проєкті вона застосовується для збереження результатів розрахунків у вигляді текстових таблиць, сумісних з електронними таблицями, що дозволяє швидко переглядати й аналізувати результати.

Бібліотека `pandas` застосовується для структурування, обробки і збереження табличних даних. Вона є ключовим інструментом для роботи з великими обсягами числової інформації. У межах розробки модуля `pandas` використовується для формування звітів, які експортуються у формат Excel (`.xlsx`).

Бібліотека `openpyxl` опосередковано використовується через `pandas` як рушій для запису даних у формат Excel. Вона забезпечує коректну генерацію файлів `.xlsx`, які можуть відкриватися в Microsoft Excel або аналогічних програмах без втрати форматування і структури.

Поєднання зазначених бібліотек дозволяє реалізувати повноцінний функціональний модуль, який охоплює збір вхідних даних, їх обробку, виведення результатів і формування звітності. Вибір саме цих бібліотек був зумовлений їх простотою інтеграції, стабільністю, сумісністю з Python та відкритістю для подальшої розробки.

3.2 Обґрунтування вибору середовища програмування

Для реалізації програмного модуля розрахунку витрат на здійснення науково-дослідної роботи необхідно було обрати зручне, стабільне та функціональне середовище програмування, яке б відповідало поставленим

вимогам до розробки прикладного програмного забезпечення з графічним інтерфейсом.

Було розглянуто кілька популярних середовищ програмування, зокрема Visual Studio Code, PyCharm, Sublime Text та Thonny. Для порівняння було враховано основні критерії, такі як підтримка Python, швидкодія, гнучкість налаштувань, інтеграція з Git, відлагодження, зручність інтерфейсу та занесено їх до таблиці 3.2.

Таблиця 3.2 – Порівняльний аналіз середовищ програмування

Критерій	Visual Studio Code	PyCharm	Sublime Text	Thonny
Підтримка Python	Відмінна (через розширення)	Вбудована	Часткова (через плагіни)	Вбудована
Швидкодія	Висока	Середня	Дуже висока	Висока
Гнучкість налаштувань	Висока	Висока	Середня	Низька
Інтеграція з Git	Повна	Повна	Обмежена	Немає
Зручність інтерфейсу	Інтуїтивна	Професійна	Мінімалістична	Проста для початківців
Кросплатформеність	Так	Так	Так	Так

Серед усіх розглянутих варіантів найбільш оптимальним середовищем для розробки став Visual Studio Code. Він поєднує в собі високу швидкодію, гнучкість у налаштуваннях, підтримку численних розширень, у тому числі для Python, Git, запуску й відлагодження коду. Також він є безкоштовним, кросплатформенним та зручним у використанні.

Таким чином, Visual Studio Code повністю відповідає вимогам проєкту, сприяє ефективній реалізації програмного модуля та забезпечує комфортну роботу на всіх етапах розробки.

3.3 Розробка функціоналу програмного модуля розрахунку витрат на здійснення науково-дослідної роботи

Програмний модуль розрахунку витрат на здійснення науково-дослідної роботи реалізований у вигляді графічного інтерфейсу користувача, який дозволяє вводити вихідні дані, виконувати розрахунки за кожним з напрямів витрат та формувати підсумковий звіт. Основна мета - забезпечити інтуїтивну взаємодію користувача з інструментами розрахунку та обробки даних у межах заданої структури НДР.

Інтерфейс користувача побудований за допомогою бібліотеки `tkinter`. Головне вікно містить меню, через яке користувач може обрати потрібний розділ витрат. Кожен пункт меню відкриває окреме вікно з формами для введення даних, розрахунковими кнопками та відображенням результатів.

На рисунку 3.1 зображено фрагмент коду, який відповідає за формування головного меню програми.

У наведеному фрагменті використовується модуль `tkinter`, зокрема елемент `Menu`, для створення меню навігації. Функція `build_global_menu(root)` відповідає за додавання усіх розділів витрат, таких як «Витрати на оплату праці», «Сировина та матеріали», «Паливо та енергія» тощо. Для кожного пункту меню задається відповідна функція, яка відкриває нове вікно із формою для введення та розрахунків.

Загальна структура програмного модуля передбачає поділ логіки на окремі функціональні блоки, кожен з яких реалізовано в окремому модулі Python. У наступному підрозділі буде детально описано кожен з функцій, що реалізують відповідні розрахунки витрат.

```

# ui/menu.py
import tkinter as tk
from tkinter import Menu
from ui.pages.report_window import open_report_window #  Додаємо импорт звіту

def build_global_menu(root):
    from ui.salary_block_ui import open_salary_block
    from modules.social_tax import SocialTaxCalculator
    from modules.material_cost import MaterialCostCalculator
    from modules.component_cost import ComponentsCostCalculator
    from modules.equipment_cost import EquipmentCostCalculator
    from modules.software_cost import SoftwareCostCalculator
    from modules.amortization_cost import AmortizationCalculator
    from modules.energy_cost import EnergyCostCalculator
    from modules.travel_expenses import TravelExpensesCalculator
    from modules.external_services import ExternalServicesCalculator
    from modules.other_expenses import OtherExpensesCalculator
    from modules.overhead_expenses import OverheadExpensesCalculator
    from modules.total_cost import TotalCostCalculator

    # Меню-бар
    menu_bar = Menu(root)
    nav_menu = Menu(menu_bar, tearoff=0)

    nav_menu.add_command(label="Витрати на оплату праці", command=lambda: open_salary_block(root))
    nav_menu.add_command(label="Відрахування на соціальні заходи", command=lambda: SocialTaxCalculator(root))
    nav_menu.add_command(label="Сировина та матеріали", command=lambda: MaterialCostCalculator(root))
    nav_menu.add_command(label="Розрахунок витрат на комплектуючі", command=lambda: ComponentsCostCalculator(root))
    nav_menu.add_command(label="Спецустаткування для наукових робіт", command=lambda: EquipmentCostCalculator(root))
    nav_menu.add_command(label="Програмне забезпечення для наукових робіт", command=lambda: SoftwareCostCalculator(root))
    nav_menu.add_command(label="Амортизація обладнання", command=lambda: AmortizationCalculator(root))
    nav_menu.add_command(label="Паливо та енергія", command=lambda: EnergyCostCalculator(root))
    nav_menu.add_command(label="Службові відрядження", command=lambda: TravelExpensesCalculator(root))
    nav_menu.add_command(label="Сторонні роботи", command=lambda: ExternalServicesCalculator(root))
    nav_menu.add_command(label="Інші витрати", command=lambda: OtherExpensesCalculator(root))
    nav_menu.add_command(label="Накладні витрати", command=lambda: OverheadExpensesCalculator(root))
    nav_menu.add_command(label="Підсумковий розрахунок", command=lambda: TotalCostCalculator(root))

    nav_menu.add_separator()
    nav_menu.add_command(label="Звіт", command=lambda: open_report_window(root)) #  Додано пункт звіту

    menu_bar.add_cascade(label="☰ Меню", menu=nav_menu)
    root.config(menu=menu_bar)

```

Рисунок 3.1 – Фрагмент коду створення головного меню програми (файл menu.py)

Функціональний блок «Витрати на оплату праці» реалізований у вигляді окремого вікна, яке викликається з головного меню. У межах цього блоку користувач послідовно виконує три розрахунки: основної заробітної плати (Z_o), додаткової заробітної плати (Z_d) та загальної заробітної плати (Z_p).

На рисунку 3.2 зображено фрагмент коду модуля, що відповідає за відкриття вікна для розрахунку заробітної плати. Вікно будується на основі бібліотеки tkinter і містить виклики функцій, що відповідають за обчислення кожної з трьох формул. Також реалізовано передачу результатів між блоками через змінні - посилання.

```

def open_salary_block(root):
    for widget in root.winfo_children():
        widget.destroy()

    build_global_menu(root)

    tk.Label(root, text="Витрати на оплату праці", font=("Arial", 16, "bold")).pack(pady=10)

    SalaryZoCalculator(root)
    SalaryZpCalculator(root)

    zdod_block = SalaryZdodCalculator(root)

    set_zdod_block_zo(zdod_block) # 🔄 передаємо в Zo
    set_zdod_block_zp(zdod_block) # ✅ передаємо в Zp

    tk.Button(root, text="Назад", command=lambda: show_main_menu(root)).pack(pady=10)

```

Рисунок 3.2 – Фрагмент модуля, що реалізує блок розрахунку заробітної плати

Функції `SalaryZoCalculator`, `SalaryZpCalculator` та `SalaryZdodCalculator` викликаються у відповідному порядку. Їхня логіка розділена по файлах, що відповідає принципу модульності й полегшує тестування та подальшу підтримку коду. Отримані значення передаються далі для обчислення соціальних внесків та підсумкового розрахунку.

Модуль «Відрахування на соціальні заходи» реалізує розрахунок обов'язкових внесків із заробітної плати відповідно до встановленого відсотка. Інтерфейс дає змогу користувачеві ввести значення основної заробітної плати дослідників, заробітної плати інших працівників, додаткової зарплати та відсоток соціального внеску.

На рисунку 3.3 представлено фрагмент коду, який реалізує вікно розрахунку соціальних відрахувань. Користувач вводить дані у відповідні поля, а результат відображається у графічному інтерфейсі після натискання кнопки.

Обчислений результат зберігається у спільному сховищі даних програми за допомогою функції `set_result`, що дозволяє повторно використовувати його в інших модулях, зокрема у підсумковому обчисленні загальних витрат. Передбачено обробку помилок введення даних, що підвищує надійність модуля.

```

def SocialTaxCalculator(root):
    for widget in root.winfo_children():
        widget.destroy()
    build_global_menu(root)

    root.title("Розрахунок  $Z_n$  – соціальні відрахування")

    def calculate_social_tax():
        try:
            Z_o = float(entry_Zo.get())
            Z_r = float(entry_Zr.get())
            Z_dod = float(entry_Zdod.get())
            H_soc = float(entry_Hsoc.get())

            Zn = (Z_o + Z_r + Z_dod) * H_soc / 100
            result_var.set(f"Відрахування  $Z_n$ : {Zn:.2f} грн")
            set_result(" $Z_n$ ", Zn) # ← Зберігаємо результат
        except ValueError:
            messagebox.showerror("Помилка", "Перевірте правильність введених чисел")

    Label(root, text="Z_o (ЗП дослідників)").grid(row=0, column=0, sticky="e")
    entry_Zo = Entry(root)
    entry_Zo.grid(row=0, column=1)

    Label(root, text="Z_p (ЗП робітників)").grid(row=1, column=0, sticky="e")
    entry_Zr = Entry(root)
    entry_Zr.grid(row=1, column=1)

    Label(root, text="Z_доп (додаткова ЗП)").grid(row=2, column=0, sticky="e")
    entry_Zdod = Entry(root)
    entry_Zdod.grid(row=2, column=1)

    Label(root, text="H_зп (%)").grid(row=3, column=0, sticky="e")
    entry_Hsoc = Entry(root)
    entry_Hsoc.insert(0, "22")
    entry_Hsoc.grid(row=3, column=1)

```

Рисунок 3.3 – Фрагмент модуля SocialTaxCalculator, що реалізує розрахунок соціального внеску

Модуль «Сировина та матеріали» реалізує функціонал для розрахунку витрат на матеріали, які використовуються під час виконання науково-дослідної роботи. Користувач має можливість ввести кількість видів матеріалів, а також відповідні параметри для кожного з них.

Після введення значень у таблицю параметрів, програма автоматично обчислює загальні витрати та відображає їх у графічному інтерфейсі. Передбачена

перевірка на коректність введення чисел, що забезпечує захист від помилок користувача.

На рисунку 3.4 представлено фрагмент коду, що реалізує функцію розрахунку витрат на матеріали, а також генерацію динамічної таблиці введення даних.

```
def MaterialCostCalculator(root):
    for widget in root.winfo_children():
        widget.destroy()

    build_global_menu(root)
    root.title("Розрахунок М – витрати на матеріали")

    # Ввід кількості рядків
    Label(root, text="Кількість видів матеріалів (n):").pack()
    row_count_var = StringVar(value="3")
    Entry(root, textvariable=row_count_var, width=5).pack()

    def calculate_material_cost():
        try:
            rows = []
            for i in range(int(row_count_var.get())):
                hj = float(entries[i]['hj'].get())
                cj = float(entries[i]['cj'].get())
                kj = float(entries[i]['kj'].get())
                bj = float(entries[i]['bj'].get())
                cebj = float(entries[i]['cebj'].get())
                rows.append(hj * cj * kj - bj * cebj)

            M = sum(rows)
            result_var.set(f"Загальні витрати на матеріали М = {M:.2f} грн")
            set_result("М", M)
        except ValueError:
            messagebox.showerror("Помилка", "Будь ласка, введіть коректні числові значення.")

    def generate_fields():
        for widget in input_frame.winfo_children():
            widget.destroy()
        entries.clear()
```

Рисунок 3.4 – Фрагмент модуля MaterialCostCalculator, що реалізує розрахунок витрат на матеріали

Модуль використовує структуру динамічного формування полів вводу залежно від кількості матеріалів. Обчислення результату реалізується за допомогою функції `calculate\_material\_cost`, а результат передається в загальне сховище для подальшого використання в підсумкових розрахунках.

Модуль «Комплектуючі» реалізує функціонал для розрахунку витрат на елементи обладнання, що використовуються в науково-дослідній роботі. Інтерфейс дозволяє вводити параметри кожного компонента, зокрема кількість одиниць, вартість за одиницю та коефіцієнт транспортування.

Після заповнення всіх полів даних користувачем програма виконує обчислення загальних витрат. Результат відображається в інтерфейсі програми, а також зберігається у внутрішньому сховищі даних для подальшого використання в підсумкових розрахунках.

На рисунку 3.5 подано фрагмент коду модуля ComponentsCostCalculator, що реалізує відповідну функціональність у вигляді динамічної таблиці та розрахунку суми по всім компонентам.

```
def ComponentsCostCalculator(root):
    for widget in root.winfo_children():
        widget.destroy()

    build_global_menu(root) # Додати меню

    root.title("Розрахунок Кс – витрати на комплектуючі")

    entries = []

    def calculate_components_cost():
        try:
            rows = []
            for i in range(int(row_count_var.get())):
                hj = float(entries[i]['hj'].get())
                cj = float(entries[i]['cj'].get())
                kj = float(entries[i]['kj'].get())
                rows.append(hj * cj * kj)
            Ke = sum(rows)
            result_var.set(f"Загальні витрати на комплектуючі Кс = {Ke:.2f} грн")
            set_result("Кс", Ke) # Зберігаємо для TotalCostCalculator
        except ValueError:
            messagebox.showerror("Помилка", "Будь ласка, введіть коректні числові значення.")

    def generate_fields():
        for widget in input_frame.winfo_children():
            widget.destroy()
        entries.clear()
        try:
            row_count = int(row_count_var.get())
            headers = ["Ні (кількість, шт)", "Ці (грн/шт)", "Кі (коэф. транспортування)"]
            for col, header in enumerate(headers):
                Label(input_frame, text=header, font=("Arial", 9, "bold")).grid(row=0, column=col, padx=5, pady=2)
```

Рисунок 3.5 – Фрагмент модуля ComponentsCostCalculator, що реалізує розрахунок витрат на комплектуючі

Модуль забезпечує обробку помилок введення чисел через блок try - ехсерт та динамічне формування таблиці за кількістю компонентів, зазначених користувачем. Це робить взаємодію з формою гнучкою та зручною навіть при великій кількості позицій у розрахунку.

Модуль «Спецустаткування для наукових робіт» використовується для обчислення вартості обладнання, що буде використане під час виконання науково-дослідної роботи. Користувач вводить дані щодо вартості одиниці обладнання, кількості та коефіцієнта, що враховує супутні витрати на монтаж або налаштування.

Після заповнення таблиці, програма обчислює балансову вартість усього устаткування. Результат зберігається у змінній для подальшого використання у підсумковому розрахунку загальних витрат на НДР.

На рисунку 3.6 показано фрагмент коду з модуля EquipmentCostCalculator, що реалізує логіку обчислення та обробки введених даних.

```
def EquipmentCostCalculator(root):
    for widget in root.winfo_children():
        widget.destroy()

    build_global_menu(root) # Додаємо меню

    root.title("Розрахунок В_д_пец – Спецустаткування для наукових робіт")

    rows = []

    def calculate_equipment_cost():
        try:
            total = 0
            for row in rows:
                price = float(row["price"].get())
                count = int(row["count"].get())
                coef = float(row["coef"].get())
                total += price * count * coef
            result_var.set(f"Балансова вартість устаткування: {total:.2f} грн")
            set_result("В_д_пец", total) # ← Зберігаємо результат
        except ValueError:
            messagebox.showerror("Помилка", "Перевірте правильність введених чисел")

    def update_rows():
        for widget in frame_rows.winfo_children():
            widget.destroy()
        rows.clear()
        try:
            count = int(entry_count.get())
        except ValueError:
            messagebox.showerror("Помилка", "Введіть кількість найменувань як число")
        return
```

Рисунок 3.6 – Фрагмент модуля EquipmentCostCalculator, що реалізує розрахунок вартості спецустаткування

Модуль реалізовано на основі бібліотеки tkinter з підтримкою обробки винятків та динамічним оновленням структури введення. Код забезпечує точність підрахунків і є придатним для масштабування під більші обсяги даних.

Модуль «Програмне забезпечення» відповідає за обчислення витрат на ліцензійні або спеціалізовані програмні продукти, які необхідні для реалізації науково-дослідної роботи. Користувач вводить кількість найменувань, а також для кожного ПЗ зазначає вартість, кількість одиниць та коригувальний коефіцієнт. Результат обчислень вартість програмного забезпечення виводиться на екран та зберігається для використання в підсумковому розрахунку. Модуль має вбудовану перевірку введених чисел на коректність і підтримує динамічне створення форми введення даних.

На рисунку 3.7 зображено фрагмент коду модуля SoftwareCostCalculator, де реалізовано введення, обробку та обчислення вартості ПЗ.

```
def SoftwareCostCalculator(root):
    for widget in root.winfo_children():
        widget.destroy()

    build_global_menu(root) # ← Додаємо глобальне меню

    root.title("Розрахунок В_пз – Програмне забезпечення")

    def calculate_B_pz():
        try:
            k = int(entry_k.get())
            total = 0.0
            for i in range(k):
                price = float(entry_grid[i][0].get()) # Ціна
                count = float(entry_grid[i][1].get()) # Кількість
                coef = float(entry_grid[i][2].get()) # Коефіцієнт
                total += price * count * coef
            result_var.set(f"Балансова вартість ПЗ: {total:.2f} грн")
            set_result("В_пз", total) # ← Зберігаємо результат
        except ValueError:
            messagebox.showerror("Помилка", "Будь ласка, введіть коректні числові значення.")

    def generate_inputs():
        for widget in frame_inputs.winfo_children():
            widget.destroy()
        try:
            k = int(entry_k.get())
        except ValueError:
            messagebox.showerror("Помилка", "Введіть кількість найменувань цілим числом.")
        return
```

Рисунок 3.7 – Фрагмент модуля SoftwareCostCalculator, що реалізує розрахунок вартості програмного забезпечення

Цей модуль демонструє підхід до структурування розрахунків у вигляді таблиці з динамічною кількістю рядків. Результати інтегруються в загальну систему обліку витрат НДР.

Модуль «Амортизація» призначений для розрахунку амортизаційних відрахувань на основі балансової вартості об'єкта, строку його використання в роках та фактичної тривалості використання в межах року (у місяцях).

Користувач заповнює відповідні поля введення, після чого система виконує розрахунок та виводить значення амортизаційних витрат у вказаному періоді. Результат зберігається у внутрішньому сховищі програми для подальшого узагальнення в підсумковому модулі.

На рисунку 3.8 представлено фрагмент коду модуля AmortizationCalculator, який реалізує обчислення відповідно до введених параметрів.

```
def calculate_amortization():
    try:
        Cz = float(entry_Cz.get())      # Цб – балансова вартість
        Tg = float(entry_Tg.get())      # Тв – строк використання, років
        t_vyk = float(entry_t_vyk.get()) # твик – тривалість використання, міс.

        A_obl = round((Cz / Tg) * (t_vyk / 12), 2)
        result_var.set(f"А_обл = {A_obl:.2f} грн")
        set_result("А_обл", A_obl)      # ← збереження в пам'ять
    except ValueError:
        messagebox.showerror("Помилка", "Перевірте правильність введених чисел")

Label(root, text="Цб (балансова вартість), грн").grid(row=0, column=0, sticky="w")
entry_Cz = Entry(root)
entry_Cz.grid(row=0, column=1)

Label(root, text="Тв (строк використання), років").grid(row=1, column=0, sticky="w")
entry_Tg = Entry(root)
entry_Tg.grid(row=1, column=1)

Label(root, text="твик (термін використання), міс.").grid(row=2, column=0, sticky="w")
entry_t_vyk = Entry(root)
entry_t_vyk.grid(row=2, column=1)
```

Рисунок 3.8 – Фрагмент модуля AmortizationCalculator, що реалізує розрахунок амортизації

Модуль також містить обробку помилок введення та реалізований у вигляді зручної форми з підписаними полями. Це дозволяє користувачу безпомилково ввести значення та отримати коректний результат розрахунку.

Модуль «Паливо та енергія» призначений для розрахунку витрат, пов'язаних зі споживанням енергетичних ресурсів у межах виконання науково-дослідної роботи. Інтерфейс дозволяє ввести параметри для кожного виду енергоресурсу, зокрема потужність, час роботи, ціну за одиницю, коефіцієнт використання та ККД. Після введення всіх необхідних даних система обчислює сумарну вартість витрат на енергію та зберігає результат для подальшого узагальнення. Передбачено динамічну генерацію введення по кожному джерелу та обробку помилок введення значень.

На рисунку 3.9 показано фрагмент коду модуля EnergyCostCalculator, де реалізовано обчислення загальних витрат на паливо та енергію.

```
def create_table():
    try:
        count = int(entry_count.get())
        for widget in frame_table.wininfo_children():
            widget.destroy()
        input_rows.clear()
        headers = ["W (кВт)", "t (год)", "Це (грн)", "К", "η", "Вартість (грн)"]
        for col, text in enumerate(headers):
            Label(frame_table, text=text, font=('Arial', 10, 'bold')).grid(row=0, column=col, padx=5, pady=2)

        for i in range(count):
            row = []
            for j in range(5):
                e = Entry(frame_table, width=10)
                e.grid(row=i+1, column=j, padx=5, pady=2)
                row.append(e)
            lbl = Label(frame_table, text="0.00")
            lbl.grid(row=i+1, column=5, padx=5, pady=2)
            row.append(lbl)
            input_rows.append(row)

    except ValueError:
        messagebox.showerror("Помилка", "Введіть ціле число кількості пристроїв")

Label(root, text="Кількість пристроїв:").grid(row=0, column=0)
entry_count = Entry(root, width=5)
entry_count.insert(0, "1")
entry_count.grid(row=0, column=1)
```

Рисунок 3.9 – Фрагмент модуля EnergyCostCalculator, що реалізує розрахунок витрат на паливо та енергію

Модуль підтримує обробку кількох рядків введення даних та обчислює загальну суму витрат з урахуванням ефективності використання енергії. Це дозволяє враховувати реальні показники споживання у процесі розрахунків.

Модуль «Службові відрядження» призначений для розрахунку витрат, пов'язаних із відправленням працівників у службові поїздки під час виконання науково-дослідної роботи. Користувач вводить показники основної та додаткової заробітної плати, а також коефіцієнт витрат на відрядження у відсотках.

На основі введених даних система виконує розрахунок витрат на відрядження за допомогою простої арифметичної формули. Результат автоматично виводиться у вікні та зберігається у змінній для подальшого використання у фінальному блоці підсумкового розрахунку.

На рисунку 3.10 показано фрагмент коду модуля TravelExpensesCalculator, в якому реалізовано дану функціональність.

```
Label(root, text="З_о (основна зарплата), грн:").grid(row=0, column=0, sticky='e')
entry_Zo = Entry(root)
entry_Zo.grid(row=0, column=1)

Label(root, text="З_р (зарплата робітників), грн:").grid(row=1, column=0, sticky='e')
entry_Zp = Entry(root)
entry_Zp.grid(row=1, column=1)

Label(root, text="Н_св (% нарахувань):").grid(row=2, column=0, sticky='e')
entry_Hsv = Entry(root)
entry_Hsv.insert(0, "5") # значення за замовчуванням
entry_Hsv.grid(row=2, column=1)
```

Рисунок 3.10 – Фрагмент модуля TravelExpensesCalculator, що реалізує розрахунок витрат на службові відрядження

Цей модуль має просту структуру та забезпечує обробку помилок введення чисел, що забезпечує стабільність роботи навіть при неточних або випадкових діях користувача.

Модуль «Сторонні роботи» реалізує розрахунок витрат, пов'язаних із залученням зовнішніх виконавців для виконання окремих частин науково-дослідної роботи. Інтерфейс передбачає введення даних про заробітну плату та норматив відрахувань, які застосовуються до цього типу робіт. Після введення значень основної та додаткової зарплати, а також коефіцієнта витрат на сторонні роботи, система автоматично обчислює загальні витрати. Результат виводиться на екран і зберігається у внутрішньому сховищі для подальших підсумкових обчислень.

На рисунку 3.11 зображено фрагмент коду модуля ExternalServicesCalculator, де реалізовано логіку цього розрахунку.

```
build_global_menu(root)
root.title("Розрахунок В_сп – витрати на сторонні роботи")

def calculate_external_services():
    try:
        Zo = float(entry_Zo.get())
        Zp = float(entry_Zp.get())
        Hsn = float(entry_Hsn.get())

        Bsn = round((Zo + Zp) * Hsn / 100, 2)
        result_var.set(f"Витрати на сторонні роботи В_сп = {Bsn:.2f} грн")
        set_result("В_сп", Bsn)
    except ValueError:
        messagebox.showerror("Помилка", "Будь ласка, введіть коректні числові значення.")

Label(root, text="В_о (основна ЗП дослідників), грн:").grid(row=0, column=0, sticky='e')
entry_Zo = Entry(root)
entry_Zo.grid(row=0, column=1)

Label(root, text="В_п (основна ЗП робітників), грн:").grid(row=1, column=0, sticky='e')
entry_Zp = Entry(root)
entry_Zp.grid(row=1, column=1)
```

Рисунок 3.11 – Фрагмент модуля ExternalServicesCalculator, що реалізує розрахунок витрат на сторонні роботи

Даний модуль є типовим прикладом розрахункової форми з обробкою виключень. Він забезпечує точність результатів і є повністю інтегрованим у структуру загальної системи модуля розрахунків.

Модуль «Інші витрати» призначений для врахування додаткових витрат, які не входять до основних категорій, але є обов’язковими для виконання науково-дослідної роботи. Користувач вводить дані про основну та додаткову заробітну плату, а також відсоток додаткових витрат. На основі введених значень виконується обчислення загальної суми інших витрат. Результат виводиться у відповідному полі інтерфейсу користувача та зберігається у загальній структурі даних програми для подальшого використання у підсумковому розрахунку.

На рисунку 3.12 подано фрагмент коду з модуля OtherExpensesCalculator, де реалізовано цей функціонал.

```
build_global_menu(root)
root.title("Розрахунок _в – інші витрати")

def calculate_other_expenses():
    try:
        Zo = float(entry_Zo.get())
        Zp = float(entry_Zp.get())
        Hiv = float(entry_Hiv.get())

        Iv = round((Zo + Zp) * Hiv / 100, 2)
        result_var.set(f"Інші витрати _в = {Iv:.2f} грн")
        set_result("_в", Iv)
    except ValueError:
        messagebox.showerror("Помилка", "Будь ласка, введіть коректні числові значення.")

Label(root, text="_о (ЗП дослідників), грн:").grid(row=0, column=0, sticky='e')
entry_Zo = Entry(root)
entry_Zo.grid(row=0, column=1)

Label(root, text="_п (ЗП робітників), грн:").grid(row=1, column=0, sticky='e')
entry_Zp = Entry(root)
entry_Zp.grid(row=1, column=1)

Label(root, text="_в (% інших витрат):").grid(row=2, column=0, sticky='e')
entry_Hiv = Entry(root)
entry_Hiv.insert(0, "2")
entry_Hiv.grid(row=2, column=1)
```

Рисунок 3.12 – Фрагмент модуля, що реалізує розрахунок інших витрат

Модуль реалізовано у вигляді компактного вікна, з простою структурою та перевіркою на коректність введення. Його використання забезпечує врахування всіх другорядних витрат, що можуть впливати на загальну суму фінансування проекту.

Модуль «Накладні витрати» відповідає за розрахунок витрат, пов'язаних із загальним функціонуванням проєкту. До таких витрат належать витрати на управління, комунальні послуги, адміністративне забезпечення тощо. Розрахунок здійснюється на основі суми основної та додаткової зарплати та відсоткового коефіцієнта накладних витрат. Після введення необхідних значень користувач отримує результат обчислення накладних витрат, який відображається у вікні та автоматично зберігається у спільному сховищі для подальшого підсумкового аналізу витрат.

На рисунку 3.13 представлено фрагмент коду модуля OverheadExpensesCalculator, де реалізовано розрахунок накладних витрат.

```
build_global_menu(root)
root.title("Розрахунок В_нзв – накладні витрати")

def calculate_overhead_expenses():
    try:
        Zo = float(entry_Zo.get())
        Zp = float(entry_Zp.get())
        Hn = float(entry_Hn.get())

        Bn = round((Zo + Zp) * Hn / 100, 2)
        result_var.set(f"Накладні витрати В_нзв = {Bn:.2f} грн")
        set_result("В_нзв", Bn)
    except ValueError:
        messagebox.showerror("Помилка", "Перевірте правильність введених чисел.")

Label(root, text="В_о (ЗП дослідників), грн:").grid(row=0, column=0, sticky='e')
entry_Zo = Entry(root)
entry_Zo.grid(row=0, column=1)

Label(root, text="В_р (ЗП робітників), грн:").grid(row=1, column=0, sticky='e')
entry_Zp = Entry(root)
entry_Zp.grid(row=1, column=1)

Label(root, text="H_нзв (% накладних витрат):").grid(row=2, column=0, sticky='e')
entry_Hn = Entry(root)
entry_Hn.insert(0, "50") # Значення за замовчуванням
entry_Hn.grid(row=2, column=1)
```

Рисунок 3.13 – Фрагмент модуля OverheadExpensesCalculator, що реалізує розрахунок накладних витрат

Завдяки простій структурі та обробці помилок, модуль забезпечує надійність розрахунків та коректність введення даних. Він є ключовим елементом у загальній системі обліку витрат НДР, оскільки накладні витрати становлять істотну частку загального бюджету.

Завершальним етапом функціонального модуля є підсумковий розрахунок, який об'єднує усі попередні обчислення. Модуль «Підсумковий розрахунок» виконує агрегацію значень, отриманих з усіх функціональних блоків витрат, та формує загальну суму витрат на науково-дослідну роботу. Дані беруться із внутрішнього сховища результатів, куди кожен модуль передає своє обчислене значення. Після перевірки введених значень, виконується обчислення суми і результат виводиться у відповідному вікні графічного інтерфейсу.

На рисунку 3.14 зображено фрагмент модуля TotalCostCalculator, який забезпечує об'єднання та обчислення загального бюджету проєкту.

```
# Відповідні ключі з data_store
keys = [
    "В_о", "В_р", "В_доп", "В_н", "М", "К_е", "В_спец", "В_пз",
    "А_обл", "В_е", "В_св", "В_сп", "П_в", "В_нзв"
]

entries = {}

def calculate_total():
    try:
        total = sum(float(entries[k].get()) for k in keys)
        result_var.set(f"Загальна сума витрат = {total:.2f} грн")
    except ValueError:
        messagebox.showerror("Помилка", "Перевірте всі введені значення.")

for i in range(len(labels)):
    Label(root, text=labels[i] + ":", anchor="e").grid(row=i, column=0, sticky="e", padx=5, pady=2)
    entry = Entry(root)
    entry.grid(row=i, column=1, padx=5, pady=2)

    value = get_result(keys[i])
    if value is not None:
        entry.insert(0, f"{value:.2f}")

    entries[keys[i]] = entry
```

Рисунок 3.14 – Фрагмент модуля, що реалізує підсумковий розрахунок витрат

Модуль має компактну реалізацію та використовує цикл для зчитування значень з усіх ключових блоків витрат. Його наявність забезпечує завершеність логіки програми та дозволяє оперативно отримати повний фінансовий результат проєкту.

Окремим елементом функціонального модуля є підсистема формування підсумкового звіту. Вона реалізована у вигляді окремого вікна, яке дає змогу переглянути проміжні результати всіх розрахунків і зберегти їх у форматі CSV або Excel для подальшого аналізу та архівування. У звіті формуються пари «позначення - назва витрати», отримані з результатів розрахунків у попередніх модулях. Для цього використовується структура даних REPORT_ITEMS, яка містить перелік усіх типів витрат. Дані зчитуються за допомогою функції get_result, після чого формуються таблиці за допомогою бібліотек csv або pandas.

На рисунку 3.15 представлено фрагмент коду модуля report_window.py, який відповідає за побудову вікна та завантаження структури звіту.

```
REPORT_ITEMS = [
    ("З_о", "Зарплата дослідників"),
    ("З_р", "Зарплата робітників"),
    ("З_доп", "Додаткова зарплата"),
    ("З_н", "Соціальні відрахування"),
    ("М", "Матеріали"),
    ("К_б", "Комплектуючі"),
    ("В_спец", "Спецстаткування"),
    ("В_пз", "Програмне забезпечення"),
    ("А_обл", "Амортизація обладнання"),
    ("В_е", "Енерговитрати"),
    ("В_св", "Службові відрядження"),
    ("В_сп", "Сторонні роботи"),
    ("І_в", "Інші витрати"),
    ("В_нзв", "Накладні витрати"),
]

def open_report_window(root):
    for widget in root.winfo_children():
        widget.destroy()

    from ui.menu import build_global_menu
    build_global_menu(root)

    root.title("Звіт про витрати")
    tk.Label(root, text="Результати розрахунку витрат", font=("Arial", 14, "bold")).pack(pady=10)

    frame = tk.Frame(root)
    frame.pack()
```

Рисунок 3.15 – Фрагмент модуля report_window.py, що реалізує побудову підсумкового звіту

Можливість формування звіту у текстовому та табличному форматі підвищує зручність використання програми. Отримані результати можуть бути використані для звітності, аудиту або подальших розрахунків у зовнішніх системах.

3.4 Тестування програмного модуля розрахунку витрат на здійснення науково-дослідної роботи

Тестування програмного забезпечення є невід’ємною складовою процесу його розробки та впровадження. Метою тестування є виявлення та усунення помилок, забезпечення стабільної роботи програми, а також перевірка відповідності програмного продукту технічному завданню. Якісне тестування дозволяє уникнути логічних, синтаксичних, функціональних та інтерфейсних помилок ще до початку реального використання продукту.

У процесі тестування програмного модуля розрахунку витрат на здійснення науково-дослідної роботи особлива увага приділяється перевірці коректності обчислень, зручності користування інтерфейсом, а також роботі збереження та виведення звітів. Оскільки модуль містить велику кількість розрахункових блоків, критично важливо забезпечити правильну взаємодію між ними та передачу даних у підсумковий розрахунок. Тестування проводиться переважно вручну, шляхом введення різних тестових даних користувачем, перевірки результатів, а також симуляції помилкових дій (наприклад, введення некоректних значень). Також перевіряється стійкість інтерфейсу, правильність відображення результатів, відсутність збоїв при переході між розділами меню та при формуванні звітів.

На рисунку 3.16 показано головне вікно, яке з’являється після запуску програми. Воно забезпечує доступ до розрахункового модуля.

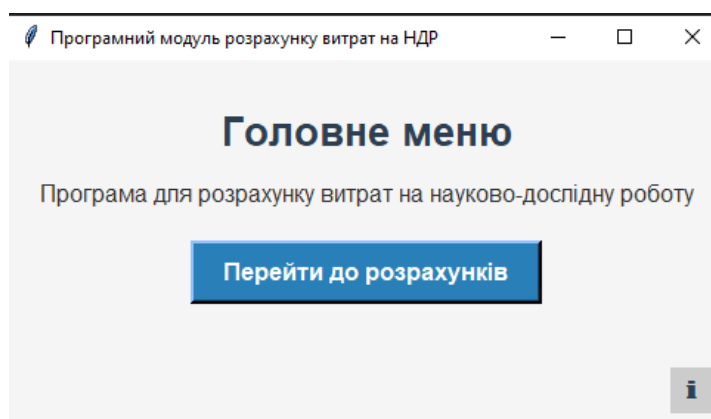


Рисунок 3.16 – Головне вікно програми після запуску

Після натискання кнопки «Перейти до розрахунків» користувач потрапляє до головної панелі розрахунків. Це основний елемент інтерфейсу, який надає доступ до всіх функціональних блоків програми. Кожна кнопка відкриває окреме вікно, у якому реалізовано розрахунок певного типу витрат, відповідно до структури кошторису науково-дослідної роботи.

На рисунку 3.17 представлено зовнішній вигляд панелі розрахунків, що містить перелік усіх доступних категорій: від витрат на оплату праці та матеріали - до сторонніх робіт і накладних витрат. Також тут присутня кнопка підсумкового розрахунку, що завершує обчислення загальних витрат.

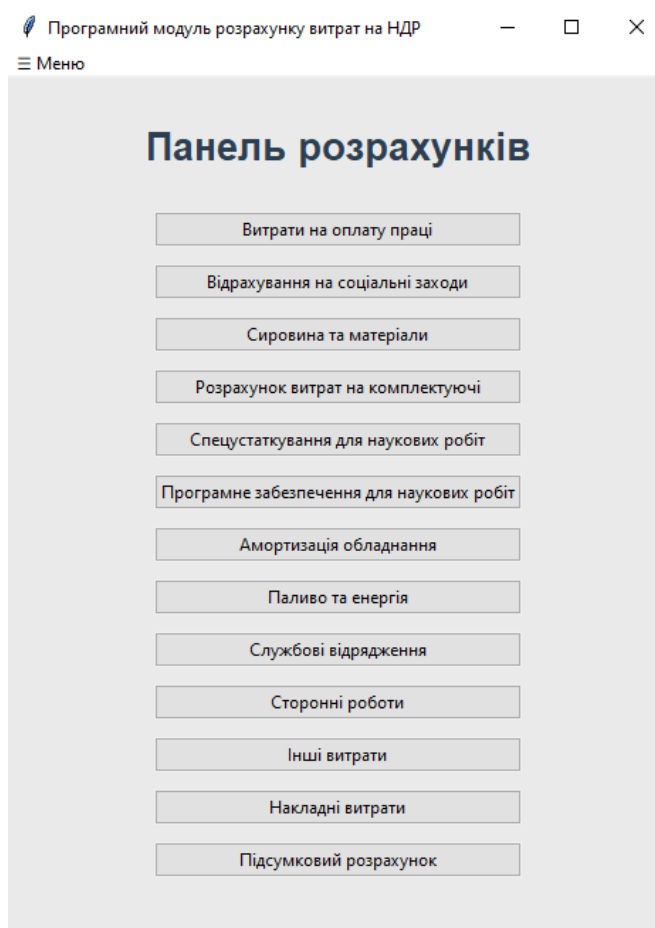


Рисунок 3.17 – Панель розрахунків після переходу з головного меню

Окрім панелі з кнопками, доступ до всіх функціональних блоків надається також через верхнє випадаюче меню «Меню». Це дублюючий навігаційний інструмент, який дозволяє користувачу перейти до будь - якого розрахункового

модуля незалежно від поточного екрану. Такий підхід підвищує зручність користування і є важливим з точки зору ергономіки інтерфейсу.

На рисунку 3.18 показано вигляд випадаючого меню з усіма розділами, що відповідають типам витрат. Останнім пунктом меню є «Звіт», який дозволяє перейти до вікна формування підсумкового звіту. Функціональність меню протестована на відповідність логіці навігації, правильність переходів та відсутність збоїв.

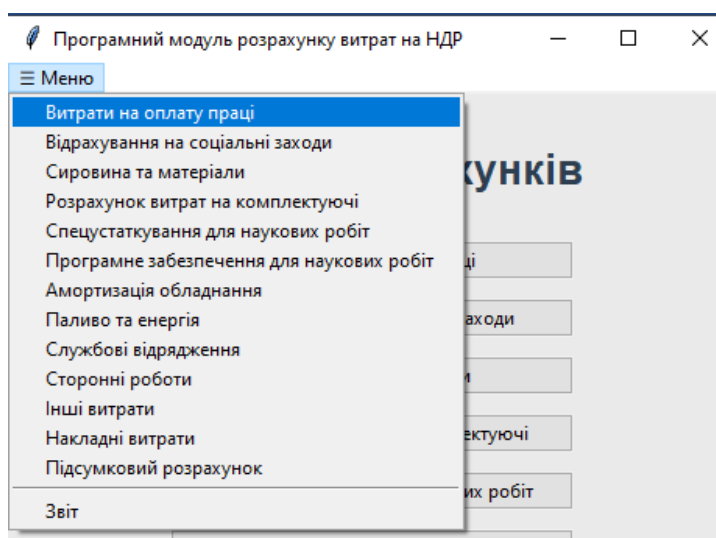


Рисунок 3.18 – Випадаюче меню з розділами розрахунків та пунктом звіту

Тестування розрахунку Z_0 - основна заробітна плата дослідників. Приклад коректного введення значень (рис. 3.19):

Поле «Оклад» = 2, Поле «Дні» = 2, Кількість робочих днів = 22. Натискаємо кнопку «Розрахувати». Результат: 0.18 грн. Обчислення виконано коректно.

Розрахунок Z_0 — основна ЗП дослідників

Кількість посад:

Посада	Оклад	Дні	Результат
Дослідник	2	2	0.18

T_r (робочих днів):

Загальна Z_0 = 0.18 грн

Рисунок 3.19 – Коректне введення для розрахунку заробітної плати дослідників

Приклад некоректного введення значення (рис. 3.20):

Поле «Оклад» = 2, Поле «Дні» = 'abc' (текстове значення), $T_p = 22$. Натискаємо кнопку «Розрахувати». Результат: у полі результату виводиться повідомлення «Помилка», загальна $Z_o = 0.00$ грн.

Розрахунок Z_o — основна ЗП дослідників

Кількість посад: 1 Оновити

Посада	Оклад	Дні	Результат
Дослідник	2	abc	Помилка

Т_p (робочих днів): 22 Розрахувати

Загальна $Z_o = 0.00$ грн

Рисунок 3.20 – Некоректне введення для заробітної плати дослідників

Тестування розрахунку Z_p - основна заробітна плата робітників. Приклад коректного введення значень (рис. 3.21). Усі числові поля заповнено правильно. Після натискання кнопки «Розрахувати» система успішно обчислює ставку і загальну суму.

Посада	Години	Розряд (Кі)	Тариф. коеф. (Кс)	Ставка (грн/год)	Результат (грн)
asd	sd	asdf	asdf		

Кількість робітників: 1 Оновити Т_p (робочих днів): 22 тзм (тривалість зміни): 8 ММ (мін. з/п): 7100 Розрахувати

Загальна $Z_p = 0.00$ грн

Рисунок 3.21 – Коректне введення для розрахунку заробітної плати робітників

Приклад некоректного введення значення (рис. 3.22):

У поля 'Години', 'Кі' або 'Кс' введено текстові значення (наприклад, 'sd', 'asdf').

Посада	Години	Розряд (Кі)	Тариф. коеф. (Кс)	Ставка (грн/год)	Результат (грн)
Робітник	sd	asdf	чиф		

Кількість робітників: 1 Оновити Т_p (робочих днів): 22 тзм (тривалість зміни): 8 ММ (мін. з/п): 7100 Розрахувати

Загальна $Z_p = 0.00$ грн

Додаткова зарплата

Помилка
Один або декілька полів містять некоректні значення. Перевірте введення.

Рисунок 3.22 – Некоректне введення для заробітної плати робітників

Тестування розрахунку Зд - додаткова заробітна плата. Модуль додаткової заробітної плати (Зд) виконує обчислення надбавки на основі сум основної заробітної плати дослідників (Z_o) та робітників (Z_p). Ці значення підставляються автоматично із попередніх обчислень. Користувач повинен ввести лише значення надбавки у відсотках. Система виконує розрахунок лише за умови правильного числового введення, і в іншому випадку повідомляє про помилку. Приклад коректного введення значень (рис. 3.23).

Програмний модуль розрахунку витрат на НДР

Витрати на оплату праці

Розрахунок Z_o — основна ЗП дослідників

Кількість посад: 1

Посада	Оклад	Дні	Результат
Робітник	50000	20	45454.55

T_p (робочих днів): 22

Загальна $Z_o = 45454.55$ грн

Посада	Години	Розряд (К)	Тариф. коеф. (Кс)	Ставка (грн/год)	Результат (грн)
Працівник	8	2	1.5	121.02	968.18

Кількість робітників: 1 T_r (робочих днів): 22 $t_{зм}$ (тривалість зміни): 8 ММ (мін. з/п): 7100

Загальна $Z_p = 968.18$ грн

Додаткова зарплата

Z_o	45454.55
Z_p	968.18
Надод (%)	12
Обчислити	
Здод =	5570.73 грн

Рисунок 3.23 – Коректне введення надбавки: обчислення виконано успішно

Приклад некоректного введення значення (рис. 3.24):

Надбавка = 'abd'. Після натискання кнопки «Обчислити» система не виконує розрахунок і виводить повідомлення про помилку.

The screenshot shows a web form with three input fields. The first field, labeled 'Zo', contains the value '45454.55'. The second field, labeled 'Zp', contains the value '968.18'. The third field, labeled 'Hдод (%)', contains the text 'abc'. Below these fields is a button labeled 'Обчислити'. At the bottom of the form, a message 'Помилка вводу' (Input error) is displayed, and a 'Назад' (Back) button is visible.

Рисунок 3.24 – Некоректне введення надбавки: відображається повідомлення «Помилка вводу»

Модуль розрахунку заробітної плати реалізований з урахуванням захисту від помилок введення. Система коректно обробляє некоректні текстові значення, не виконує обчислення у разі помилки та інформує користувача відповідним повідомленням. Така поведінка свідчить про наявність механізмів обробки винятків і забезпечує стійкість і надійність модуля у типових ситуаціях некоректної взаємодії.

Після завершення усіх обчислень користувач може сформувати підсумковий звіт, який включає значення по всіх статтях витрат. Після натискання кнопки «Звіт» відкривається вікно з табличним поданням результатів та підсумковою сумою витрат. Програма також надає можливість експортувати ці дані у формат CSV або Excel за вибором користувача.

На рисунку 3.25 зображено підсумкове меню з усіма розрахованими величинами до моменту генерації звіту.

Зарплата дослідників:	115812.00
Зарплата робітників:	2783.52
Додаткова зарплата:	14231.46
Соціальні відрахування:	29221.94
Матеріали:	1476.00
Комплектуючі:	5544.00
Спецустаткування:	5544.00
Програмне забезпечення:	396.00
Амортизація обладнання:	1161.50
Енерговитрати:	8064.00
Службові відрядження:	2222.80
Сторонні роботи:	1985.30
Інші витрати:	1135.52
Накладні витрати:	3428.00

Розрахувати

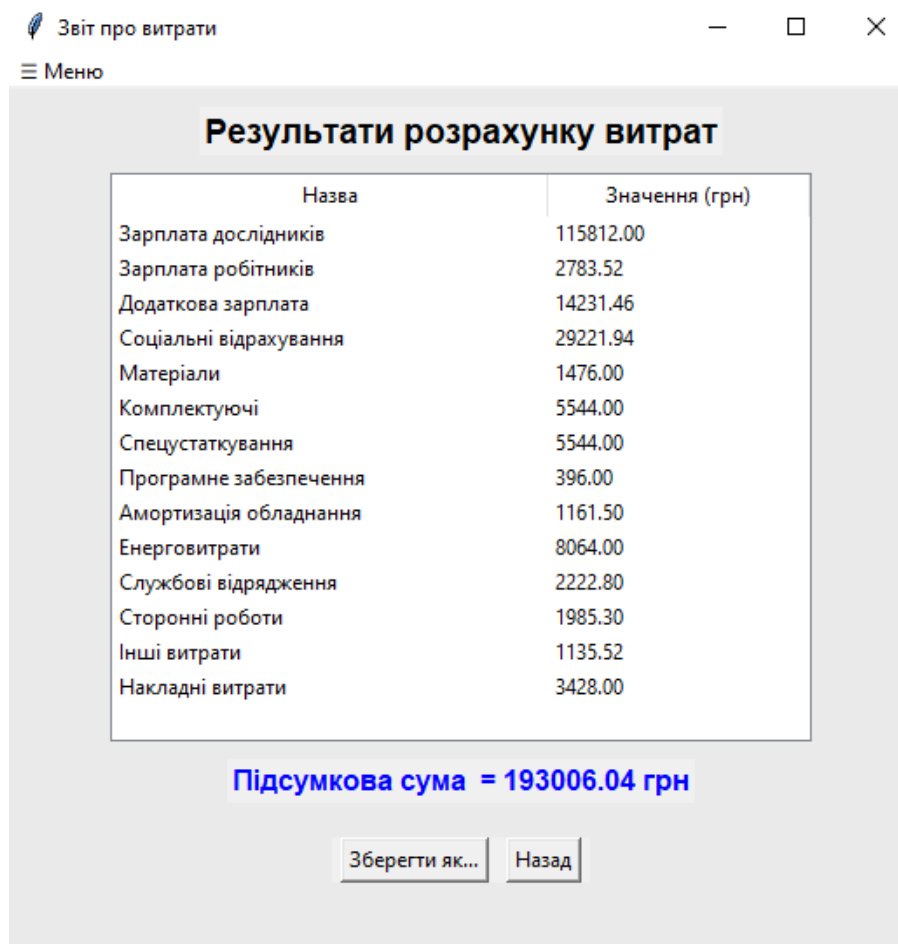
Загальна сума витрат = 193006.04 грн

Назад

Рисунок 3.25 – Підсумкове вікно програми з розрахованими даними

Для переходу до формування підсумкового звіту користувач має натиснути кнопку «Звіт», яка розміщена у випадаючому меню головної панелі (іконка з трьома горизонтальними рисками у верхньому лівому куті вікна). Такий підхід забезпечує компактність інтерфейсу та дозволяє уникнути зайвих елементів на основному екрані.

На рисунку 3.26 зображено вікно сформованого звіту з результатами розрахунків.



Назва	Значення (грн)
Зарплата дослідників	115812.00
Зарплата робітників	2783.52
Додаткова зарплата	14231.46
Соціальні відрахування	29221.94
Матеріали	1476.00
Комплектуючі	5544.00
Спецустаткування	5544.00
Програмне забезпечення	396.00
Амортизація обладнання	1161.50
Енерговитрати	8064.00
Службові відрядження	2222.80
Сторонні роботи	1985.30
Інші витрати	1135.52
Накладні витрати	3428.00

Підсумкова сума = 193006.04 грн

Зберегти як... Назад

Рисунок 3.26 – Вікно сформованого звіту з результатами розрахунків

Кнопка «Зберегти як...» дозволяє обрати бажаний формат файлу. Користувач може зберегти звіт у форматі CSV або Excel, що забезпечує зручність подальшого аналізу або архівації результатів (рис. 3.27).

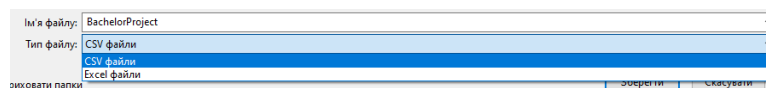


Рисунок 3.27 – Вибір формату збереження підсумкового звіту

Механізм формування та збереження звіту реалізований зручно та інтуїтивно зрозуміло для користувача. Програма дає змогу отримати візуальне подання усіх результатів та зберегти їх у потрібному форматі, що підтверджує повноту та завершеність функціоналу.

Після вибору формату файлу та підтвердження збереження з'являється повідомлення «Звіт успішно збережено!» (рис. 3.28). Це дає користувачеві зрозумілий сигнал про завершення операції.

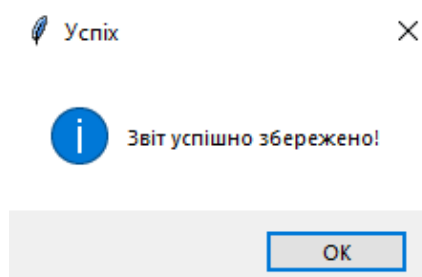


Рисунок 3.28 – Повідомлення про успішне збереження звіту

На рисунку 3.29 наведено приклад збереженого звіту у форматі Excel. Таблиця містить назви усіх розрахованих статей витрат, відповідні значення та підсумок у нижній частині.

	А	В
1	Результати розрахунку витрат	
2		
3	Назва	Значення (грн)
4	Зарплата дослідників	115 812,00 грн
5	Зарплата робітників	2 783,52 грн
6	Додаткова зарплата	14 231,46 грн
7	Соціальні відрахування	29 221,94 грн
8	Матеріали	1 476,00 грн
9	Комплектуючі	5 544,00 грн
10	Спецустаткування	5 544,00 грн
11	Програмне забезпечення	396,00 грн
12	Амортизація обладнання	1 161,50 грн
13	Енерговитрати	8 064,00 грн
14	Службові відрядження	2 222,80 грн
15	Сторонні роботи	1 985,30 грн
16	Інші витрати	1 135,52 грн
17	Накладні витрати	3 428,00 грн
18	Підсумок	193 006,04 грн

Рисунок 3.29 – Приклад збереженого звіту у форматі Excel

Для оцінки ефективності розробленого програмного модуля було проведено порівняння з існуючими системами управління проектами, а саме: PLANTA Project

та OpenProject. У таблиці 3.3 наведено порівняння за основними критеріями, що стосуються зручності використання, функціональності, доступності та відкритості коду.

Таблиця 3.3 – Порівняльний аналіз функціоналу розробленого програмного модуля з конкурентними системами

Критерій	Розроблений модуль	PLANTA Project	OpenProject
Розрахунок витрат на проведення НДР	+	Обмежено	Обмежено
Наявність графічного інтерфейсу	+	+	+
Гнучкість у введенні даних	+	-	-
Автоматичне збереження результатів	+	+	+
Формування підсумкового звіту	+	+	+
Експорт у CSV/Excel	+	+	-
Мова реалізації	Python	Java	Ruby
Локалізація (українська мова)	+	-	-
Простота використання	Висока	Середня	Середня
Можливість модифікації коду	Відкрита структура	Закритий код	Частково доступна

Результати порівняння свідчать про те, що розроблений програмний модуль має розширений функціонал у порівнянні з конкурентними системами. Зокрема, він забезпечує гнучке введення даних, простоту у використанні та можливість модифікації вихідного коду. Розширення функціональних можливостей досягнуто

завдяки адаптації до специфіки розрахунків витрат саме на науково-дослідну роботу відповідно до національної нормативної бази. Програмний модуль враховує чинні значення мінімальної заробітної плати, вартості енергії, відсотків соціальних відрахувань тощо, що робить обчислення максимально наближеними до реальних умов. Гнучкість інтерфейсу реалізована у вигляді можливості редагування будь-яких вхідних параметрів користувачем: чисельності персоналу, розряду, коефіцієнтів, витрат на матеріали, устаткування, ПЗ та інші статті. Такий підхід дозволяє адаптувати розрахунки під конкретний проєкт без необхідності змін у вихідному коді, що значно підвищує універсальність і практичну цінність розробки.

3.5 Висновок до розділу 3

У цьому розділі було здійснено розробку функціонального програмного модуля для розрахунку витрат на виконання науково-дослідної роботи. На початку було обґрунтовано вибір мови програмування та середовища розробки. Порівняльний аналіз популярних мов та середовищ показав, що Python у поєднанні з бібліотекою Tkinter є оптимальним рішенням для реалізації графічного інтерфейсу та виконання обчислень. Використання додаткових бібліотек, таких як csv та pandas, забезпечило можливість експорту результатів у зручні формати. У ході реалізації модуля було створено зручний інтерфейс користувача з розділенням на окремі функціональні блоки для кожного виду витрат. Всі обчислення виконуються автоматично після введення вхідних даних. Для кожного елементу витрат реалізовані окремі форми та модулі, що забезпечують масштабованість і гнучкість системи.

Тестування підтвердило коректність функціонування всіх модулів, наявність обробки помилок при некоректному введенні даних, а також стабільну роботу функцій збереження та формування звіту. Результати розрахунків виводяться у підсумкове вікно, після чого користувач має змогу зберегти звіт у форматі CSV або

Excel. Проведене порівняння з існуючими системами (PLANTA Project, OpenProject) показало, що розроблений програмний модуль має переваги у простоті використання, гнучкості, локалізації та можливості модифікації. Таким чином у процесі розробки було реалізовано три додаткові функції, адаптацію до національних нормативів розрахунку витрат, редагування вхідних параметрів користувачем, а також локалізацію інтерфейсу українською мовою. Розрахунок витрат на проведення науково-дослідної роботи гнучкість у введенні даних. Це свідчить про доцільність його застосування в навчальних закладах, наукових установах та малих організаціях, що виконують розрахунки витрат на НДР.

ВИСНОВКИ

Поставлені перед бакалаврською кваліфікаційною роботою задачі виконано в повному обсязі, а саме:

1. Проведено аналіз сучасних підходів до автоматизованого розрахунку витрат на науково-дослідну роботу. Розглянуто основні методи кошторисного планування, такі як нормативний, факторний, аналоговий та експертний. Проаналізовано особливості їх застосування у сфері наукових досліджень.

2. Здійснено огляд існуючих програмних рішень та визначено їхні переваги й недоліки. Обґрунтовано необхідність розробки удосконаленого програмного модуля, адаптованого до українських реалій та специфіки вітчизняних нормативних документів.

3. Спроектовано модульну структуру програмного забезпечення з поділом на окремі функціональні блоки: заробітна плата, соціальні нарахування, амортизація, витрати на матеріали, енергію, програмне забезпечення, інші витрати. Для кожного з блоків визначено формули обчислення, типи вхідних даних та алгоритми обробки.

4. Реалізацію модуля виконано мовою програмування Python із використанням бібліотек tkinter, csv та pandas, що забезпечило побудову графічного інтерфейсу, логіку розрахунків і можливість збереження результатів.

5. Тестування підтвердило коректність роботи модуля та його відповідність поставленим завданням. У порівнянні з конкурентними системами, розроблений модуль відрізняється розширеним функціоналом і включає 3 додаткових функції, а саме: адаптацію до національних нормативів розрахунку витрат, редагування вхідних параметрів користувачем, а також локалізацію інтерфейсу українською мовою.

6. Розроблено інструкцію користувача, яка містить опис функціональних можливостей, порядок введення даних, рекомендації щодо використання модулів, а також порядок збереження та експорту результатів у форматах CSV та Excel.

Таким чином, мету бакалаврської роботи досягнуто.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Козловський В. О. Техніко-економічні обґрунтування та економічні розрахунки в дипломних проектах та роботах: навчальний посібник / Козловський В. О. – Вінниця: ВДТУ, 2003. – 75 с.

2. Лопухова К. А. Облік витрат на виконання науково-дослідних та дослідно - конструкторських робіт / К. А. Лопухова // Наукові записки кафедри обліку і оподаткування. – 2023. – № 2. – С. 45–49. – Наук. керівник: Масіна Л. О.

3. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад: В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця: ВНТУ, 2021. – 42 с.

4. Норми тривалості робочого часу. – [Електронний ресурс]. – Режим доступу: <https://services.dtki.ua/catalogues/worktime>. – [Назва з екрана]. Дата звернення: 12.05.2025 р.

5. Валявський О.С., Богач І.В. Інформаційна технологія для обліку робочого часу працівників // Матеріали Всеукраїнської науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/22780>.

6. Середньозважена ціна акцептованої на РДН електроенергії (оптовий ринок). – [Електронний ресурс]. – Режим доступу: <https://www.oree.com.ua/index.php/newsctr/n/6582>. – [Назва з екрана]. Дата звернення: 12.05.2025 р.

7. Середньомісячна заробітна плата за видами економічної діяльності промисловості за період з початку року. – [Електронний ресурс]. – Режим доступу: http://www.ukrstat.gov.ua/operativ/operativ2005/gdn/Zarp_prom_p/Zp_pr_p_u/arh_prp_u.html. – [Назва з екрана]. Дата звернення: 12.05.2020 р.

8. Про затвердження «Методичних рекомендацій визначення кошторисної вартості НДДКР для організацій (підприємств) різних форм власності та господарювання». – [Електронний ресурс]. – Режим доступу:

<https://zakon.rada.gov.ua/rada/show/v0119290-06#Text>. – [Назва з екрана]. Дата звернення: 12.05.2025р.

9. Про затвердження Методики визначення вартості наукових робіт. – [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/rada/show/v0545666-08#top>. – [Назва з екрана]. Дата звернення: 12.05.2025 р.

10. Project Management Institute. Expert Judgment / ProjectManagement.com – [Електронний ресурс]. – Режим доступу: <https://www.projectmanagement.com/wikis/344587/expert-judgment> . Дата звернення: 12.05.2025р.

11. Як розраховується прибуток [Електронний ресурс]. – URL: <https://knowledge-base.joinposter.com/uk-ua/>.

12. COCOMO II – Constructive Cost Model [Електронний ресурс]. – Режим доступу: <http://softwarecost.org/tools/COCOMO/> – Дата звернення: 20.05.2025.

13. PLANTA Project [Електронний ресурс]. – Режим доступу: <https://www.planta.de/en/> – Дата звернення: 20.05.2025.

14. Microsoft Project [Електронний ресурс]. – Режим доступу: <https://www.microsoft.com/uk-ua/microsoft-365/project/project-management/> – Дата звернення: 20.05.2025.

15. OpenProject [Електронний ресурс]. – Режим доступу: <https://www.openproject.org/> – Дата звернення: 20.05.2025.

16. Методологія наукових досліджень: навчальний посібник / за ред. В. П. Горина. Тернопіль: ФОП Осадца Ю. В., 2023. 170 с.

17. Алгоритми та структури даних: навчальний посібник [Електронний ресурс] / Коваленко О. О., Ткаченко О. М., Чехмestрук Р. Ю. – Вінниця : ВНТУ, 2025. – PDF, 113 с.

18. Schaeffer, J.; Burch, N.; Y. Björnsson; Kishimoto, A.; Müller, M.; Lake, R.; Lu, P.; Sutphen, S. (2007). Checkers is Solved (PDF). Science. 317 (5844): 1518—22. doi:10.1126/science.1144079. PMID 17641166.

19 Lee S. Discover Modular Design Strategies for Scalable Software Solutions [Електронний ресурс]. – Режим доступу: <https://numberanalytics.com/blog/modular-design-strategies> – Дата звернення: 20.05.2025.

20. Методичні вказівки до самостійної роботи студентів з проектування UML діаграм в ході виконання курсових робіт з дисципліни «Об'єктноорієнтоване програмування» для студентів спеціальності 121 – «Інженерія програмного забезпечення» [Електронний ресурс] / Уклад. Д. І. Кательніков, О. О. Дудник, А. В. Денисюк – Вінниця: ВНТУ, 2021. – 28 с.

21. The Python Tutorial. URL: <https://docs.python.org/3/tutorial/index.html> – Дата звернення: 20.05.2025

22. GeeksforGeeks. Introduction to Java. [Електронний ресурс]. – Режим доступу: <https://www.geeksforgeeks.org/introduction-to-java/> – Дата звернення: 20.05.2025

23. Microsoft Learn. Overview: A tour of the C# language. [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview> – Дата звернення: 20.05.2025

24. Clouwood / iMakeable. Investigating Benefits and Drawbacks of Electron.js vs. Native [Електронний ресурс]. – Режим доступу: <https://clouwood.com/be-future-ready-investigating-the-benefits-and-drawbacks-of-electron-js-vs-native-development-for-your-app/> – Дата звернення: 20.05.2025

25. Методичні вказівки до виконання бакалаврської кваліфікаційної роботи для студентів денної та заочної форм навчання спеціальності 122 - «Комп'ютерні науки» / Укладачі А.А.Яровий, О.В.Сілагін, І.В.Богач. – Вінниця: ВНТУ, 2022. – 47 с.

26. BachelorProject – розрахунок витрат на здійснення науково-дослідної роботи [Електронний ресурс] / kazik112. – Режим доступу: <https://github.com/kazik112/BachelorProject> – Дата звернення: 20.05.2025

Додаток А (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ
ТЕКСТОВИХ ЗАПОЗИЧЕНЬНазва роботи: Програмний модуль розрахунку витрат на здійснення науково-дослідної роботиТип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)Підрозділ кафедра комп'ютерних наук
(кафедра, факультет, навчальна група)Коефіцієнт подібності текстових запозичень, виявлених у роботі
системою StrikePlagiarism 14,87 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

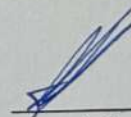
Експертна комісія:

Яровий А.А., зав. каф. КН

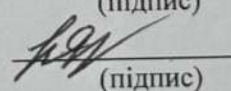
(прізвище, ініціали, посада)

Колесницький О.К., проф. каф КН

(прізвище, ініціали, посада)



(підпис)



(підпис)

Особа, відповідальна за перевірку


(підпис)Озеранський В.С.

(прізвище, ініціали)

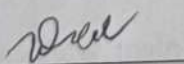
З висновком експертної комісії ознайомлений(-на)

Керівник


(підпис)Шевчук О.Ф.

(прізвище, ініціали, посада)

Здобувач


(підпис)Казимірський Р.С.

(прізвище, ініціали)

Додаток Б (обов'язковий)

ЛІСТИНГ ПРОГРАМИ

Головне меню

```
# ui/main_window.py

import tkinter as tk
from ui.navigation import show_main_menu

def MainWindow(root):
    for widget in root.winfo_children():
        widget.destroy()

    root.title("Програмний модуль розрахунку витрат на НДР")
    root.geometry("500x350")
    root.configure(bg="#f5f5f5")

    # Заголовок
    tk.Label(
        root,
        text="Головне меню",
        font=("Arial", 20, "bold"),
        bg="#f5f5f5",
        fg="#2c3e50"
    ).pack(pady=(30, 10))

    # Пояснення
    tk.Label(
        root,
        text="Програма для розрахунку витрат на науково-дослідну роботу",
        font=("Arial", 12),
        bg="#f5f5f5",
        fg="#333"
    ).pack(pady=(0, 20))

    # Кнопка переходу до розрахунків
    tk.Button(
        root,
        text="Перейти до розрахунків",
        font=("Arial", 12, "bold"),
        bg="#2980b9",
        fg="white",
        relief="raised",
        bd=3,
        padx=15,
        pady=5,
        command=lambda: show_main_menu(root)
    ).pack()
```

```

# Кнопка i — інформація (в правому нижньому куті)
info_btn = tk.Button(
    root,
    text="i",
    font=("Arial", 13, "bold"),
    bg="#cccccc",
    fg="#2c3e50",
    relief="flat",
    bd=1,
    width=2,
    height=1,
    cursor="hand2",
    activebackground="#bbbbbb",
    command=lambda: show_info_window(root)
)
info_btn.place(relx=1.0, rely=1.0, x=-10, y=-10, anchor="se")

# Спливаюча підказка
Tooltip(info_btn, "Дізнатись більше про програму")

def show_info_window(root):
    info_window = tk.Toplevel(root)
    info_window.title("Інформація про програму")
    info_window.geometry("450x300")
    info_window.configure(bg="#ecf0f1")

    tk.Label(
        info_window,
        text="Про програму",
        font=("Arial", 14, "bold"),
        bg="#ecf0f1",
        fg="#2c3e50"
    ).pack(pady=(20, 10))

    tk.Label(
        info_window,
        text=(
            "Цей програмний модуль призначений для розрахунку витрат на виконання "
            "науково-дослідних робіт (НДР). Програма дозволяє поетапно обчислити всі "
            "складові витрат, включаючи заробітну плату, соціальні нарахування, витрати "
            "на матеріали, енергію, відрядження, сторонні роботи та інші.\n\n"
            "Результати автоматично зберігаються та доступні у підсумковому звіті."
        ),
        font=("Arial", 11),
        bg="#ecf0f1",
        wraplength=400,
        justify="left"
    ).pack(pady=10, padx=20)

```

```
tk.Button(info_window, text="Закрити", command=info_window.destroy).pack(pady=10)
```

```
# Tooltip-клас
```

```
class Tooltip:
```

```
    def __init__(self, widget, text):
        self.widget = widget
        self.text = text
        self.tip_window = None
        widget.bind("< Enter>", self.show_tooltip)
        widget.bind("< Leave>", self.hide_tooltip)
```

```
    def show_tooltip(self, event=None):
        if self.tip_window or not self.text:
            return
        x = self.widget.winfo_rootx() - 195
        y = self.widget.winfo_rooty() + 5
        self.tip_window = tw = tk.Toplevel(self.widget)
        tw.wm_overrideredirect(True)
        tw.wm_geometry(f"#{x}#{y}")
        label = tk.Label(
            tw,
            text=self.text,
            justify="left",
            background="#ffffe0",
            relief="solid",
            borderwidth=1,
            font=("Arial", 9)
        )
        label.pack(ipadx=5, ipady=2)
```

```
    def hide_tooltip(self, event=None):
        if self.tip_window:
            self.tip_window.destroy()
            self.tip_window = None
```

```
Навігатор (navigation.py)
```

```
import tkinter as tk
```

```
from tkinter import ttk
```

```
from ui.menu import build_global_menu
```

```
def show_main_menu(root):
    from ui.salary_block_ui import open_salary_block
    from modules.social_tax import SocialTaxCalculator
    from modules.material_cost import MaterialCostCalculator
    from modules.component_cost import ComponentsCostCalculator
    from modules.equipment_cost import EquipmentCostCalculator
    from modules.software_cost import SoftwareCostCalculator
```

```

from modules.amortization_cost import AmortizationCalculator
from modules.energy_cost import EnergyCostCalculator
from modules.travel_expenses import TravelExpensesCalculator
from modules.external_services import ExternalServicesCalculator
from modules.other_expenses import OtherExpensesCalculator
from modules.overhead_expenses import OverheadExpensesCalculator
from modules.total_cost import TotalCostCalculator

for widget in root.winfo_children():
    widget.destroy()

root.title("Програмний модуль розрахунку витрат на НДР")
root.configure(bg="#eaeaea")

build_global_menu(root)

outer_frame = tk.Frame(root, bg="#eaeaea", padx=30, pady=30)
outer_frame.pack(expand=True, fill="both")

title = tk.Label(
    outer_frame,
    text="Панель розрахунків",
    font=("Arial", 20, "bold"),
    bg="#eaeaea",
    fg="#2c3e50"
)
title.pack(pady=(0, 20))

button_frame = tk.Frame(outer_frame, bg="#eaeaea")
button_frame.pack()

buttons = [
    ("Витрати на оплату праці", lambda: open_salary_block(root)),
    ("Відрахування на соціальні заходи", lambda: SocialTaxCalculator(root)),
    ("Сировина та матеріали", lambda: MaterialCostCalculator(root)),
    ("Розрахунок витрат на комплектуючі", lambda: ComponentsCostCalculator(root)),
    ("Спецустаткування для наукових робіт", lambda: EquipmentCostCalculator(root)),
    ("Програмне забезпечення для наукових робіт", lambda: SoftwareCostCalculator(root)),
    ("Амортизація обладнання", lambda: AmortizationCalculator(root)),
    ("Паливо та енергія", lambda: EnergyCostCalculator(root)),
    ("Службові відрядження", lambda: TravelExpensesCalculator(root)),
    ("Сторонні роботи", lambda: ExternalServicesCalculator(root)),
    ("Інші витрати", lambda: OtherExpensesCalculator(root)),
    ("Накладні витрати", lambda: OverheadExpensesCalculator(root)),
    ("Підсумковий розрахунок", lambda: TotalCostCalculator(root)),
]

for text, cmd in buttons:
    ttk.Button(button_frame, text=text, command=cmd).pack(pady=6, padx=10, fill="x")

```

Меню розрахунку витрат (menu.py)

```
# ui/menu.py
import tkinter as tk
from tkinter import Menu
from ui.pages.report_window import open_report_window # ✓ Додаємо імпорт звіту

def build_global_menu(root):
    from ui.salary_block_ui import open_salary_block
    from modules.social_tax import SocialTaxCalculator
    from modules.material_cost import MaterialCostCalculator
    from modules.component_cost import ComponentsCostCalculator
    from modules.equipment_cost import EquipmentCostCalculator
    from modules.software_cost import SoftwareCostCalculator
    from modules.amortization_cost import AmortizationCalculator
    from modules.energy_cost import EnergyCostCalculator
    from modules.travel_expenses import TravelExpensesCalculator
    from modules.external_services import ExternalServicesCalculator
    from modules.other_expenses import OtherExpensesCalculator
    from modules.overhead_expenses import OverheadExpensesCalculator
    from modules.total_cost import TotalCostCalculator

    # Меню-бар
    menu_bar = Menu(root)
    nav_menu = Menu(menu_bar, tearoff=0)

    nav_menu.add_command(label="Витрати на оплату праці", command=lambda:
open_salary_block(root))
    nav_menu.add_command(label="Відрахування на соціальні заходи", command=lambda:
SocialTaxCalculator(root))
    nav_menu.add_command(label="Сировина та матеріали", command=lambda:
MaterialCostCalculator(root))
    nav_menu.add_command(label="Розрахунок витрат на комплектуючі", command=lambda:
ComponentsCostCalculator(root))
    nav_menu.add_command(label="Спецустаткування для наукових робіт", command=lambda:
EquipmentCostCalculator(root))
    nav_menu.add_command(label="Програмне забезпечення для наукових робіт",
command=lambda: SoftwareCostCalculator(root))
    nav_menu.add_command(label="Амортизація обладнання", command=lambda:
AmortizationCalculator(root))
    nav_menu.add_command(label="Паливо та енергія", command=lambda:
EnergyCostCalculator(root))
    nav_menu.add_command(label="Службові відрядження", command=lambda:
TravelExpensesCalculator(root))
    nav_menu.add_command(label="Сторонні роботи", command=lambda:
ExternalServicesCalculator(root))
    nav_menu.add_command(label="Інші витрати", command=lambda:
OtherExpensesCalculator(root))
```

```

    nav_menu.add_command(label="Накладні витрати", command=lambda:
OverheadExpensesCalculator(root))
    nav_menu.add_command(label="Підсумковий розрахунок", command=lambda:
TotalCostCalculator(root))

    nav_menu.add_separator()
    nav_menu.add_command(label="Звіт", command=lambda: open_report_window(root)) # ✓
Додано пункт звіту

    menu_bar.add_cascade(label="☰ Меню", menu=nav_menu)
    root.config(menu=menu_bar)

```

Блок розрахунку заробітної плати (salary_block.py)

```

import tkinter as tk

from modules.salary_zo import SalaryZoCalculator, set_zdod_block as set_zdod_block_zo
from modules.salary_zp import SalaryZpCalculator, set_zdod_block as set_zdod_block_zp
from modules.salary_zdod import SalaryZdodCalculator

from ui.navigation import show_main_menu
from ui.menu import build_global_menu

def open_salary_block(root):
    for widget in root.winfo_children():
        widget.destroy()

    build_global_menu(root)

    tk.Label(root, text="Витрати на оплату праці", font=("Arial", 16, "bold")).pack(pady=10)

    SalaryZoCalculator(root)
    SalaryZpCalculator(root)

    zdod_block = SalaryZdodCalculator(root)

    set_zdod_block_zo(zdod_block) # ↻ передаємо в Zo
    set_zdod_block_zp(zdod_block) # ✓ передаємо в Zp

    tk.Button(root, text="Назад", command=lambda: show_main_menu(root)).pack(pady=10)

```

Блок формування звіту (report_window.py)

```

import tkinter as tk
from tkinter import ttk, filedialog, messagebox
from modules.data_store import get_result
from ui.routes import show_main_menu

```

```

import csv
import pandas as pd

REPORT_ITEMS = [
    ("З_о", "Зарплата дослідників"),
    ("З_р", "Зарплата робітників"),
    ("З_дод", "Додаткова зарплата"),
    ("З_н", "Соціальні відрахування"),
    ("М", "Матеріали"),
    ("К_е", "Комплектуючі"),
    ("В_спец", "Спецустаткування"),
    ("В_пз", "Програмне забезпечення"),
    ("А_обл", "Амортизація обладнання"),
    ("В_е", "Енерговитрати"),
    ("В_св", "Службові відрядження"),
    ("В_сп", "Сторонні роботи"),
    ("І_в", "Інші витрати"),
    ("В_нзв", "Накладні витрати"),
]

def open_report_window(root):
    for widget in root.winfo_children():
        widget.destroy()

    from ui.menu import build_global_menu
    build_global_menu(root)

    root.title("Звіт про витрати")
    tk.Label(root, text="Результати розрахунку витрат", font=("Arial", 14, "bold")).pack(pady=10)

    frame = tk.Frame(root)
    frame.pack()

    tree = ttk.Treeview(frame, columns=("name", "value"), show="headings", height=15)
    tree.heading("name", text="Назва")
    tree.heading("value", text="Значення (грн)")
    tree.column("name", width=250)
    tree.column("value", width=150)
    tree.pack()

    results = []
    total = 0.0
    for key, label in REPORT_ITEMS:
        value = get_result(key)
        if value is not None:
            results.append((label, round(value, 2)))
            tree.insert("", "end", values=(label, f"{value:.2f}"))
            total += value
        else:
            results.append((label, 0.00))

```

```

tree.insert("", "end", values=(label, "-"))

total_label = tk.Label(root, text=f"Підсумкова сума = {total:.2f} грн", font=("Arial", 12, "bold"),
fg="blue")
total_label.pack(pady=10)

def save_report():
    filetypes = [
        ("CSV файли", "*.csv"),
        ("Excel файли", "*.xlsx"),
    ]
    path = filedialog.asksaveasfilename(
        title="Зберегти звіт",
        defaultextension=".csv",
        filetypes=filetypes
    )
    if not path:
        return

    try:
        if path.endswith(".csv"):
            with open(path, "w", newline="", encoding="utf-8") as f:
                writer = csv.writer(f)
                writer.writerow(["Назва", "Значення (грн)"])
                for name, value in results:
                    writer.writerow([name, value])
                writer.writerow(["Підсумок", round(total, 2)])
        elif path.endswith(".xlsx"):
            from openpyxl import Workbook
            from openpyxl.styles import Font, Alignment, Border, Side, NamedStyle

            wb = Workbook()
            ws = wb.active
            ws.title = "Звіт"

            bold_font = Font(bold=True)
            currency_style = NamedStyle(name="currency")
            currency_style.number_format = '#,##0.00 "грн"'
            border = Border(bottom=Side(style='thin'))

            # Заголовок
            ws.merge_cells("A1:B1")
            ws["A1"] = "Результати розрахунку витрат"
            ws["A1"].font = Font(size=14, bold=True)
            ws["A1"].alignment = Alignment(horizontal="center")

            # Шапка таблиці
            ws["A3"] = "Назва"
            ws["B3"] = "Значення (грн)"
            ws["A3"].font = bold_font

```

```

ws["B3"].font = bold_font
ws["A3"].border = border
ws["B3"].border = border

# Основні значення
for idx, (name, value) in enumerate(results, start=4):
    ws[f"A{idx}"] = name
    ws[f"B{idx}"] = value
    ws[f"B{idx}"].style = currency_style

# Підсумок
total_row = len(results) + 4
ws[f"A{total_row}"] = "Підсумок"
ws[f"A{total_row}"].font = bold_font
ws[f"B{total_row}"] = total
ws[f"B{total_row}"].font = bold_font
ws[f"B{total_row}"].style = currency_style

# Ширина колонок
ws.column_dimensions["A"].width = 35
ws.column_dimensions["B"].width = 20

wb.save(path)

messagebox.showinfo("Успіх", "Звіт успішно збережено!")
except Exception as e:
    messagebox.showerror("Помилка", f"Не вдалося зберегти: {e}")

buttons = tk.Frame(root)
buttons.pack(pady=10)

tk.Button(buttons, text="Зберегти як...", command=save_report).pack(side="left", padx=5)
tk.Button(buttons, text="Назад", command=lambda: show_main_menu(root)).pack(side="left",
padx=5)

```

Додаток В (довідниковий)

ІНСТРУКЦІЯ КОРИСТУВАЧА

Розроблений програмний модуль має простий та інтуїтивно зрозумілий веб-інтерфейс. Для запуску і роботи з модулем користувачеві достатньо виконати такі кроки:

1. Запуск застосунку

У каталозі з файлом `main.py` потрібно виконати:

```
PS C:\Users\Ruslan\Documents> python main.py
```

Рисунок В.1 – виконання команди `python main.py`

Після цього автоматично відкриється головне меню.

2. Опис основного функціоналу

- Початок розрахунків

На головній сторінці (рис В.2) користувач переходить в меню розрахунку витрат та обирає потрібний розділ . На рисунку В.3 зображено вибір розділів.

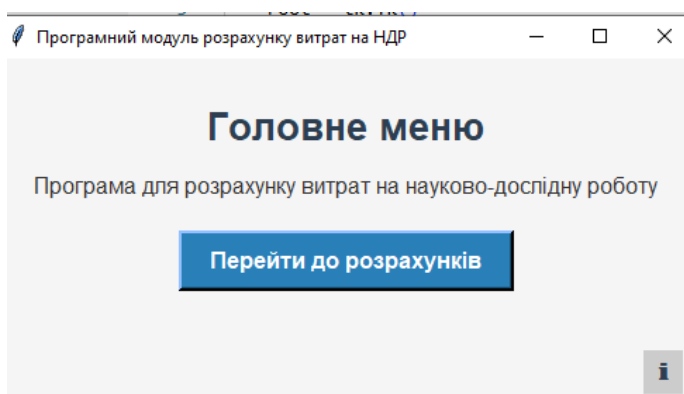


Рисунок В.2 – Перехід до розрахунків

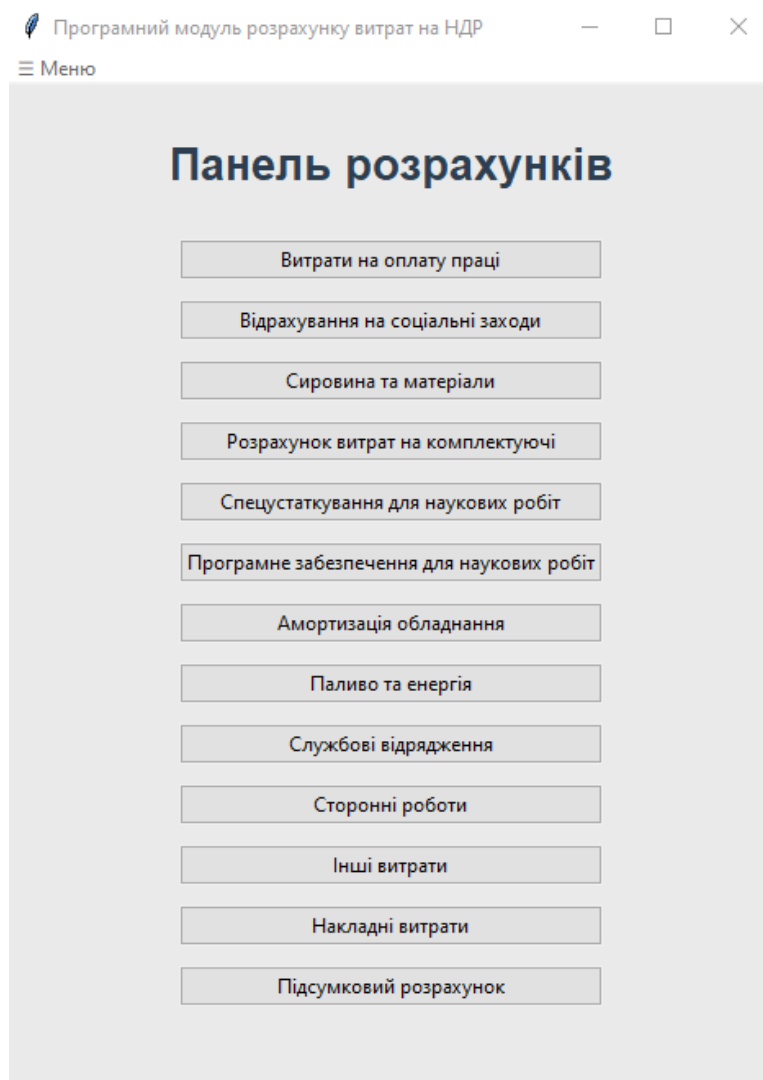


Рисунок В.3 – Вибір розділу розрахунку

Після розрахунків, до прикладу, витрати на оплату праці, (рис В.4), дані атоматично зберігаються (рис В.5).

Програмний модуль розрахунку витрат на НДР

Меню

Витрати на оплату праці

Розрахунок З_о — основна ЗП дослідників

Кількість посад: 1

Посада	Оклад	Дні	Результат
Робітник	50000	20	45454.55

Т_р (робочих днів): 22

Загальна З_о = 45454.55 грн

Посада	Години	Розряд (Кі)	Тариф. коеф. (Кс)	Ставка (грн/год)	Результат (грн)
Працівник	8	2	1.5	121.02	968.18

Кількість робітників: 1 Т_р (робочих днів): 22 тзм (тривалість зміни): 8 ММ (мін. з/п): 7100

Загальна З_р = 968.18 грн

Додаткова зарплата

З_о
45454.55

З_р
968.18

Ндод (%)
12

Здод = 5570.73 грн

Рисунок В.4 – Витрати на оплату праці

Підсумковий розрахунок заг...

Зарплата дослідників:	115812.00
Зарплата робітників:	2783.52
Додаткова зарплата:	14231.46
Соціальні відрахування:	29221.94
Матеріали:	1476.00
Комплектуючі:	5544.00
Спецустаткування:	5544.00
Програмне забезпечення:	396.00
Амортизація обладнання:	1161.50
Енерговитрати:	8064.00
Службові відрядження:	2222.80
Сторонні роботи:	1985.30
Інші витрати:	1135.52
Накладні витрати:	3428.00

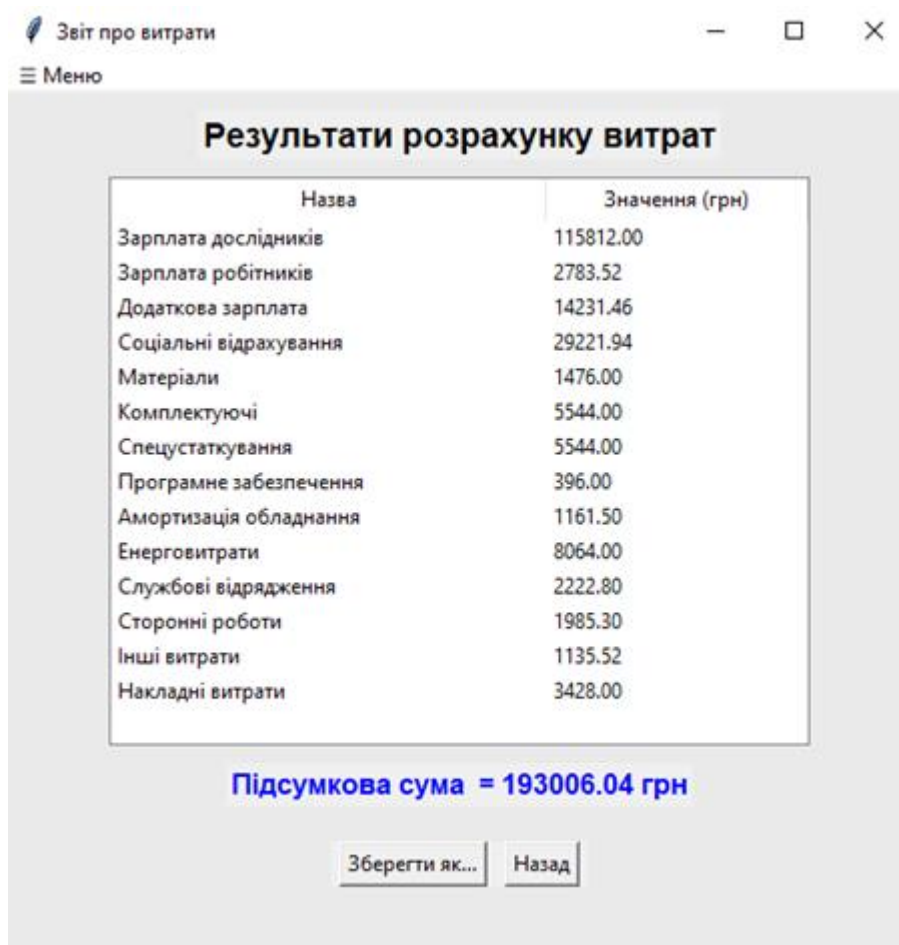
Загальна сума витрат = 193006.04 грн

Рисунок В.5 – Підсумкове вікно програми з розрахованими даними

- Збереження результатів

Для переходу до формування підсумкового звіту користувач має натиснути кнопку «Звіт», яка розміщена у випадаючому меню головної панелі (іконка з трьома горизонтальними рисками у верхньому лівому куті вікна).

На рисунку В.6 зображено вікно сформованого звіту з результатами розрахунків.



Назва	Значення (грн)
Зарплата дослідників	115812.00
Зарплата робітників	2783.52
Додаткова зарплата	14231.46
Соціальні відрахування	29221.94
Матеріали	1476.00
Комплектуючі	5544.00
Спецустаткування	5544.00
Програмне забезпечення	396.00
Амортизація обладнання	1161.50
Енерговитрати	8064.00
Службові відрядження	2222.80
Сторонні роботи	1985.30
Інші витрати	1135.52
Накладні витрати	3428.00

Підсумкова сума = 193006.04 грн

Зберегти як... Назад

Рисунок В.6 – Вікно сформованого звіту з результатами розрахунків

З допомогою кнопки «Зберегти як...» користувач може обрати бажаний формат файлу (рис. В.7).

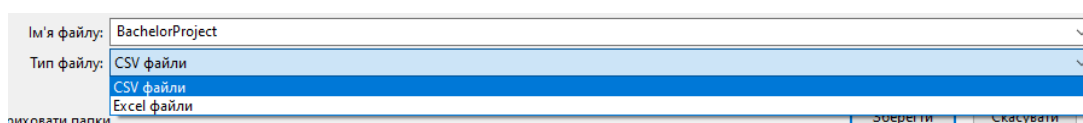


Рисунок В.7 – Вибір формату збереження підсумкового звіту

На рисунку В.8 наведено приклад збереженого звіту у форматі Excel. Таблиця містить назви усіх розрахованих статей витрат, відповідні значення та підсумок у нижній частині.

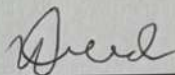
	А	В
1	Результати розрахунку витрат	
2		
3	Назва	Значення (грн)
4	Зарплата дослідників	115 812,00 грн
5	Зарплата робітників	2 783,52 грн
6	Додаткова зарплата	14 231,46 грн
7	Соціальні відрахування	29 221,94 грн
8	Матеріали	1 476,00 грн
9	Комплектуючі	5 544,00 грн
10	Спецустаткування	5 544,00 грн
11	Програмне забезпечення	396,00 грн
12	Амортизація обладнання	1 161,50 грн
13	Енерговитрати	8 064,00 грн
14	Службові відрядження	2 222,80 грн
15	Сторонні роботи	1 985,30 грн
16	Інші витрати	1 135,52 грн
17	Накладні витрати	3 428,00 грн
18	Підсумок	193 006,04 грн
19		

Рисунок В.8 – Приклад збереженого звіту у форматі Excel

Додаток Г (обов'язковий)

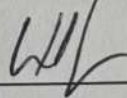
**ІЛЮСТРАТИВНА ЧАСТИНА
«ПРОГРАМНИЙ МОДУЛЬ РОЗРАХУНКУ ВИТРАТ НА ЗДІЙСНЕННЯ
НАУКОВО-ДОСЛІДНОЇ РОБОТИ»**

Виконав: студент 4 курсу, групи 4КН-216
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)



Казимірський Р. С.

(прізвище та ініціали)

Керівник: к.ф-м.н., доц. каф. КН

Шевчук О.Ф.

(прізвище та ініціали)

«3» Червня 2025 р.

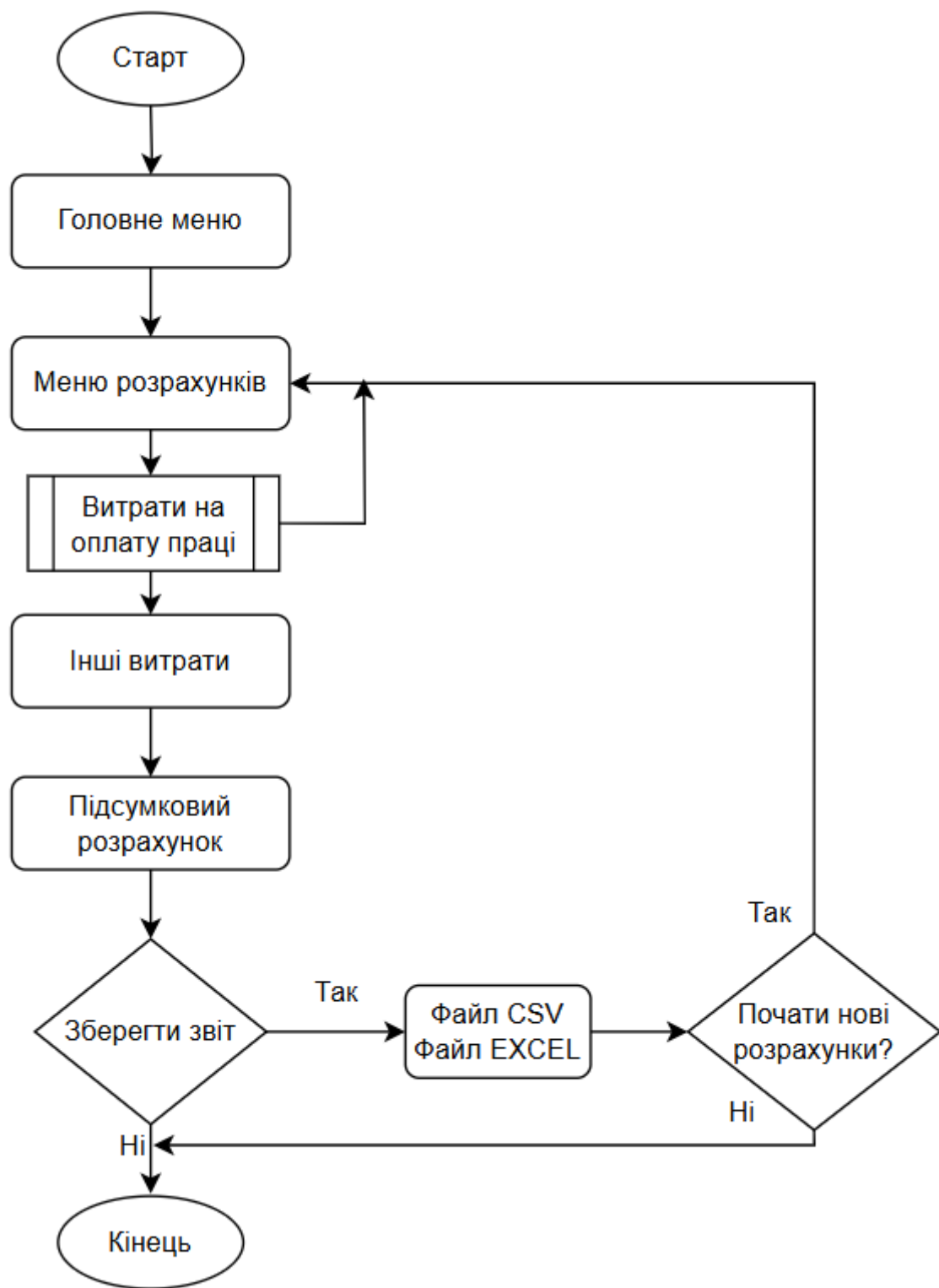


Рисунок Г.1 – UML - діаграма діяльності алгоритму розрахунку загальних витрат на здійснення НДР

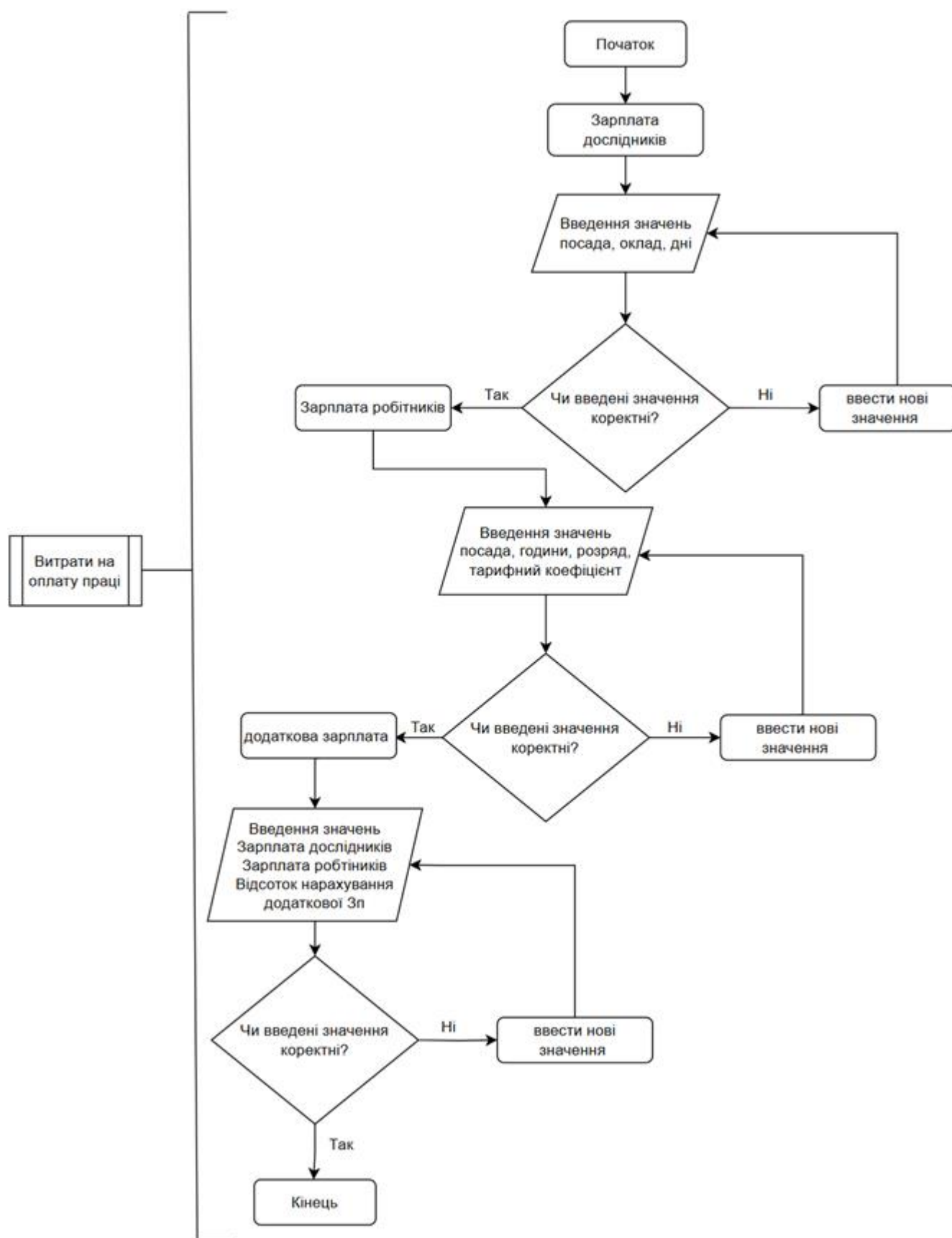


Рисунок Г.2 – UML - діаграма діяльності алгоритму розрахунку заробітної плати

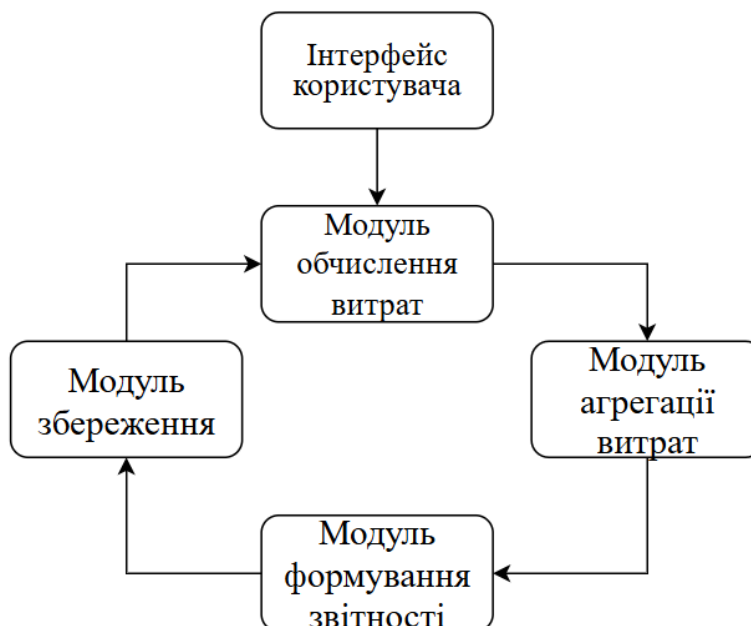


Рисунок Г.3 – Структура програмного модуля розрахунку витрат

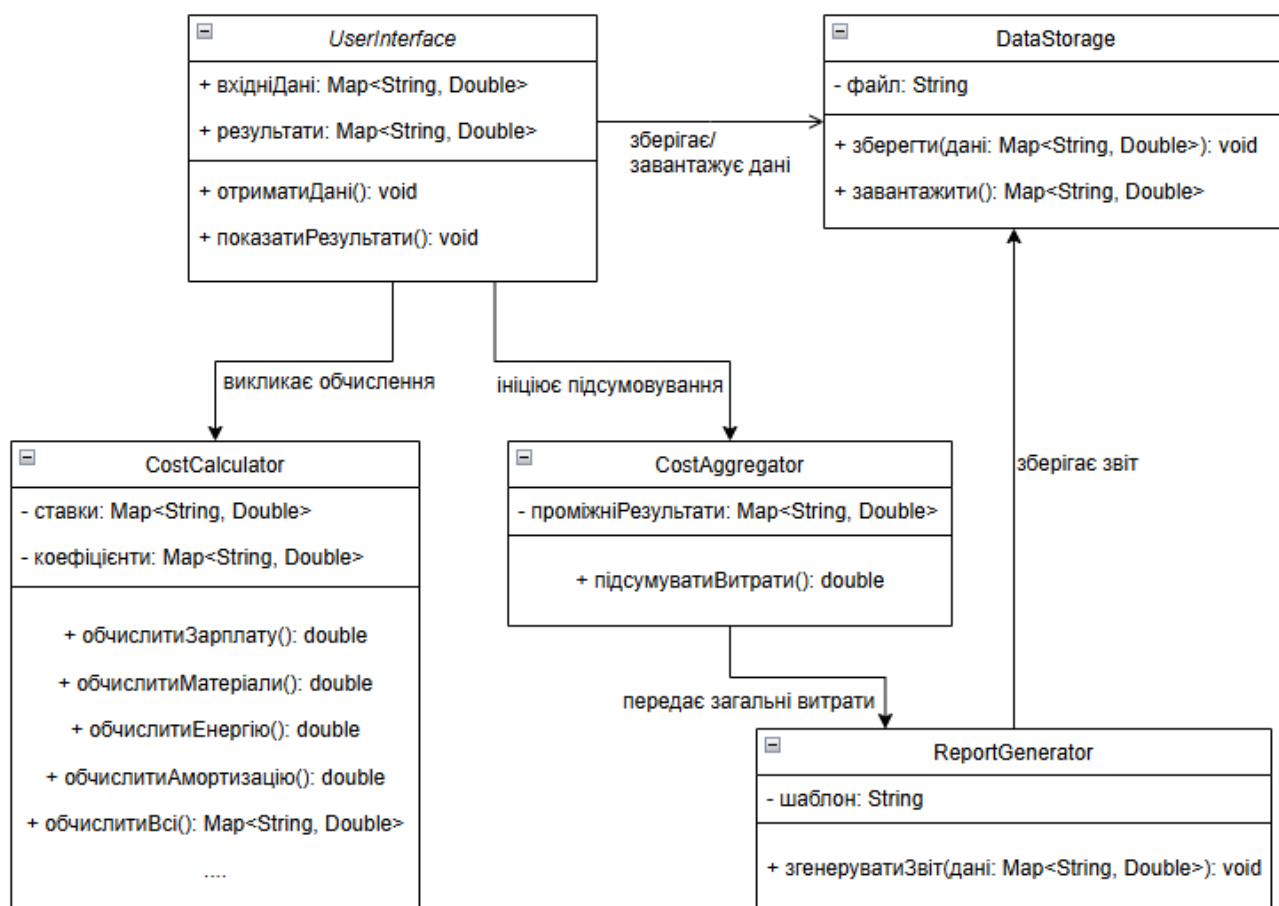


Рисунок Г.4 – UML - діаграма класів програмного модуля розрахунку витрат на НДР

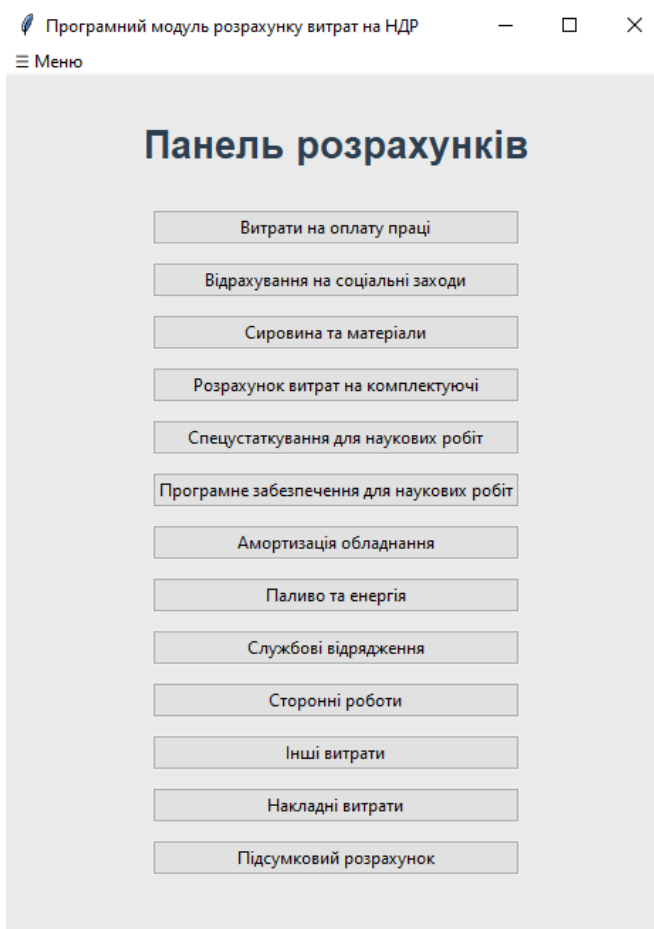


Рисунок Г.5 – Панель розрахунків після переходу з головного меню

Програмний модуль розрахунку витрат на НДР

Меню

Панель розрахунків

- Витрати на оплату праці
- Відрахування на соціальні заходи
- Сировина та матеріали
- Розрахунок витрат на комплектуючі
- Спецстаткування для наукових робіт
- Програмне забезпечення для наукових робіт
- Амортизація обладнання
- Паливо та енергія
- Службові відрядження
- Сторонні роботи
- Інші витрати
- Накладні витрати
- Підсумковий розрахунок

Рисунок Г.6 – Панель розрахунків витрат на оплату праці

Підсумковий розрахунок загальних витрат

Зарплата дослідників:	115812.00
Зарплата робітників:	2783.52
Додаткова зарплата:	14231.46
Соціальні відрахування:	29221.94
Матеріали:	1476.00
Комплектуючі:	5544.00
Спецустановлення:	5544.00
Програмне забезпечення:	396.00
Амортизація обладнання:	1161.50
Енерговитрати:	8064.00
Службові відрядження:	2222.80
Сторонні роботи:	1985.30
Інші витрати:	1135.52
Накладні витрати:	3428.00

Розрахувати

Загальна сума витрат = 193006.04 грн

Назад

Рисунок Г.7 – Підсумкове вікно програми з розрахованими даними

Звіт про витрати

Меню

Результати розрахунку витрат

Назва	Значення (грн)
Зарплата дослідників	115812.00
Зарплата робітників	2783.52
Додаткова зарплата	14231.46
Соціальні відрахування	29221.94
Матеріали	1476.00
Комплектуючі	5544.00
Спецустановлення	5544.00
Програмне забезпечення	396.00
Амортизація обладнання	1161.50
Енерговитрати	8064.00
Службові відрядження	2222.80
Сторонні роботи	1985.30
Інші витрати	1135.52
Накладні витрати	3428.00

Підсумкова сума = 193006.04 грн

Зберегти як... **Назад**

Рисунок Г.8– Вікно сформованого звіту з результатами розрахунків

	A	B
1	Результати розрахунку витрат	
2		
3	Назва	Значення (грн)
4	Зарплата дослідників	115 812,00 грн
5	Зарплата робітників	2 783,52 грн
6	Додаткова зарплата	14 231,46 грн
7	Соціальні відрахування	29 221,94 грн
8	Матеріали	1 476,00 грн
9	Комплектуючі	5 544,00 грн
10	Спецустаткування	5 544,00 грн
11	Програмне забезпечення	396,00 грн
12	Амортизація обладнання	1 161,50 грн
13	Енерговитрати	8 064,00 грн
14	Службові відрядження	2 222,80 грн
15	Сторонні роботи	1 985,30 грн
16	Інші витрати	1 135,52 грн
17	Накладні витрати	3 428,00 грн
18	Підсумок	193 006,04 грн
19		

Рисунок Г.9 – Приклад збереженого звіту у форматі Excel