

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

**" Віртуальний симулятор структури змінних
електромагнітних полів в хвилеводі із курсу
«ТЕОРІЯ ПОЛЯ»"**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Берест О.Б.

Студента групи ІН.м–02

Горячев М.Д.

СУМИ 2021

Сумський державний університет

Факультет ЕлІТ Кафедра Комп'ютерних наук

Спеціальність 122 «Інформатика»

Затверджую:

зав.кафедрою _____

“ _____ ” _____ 2021 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ**

Горячеву Максиму Денисовичу

1. Тема проекту (роботи) "Віртуальний симулятор структури змінних електромагнітних полів в хвилеводі із курсу «ТЕОРІЯ ПОЛЯ»"

затверджую наказом по інституту від “ _____ ” _____ 20 ____ р. № _____

2. Термін здачі студентом закінченого проекту (роботи) _____

3. Вхідні данні до проекту (роботи) _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблематики області дослідження; 2) Вибір методів вирішення та постановка завдання; 3) Дизайн додатку; 4) Теоретичні засади для вирішення поставленої задачі; 5) Розроблення і імплементація; 6) Тестування та інструкція користувача

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	Аналіз проблематики області дослідження	20.10.2021	
2.	Вибір методів вирішення та постановка завдання	26.10.2021	
3.	Дизайн додатку	12.11.2021	
4.	Теоретичні засади для вирішення поставленої задачі	20.11.2021	
5.	Розроблення і імплементація	28.11.2021	
6.	Тестування та інструкція користувача	29.11.2021	
7.	Оформлення пояснювальної записки до дипломної роботи	07.12.2021	

Студент – дипломник

Керівник проекту

РЕФЕРАТ

Записка: 86 стор., 42 рис., 6 табл., Додаток А, Додаток Б, 19 джерел.

Об'єкт дослідження — інформаційна система, яка моделює процес побудови структури змінних електромагнітних полів в хвилеводі.

Мета роботи — створити симулятор визначених процесів для курсової роботи «Розрахунок структури електромагнітних полів» із курсу «ТЕОРІЯ ПОЛЯ», щоб покращити точність и швидкість виконання цієї роботи.

Методи дослідження — Інформаційний, графічний, методи програмування

Результати — Проведено огляд аналогів і технологій програмних розрахунків процесів з подібним функціоналом. Розроблений та впроваджений програмний продукт для платформи Windows .

СИМУЛЯТОРИ, ВІРТУАЛЬНІ, МОДЕЛЬ, КОМП'ЮТЕРНІ
СИМУЛЯТОРИ, ДОДАТОК..

ЗМІСТ

СКОРОЧЕННЯ І УМОВНІ ЗНАЧЕННЯ	7
ВСТУП	8
1. АНАЛІЗ ПРОБЛЕМАТИКИ ОБЛАСТІ ДОСЛІДЖЕННЯ	10
1.1. Як симулятори змінили навчання та наукові експерименти.....	10
1.2. Технології віртуальних стимуляторів.....	12
1.2.1. Віртуальні лабораторії.....	12
1.2.2. Віртуальні (комп'ютерні).....	15
1.2.3. Комп'ютерні моделі об'єкта, що вивчається.....	16
1.3. Огляд існуючих програмних рішень.....	18
1.3.1. The Abyssal Engine.....	18
1.3.2. Cafu.....	18
1.3.3. CityEngine.....	19
1.3.4. Tracker.....	20
1.3.5. Graph.....	20
1.3.6. GeoGebra.....	21
1.4. Огляд технологій розробки стимуляторів.....	22
1.4.1. Симулятори на HTML5 для різних браузерів (Chrome, Opera і т.п.).....	22
1.4.2. Комп'ютерні симулятори.....	22
1.4.3. Swift.....	23
1.4.4. Java.....	23
2. ВИБІР МЕТОДІВ РІШЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ	24
2.1. Визначення та концепція методів реалізації мети роботи.....	24
2.2. Сценарії інформаційної системи програми.....	25
2.3. Вимоги до програмного продукту.....	27
2.4. Модель програмного продукту.....	29
2.4.1. Структурно-функціональна модель процесу.....	29
2.4.2. Основний функціонал програми.....	30

2.5.	Модель варіантів використання (ВВ).....	30
3.	ДИЗАЙН ДОДАТКУ	33
3.1.	Аналіз потреб щодо реалізації функціоналу програми.....	33
3.2.	Ескізи інтерфейсів які будуть використовуватись.....	35
3.3.	Шаблон Template MS Word.....	37
4.	ТЕОРЕТИЧНІ ЗАСАДИ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ	40
4.1.	Математична модель порожніх металевих спрямовуючих систем (хвилеводів)	40
4.1.1.	Постановка задачі моделювання.....	40
4.1.2.	Теоретическое исследование математической модели.....	42
4.2.	Послідовність розрахунку складових електромагнітного поля.....	48
5.	РОЗРОБЛЕННЯ І ІМПЛІМЕНТАЦІЯ	50
5.1.	Архітектура програмного продукту.....	50
5.2.	Проектування програми.....	53
5.3.	Короткий опис програмної реалізації.....	55
6.	ТЕСТУВАННЯ ТА ІНСТУРКЦІЯ КОРИСТУВАЧА	64
6.1.	Тестування програми та узагальнення роботи.....	64
	ВИСНОВКИ	69
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71
	ДОДАТОК А.....	74
	ДОДАТОК Б.....	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Моделювання — це процес створення моделі для подальшого дослідження об'єкта (процесу, явища).

Інформаційна модель — це опис важливих для певного дослідження властивостей об'єкта (явища, процесу).

Математична модель — це опис прототипу за допомогою математичних співвідношень, формул, рівнянь.

Комп'ютерна модель — це інформаційна модель, реалізована за допомогою програмного середовища (текстового або графічного редактора, редактора презентацій, середовища програмування тощо).

Графічна комп'ютерна модель — це опис об'єкта (явища, процесу) у вигляді схем, діаграм, рисунків, створений за допомогою комп'ютерних технологій

Структурно-функціональна модель — це опис об'єкта (явища, процесу) у вигляді схем і таблиць. Результати дослідження моделі дозволяють виявити найсуттєвіші властивості об'єкта, що впливають на його поведінку, вибрати оптимальні способи керування об'єктом (наприклад, космічною станцією).

Імітаційна модель — програма або комплекс програм, що реалізує алгоритм функціонування об'єкта в різних умовах

комп'ютерний експеримент — дослідження математичної моделі за допомогою комп'ютера

$f_{кр}$ (критична частота) — це гранична частота, що розділяє діапазон нормального поширення $f > f_{кр}$ електромагнітних полів (без загасання) у вигляді хвиль і діапазон відсічення $f < f_{кр}$ (поширення хвиль із експоненціальним загасанням).

ВСТУП

Останні роки характеризуються модернізуванню інформатизації освіти та розвитку віртуальних складових технологій навчання: нові формати подання матеріалу, доступ до всіх можливих освітніх ресурсів в Інтернеті, а також нові форми організації навчання (вебінари, відеолекції, віртуальні лабораторії, тренажери, симулятори). У різні сфери діяльності людини стрімко впроваджується велика кількість віртуальних технологій. Одним із національною стратегією розвитку освіти в Україні є пріоритетом розвитку освіти було визначено впровадження сучасних інформаційно-комунікаційних технологій.

Беручи до уваги сучасній електроніці і надвисокочастотних(НВЧ) технологіях і спираючись на підручники, посібники та електронні каталоги, що з'являються останніми роками, можна прийти до висновку, що структура курсових робіт студентів ґрунтуватиметься на переході від окремих видів полів (електростатичного, електричного і магнітного полів постійного струму) до загальних рівнянь електромагнітних полів. Такий підхід дозволяє на власні очі впевнитися у справедливості тих чи інших законів, зрозуміти їх фізичний зміст та вдосконалити і поглибити практичні навички в розрахунках електромагнітних полів. Ефективність освітнього процесу можна підвищити завдяки застосуванню сучасних інформаційно–комунікаційних технологій. Такі технології сприяють підвищенню інформативно–комунікативної компетентності студентів, активізації пізнавальної діяльності та розвитку їхньої самостійності в опануванні знань, посиленню позитивної мотивації навчання завдяки формуванню умінь та навичок роботи із сучасними інтерактивними технологіями. Подальші поліпшення навчання досягаються безпосереднім виконанням практичних завдань. У всьому світі існує тенденція розробки

навчальних симуляторів , тому для полегшення виконання завдань навчання використовують симулятори тих та інших процесів.

Такий підхід дозволяє вивчити основні закони теорії електромагнітного поля, зрозуміти загальні принципи роботи і методи розрахунку різноманітних радіофізичних і електрофізичних пристроїв, а також їх елементів..

Основна мета розробки таких симуляторів це спрощення роботи в декілька разів, що підвищує швидкість та точність результату. Завдання симуляторів візуалізувати вже набуті знання. Особливістю симуляторів є використання анімації, яка є корисною для того, щоб допомогти студентам уявити динаміку фізичних процесів. Студенти можуть легко змінювати необхідний параметр і тут же спостерігати за його впливом на стан системи. Ілюстрація можливості дослідження фізичних властивостей наведена на Рис.1

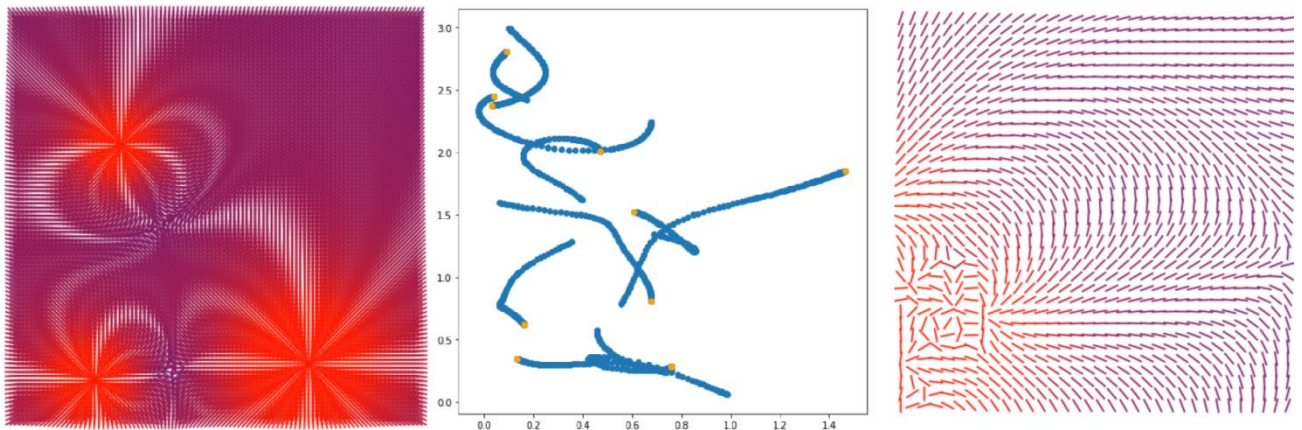


Рисунок 1 — Візуалізація векторного поля напруги електромагнітного поля

Інтерактивне комп'ютерне моделювання за допомогою симуляторів є унікальним, потужним, доступним та технологічним інструментом для вивчення фізичних властивостей.

1 АНАЛІЗ ПРОБЛЕМАТИКИ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Як симулятори змінили навчання та наукові експерименти

Симуляція — імітація певної реальної речі, ситуації чи процесу. Процес симуляції зазвичай включає відтворення деяких ключових властивостей чи поведінки обраної фізичної або абстрактної системи. Рис. 1.1-1.3



Рисунок 1.1 —
Симулятор коня часів
першої світової



Рисунок 1.2 — Студенти-
медики на заняттях



Рисунок 1.3 —
Віртуальний симулятор
автомобіля

Приклад фантастичного віртуального симулятора - фільм "Матриця" режисерів Вачовські. Серед прихильників ідеї симуляції є й підприємець Ілон Маск, який у червні 2016 року заявив, що ймовірність цього складає "мільярд до одного".

А технічний директор Google з питань штучного інтелекту Реймонд Курцвейл припускає, що "весь наш Всесвіт - лише науковий експеримент учнів молодших класів з іншого всесвіту".

Симуляцію проводять з різною метою — тренування та навчання персоналу, тестування технології в граничних умовах, тестування безпеки, розваги (відеоігри, симуляція невагомості). Також використовується науковцями для здійснення експериментів, які неможливі в реальності. Також

симуляцію використовують, щоб продемонструвати можливі ефекти певних дій. Зазвичай симуляцію проводять, коли експерименти над реальною системою неможливі через її недосяжність, небезпеку або високу вартість таких експериментів. [1].

Незаперечним фактом є те, що під час вивчення фізики велике значення має навчальний експеримент.



Рисунок 1.4 — Блискавка в лабораторії.

Безсумнівно, реальні фізичні досліди (Рис. 1.4) завжди цікавіші за віртуальні, але моделювати фізику в комп'ютері набагато простіше і дешевше (в даному випадку - взагалі безкоштовно) і можна використовувати такі цікаві ефекти, як уповільнення часу, масштабування і т.д. Студенти, які вивчають природничі науки, технології, інженерію та математику (STEM), повинні мати справу з складними моделями реальних систем. Незалежно від того, чи відносяться ці моделі до інфраструктурних систем, телекомунікаційних мереж, комп'ютерних систем або ланцюжків поставок, студентам STEM необхідно пройти спеціалізовану підготовку для прийняття обґрунтованих рішень, які впливають на складні системи з тисячами змінних і взаємодій між їх елементами.

1.2 Технології віртуальних стимуляторів

Завдання створення навчальних або наукових симуляторів часто дуже відрізняються. Відмінності характерні як самого навчального інтерфейсу, так середовища, в якому програма повинна симулювати взаємодії. Моделювання використовує математичний опис, або модель, реальної системи у формі комп'ютерної програми. Ця модель складається з рівнянь, які дублюють функціональні взаємозв'язки всередині реальної системи. При запуску програми результуюча математична динаміка формує аналог поведінки реальної системи, а результати представлені у вигляді даних. Моделювання також може приймати форму зображення комп'ютерної графіки, яке представляє динамічні процеси в анімований послідовності.

За реалізацією технологічно симулятори умовно можливо розділити на групи:

1) віртуальні лабораторії – програмно-апаратний комплекс (електронне середовище), що дозволяє проводити досліди без безпосереднього контакту з реальною установкою, лабораторією, обладнанням або за повної їх відсутності (наприклад, проведення лабораторних робіт, фізичних, хімічних дослідів тощо);

2) віртуальні (комп'ютерні) тренажери – електронне середовище для виконання професійних завдань, відпрацювання практичних умінь;

3) комп'ютерні моделі об'єкта, що вивчається – заміщення об'єкта дослідження, конкретних предметів, явищ з метою вивчення їх властивостей, отримання необхідної інформації про об'єкт.

1.2.1 Віртуальні лабораторії

Це комп'ютерна симуляція, яка максимально наближена до реальності імітація фізичних процесів та застосовує моделювання об'єктів (явищ, процесів) за допомогою комп'ютера. Застосування їх у професійній освіті дозволяє студентам та науковцям освоювати або удосконалювати теоретичні знання, необхідні практичні вміння у безпечних умовах, з меншими витратами

(тимчасовими, економічними та ін.), за недоступності необхідного обладнання, специфіки досліджуваного явища (масштаб, тривалість перебігу процесу та ін.), знижує ризик при помилкових діях, дозволяє опрацьовувати ситуацію кілька разів з огляду на попередній досвід, а також дозволяє задавати різноманітні умови діяльності з різним рівнем складності. Комп'ютерне моделювання використовується для вивчення динамічної поведінки об'єктів або систем у відповідь на умови, які не можуть бути легко або безпечно застосовані в реальному житті.

Наприклад, ядерний вибух може бути описаний математичною моделлю, яка включає такі змінні, як тепло, швидкість і радіоактивні викиди. Потім можна використовувати додаткові математичні рівняння для коригування моделі з урахуванням змін певних змінних, таких як кількість матеріалу, що розщеплюється, який справив вибух. Моделювання особливо корисно для того, щоб дозволити спостерігачам вимірювати і прогнозувати, як на функціонування всієї системи може вплинути зміна окремих компонентів в цій системі. Рис. 1.5-1.7. За допомогою комп'ютера можна моделювати явища або процеси, які не відбувалися в реальності (наприклад, вибух на хімічному підприємстві), дуже швидкоплинні (вибух наднової зірки) або довготривалі процеси (зокрема, геологічні) тощо.

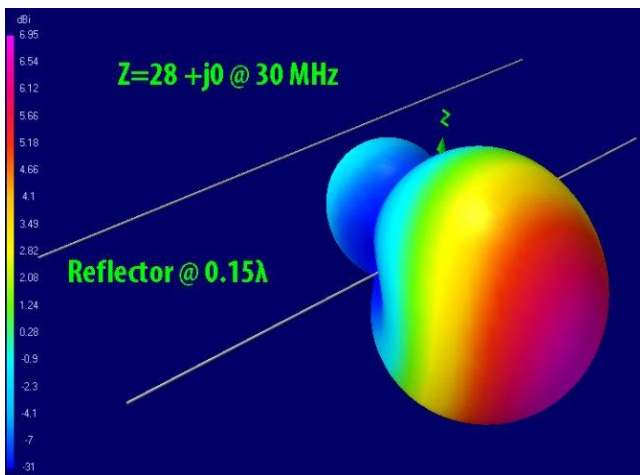


Рисунок 1.5 — Зміна діаграми спрямованості антени від частоти

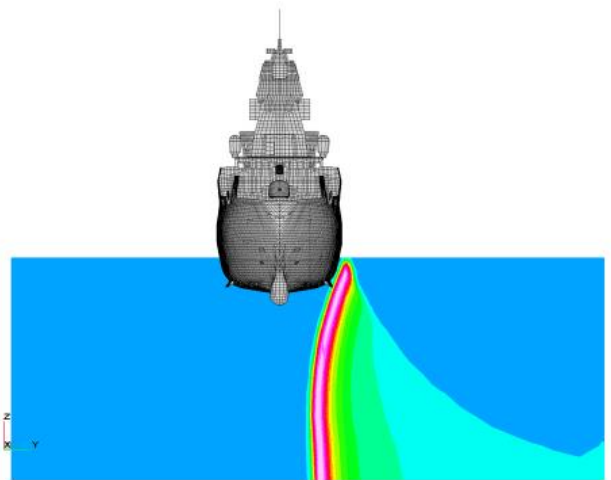


Рисунок 1.6 — Дії вибухової хвилі на корпус корабля

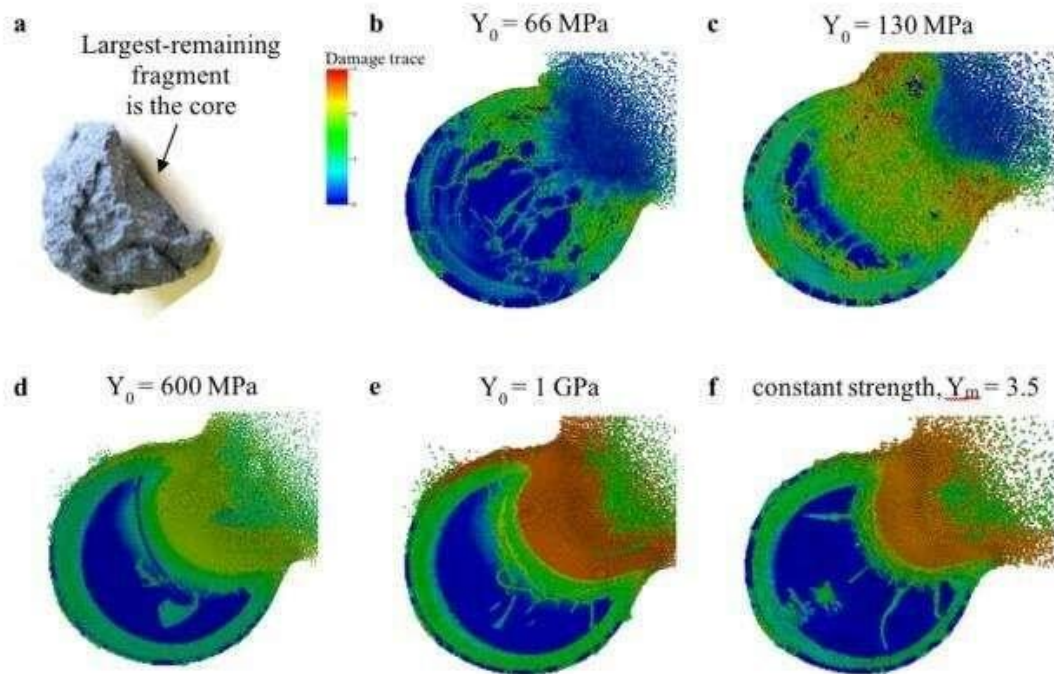


Рисунок 1.7 — У Ліверморській лабораторії моделюють можливість відхилення астероїда за допомогою ядерного вибуху

1.2.2 Віртуальні (комп'ютерні)

Тренажери використовують імітаційну модель інформаційні моделі реальних об'єктів і можуть бути розділені на групи відповідно до виконуваних функцій:

1. Тренажери для набуття знань - Електронні підручники (ЕУ). Завдяки широкому використанню мультимедійних засобів (графіки, анімації, звуку) ефективність навчання значно підвищується. Сучасні технології дозволяють легко доповнювати математичні формули, присутні в ЕУ "спливаючими" підказками, і графічними ілюстраціями з контекстуальними поясненнями.

2. Контролюючі тренажери - програми тестування, призначені для перевірки знань студента з лабораторних тем. Їх можна використовувати для самопідготовки або для отримання теоретичного допуску до роботи. Тести можуть включати питання, що дозволяють визначити ступінь готовності студента до осмисленої роботи з тренажером. Для посилення контролюючого ефекту результати проходження тестів оцінюються в балах, студентові повідомляється кількість пройдених тестів і сума штрафних балів. Для мінімізації вгадування відповідей в програмі блокується вивід на екран інформації з результатом кожного окремого тесту.

3. Тренажери для навчання навичкам - мультимедійні анімаційні імітатори, призначені для імітації зміни станів фізичного обладнання (приладів, пристроїв) за різних умов, що створює ілюзію дій з фізичною апаратурою. Їх головною особливістю є найбільш повне відтворення зовнішнього вигляду фізичних пристроїв (передніх панелей, шкал, стрілок та інших елементів які відображують та записують) і елементів управління (кнопок, тумблерів, перемикачів), а також рух окремих елементів відповідно до дій користувача на основі створення анімаційних об'єктів і складних сцен. Студент має можливість детально розглянути та вивчити технічний пристрій, ознайомитися з його

деталіями, а також виконати обмежений набір дій, пов'язаних з розбиранням або настройкою приладу.

До імітаційних моделей відносять також ігрові моделі, що дозволяють відтворювати поведінку об'єкта в ігрових ситуаціях з урахуванням реакції з боку суперника (Рис.1.8).



Рисунок 1.8 — ігрові моделі

1.2.3 Комп'ютерні моделі об'єкта, що вивчається

При проведенні експериментів і виконання розрахунків виникає необхідність побудови матеріальних або інформаційних моделей реальних об'єктів – прототипів моделі [2]

Матеріальні моделі відтворюють фізичні, геометричні та інші властивості об'єктів моделювання (глобус, модель молекули, макет літака, будинка тощо).

Модель повинна враховувати тільки ті властивості об'єкта, які важливі для вирішення певної задачі.

Якщо розв'язувати задачу на знаходження шляху, пройденого за певний час автомобілем, що рухається з постійною швидкістю, то враховують тільки фізичні характеристики руху автомобіля — швидкість руху і час, а такі дані, як колір, марка чи ціна автомобіля, ігноруються як несуттєві.

Одним із типів інформаційних моделей є математичні моделі. Такі математичні моделі використовують у задачах, які потребують обчислень.

Розрахункові моделі (Рис. 1.9) дозволяють визначити числові значення властивостей об'єкта на основі різних вихідних даних. Аналіз цих значень допомагає спрогнозувати стан об'єкта в майбутньому: модель атмосфери Землі дозволяє передбачити погоду, модель екосистеми — визначити межі користування природними ресурсами.



Рисунок 1.9 — Розрахункова модель рівноприскореного прямолінійного руху

Комп'ютерний експеримент — це потужний інструмент для вивчення реального об'єкта. Після побудови, зокрема, математичної моделі оцінюється її відповідність реальному об'єкту. Деякі параметри моделі використовуються для розрахунку інших її параметрів і порівняння їх з реальними параметрами об'єкта. Якщо параметри моделі та її прототипу не збігаються, вони вказують на те, що математична модель не відповідає прототипу. Після перевірки правильності моделі її подальше дослідження надає нову інформацію про реальний об'єкт.

Існує велика кількість симуляторів здатних моделювати будь-які об'єкти, процесі та явища. Але потрібно пам'ятати, що результати проведеного комп'ютерного експерименту, імітація фізичних процесів та віртуальні тренажери, характеризують модель, а не прототип. Якщо на етапі проведення експерименту результати виявилися невірогідними, виникає потреба вибору іншого методу дослідження, уточнення моделі або навіть внесення змін у постановку задачі щодо розробки програмного продукту.

1.3 Огляд існуючих програмних рішень

У наші дні, чим більше інструментів у вас в наборі, тим краще. Тобто має сенс знати більше одного способу реалізації рішення, тому що в деяких випадках одна реалізація може мати переваги перед іншими. Так як говориться про технологій, маємо розуміти та знати вже існуючі представників

1.3.1 The Abyssal Engine

Abyssal Engine-це потужний професійний інструментарій для створення ігор і система рендеринга для створення якісного 3D-програмного забезпечення. Технологія призначена для скорочення часу розробки, надаючи розробникам потужний набір інструментів, необхідних для створення чудових візуальних ефектів і занурення в онлайн - або оффлайн-світи. (Рис. 1.10)



Рисунок 1.10 — Інтерфейс Abyssal Engine

1.3.2 Cafu

Движок Cafu-це універсальний, сучасний 3D-графічний движок і набір для розробки ігор, який дозволяє швидко почати роботу. Cafu доступний для всіх як 100% вільного програмного забезпечення під ліцензією GNU General Public License (GPLv3). Його можна використовувати для створення різних 3D-додатків, включаючи ігри, Симулятори, навчальне та архітектурне програмне забезпечення. Написані на C++, інструменти, бібліотеки та фреймворк були розроблені для полегшення розробки нових ігор та інших 3D-додатків. Вони

активно і безперервно розвиваються, щоб надавати новітні технології. (Рис. 1.11)

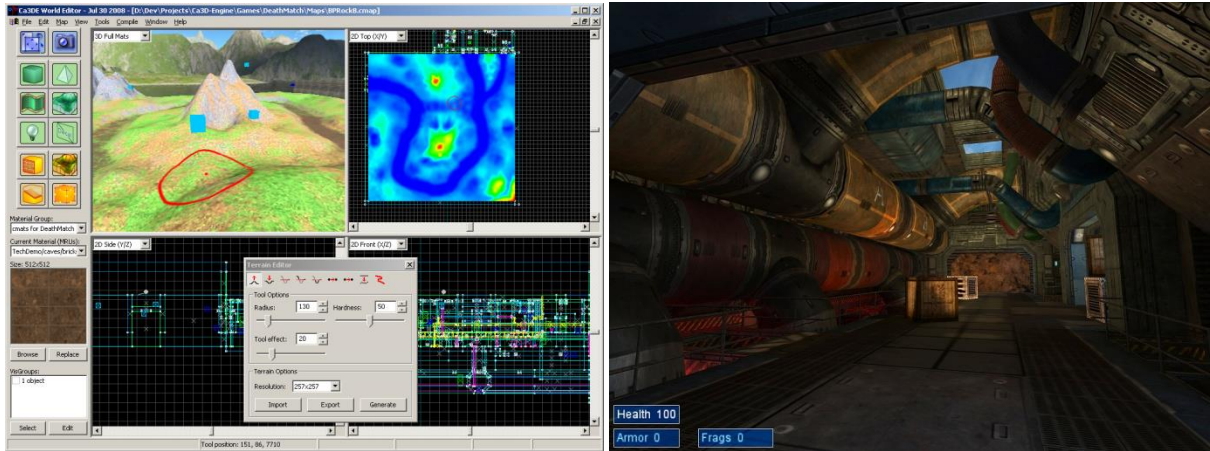


Рисунок 1.11 — Інтерфейс SaFu

1.3.3 CityEngine

CityEngine надає професійним користувачам в області розваг, архітектури, міського планування і загального створення 3D-контенту унікальне рішення для раннього проектування і моделювання для ефективної візуалізації 3D-міського середовища. (Рис. 1.12)

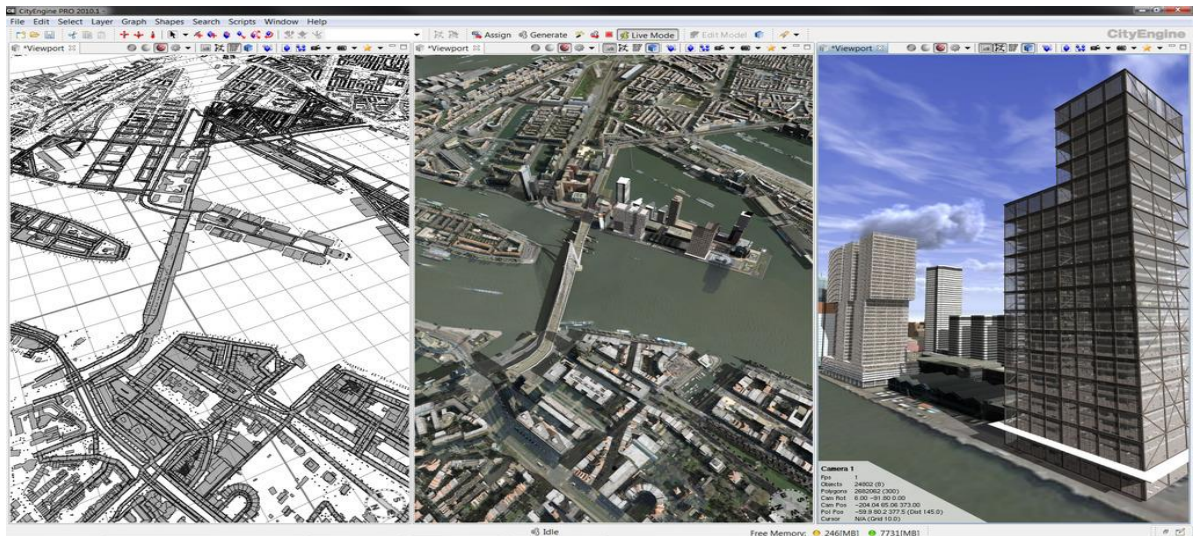


Рисунок 1.12 — Інтерфейс CityEngine

1.3.4 *Tracker* — безплатний інструмент, що надає змогу моделювати та аналізувати рух об'єктів на відео чи зображеннях. (Рис. 1.13) Програмний засіб призначений для використання у фізиці. Завантажити програму можна безплатно зі сторінки [3].

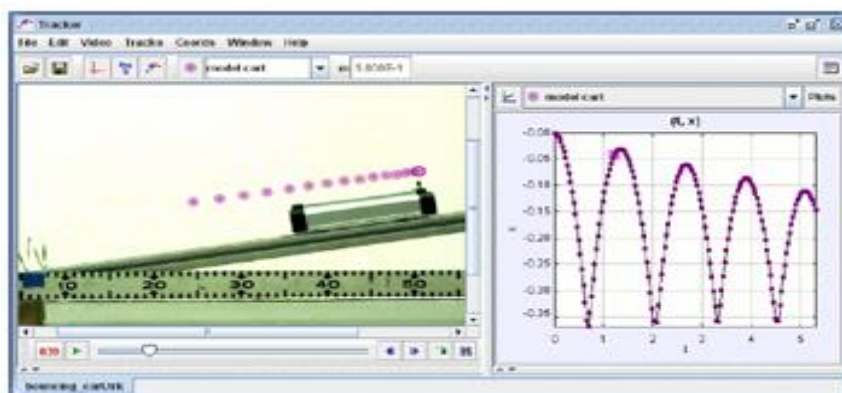


Рисунок 1.13 — Інтерфейс симулятора *Tracker*

1.3.5 *Graph* — програма з відкритим кодом і призначена для побудови математичних графіків (Рис. 1.14). Програма вмє будувати графіки різних функцій і рівнянь, створювати таблиці, креслити графіки, які проходять через задані точки, проводити обчислення екстремумів і нульових точок тощо. Завантажити програму можна безплатно зі сторінки [4].

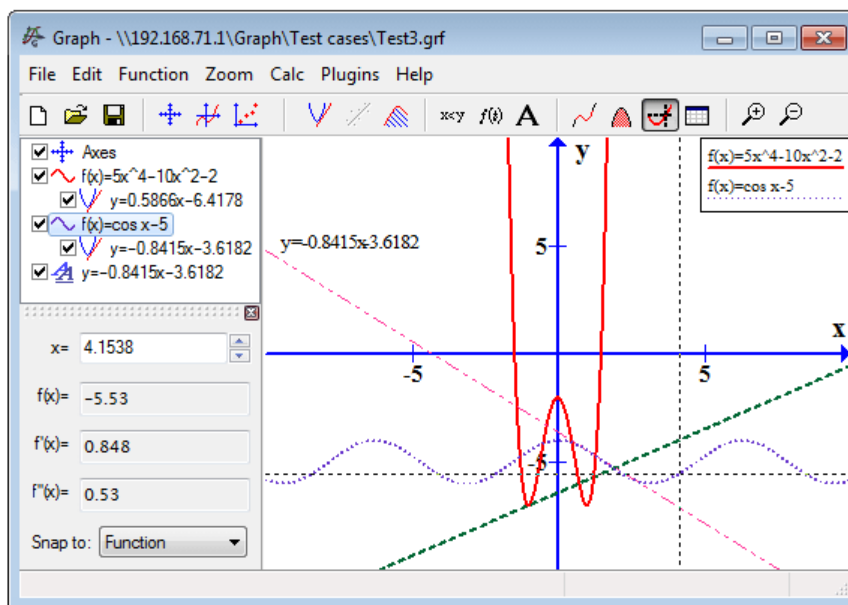


Рисунок 1.14 — Інтерфейс симулятора *Graph*

1.3.6. *GeoGebra* — програма, призначена для вчителів середньої школи, а також для учнів. Вона дає змогу легко вибудовувати графіки, фігури та вектори, розв'язувати графічно рівняння тощо. Усі елементи, які входять до складу, є динамічними й можуть бути змінені за 2–3 кліка. Крім того, програма *GeoGebra* виконує різні обчислення, наприклад, можна обчислити інтеграл, площу фігури, довжину траєкторії або досліджувати будь-яку функцію. Програма *GeoGebra* (Рис. 1.15.) — це повністю безоплатний продукт і, може бути завантажений зі сторінки [5].

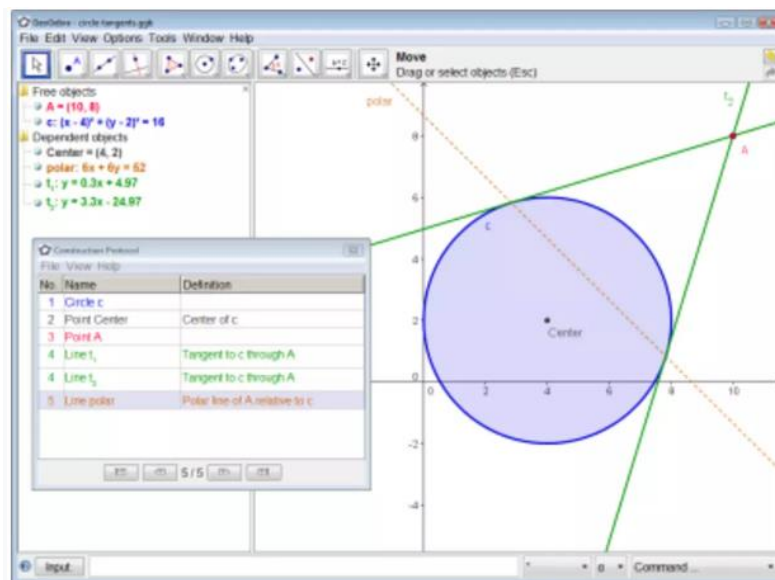


Рисунок 1.15 — Інтерфейс симулятора *GeoGebra*

Усі зазначені засоби стануть у нагоді під час аналізу отриманих результатів і в жодному разі не є альтернативою реальному експерименту в умовах фізичного кабінету. Навпаки, вони мають лише розширити можливості викладачів та студентів в організації дослідницької діяльності.

Будучи доповненням до традиційних методів навчання, програмне забезпечення для моделювання та серйозні ігри все частіше використовуються в якості освітніх інструментів протягом останніх десятиліть. Триваючий розвиток технологій і мов програмування зробило моделювання передовим інструментом, який з високим ступенем точності відображає реальні ситуації.

Особливу увагу заслуговує сайт університету Колорадо (phet.colorado.edu), де безплатно пропонуються для навчання різноманітні інтерактивні симуляції.

1.4 Огляд технологій розробки симуляторів

У кожній мові програмування свої переваги та призначення, тому не варто думати, що якась краща за інші — всі вони для вирішення різних завдань. Багато хто, наприклад, пишуть більшу частину проекту на одному, а високонавантаженому — на іншому.

Щоб вибрати, яка мова підійде нам, потрібно розібратися в декількох моментах.

- Swift — симулятори для iOS або macOS.
- PHP та JavaScript – Симулятори на HTML5.
- C#, Java - невимогливі стимулятори
- C або C++ – великі вимогливі симулятори.

1.4.1 Симулятори на HTML5 для різних браузерів (Chrome, Opera і т.п.)

Вони хоч і не такі круті, як симулятори для комп'ютерів, але теж затягують. Причина в умілій роботі дизайнерів - вони продумують механіки так, щоб ви отримували від стимулятора більше комфорту. Тут вже можна підключити JavaScript — він дозволяє зберігати в змінних дані об'єкту, а обробляти графіку за допомогою Canvas. Якщо прикрутити PHP, вдасться створити базу даних, побудувати захист і реалізувати розрахований на багато користувачів режим.

1.4.2 Комп'ютерні симулятори. У таких симуляторах ведеться робота над освітленням, тінями, частинками, руйнівністю — усім, що є важливим для конкретного проекту. Одна людина не потягне все це за рік чи навіть два, тому таке під силу лише великим студіям. Звичайно, багато що вирішується в движках, але робота все одно колосальна. Без движків також можна обійтися: World of WarCraft був написаний на C++, а MineCraft створений однією

людиною на Java, після чого гру купила компанія Microsoft за 2,5 мільярда доларів.

1.4.3. Swift — відкрита мультипарадигмальна компілювана мова програмування загального призначення. Створений компанією Apple насамперед для розробників iOS та macOS.

1.4.4. Java

Кроссплатформенності. Java запускається всередині Java Virtual Machine (JVM), а це означає, що ваш код не звертається безпосередньо до операційної системи, тобто теоретично може працювати будь-де, де є ця сама JVM, чи то Windows, Linux, MacOS, мобільні платформи, а також у робототехніці та вбудовуваних системах. Java створений для масштабованості, тому він такий популярний серед великих проектів і стартапів, що ростуть (Twitter перейшов з Ruby на Java). Оскільки Java є мовою статичної типізації, її простіше та швидше підтримувати з меншою кількістю помилок. Він також має зворотну сумісність, що означає, що старі версії додатків, як і раніше, працюватимуть ідеально навіть після переходу на нові версії мови.

Враховуючи невеликий обсяг розрахунків нам потрібен невимогливі до ресурсів симулятор, якій доцільно реалізувати на мові Java .

2 ВИБІР МЕТОДІВ РІШЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ

2.1 Визначення та концепція методів реалізації мети роботи

Здійснивши огляд всіх можливих до застосування технологій симуляторів та враховуючі визначеність об'єкту інформаційної моделі, в нашому випадку це хвилевод, то для дослідження структури змінних електромагнітних полів в із курсу «ТЕОРІЯ ПОЛЯ» нам достатньо побудувати математичну модель, графічну комп'ютерну модель та провести комп'ютерний експеримент[6]. Симулятор будемо реалізовувати на мові JavaFX за методом комп'ютерної симуляції.

Мета роботи – створити симулятор визначених процесів для курсової роботи «Розрахунок структури електромагнітних полів» із курсу «ТЕОРІЯ ПОЛЯ», щоб покращити точність и швидкість виконання цієї роботи. Задля досягнення основної мети потрібно вирішити наступні питання: визначення функціональних вимог, розробка дизайну та інтерфейсу програми, створення алгоритму розрахунків.

Метою є програма, яка буде дозволяти користувачеві вводити дані у відповідні поля головної сторінки інтерфейсу, зчитувати ці дані із інтерфейсу програми, робити обробку даних за заданими параметрами з використанням математичних формул за визначеним алгоритмом. Результати розрахунків потрібно виводити на екран та за запитом користувача зберігати у файлі MS Word.

Задачею для реалізації поставленої мети є:

1. Створення програмного коду, якій буде реалізовувати інтерфейс користувача, щодо вводу управляючих параметрів та виводу розрахованих даних та графіків.
2. Створення програмного коду, якій буде здійснювати розрахунок параметрів відповідно алгоритму;

3. Створення програмного коду, який буде результати зберігати у файлі MS Word;

2.2 Сценарії інформаційної системи програми

Поверхневий опис у вільній формі усіх сценаріїв (головного і альтернативних)

Користувач потрапив у програму. Користувачу доступна головна сторінка програми та можливість вибору однієї із сторінок «*Архів записів*» або «*Новий Розрахунок*». Одночасно додаток установлює зв'язок з базою даних для подальших завантажень .

Початковою сторінкою є «*Архів записів*» . При активації цієї сторінки програма здійснює запит до бази даних для завантаження усіх наявних записів із БД у компонент для відображення списку записів, який розташовано на сторінці.

Користувач має можливість на цій сторінці при позначенні відповідного запису (установці курсору на запис) переглянути інформацію про раніше зроблені розрахунки, а саме виведення вихідних даних та графіків цього запису в інтерфейс програми. Програма здійснює запит до БД, отримані дані розміщує у відповідних групах вихідних значень та графіків.

За потребою користувач має можливість зберегти окремо усі дані вибраного запису у форматі програми MS Word , натиснувши кнопку «*Зберегти у форматі Word*». Програма автоматично створює, за встановленим шаблоном, файл у форматі програми MS Word та записує в нього дані вибраного запису розрахунку.

При необхідності проведення нового розрахунку користувач переходить на сторінку «*Новий Розрахунок*». На цій сторінці розташовуються поля вводу вхідних значень для розрахунку. Користувач повинен заповнити усі наявні поля. Користувач може відмовитись від введення покинувши сторінку. Дані ,які він вже ввів не зберігаються.

Після введення даних у поля вводу, для продовження, користувач натискає кнопку «Розрахувати». Програма здійснює перевірку валідності даних, що вводяться користувачем у поля сторінці. Далі програма здійснює розрахунок за порядком передбаченим математичною моделлю. Після розрахунку вихідних даних програма створює запит до БД на видалення запису з найменшою датою та запит на створення нового запису з поточною датою. Програма перенаправляє користувача на початкову сторінку для перегляду записів.

Альтернативний сценарій. У випадку необхідності створення повторного розрахунку на підставі вхідних даних запису, якій переглядається, користувач натискає кнопку повторній розрахунок та переходить на сторінку «Новий розрахунок». При цьому усі вхідні дані цього запису автоматично заповнюють відповідні поля на сторінці «Новий розрахунок».

Відповідно до Сценарію інформаційної системи програми визначимо схему взаємодії сценаріїв системи (Рис 2.1)



Рисунок 2.1 — Схема взаємодії сценаріїв системи

2.3 Вимоги до програмного продукту

Вхідними даними є :

1. БД MySQL з встановленою структурою:

Таблиця 2.1 – Структура даних

Атрибут даних щодо результату	Тип даних
дата результату	поле формату «Дата-время»
тип хвилі	текстове поле 1 знак
структура поля m	цифрове поле 2 знака
структура поля n	цифрове поле 2 знака
початкова комплексна амплітуда	цифрове поле 8 знаків
розмір хвилеводу a	цифрове поле 2 знака
розмір хвилеводу b	цифрове поле 2 знака
довжина хвилі в необмеженому просторі	цифрове поле 2 знака
діелектрична проникність	цифрове поле 2 знака
поля E , від координат хвилеводу по осі x	Масив double
поля H , від координат хвилеводу по осі x	Масив double
поля E , від координат хвилеводу по осі y	Масив double
поля H , від координат хвилеводу по осі y	Масив double
поля E , від координат хвилеводу по осі z	Масив double
поля H , від координат хвилеводу по осі z	Масив double

2. Користувацький інтерфейс, щодо вводу/виводу значення параметрів для розрахунків (блок 1):

- 1) тип хвилі (три групи хвиль) – список фіксованих значень;
- 2) структура поля в поперечних площинах хвилеводу для E - і H -хвиль (визначається цілими числами $m = 0, 1, 2, 3, \dots$ і $n = 0, 1, 2, 3, \dots$) - тільки цілі позитивні числа;
- 3) початкова комплексна амплітуда E_0 - тільки цілі позитивні числа;
- 4) розмір хвилеводу – - тільки цілі позитивні числа ;
- 5) довжина хвилі в необмеженому просторі - тільки цілі позитивні числа;
- 6) діелектрична проникність - тільки цілі позитивні числа.

3. Користувацький інтерфейс, щодо набору вихідних характеристик полів, якій користувач повинен вибрати для побудови графіків (блок 2):

1) залежності компонентів поля, окремо H та E , від координат хвилеводу по осях x , y , z (б епюр розподілу електричного й магнітного полів хвилі) - опція вибору;

2) залежності фазової і групової швидкостей розповсюдження хвилі від частоти - опція вибору.

Вихідними даними є

1. Розраховані значення (блок 3):

1) довжина хвилі у хвилеводі

2) еквівалентний опір хвилеводу для $E(H)$ -хвилі

3) критична частота

4) швидкості електромагнітної хвилі в необмеженому середовищі з відповідними параметрами діелектричного заповнення хвилеводу

5) фазова швидкість

6) групова швидкість

2. Графіки (блок 4)

1) вибрані залежності компонентів поля, від координат хвилеводу по осях x , y , z

2) залежність фазової і групової швидкостей розповсюдження хвилі від частоти

2.4 Модель програмного продукту (ПП)

Повна модель ПП розробляється на основі методології Rational Unified Process [7]. Повна модель ПП буде складатись із кількох елементів:

2.4.1 Структурно-функціональна модель процесу [8] [9], яка реалізується у розроблюваній програмі. Рис.2.2 , Рис.2.3



Рисунок 2.2 — Концептуальна модель - діаграма процесу обробки даних щодо хвиль в прямокутному хвилеводі

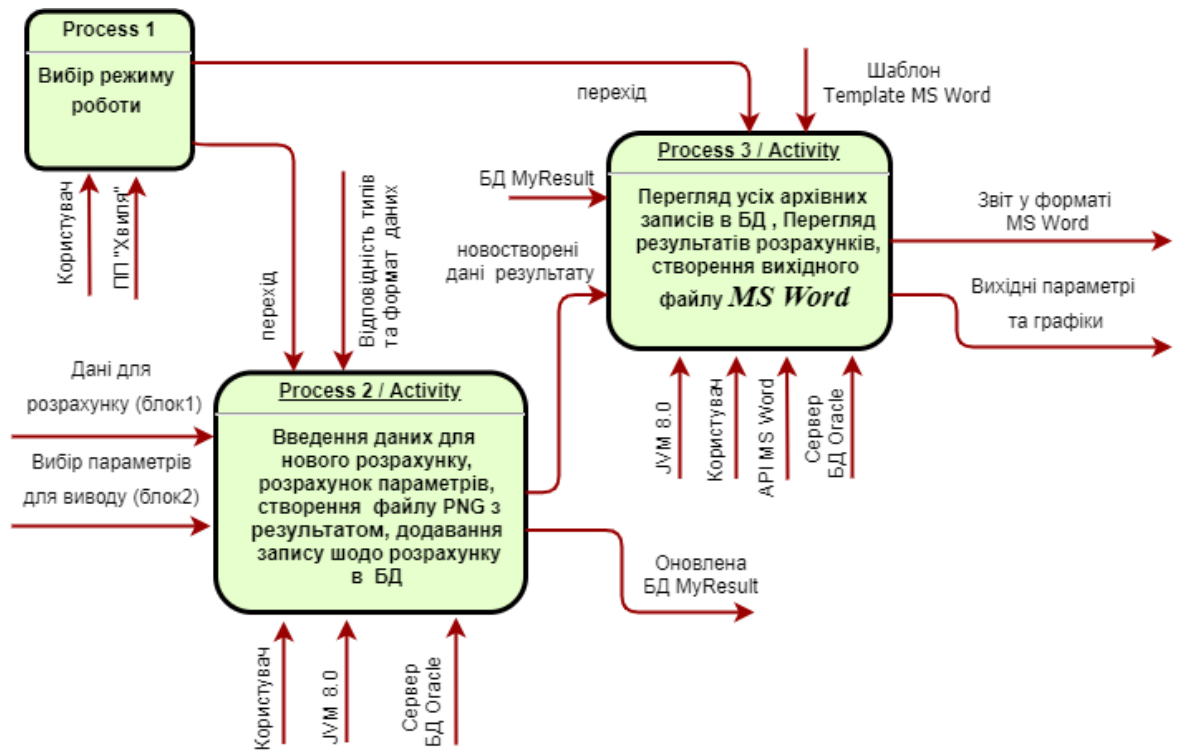


Рисунок 2.3 — Модель декомпозиції першого рівня в нотації IDEF

2.4.2 Основний функціонал програми, якій дає :

- можливість переглянути інформацію про раніше зроблені розрахунки ;
- введення вхідної інформації щодо параметрів модулювання експерименту ;
- перевірку валідності даних, що вводяться користувачем у поля форм ;
- розрахунок параметрів встановлених та заданих користувачем;
- виведення вихідних даних в інтерфейс програми.

2.5 Модель варіантів використання (VV)

Опис акторів

- Користувач - особа, яка використовує БД MyResult щодо введення нових записів по результатам симуляції, а також щодо пошуку результатів попередніх симуляції, за заданими параметрами дати існуючих записів;

b) MS Word - програма MS Word для збереження файлу із результатами вибору користувачем інформації про результати симуляції.

c) OracleSQL - СКБД Oracle для керування БД MyResult із інформацією результати симуляції.

Для реалізації функціоналу програми визначено 4 ВВ.

Таблиця 2.2 – Варіанти використання

ВВ	Актор, який приймає участь у ВВ		
	Customer	MS Word	Sql Oracle
UC01 Load Result (Загрузка архіву та виведення списку записів результатів)			×
UC02 Read Result(перегляд результату)	×		×
UC03 Add Result (Створення нового розрахунку та додавання до архіву)	×		×
UC04 Write Data (Запис у Word)	×	×	

Порядок взаємодії варіантів використання програмного продукту вказаний на Рис.2.4

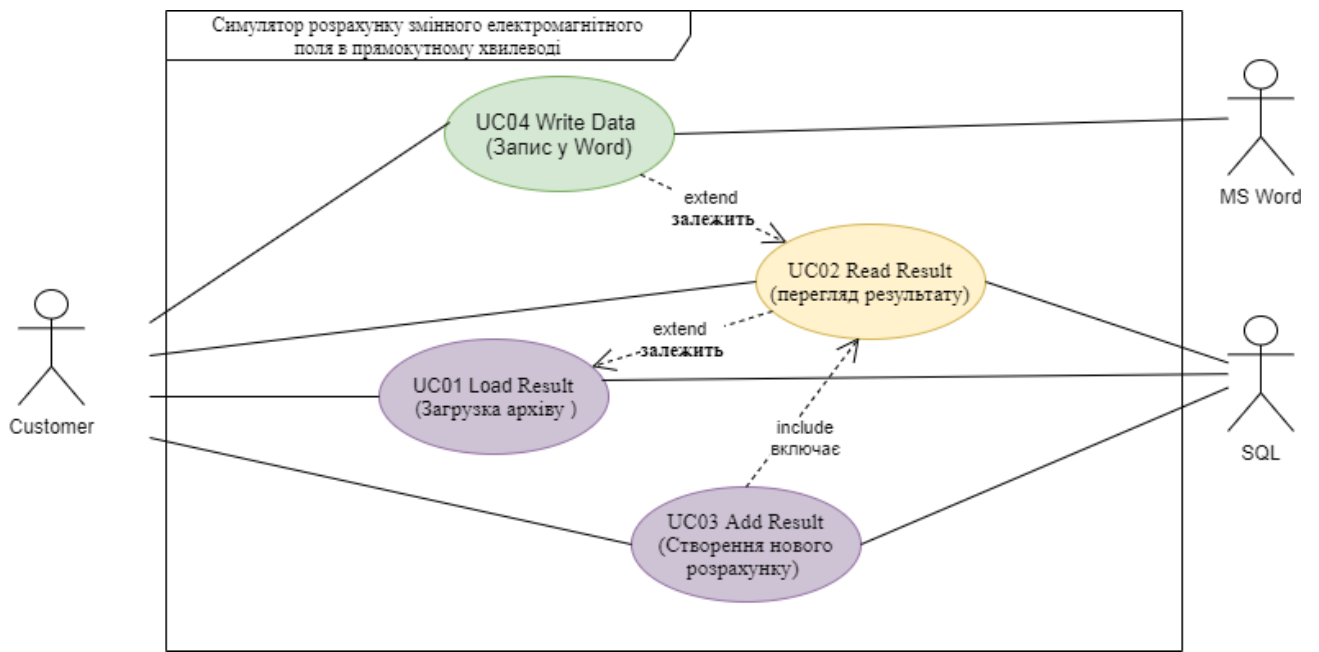


Рисунок 2.4 — Діаграма варіантів використання ПП

Проведено визначення методів реалізації на підставі концептуальної моделі побудови програмного продукту. Визначено структуру вхідних даних та потреба в користувацькому інтерфейсі. Створена модель програмного продукту з побудовою моделі процесу, визначенням функціоналу програмного продукту та моделі варіантів використання, за допомогою онлайн - редактора [10]. Сформовані вимоги до реалізації програмного продукту.

3. ДИЗАЙН ДОДАТКУ

Правильне розміщення елементів в інтерфейсі підвищує його зручність і робить продукт (ресурс) більш привабливим для користувачів. Для цього на етапі розробки дизайну спочатку створюють прототип—чорно-білий макет, спрощену схему інтерфейсу програми. Він містить всі головні елементи, які представлені у вигляді блоків, які допоможуть оцінити основу концепт. Після створення прототипу, виявляється більшість деталей із раніше не помічених потреб, а також заповнюються функціональні прогалини.

3.1 Аналіз потреб щодо реалізації функціоналу програми

Для визначення необхідних для структури програми й макету інтерфейсу елементів виводу інформації та керування, зробимо повній опис варіантів використання (табл.3.1).

Таблиця 3.1 – Опис варіантів використання (ВВ) програмного продукту (ПП) :

Назва ВВ	Короткий опис	Посилання на SR	Остаточний опис сценарію ВВ
UC01 Load Result (Загрузка архіву та виведення списку записів результатів	ВВ показує раніше створених розрахунків у формі перегляду програми	SRS-0200	Customer зацікавлені у перегляді даних, котрі зберігаються в БД MyResult.db UC01.1 БД MyResult.sql не повинна бути порожньою та розміщена на сервері БД OracleSQL. Customer запустив виконавчий файл програми "Хвіли". Вона відкриває початкову форму програми "Архів розрахунків". UC01.2 Програма завантажила файл бази даних, зчитує із БД MyResult.db дані про записи розрахунків та показує у табличній формі у блоку "записи розрахунків", а в блоки результатів розрахунку завантажуються параметри останнього запису. UC01.3 виконує потік UC02.2 UC01.4 Якщо вхідні списки розрахунків не можуть бути виведені, користувач отримує повідомлення про помилку. UC01.5 Відкрита початкова форма програми "Архів розрахунків", де блок "записи розрахунків" заповнений даними із БД MyResult.db у формі списку, а в блоках щодо результатів розрахунку виведено параметри останнього в списку розрахунку.

<p>UC02 Read Result (перегляд результату)</p>	<p>BB дозволяє користувач у переглядати інформацію про інші розрахунки</p>	<p>SRS-0300</p>	<p>Customer зацікавлені у перегляді параметрів записів, котрі уже оброблені.</p> <p>UC02.1 Програма працює. UC01 виконаний. Відкрита форма програми "Архів розрахунків".</p> <p>UC02.2 Користувач у списку розрахунків у блоку «записи розрахунків» форми програми "Архів розрахунки" обирає рядок конкретного розрахунку, якій потрібно вивести в блоках результатів розрахунку, а в блоки з результатами розрахунку завантажуються параметри цього запису.</p> <p>UC02.3 Команда "Новий розрахунок" (субпотік UC03.2-1):</p> <p>UC02.3-1 натискання кнопки "Повторний розрахунок".</p> <p>UC02.4 Якщо обрані дані не можуть бути виведені, користувач повідомляється про це.</p> <p>UC02.5 Список значень параметрів (в блоках результатів розрахунків на формі "Архів розрахунків") містить оновлені дані про відповідні параметри обраного запису.</p>
<p>UC03 Add Result (Створення нового розрахунку та додавання до архіву)</p>	<p>BB дозволяє користувач у провести розрахунок та додати у запис результатів (дані відображаються у програмі та записуються у файл БД MyBook.sql).</p>	<p>SRS-0400, SRS-0500</p>	<p>Customer зацікавлений у проведенні розрахунку та додаванні запису результатів у список результатів БД MyResult.db.</p> <p>UC03.1 Програма працює. UC01 виконаний. Відкрита форма програми "Новий розрахунок". Customer має дані щодо проведення розрахунку. Жодні параметри у списку параметрів на формі "Новий розрахунок" не визначені та вибрані.</p> <p>UC03.2 Customer вносить вхідні дані для розрахунку</p> <p>UC03.2-1 Ці записи (окрім ID) вносяться у поля блоку "параметрів для розрахунків". Customer редагує інформацію та виконує команду "Розрахувати"(субпотік UC03.3-1). Дані про новий розрахунок записуються у кінець БД MyResult.db, найстаріший запис видаляється (альтернативний потік UC03.4-2). Здійснюється безумовний перехід на форму "Архів розрахунків"</p> <p>UC03.3 Команда "Розрахувати":</p> <p>UC03.3-1 натискання кнопки "Розрахувати"</p> <p>UC03.4</p> <p>UC03.4-1 Якщо розрахунок не може бути проведений, користувач повідомляється про це.</p> <p>UC03.4-2 Якщо розрахунок не може бути доданий до БД або видалений, користувач повідомляється про це.</p> <p>UC04.5 Список розрахунків у формі програми "Архів розрахунків" містить оновлені дані у блоку "записи розрахунків", а в блоки результатів розрахунків виведено параметри цього розрахунку.</p>

UC04 Write Data (Запис у Word)	ВВ дозволяє користувач у-клієнту зберегти параметри розрахунку, у файл Word на основі шаблону Шаблон_Результат.dot	SRS-0600	Customer зацікавлений у записі інформації, яка була отримана при розрахунку (UC02 та UC03), у звіт, сформований на основі шаблону Template MS Word (Шаблон_Результат.dot). UC04.1 Програма працює. Відкрита форма програми "Архів розрахунків". UC02 або UC03 виконаний. UC04.2 Customer виконує команду "Зберегти" (субпотік UC04.3-1). Параметри розрахунку із БД MyResult.db записуються у файл формату Word, сформований на основі шаблону Template MS Word (Шаблон_Результат.dot) (альтернативні потоки UC04.4-1 та UC04.4-2). UC04.3 Команда "Зберегти": UC04.3-1 натисканням кнопки "Зберегти". UC04.4 UC04.4-1 Якщо файл для запису параметрів розрахунку не може бути створений, користувач повідомляється про це. UC04.4-2 Якщо файл із записаними параметрів розрахунку не може бути збережений, користувач повідомляється про це. UC04.5 Параметри розрахунку до заданого запису розрахунків збережені у звіті на основі шаблону Template MS Word.
--------------------------------	--	----------	--

3.2 Ескізи інтерфейсів які будуть використовуватись

Виходячи з варіантів використання, їх порядку взаємодії, процесів обробки даних складемо список форм, із зазначенням кількості форм, їх структури та елементів керування, які приймаємо в ході проектування користувацького інтерфейсу (Табл. 3.2).

Таблиця 3.2 Структури та елементи керування

ВВ	Пункт меню, що виконує ВВ	Кнопка, яка ініціює ВВ
UC01 Load File (Загрузка архіву та виведення списку записів результатів)	<i>Архів записів</i>	Відсутня
UC02 Read Result (перегляд результату)	<i>Архів записів</i>	Покажчик рядка запису
UC03 Add Result (Створення нового розрахунку та додавання до архіву)	<i>Архів записів</i>	«Повторний розрахунок»
	<i>Новий Розрахунок</i>	«Розрахувати»
UC04 Write Data (Запис у Word)	<i>Архів записів</i>	«Зберегти у форматі Word»

Щоб побудувати простий користувацький інтерфейс відмовимось від створення лінійної або покрокової навігації. Нам потрібно два пункту меню з власними формами, в якому будуть підпункти меню в вигляді кнопок в середині

форми. Прототип загального інтерфейсу (User Interface, UI) програми показано на Рис. 3.1. Елементи користувацького інтерфейсу в формах визначені в таб.3.3.

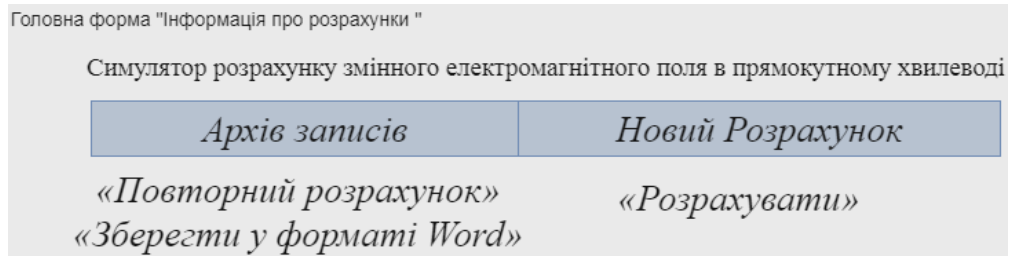


Рисунок 3.1 — Прототип загального інтерфейсу меню програми

Таблиця 3.3 – Елементи користувацького інтерфейсу

Змінні та елементи користувацького інтерфейсу	Примітки
форма програми "Архів записів" InfoForm	містить: <ul style="list-style-type: none"> - назву меню; - компонент для відображення списку записів; - група (блок1) для виводу значення вхідних параметрів проведеного розрахунку ; - група (блок3) вихідних значень проведеного розрахунку ; - група (блок4) вихідних графіків проведеного розрахунку ; - кнопку «Повторний розрахунок»; - кнопку «Зберегти у форматі Word».
форма програми "Новий Розрахунок" ResultForm	містить: <ul style="list-style-type: none"> - назву меню; - група (блок1) для вводу значення вхідних параметрів нового розрахунку ; - група (блок2) відбір вихідних значень для проведення розрахунків ; - кнопку «Розрахувати».
кнопка RepeatResultBtn форми InfoForm	для переходу до форми "Новий Розрахунок" та завантаження до блоку1 значення вхідних параметрів з запису, на якій вказував покажчик у компоненті для відображення списку записів.
кнопка saveResultBtn форми InfoForm	для збереження результату одного розрахунку у форматі MS Word.
кнопка NewResultBtn форми ResultForm	для проведення розрахунку вихідних значень і графіків, видалення найстарішого запису та збереження результату розрахунків у БД.

Прототипи форми "Архів записів" та форми "Новий Розрахунок" з елементами користувацького інтерфейсу наведені на Рис. 3.2. та Рис. 3.3.

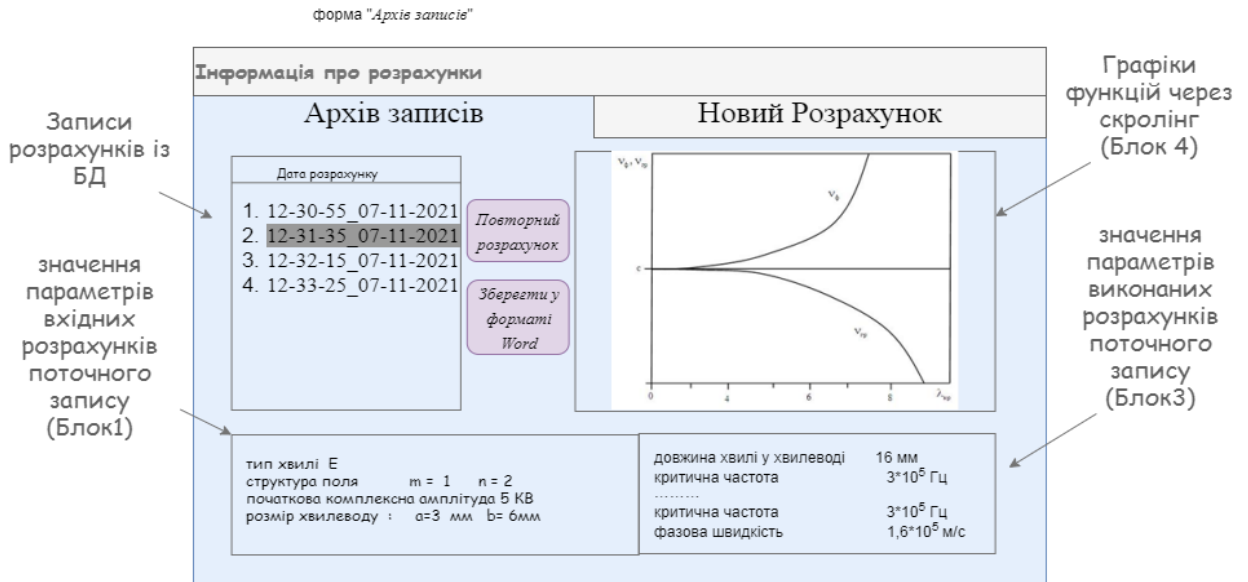


Рисунок 3.2 — Прототип інтерфейсу форми "Архів записів"

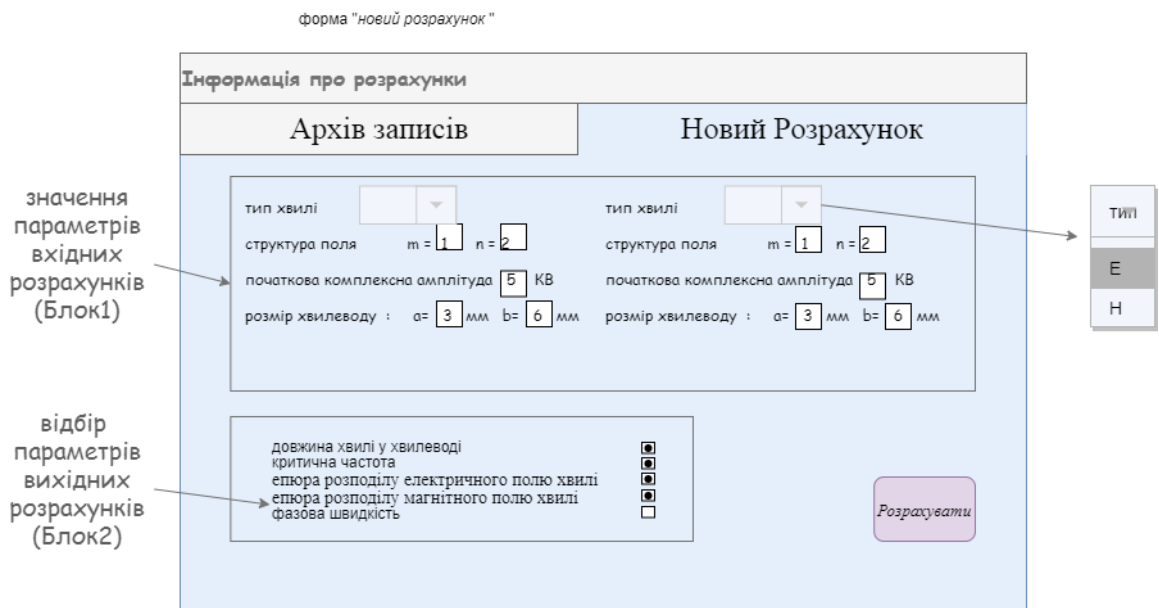


Рисунок 3.3 — Прототип інтерфейсу форми "Новий Розрахунок"

3.3 Шаблон Template MS Word

Для використання результатів розрахунків поза програмного продукту (ПП) зручно створити документ у форматі програми MS Word. Для досягнення однаковості вихідних документів застосуємо шаблони, які визначають основну структуру документа MS Word і містять параметри документа: шрифти, автотекст, макроси, параметри сторінки, форматування, стилі і т.д. (Рис. 3.4).

Нам необхідно розмістити на листі (розмір А4) наступні вихідні дані :

- вхідні значення параметрів для розрахунків (блок 1)
- числові значення параметрів, які розрахованих нашим ПП (блок 3);
- графіки (до 8 епюр), які побудовані нашим ПП (блок 4).

Віртуальний симулятор

структури змінних електромагнітних полів в хвилеводі

запис № 12-12-45_10-11-2021

Вхідні значення параметрів для розрахунків

тип хвилі	E(H)		
початкова комплексна амплітуда E_0	5		КВольт
структура поля в поперечних площинах хвилеводу	m = 1	n = 2	
розмір хвилеводу (a x b)	a = 3	b = 5	мм
довжина хвилі в необмеженому просторі λ	4,2		мм
діелектрична проникність середовища в середині хвилеводу ϵ_n (при $\mu_n=1$)	2		

Розраховані значення

довжина хвилі у хвилеводі λ_c	5,6	мм
еквівалентний опір хвилеводу для E(H)-хвилі Z_{eq}	377	Ом
критична довжина хвилі $\lambda_{кр.}$	13,6	мм
швидкості електромагнітної хвилі в необмеженому середовищі з відповідними параметрами діелектричного заповнення хвилеводу v	$4,55 \cdot 10^8$	м/с
фазова швидкість V_{ph}	$6,14 \cdot 10^8$	м/с
групова швидкість V_{gr}	$1,74 \cdot 10^8$	м/с

Епюри розподілу електричного й магнітного полів хвилі E₂₂.

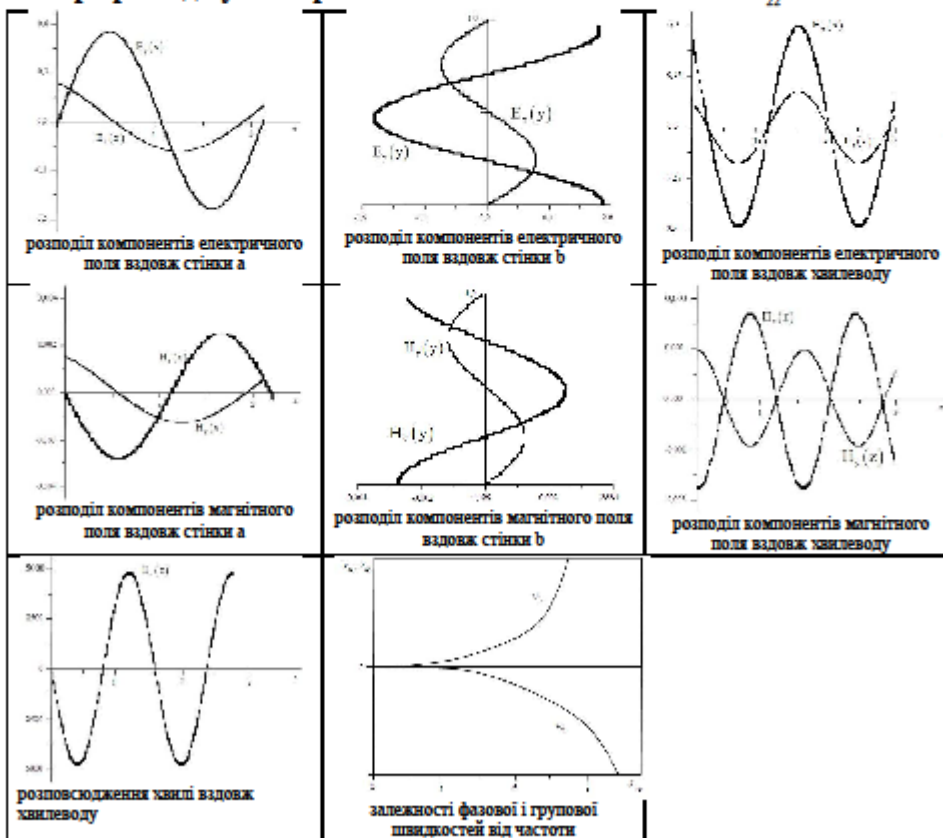


Рис. 3.4 Прототип шаблону вихідного файлу MS Word

Створення прототипу — це один із кроків, який передує запуску фінальної пропозиції. Фактично, створення прототипу вашого майбутнього продукту дає можливість оцінити технічну доцільність. Прототипування робить можливою конкретизацію ідеї і допомагає зрозуміти, які аспекти ідеї буде важко, або взагалі неможливо реалізувати. Створення прототипу може таким чином виявити непередбачені фізичні, технічні або фінансові обмеження. Наявність прототипу робить можливою презентацію нашої майбутньої пропозиції потенційним клієнтам в конкретний спосіб. Наявність прототипу спрощує розуміння завдань кожному члену команди. Це дає нам можливість робити більш реалістичне планування термінів розробки і більш точно визначати бюджет розробки.

4. ТЕОРЕТИЧНІ ЗАСАДИ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

4.1. Математична модель порожніх металевих спрямовуючих систем (хвилеводів)

Моделювання це метод опосередкованого пізнання дійсності з допомогою об'єктів-заступників. Модель у ньому постає як інструмент пізнання, з допомогою якого вивчається цей об'єкт [14]. Перенесення результатів, отриманих у ході побудови та дослідження моделі, на оригінал можливе тому, що модель у певному сенсі відтворює (відображає, описує, імітує) деякі характеристики оригіналу, які цікавлять дослідників.

4.1.1 Постановка задачі моделювання.

Для заданого типу хвилі, що поширюється в прямокутному хвилеводі, одержати характеристики поздовжніх і поперечних компонент полів, миттєві значення також. Розглянемо систему зображену на Рис. 4.1.

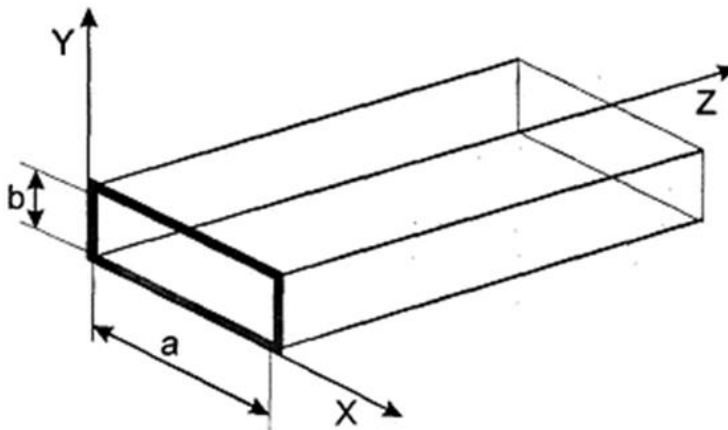


Рисунок 4.1 — Прямокутний хвилевод

Припустимо, що довжина хвилеводу в напрямку осі z нескінченна, поперечні розміри хвилеводу визначаються по осях x і y . Тип хвилі (дві групи хвиль) може бути або E або H -хвилею. Структура поля в поперечних площинах хвилеводу для E і H -хвиль визначається числами m і n . Число m характеризує число стоячих півхвиль по осі x , а число n – число стоячих півхвиль по осі y .

Розмір хвилеводу визначається числами a і b . Процеси поширення електромагнітних хвиль у порожнині металевих спрямовуючих систем (хвилеводів) будемо розглядати, вважаючи, що їх стінки виконані з надпровідного матеріалу (провідність $\gamma \rightarrow \infty$). При цій умові напруженість електричного поля на стінках системи дорівнюватиме нулю. Порожнина системи може бути заповнена діелектриком без втрат з абсолютною діелектричною проникністю ϵ_a . В усіх випадках вважаємо, що параметр $\mu = 1$. Для моделювання також задається довжина хвилі в необмеженому просторі з початковою амплітудою поля $E_m = 5$ кВ/см.

Для відповіді на питання завдання побудуємо математичну модель цього фізичного процесу.

Електромагнітні коливання в хвилеспрямовуючій системі описуються рішенням рівнянь Максвелла, що задовольняє граничну умову, яка ставиться на металевій поверхні, що обмежує систему. Найпростішою моделлю є ситуація, коли метал є ідеальним провідником, і на поверхні хвилеводу дотична складова вектора електричної напруженості звертається в нуль, т. ч. в математичній моделі на межі області ставиться умова Діріхле:

$$\begin{cases} \operatorname{rot} \mathbf{H} = -ik\mathbf{E}, \\ \operatorname{rot} \mathbf{E} = ik\mathbf{H}, \\ [\mathbf{n} \times \mathbf{E}]|_S = 0, \end{cases}$$

де S -бокова поверхня хвилеводу.

Будь-яке поле в регулярному хвилеводі в області, вільної від зовнішніх струмів і зарядів, може бути представлене у вигляді суперпозиції поперечно-магнітних (ТМ) та поперечно-електричних (ТЕ) хвиль [11]. У разі хвилеводу прямокутного перерізу $S \in (0, a) \times (0, b)$ базисні функції, за якими розкладаються поперечні компоненти електромагнітного поля мають вигляд:

(е) - електричний тип ТМ:

$$\begin{cases} \Delta_{\perp} f_{nm}^{(e)} + \kappa_{nm}^2 f_{nm}^{(e)} = 0, & (x, y) \in (0, a) \times (0, b), \\ f_{nm}^{(e)} \Big|_{x=0;a} = f_{nm}^{(e)} \Big|_{y=0;b} = 0, \end{cases} \quad (1)$$

$$\kappa_{nm} = \sqrt{\left(\frac{\pi n}{a}\right)^2 + \left(\frac{\pi m}{b}\right)^2}, \quad n = \overline{1, N}, m = \overline{1, M},$$

$$f_{nm}^{(e)} = \frac{2}{\sqrt{ab}} \sin\left(\frac{\pi n}{a}x\right) \sin\left(\frac{\pi m}{b}y\right),$$

$$\mathbf{G}_{n,m}^{(e2)} = \frac{\nabla f_{nm}^{(e)}}{\kappa_{nm}}, \quad \mathbf{G}_{nm}^{(h3)} = \left[\mathbf{z}_0 \times \mathbf{G}_{nm}^{(e2)} \right],$$

(h) - магнітний тип ТЕ:

$$\begin{cases} \Delta_{\perp} f_{nm}^{(h)} + \kappa_{nm}^2 f_{nm}^{(h)} = 0, & (x, y) \in (0, a) \times (0, b), \\ \frac{\partial f_{nm}^{(h)}}{\partial x} \Big|_{x=0;a} = \frac{\partial f_{nm}^{(h)}}{\partial y} \Big|_{y=0;b} = 0, \end{cases} \quad (2)$$

$$\kappa_{nm} = \sqrt{\left(\frac{\pi n}{a}\right)^2 + \left(\frac{\pi m}{b}\right)^2},$$

$$n = \overline{n_0, N}, m = \overline{m_0, M}, n_0 + m_0 = 1,$$

$$f_{nm}^{(h)} = \frac{2}{\sqrt{ab}} \cos\left(\frac{\pi n}{a}x\right) \cos\left(\frac{\pi m}{b}y\right),$$

$$\mathbf{G}_{n,m}^{(h2)} = \frac{\nabla f_{nm}^{(h)}}{\kappa_{nm}}, \quad \mathbf{G}_{nm}^{(e3)} = \left[\mathbf{G}_{nm}^{(h2)} \mathbf{z}_0 \right].$$

рівняння (1) або (2) – це математична модель гармонійних коливань хвилі, яка поширюється у хвилеводі.

4.1.2 Теоретическое исследование математической модели

Електромагнітне поле в системі описується однорідними хвильовими рівняннями Гельмгольца:

$$\begin{aligned}\nabla^2 \dot{H} + k^2 \dot{H} &= 0, \\ \nabla^2 \dot{E} + k^2 \dot{E} &= 0,\end{aligned}\quad (3)$$

де \dot{H} і \dot{E} – комплексні амплітуди полів;

$k = 2\pi/\lambda = \omega\sqrt{\epsilon_a\mu_a}$ – хвильове число;

λ – довжина хвилі в необмеженому просторі;

ϵ_a і μ_a – відповідно абсолютні електрична й магнітна проникності;

ω – кругова частота.

Хвильові рівняння (2) еквівалентні шести скалярним рівнянням, що для хвилеводів без втрат мають такий загальний вигляд: $\nabla^2 \dot{F} + k^2 \dot{F} = 0$. (4)

Тут під $\dot{F}(x, y, z, t)$ необхідно розуміти будь-яку з проекцій вектора електричного або магнітного поля на координатній осі. Часова залежність $\dot{F}(t)$ вже визначена використанням методом комплексних амплітуд – $\dot{F}(x, y, z, t) = \dot{F}_m(x, y, z)e^{j\omega t}$. Поширювані у хвилеводі хвилі є суперпозицією біжучої (падаючої і відбитої) хвиль вздовж осі z і стоячих хвиль в напрямках x, y , що утворюються внаслідок багаторазових відбиттів від стінок хвилеводу:

$$\dot{F}(x, y, z, t) = \dot{F}^+(x, y)e^{j(\omega t - k_p z)} + \dot{F}^-(x, y)e^{j(\omega t + k_p z)}, \quad (5)$$

де $k_p = \frac{2\pi}{\Lambda}$ – поздовжній коефіцієнт поширення або поздовжнє хвильове число;

Λ – довжина хвилі в хвилеводі, яка в загальному випадку не дорівнює довжині хвилі λ в необмеженому просторі.

Функції $\dot{F}^\pm(x, y)$ характеризують розподіл полів у поперечній площині хвилеводу і відіграють роль амплітуд падаючої і відбитої хвиль, ці функції подібні і відрізняються лише сталим множником, оскільки задовольняють одне

диференціальне рівняння (4). Оскільки при постановці задачі ми припустили, що довжина хвилеводу в напрямку осі z нескінченна, то для визначення структури полів достатньо обмежитися першим доданком у (5). Підставляючи його у (3), із урахуванням того, що $\frac{\partial^2}{\partial z^2} = -k_p^2$, маємо:

$$\nabla_{x,y}^2 \dot{F}^+(x,y) + (k^2 - k_p^2) \dot{F}^+(x,y) = 0. \quad (6)$$

Тут $\nabla_{x,y}^2$ – оператор Лапласа, що вміщує в собі похідні лише за поперечними координатами. Проаналізуємо різницю $(k^2 - k_p^2)$, прирівнявши її до виразу $(2\pi/\lambda_{\text{ед}})^2$, що за формою збігається з виразами хвильових чисел через

відповідні довжини хвиль:
$$k^2 - k_p^2 = \left(\frac{2\pi}{\lambda}\right)^2 - \left(\frac{2\pi}{\Lambda}\right)^2 = \left(2\pi/\lambda_{\text{ед}}\right)^2.$$

Скоротивши всі члени цієї рівності на $(2\pi)^2$, запишемо його відносно довжини хвилі в хвилеводі:

$$\Lambda = \frac{\lambda}{\sqrt{1 - (\lambda/\lambda_{\text{ед}})^2}}. \quad (7)$$

Оскільки фазова швидкість поширення хвилі в хвилеводі пов'язана з довжиною хвилі співвідношенням $v_0 = \Lambda \cdot f$, аналогічним $c = \lambda \cdot f$ (c – швидкість світла; f – частота), то для повітряного заповнення хвилеводу ($\varepsilon = \mu = 1$) маємо швидкість поширення хвилі світла в хвилеводі:

$$v_c = \frac{c}{\sqrt{1 - (\lambda/\lambda_{\text{ед}})^2}} \quad (8)$$

Критична довжина хвилі $\lambda_{\text{ед}}$, це деяке граничне значення довжини хвилі у вільному просторі, що розділяє діапазон нормального поширення електромагнітних полів у вигляді хвиль і так званий діапазон відсічення (поширення хвиль із експоненціальним загасанням).

При $k^2 - k_p^2 = 0$ із (5) випливає рівняння: $\nabla_{x,y}^2 \dot{F}^+ = 0$, що свідчить про відсутність дисперсії ($v_0 = c$, а $\Lambda = \lambda$). Таке поширення хвиль можливе лише в хвилеводах, які допускають існування статичних полів, що містять як мінімум два провідники: коаксіальний хвилевід, полоскова лінія і т. д.

З'ясуємо три можливі ситуації (табл.4.1):

Таблиця 4.1 Варіанти

№ пп.	Складники поля		Тип хвилі		Назва
	E_z	H_z	Позначення	Попередня та англomовна література	
1	$E_z = 0$	$H_z = 0$	T	TEM	Поперечна
2	$E_z \neq 0$	$H_z = 0$	E	TM	Поздовжня електрична (поперечна магнітна)
3	$E_z = 0$	$H_z \neq 0$	H	TE	Поздовжня магнітна (поперечна електрична)

T-хвилі характеризуються відсутністю дисперсії та обох поздовжніх складових електричного і магнітного полів ($E_z = H_z = 0$). Опір хвилеводу для цих хвиль дорівнює хвильовому опору в необмеженому середовищі:

$$Z_c^0 = Z_c = \sqrt{\frac{\mu_a}{\epsilon_a}}.$$

H-хвилі характеризуються відсутністю поздовжньої складової електричного поля ($E_z = 0$). Однією із особливостей таких хвиль є те, що відношення поперечних складових електричного і магнітного полів не залежить від координат. Це співвідношення називається еквівалентним опором хвилеводу, причому:

$$Z_c^H = \frac{|\dot{E}_x|}{|\dot{H}_y|} = \frac{|\dot{E}_y|}{|\dot{H}_x|} = \frac{\omega \mu_a}{k_p} = Z_c \frac{\Lambda}{\lambda} = \frac{Z_c}{\sqrt{1 - (f_{\text{сд}}/f)^2}}. \quad (9)$$

$Z_c = 377 \sqrt{\mu/\epsilon}$ – хвильовий опір необмеженого середовища;

$f_{\text{сд}}$ – критична частота.

E-хвилі характеризуються відсутністю поздовжньої складової магнітного поля ($H_z = 0$). Еквівалентний опір хвилеводу для цих хвиль:

$$Z_c^E = \frac{k_p}{\omega \epsilon_a} = Z_c \frac{\lambda}{\Lambda} = Z_c \sqrt{1 - (f_{\text{сд}}/f)^2}. \quad (10)$$

Для інтегруванні рівнянь Гельмгольца є метод розділення змінних, суть якого полягає в тому, що шукану функцію подають як добуток функцій, кожна з яких залежить лише від однієї координати [12;13].

При визначенні структури полів у хвилеводах вихідними є рівняння (1) або рівняння (2). Даному випадку для сталих поділу виконується рівність $k_x^2 - k_y^2 = k^2 - k_p^2$. Тому сталі поділу k_x і k_y (за аналогією з k і k_p) іноді називають поперечними хвильовими числами, незважаючи на те, що в поперечному напрямку поширення хвиль не відбувається.

Розв'язок рівнянь (1), (5) може бути подано в такому вигляді:

$$\dot{F}(x, y, z) = \dot{F}^+(x, y)e^{-jk_p z}, \quad \dot{F}^+(x, y) = X(x)Y(y), \quad \text{де } X(x) = C_x \frac{\sin}{\cos}(k_x x + \varphi_x), \quad Y(y) = C_y \frac{\sin}{\cos}(k_y y + \varphi_y).$$

Подвійний запис означає, що, виходячи зі зручності, можна вибирати будь-яку із функцій, поки сталі інтегрування φ_x і φ_y не визначені. Виходячи із вищесказаного, загальний вигляд розв'язку хвильових рівнянь для будь-якої зі складових електричного і магнітного полів може бути поданий у такій формі:

$$\dot{F}^+(x, y, z, t) = \dot{F}^+(x, y)e^{j(\omega t - k_p z)} = C_x \frac{\sin}{\cos}(k_x x + \varphi_x) \frac{\sin}{\cos}(k_y y + \varphi_y) e^{j(\omega t - k_p z)}.$$

Це рівняння показує, що гармонійні зміни напруженості аналогічні полям стоячої хвилі в поперечному перерізі, а у поздовжньому напрямку в хвилеводі існує хвильовий процес, що характеризується поздовжнім хвильовим числом k_p . Сталі інтегрування k_x , k_y , φ_x і φ_y можуть бути знайдені із граничних умов, які для прямокутного хвилеводу нескінченної довжини без втрат у стінках і в заповнювальному його діелектрику формулюються таким чином:

$$E_y = E_z = 0 \quad \text{при } x = 0, x = a \quad (\text{розмір широкої стінки хвилеводу по осі } x),$$

$$E_x = E_z = 0 \quad \text{при } y = 0, y = b \quad (\text{розмір висоти хвилеводу по осі } y).$$

Щоб не визначати сталі інтегрування для кожної складової поля, як правило користуються таким прийомом: рішення для поздовжньої складової поля записують у формі (11):

$$\begin{aligned} E_z &= j\dot{E}_0 \sin(k_x x + \varphi_x) \sin(k_y y + \varphi_y), \\ H_z &= j\dot{H}_0 \sin(k_x x + \varphi_x) \sin(k_y y + \varphi_y), \end{aligned} \quad (12)$$

де \dot{E}_0 і \dot{H}_0 – комплексні амплітуди полів, які визначаються початковими умовами (хвильовий множник у цьому записі опущений), а поперечні компоненти знаходяться з перших двох рівнянь Максвелла для гармонійних процесів у проєкціях на осі координат шляхом вираження з них поперечних компонент полів і підстановки в них значень (12). При цьому можна обійтися без формулювання граничних умов для магнітного поля. Аналітичні вирази для складового поля хвилі одержуємо з (12):

$$\begin{aligned} \dot{E}_x &= \dot{E}_0 k_p k_x \cos(k_x x + \varphi_x) \sin(k_y y + \varphi_y) / (k^2 - k_p^2), \\ \dot{E}_y &= \dot{E}_0 k_p k_y \sin(k_x x + \varphi_x) \cos(k_y y + \varphi_y) / (k^2 - k_p^2), \\ \dot{H}_x &= -\dot{E}_0 k_y \omega \varepsilon_a \sin(k_x x + \varphi_x) \cos(k_y y + \varphi_y) / (k^2 - k_p^2), \\ \dot{H}_y &= \dot{E}_0 k_x \omega \varepsilon_a \cos(k_x x + \varphi_x) \sin(k_y y + \varphi_y) / (k^2 - k_p^2), \end{aligned} \quad (13)$$

Відповідно до граничних умов на стінках хвилеводу $\dot{E}_y = 0$ при $x=0$ й $x=a$ одержуємо $\varphi_x = 0$ і $k_x = m\pi/a$, де $m=0, 1, 2, \dots$. Беручи, що $\dot{E}_x = 0$ також на широких стінках хвилеводу ($y=0, y=b$), одержуємо $\varphi_y = 0$ й $k_y = n\pi/b$, де $n = 0, 1, 2, \dots$

Остаточні вирази для складових поля після підстановки сталих набирають вигляду:

$$\dot{E}_x = \dot{E}_0 \frac{k_x k_p}{k_x^2 + k_y^2} \cos \frac{m\pi}{a} x \sin \frac{n\pi}{b} y, \quad \dot{E}_y = \dot{E}_0 \frac{k_x k_p}{k_x^2 + k_y^2} \sin \frac{m\pi}{a} x \cos \frac{n\pi}{b} y, \quad (14)$$

$$\dot{H}_x = -\frac{\dot{E}_0}{Z_a^{EH}} \frac{k_y k_p}{k_x^2 + k_y^2} \sin \frac{m\pi}{a} x \cos \frac{n\pi}{b} y, \quad \dot{H}_y = \frac{\dot{E}_0}{Z_a^{EH}} \frac{k_x k_p}{k_x^2 + k_y^2} \cos \frac{m\pi}{a} x \sin \frac{n\pi}{b} y,$$

$$\dot{E}_z = j\dot{E}_0 \sin \frac{m\pi}{a} x \sin \frac{n\pi}{b} y, \quad \dot{H}_z = j\dot{E}_0 \cos \frac{m\pi}{a} x \cos \frac{n\pi}{b} y \quad (15)$$

Для відновлення дійсних значень необхідно компоненти полів домножити на опущений раніше хвильовий множник $e^{j(\omega t - k_p z)}$, перейти за формулою Ейлера до тригонометричної форми запису й взяти дійсну частину отриманого виразу. Продемонструємо цю операцію на прикладі E_x компоненти:

$$E_x = \operatorname{Re}(\dot{E}_x e^{j(\omega t - k_p z)}) = \operatorname{Re}\left\{E_0 \frac{k_x k_p}{\Lambda k_x^2 + k_y^2} \times \cos \frac{2\pi}{a} x \sin \frac{2\pi}{b} y \left[\cos(\omega t - k_p z) - \sin(\omega t - k_p z) \right] \right\},$$

Фазова швидкість визначається з (8), а групова в загальному випадку визначається таким співвідношенням: $v_{\text{гд}} = v \sqrt{1 - (\lambda/\lambda_{\text{кд}})^2}$, (16)

де $v = 1/\sqrt{\epsilon_a \mu_a}$ – швидкості електромагнітної хвилі в необмеженому середовищі з параметрами ϵ_a й μ_a відповідними параметрами діелектричного заповнення хвилеводу. $\lambda_{\text{кр}}, v_{\text{ф}}, \Lambda$ і k_p для виразів складових поля дорівнюють:

$$\lambda_{\text{кр}} = \frac{2}{\sqrt{\left(\frac{m}{a}\right)^2 + \left(\frac{n}{b}\right)^2}} \quad v_{\text{ф}} = \frac{c}{\sqrt{\epsilon \mu} \cdot \sqrt{1 - \left(\frac{\lambda}{\lambda_{\text{кр}}}\right)^2}}, \quad \Lambda = \frac{\lambda}{\sqrt{\epsilon \mu - (\lambda/\lambda_{\text{кд}})^2}}, \quad k_p = \frac{2\pi}{\Lambda}$$

Формула (14) – це окреме рішення математичної моделі (1) та (2). Ця формула є характеристикою коливань. Зазначимо, що при побудові цієї моделі використано ряд припущень: наприклад, вважається, що метал є ідеальним провідником, і на поверхні хвилеводу дотична складова вектора електричної напруженості звертається в нуль.

4.2. Послідовність розрахунку складових електромагнітного поля

Методика розрахунку електромагнітного поля у напрямних системах:

1. Визначаємо граничні умови до загальних рішень виразів складових поля нашого хвилеводу $\lambda_{\text{кр}}, Z_c, k_x, k_y$.
2. Відповідно до типу хвилі, знаходимо параметри хвилі у середовищі діелектричного заповнення хвилеводу $\Lambda, k_p, v_{\text{ф}}, v_{\text{гр}}, v$ (із 16), Z_c^H або Z_c^E .
3. Вирішуючи рівняння приватного рішення (14), знаходимо рішення (масив миттєвих значення вздовж стінок хвилеводу) для функції

розподілу поздовжніх складових електричного (E_x, E_y) та магнітного (H_x, H_y) полів. Також знаходимо E_z або H_z (масив миттєвих значення вздовж хвилеводу відповідно до типу хвилі) (15).

4. Перейти за формулою Ейлера до тригонометричної форми запису й взяти дійсну частину отриманого виразу, для чого необхідно компоненти полів домножити на опущений раніше хвильовий множник $e^{j(\omega t - k_p z)}$ ($[\cos(\omega t - k_p z) - \sin(\omega t - k_p z)]$).
5. Побудувати графічні залежності компонентів поля від координат хвилеводу за допомогою масивів миттєвих значень.
6. Створити масив проміжних значень v_f, v_{gr} для різної довжини хвиль. Побудувати графічні залежності фазової і групової швидкостей від частоти (довжини хвилі) вказавши v та $\lambda_{кр}$.

Математична модель, будучи абстрактним образом модельованого об'єкта або процесу, не може бути його повним аналогом. Досить подібності в тих елементах, які визначають мету дослідження. Формальних прийомів, що дозволяють автоматично, "бездумно", створювати адекватні математичні моделі, немає. Остаточне судження про адекватність моделі дає практика, тобто зіставлення моделі з чинним об'єктом.

5. РОЗРОБЛЕННЯ І ІМПЛІМЕНТАЦІЯ

5.1 Архітектура програмного продукту

За допомогою дворівневої архітектури, в якій клієнт взаємодіє із сервером безпосередньо, будемо створювати програму, в якій Java код взаємодіє безпосередньо з базою даних SQL з використанням API зв'язку з базами даних Java. (Рис.5.1)

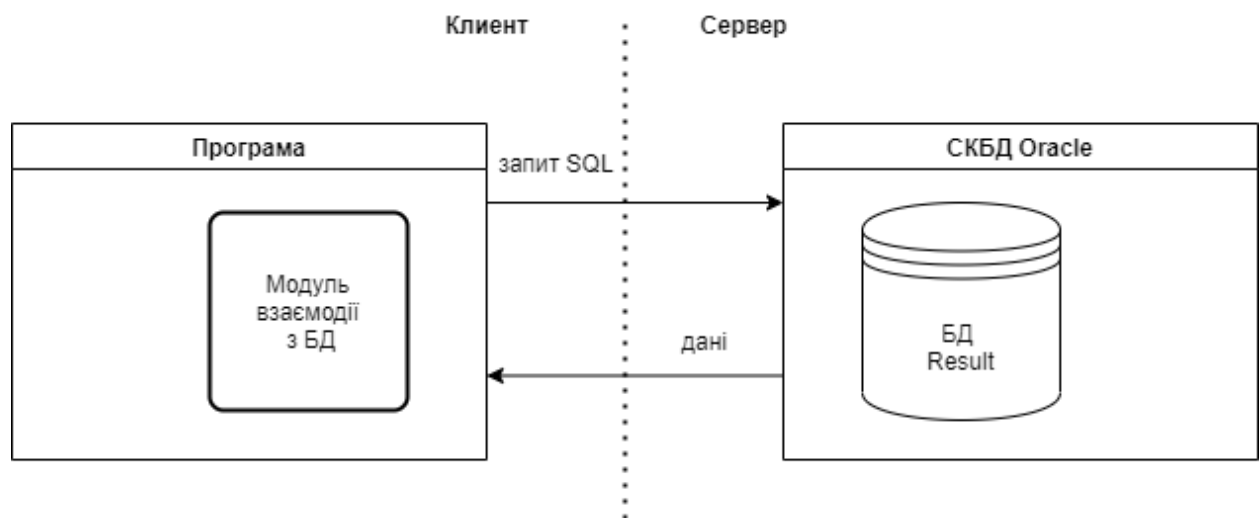


Рисунок 5.1 — Дволанкова клієнт-серверна архітектура програми

Враховуючи модель варіантів використання, а саме Опис варіантів використання та схеми взаємодії сценаріїв системи (табл.3.1) визначимо архітектуру ПП (Рис.5.2), а також модулі і функції необхідні при розробці ПП.

Так як розділення на модулі умовне, поділимо на модулі відповідно наших завдань.

Основні модулі та функції

1. модуль БД для завантаження, перегляду та запису результатів до БД;
2. функція з'єднання з БД;
3. функція отримання даних з бази (запити на окремі записи)
4. функція додавання записів;
5. функція видалення записів;
6. модуль користувача, щодо навігації та обробки подій;
7. функція перегляду списку усіх записів;
8. функція виводу значень поточного запису;
9. функція виходу з сторінки (панелі керування) користувача
10. модуль розрахунку
11. функції послідовного розрахунку значень параметрів

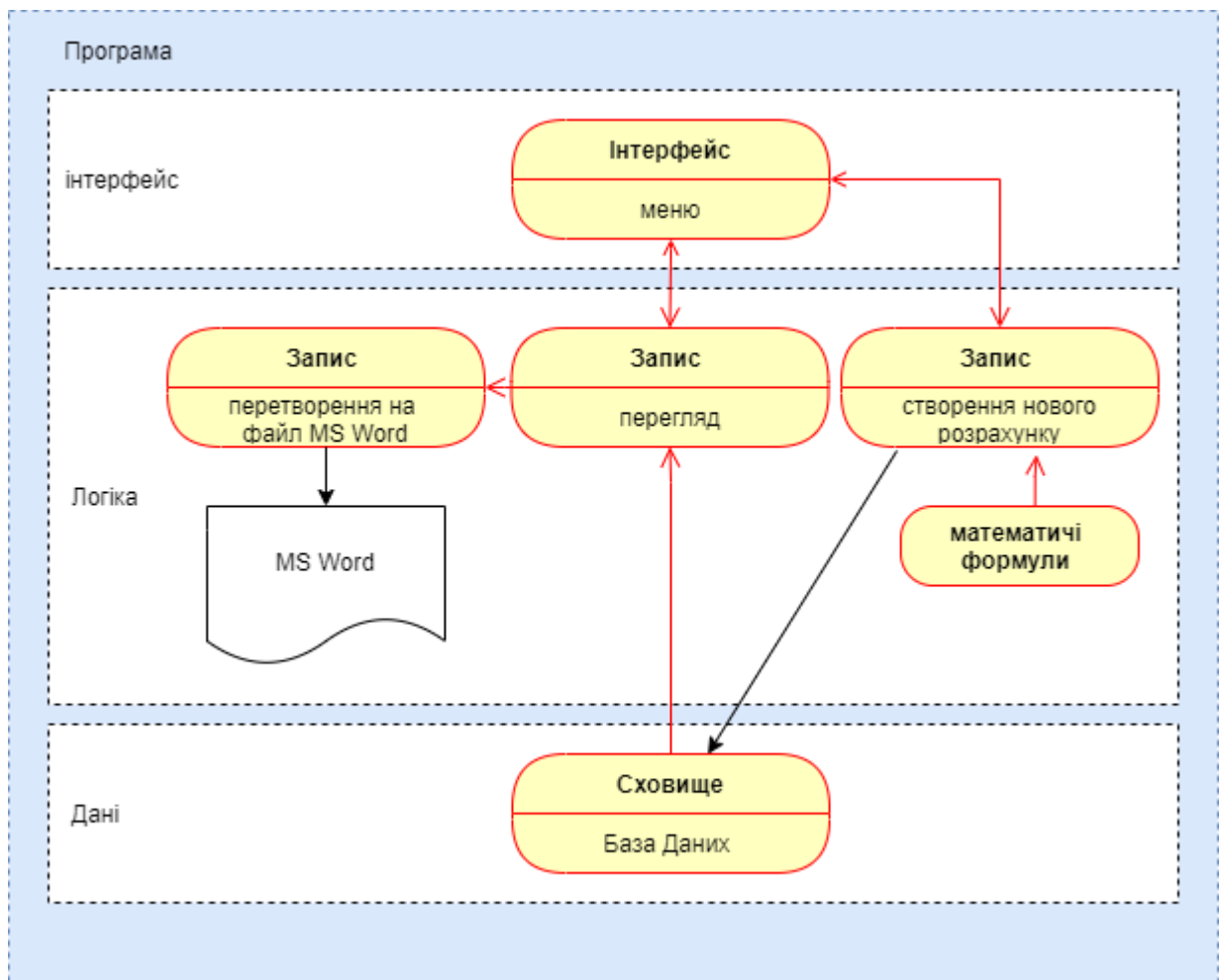


Рисунок 5.2 — Трирівнева архітектура програми

Ми будемо використовувати реляційну модель баз даних [15], яка являє собою двовимірний масив та складає одну таблицю. Кожний рядок такої таблиці називається записом, кожний стовпець — полем. Опис структури таблиць бази даних (внутрішня схема даних) вже було показано в табл. 2.2

Ця модель бази даних дозволяє виконувати такі дії:

- сортування даних за датою;
- вибірка даних окремого запису симуляції за датою ;

5.2 Проектування програми

База даних і взаємозв'язки між елементами архітектури є основною моделлю програми. Проектування програми. Будь-яку програму на Java можна як набір взаємодіючих між собою об'єктів.

Застосовуючи прийоми об'єктно-орієнтованого проектування [16], в процесі проектування було виділено такі об'єкти :

1. Програма (Application) , як об'єкт з життєвим циклом
2. Інтерфейс (Interface), як об'єкти форми представлення
3. Запис (Record), як об'єкт даних результату розрахунку
4. Сховище (Storage), як об'єкт взаємодії з БД
5. Абстракція «Об'єкт обчислення»

Абстракція «Об'єкт обчислення» нам потрібна, як базова для записів з подальшою побудовою графіків шляхом многократного пере визначення.

Шаблоном або описом об'єкта є клас, а об'єкт є екземпляр цього класу.

Java є об'єктно-орієнтованою мовою, тому такі поняття як "клас" та "об'єкт" відіграють у ньому ключову роль. Сформулюємо основні класи нашої програми з урахуванням потрібних функцій (Рис.5.3):

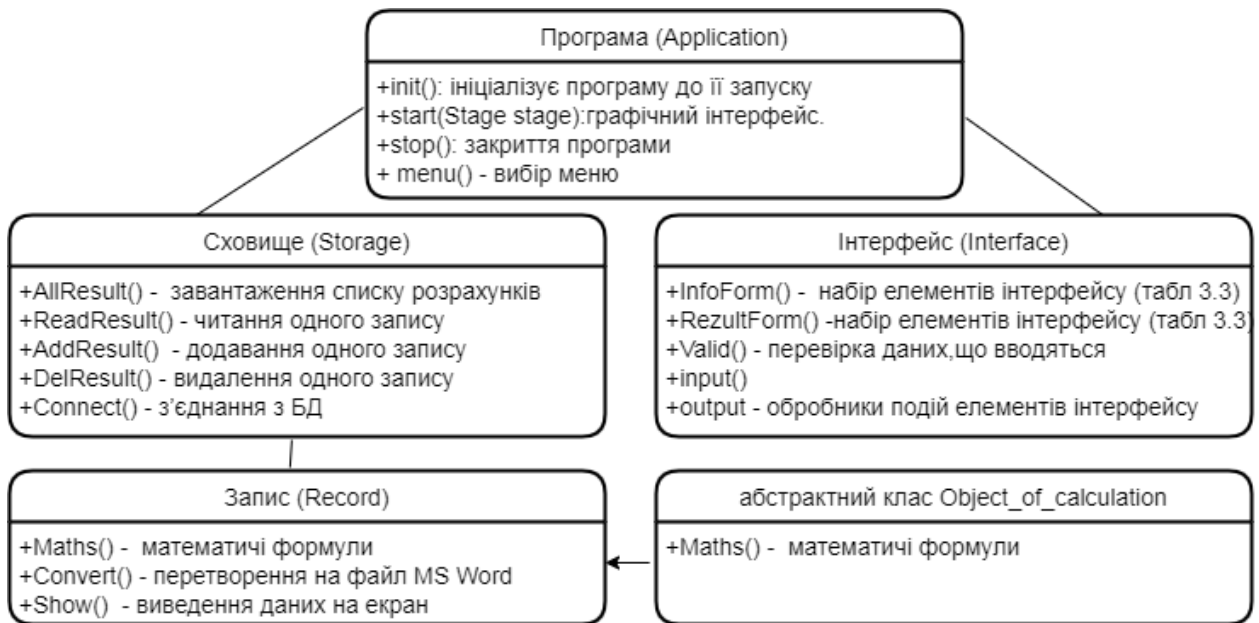


Рисунок 5.3 — Об'єкти Java

Алгоритм рішення задачі наступний:

1. Підключаємо необхідні бібліотеки й формуємо змінні та константи
2. З'єднуємося з БД файл для читання, зчитуємо з нього дані списку розрахунків і зберігаємо їх в масив;
3. Створюємо клас «Запис», похідний від абстрактного класу, додаємо до нього елементи даних з останнього запису БД, виводимо дані в інтерфейс користувача;
4. Створюємо абстрактний клас «Об'єкт обчислення» з елементом даних «Обчислювальний параметр», створюємо клас «Розрахунок», похідний від абстрактного класу, додаємо до нього елементи даних: ε_a , λ , m, n , E_0 ;
5. Розраховуємо граничні умови до загальних рішень виразів складових поля нашого хвилеводу $\lambda_{кр}, Z_c, k_x, k_y$;
6. Розраховуємо складові електромагнітного поля Відповідно до типу хвилі, знаходимо параметри хвилі у середовищі діелектричного заповнення хвилеводу $\Lambda, k_p, v_f, v_{гр}, v, Z_c^H$ або Z_c^E ;

7. Вирішуючи рівняння багаторазово, знаходимо рішення (масив миттєвих значення вздовж стінок хвилеводу) для функції розподілу поздовжніх складових електричного та магнітного полів. Перетворюємо масив на графіки у форматі PNG;

8. Створюємо клас «Новий_Запис», похідний від абстрактного класу, додаємо до нього елементи даних: вхідні , розраховані та графіки у форматі PNG;

9. З'єднуємося з БД файл для запису і зберігаємо їх;

10. Повторюємо процедури зчитування, розрахунку та виведення для інших обраних записів.

5.3 Короткий опис програмної реалізації

Нами буде використовуватися традиційна для Java модель «Модель-Вигляд-Контроллер» (Model-View-Controller, MVC)[19]. Розглянемо загальний підхід до реалізації функціоналу програми. Структура проекту вказана на Рис.5.4.

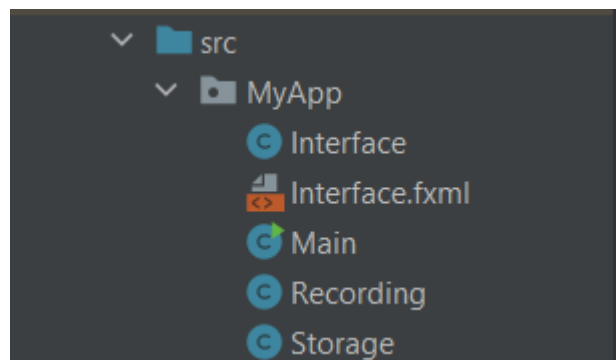


Рисунок 5.4 — Структура папок проекту

Приклади створення класів зазначених на Рис. 5.5-2.11

```
/**
 * @author Максим Горячев
 */
public class Main extends Application {

    public static void main(String[] args) { launch(args); }

    public void start(Stage primaryStage) throws IOException {
        Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("Interface.fxml")));
        Scene scene = new Scene(root);
        primaryStage.setScene(scene);
        primaryStage.setTitle("Simulator_from_MAX");
        InputStream iconStream = getClass().getResourceAsStream("icon.png");
        assert iconStream != null;
        Image image = new Image(iconStream);
        primaryStage.getIcons().add(image);
        primaryStage.show();
    }
}
```

Рисунок 5.5 — Клас Main

При розробці класів доступу до бази даних ми використаємо стандартні інструменти доступу до бази даних мови Java, а саме JDBC API [18].

Готуємо вхідні дані для запиту до бази даних та викликаємо методи класів доступу до бази даних. Для цього сформуємо підключення до бази даних у класі `ConnectToBase()`, де зчитуємо параметри підключення до бази даних та формуємо сам об'єкт підключення. Для формування запиту до бази даних використаємо підготовлені запити, щоб уникнути SQL-ін'єкції. Приклад реалізації класу `Storage`, що здійснює операцію підключення, читування та запису в базу даних наведено нижче Рис.5.6.


```

8      /**
9       * Клас-модель Бази Даних (Recording).
10     * @author Максим Горячев
11     */
12     public class Storage {
13         // private final SimpleIntegerProperty Property_id;
14         // private final int id; //индетификатор запису
15         private final SimpleStringProperty name_data; //дата запису
16         private final ObservableList<Storage> RecordData = FXCollections.observableArra
17         String[] Data = new String[16];
18         //зав
19         public Storage(/*int id,*/String name_data ){
20             // this.Property_id = new SimpleIntegerProperty(id);//индетификатор з
21             this.name_data = new SimpleStringProperty(name_data); //дата запису
22             // this.id = id;
23             // Data[0]=String.valueOf(id);
24             Data[1]=name_data;
25         }
26         public void initialize() {
27             ConnectToBase(); //підключення до БД
28         }

```

Рисунок 5.6 — Клас Storage

Розглянемо приклад (Рис.5.7) реалізації класу Recording, що представляє собою сутність запису результату. Даний клас містить поля, конструктор без параметрів та методи для встановлення й отримання значень для полів об'єкта класу.

```

/**
 * Клас-модель для запису (Recording).
 * @author Максим Горячев
 */
public class Recording {
    private int id; //идентификатор запису
    private String name_data; //дата запису
    private String wave_type; //тип хвилі текстове поле 1 знак
    private int pole_m; //структура поля m цифрове поле 2 знака
    private int pole_n; //структура поля n цифрове поле 2 знака
    private double size_a; //розмір хвилеводу a цифрове поле 2 знака
    private double size_b; //розмір хвилеводу b цифрове поле 2 знака
    private double wavelength_vacuum; //довжина хвилі в необмеженому просторі цифрове поле 2 знака
    private int power_E; //початкова комплексна амплітуда цифрове поле 8 знаків
    private int dielectric_constant; //діелектрична проникність цифрове поле 2 знака
    private double wavelength_dielectric; //Довжина хвилі у діелектричне заповненому хвилеводі цифрове поле 2 знака
    private double Critical_wavelength; //Критична довжина хвилі цифрове поле 2 знака
    private int Equivalent_resistance; //Еквівалентний опір хвилеводу цифрове поле 2 знака
    private int Wave_speed; //Швидкість поширення хвилі світла в хвилеводі цифрове поле 2 знака
    private int Group_speed; //Групова швидкість цифрове поле 2 знака
    private int Phase_speed; //Фазова швидкість цифрове поле 2 знака
}

```

Рисунок 5.7 — Клас Recording

Для виконання основних операцій розрахунку створемо клас Interface та будемо викликати методи цього класу для обробки подій на формі інтерфейсу. Нижче наведено приклад класу Interface Рис. 5.8

```

17  /**
18     * Клас-модель для інтерфейсу (Interface).
19     * @author Максим Горячев
20     */
21     public class Interface implements Initializable {
22         //Змінні розмітки Interface файл FXML
23         public TabPane tabPane;
24         @FXML private TableView<Storage> table; // елемент для виводу списку записів
25
26         @FXML private Button fxButton1;// елемент для переходу на повторний розрахунок
27         @FXML private Button fxButton2;// елемент для виводу в файл MS Word
28         @FXML private Button fxButton3;// елемент для початку розрахунку
29
30         // елемент - метки для виводу даних в вкладку "Архів розрахунків"
31         @FXML private Label lb1 ; @FXML private Label lb2 ; @FXML private Label lb3 ; @FXML private Label lb4 ;
32         @FXML private Label lb5 ; @FXML private Label lb6 ; @FXML private Label lb7 ; @FXML private Label lb8 ;
33         @FXML private Label lb9 ; @FXML private Label lb10 ; @FXML private Label lb11 ;@FXML private Label lb12 ;
34         @FXML private Label lb13 ;@FXML private Label lb14 ; @FXML private Label lb15 ;
35
36         private final Label[] labels =new Label[32];
37
38         // елемент - метки для виводу даних в вкладку "Новий розрахунок"
39         @FXML private ComboBox<String> lb22;// елемент для вибору типу хвили
40
41         // елемент - метки для виводу даних в вкладку "Новий розрахунок"
42         @FXML private ComboBox<String> lb22;// елемент для вибору типу хвили
43
44         // елемент - метки для виводу даних в таблицю записів
45         @FXML private TableColumn<Storage, Integer> idColumn ;
46         @FXML private TableColumn<Storage, String> name_data_Column ;
47         ObservableList<Storage> recData = new Storage( id: 0, name_data: null).LoadData();
48
49         // елемент - записів для виводу даних в поле графіки
50         final NumberAxis yAxis = new NumberAxis(); final NumberAxis xAxis = new NumberAxis();
51         @FXML private LineChart<Double, Double> lineGraph;
52         private double range;
53
54         public void Wave_setType() {// обробник кнопки -тип хвили
55             int index = lb22.getSelectionModel().getSelectedIndex();// отримуємо номер вибраного елемента
56             System.out.println(index); // вивід в консоль для відлагодження
57         }
58         public void click1() { // обробник кнопки -повторний розрахунок
59             tabPane.getSelectionModel().select( index: 1); //перехід на вкладку "Новий розрахунок"
60             System.out.println("Button1 press"); // вивід в консоль для відлагодження
61             fxButton1.setText("Повторити \nрозрахунок");
62         }
63         public void click2() { // обробник кнопки -виводу в файл MS Word
64             System.out.println("Button2 press");// вивід в консоль для відлагодження

```

```

68     fxButton3.setText("Розрахувати");
69 }
70
71 public void plotLine() {
72     XYChart.Series<Double,Double> series = new XYChart.Series<>();
73     XYChart.Series<Double,Double> series1 = new XYChart.Series<>();
74     series.setName("Розподіл компонент Е поля вздовж стінки");series1.setName("Розподіл компонент Н поля вздовж стінки");
75     System.out.println(series);
76     Recording ma0 = new Recording();Recording ma1 = new Recording();
77     for (double x = 0; x <= range; x = x + 0.01) {
78
79         series.getData().add(new XYChart.Data<>(x, ma0.Match(x)));
80         series1.getData().add(new XYChart.Data<>(x, ma1.Match1(x)));
81
82     }
83     lineGraph.getData().addAll(series,series1);
84     System.out.println(lineGraph);
85 }
86 public void clear() { lineGraph.getData().clear(); }
89 public void setData(int nom) {
90     System.out.println(recData.get(nom).Data[1]);
91     String[] arr = new Storage(nom, recData.get(nom).Data[1]).ReadOneDataRecords();
92     for (int i = 1; i < 16; i++) {arr.length
93         // System.out.println(labels[i]);
94         labels[i].setText(String.valueOf(arr[i]));
95
96     }
97
98     @Override
99     public void initialize(URL location, ResourceBundle resources) {
100         //збираємо усі потрібні метки для виводу даних в масив
101         labels[1]= lb1; labels[2]= lb2; labels[3]= lb3; labels[4]= lb4;labels[5]= lb5; labels[6]= lb6; labels[7]= lb7; labels[8]= lb8;
102         labels[9]= lb9; labels[10]= lb10; labels[11]= lb11; labels[12]= lb12;labels[13]= lb13; labels[14]= lb14; labels[15]= lb15;
103         // отримуємо вибраний елемент
104         ObservableList<String> Items = FXCollections.observableArrayList( ...items: "E", "H");
105         lb22.setItems(Items); lb22.setValue("E"); //устанавливаем выбранный элемент по умолчанию
106
107         // устанавливаем тип и значение которое должно храниться в колонке
108         idColumn.setCellValueFactory(new PropertyValueFactory<>("Id"));
109         name_data_Column.setCellValueFactory(new PropertyValueFactory<>("Name"));
110         //заполняем таблицу данными початковими
111         table.setItems(recData); //заповнюємо TableView записами
112         setData(0); //заповнюємо форму Interface даними
113         this.range=10.0; plotLine();
114         // Додаємо слухача, на кліку мишки на запис, запускає оновлення даних + виводить текст на консоль
115         table.addEventHandler(MouseEvent.MOUSE_CLICKED, mouseEvent -> { int pos=table.getSelectionModel().getSelectedIndex();
116             setData(pos); //заповнюємо форму Interface даними новими
117             System.out.println("INDEX:"+pos);
118         });
119     }
120 }

```

Рисунок 5.8 — Клас Interface(Контролер)

Використовуючи фреймворк JavaFX Scene Builder 2.0 створено макет нашої програми Рис.5.9-5.10, відповідно до прототипу дизайну вказаному в розділі 3

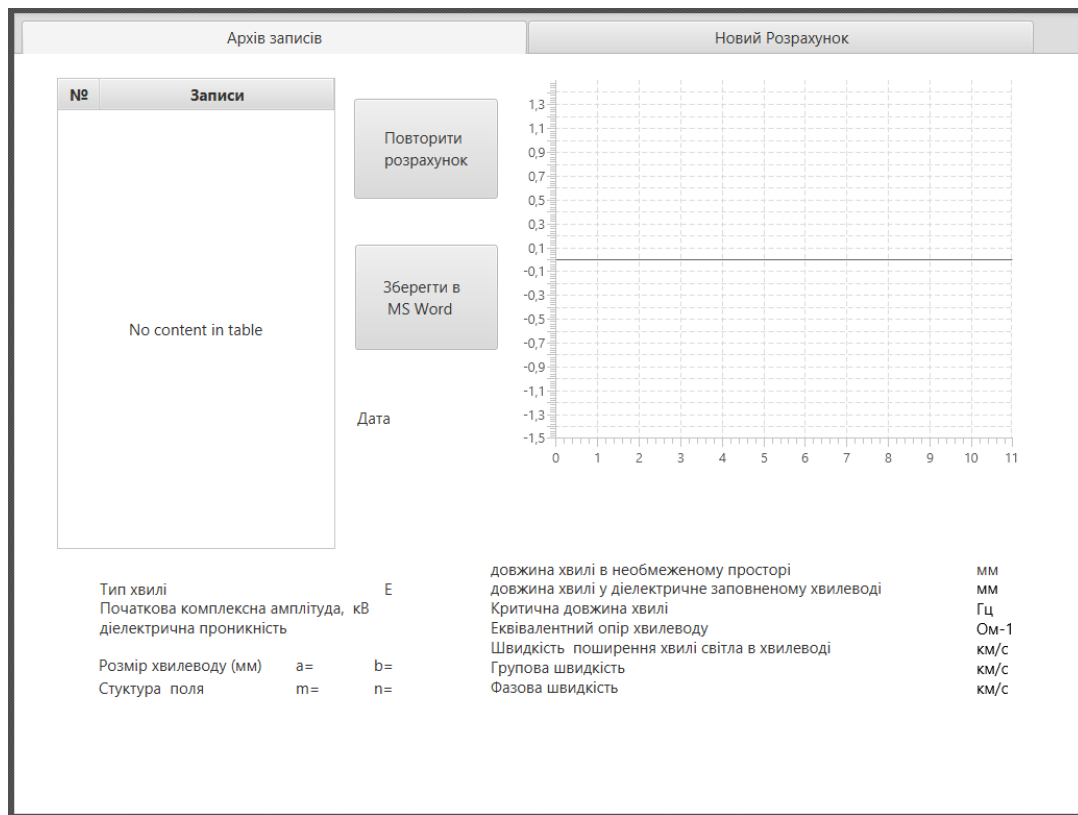


Рисунок 5.9 Готовий дизайн вкладки “Архів записів”\

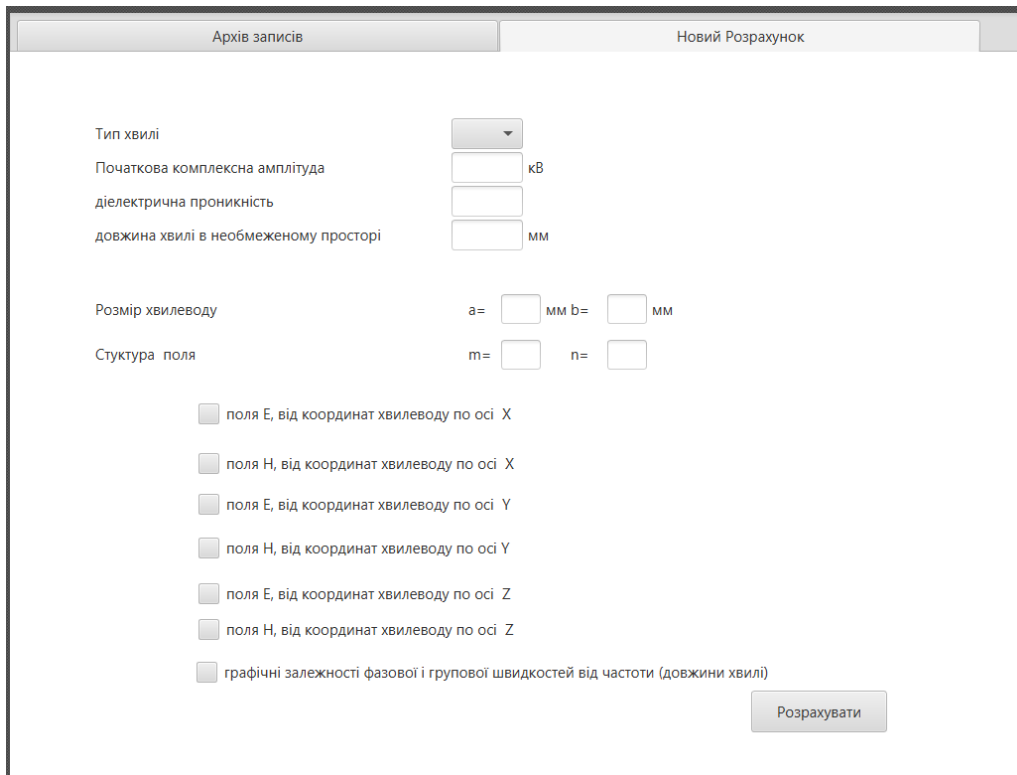


Рисунок 5.10 — Готовий дизайн вкладки “Новий Розрахунок”

Програмний код макету(View) приведено в додатку А. В результаті запуску програми, після відпрацювання процедур завантаження та відображення записів даних отримуємо на Рис. 5.11. При переході на іншу вкладку, спрацьовує процедура відображенні іншої вкладки. Рис. 5.12.

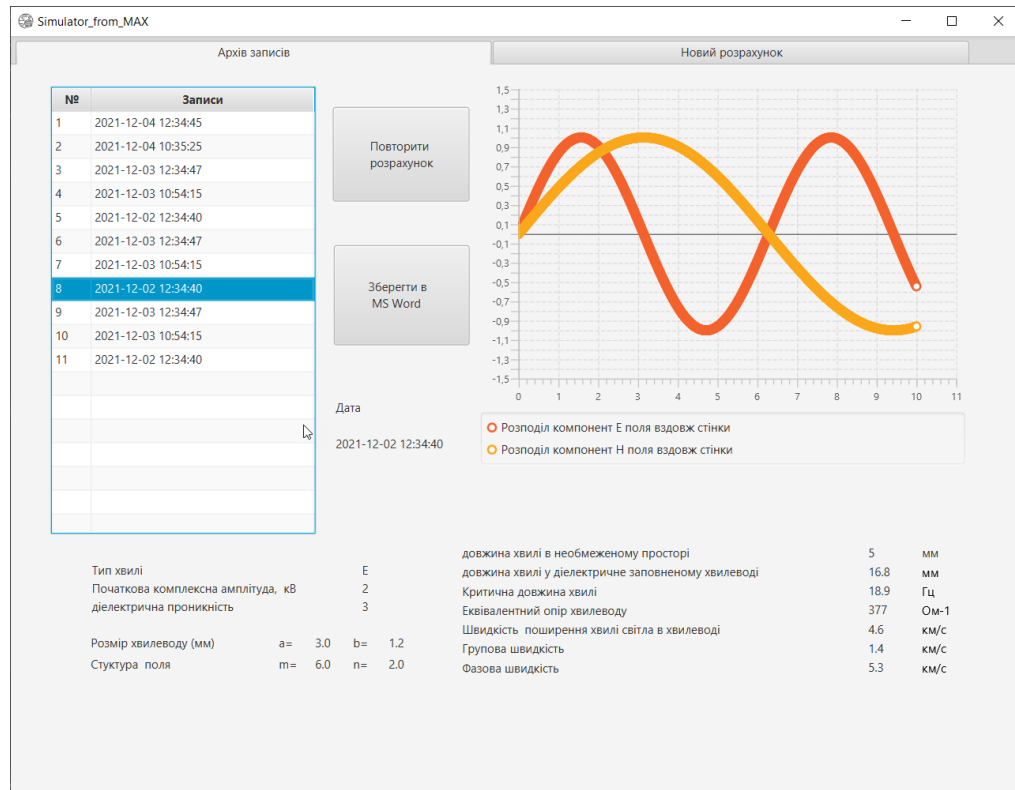


Рисунок 5.11 — Початкове вікно додатку

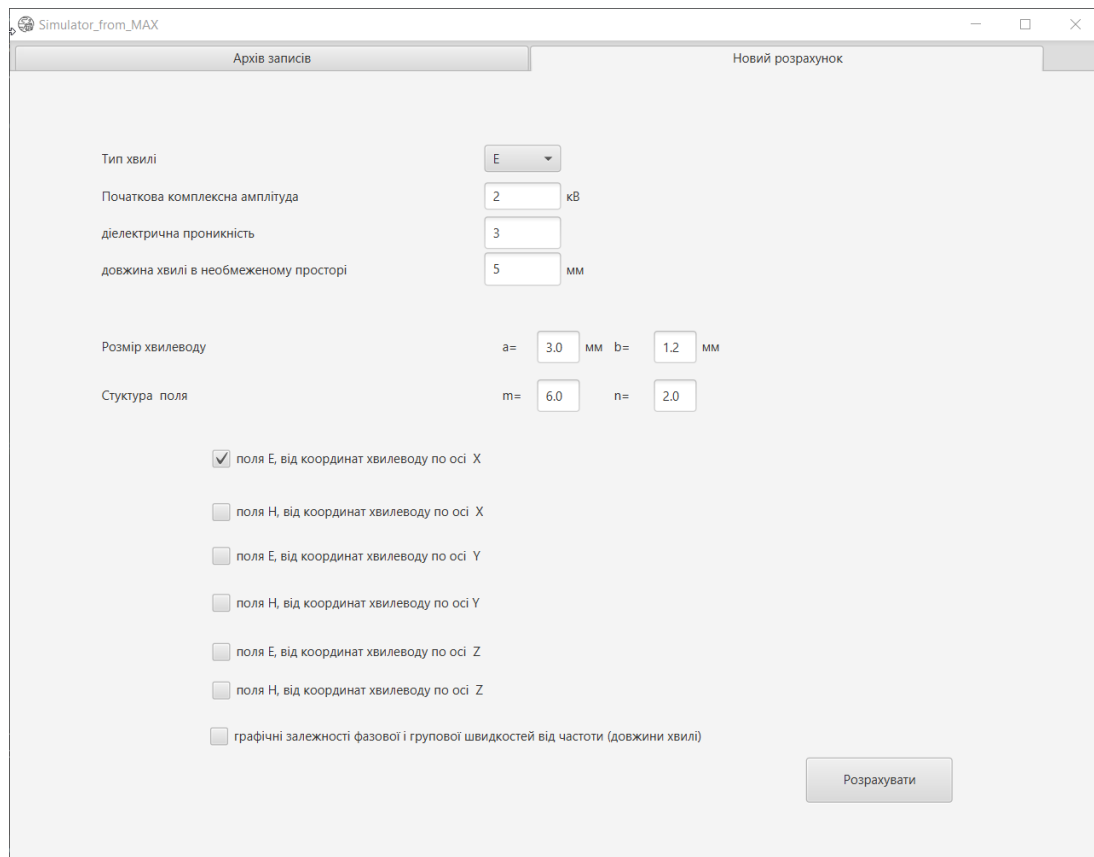


Рисунок 5.12 — Вікно програми для розрахунку

В цьому розділі нами було створено програмний код та інтерфейс робочого додатку. А також описані змінні дані та методи їх обробки відповідно до математичної моделі та алгоритму.

6 ТЕСТУВАННЯ ТА ІНСТРУКЦІЯ КОРИСТУВАЧА

6.1 Тестування програми та узагальнення роботи

Для тестування нам необхідно провести наповнення контентом програми Simulator_from_MAX. С цією метою ми здійснено введення даних в програму на прикладі одного розрахунку. При запуску відкривається вікно “Архів записів”. В цій вкладці має змогу повторити будь-який розрахунок, який є в архіві. Для цього нам потрібно вибрати запис з таблиці:

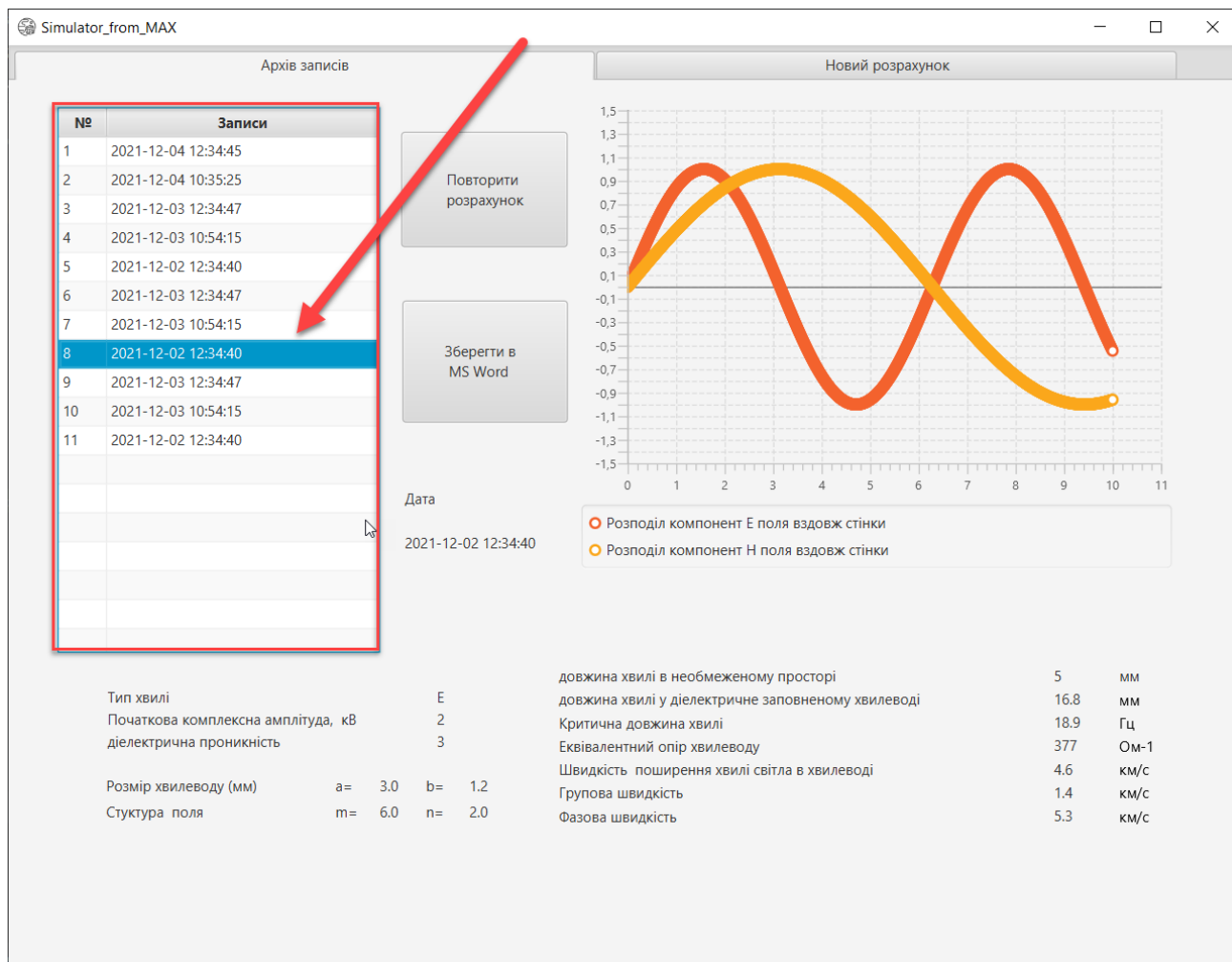


Рисунок 6.1 — Робоче вікно програми “Архів записів”

При виборі запису, відображуються раніше розраховані статичні значення та динамічний графік поля.

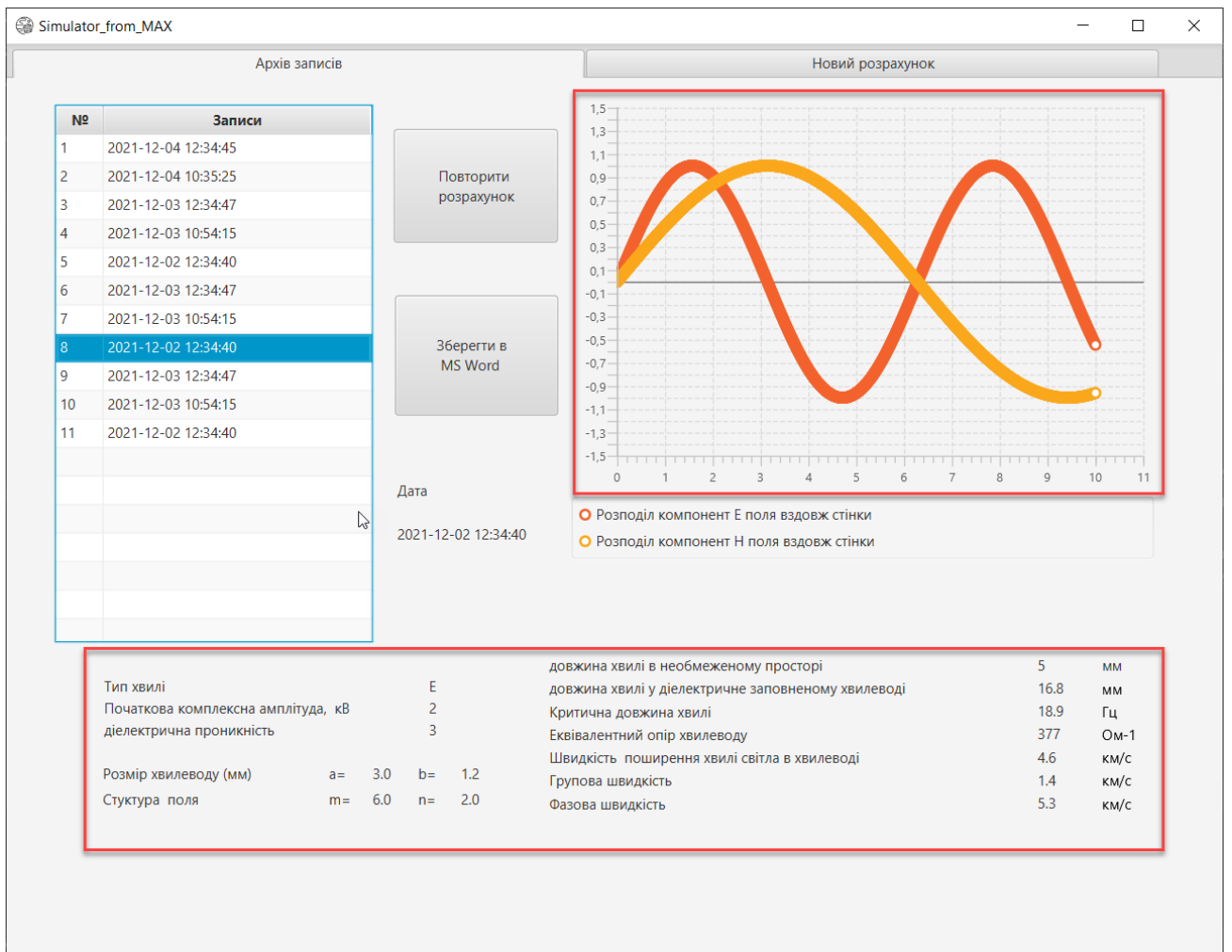


Рисунок 6.2 Робоче вікно програми “Архів записів”

Якщо нас не задовольняє значення або графік, чи бажаємо змінити один окремий параметр в розрахунку з метою повторного аналізу, натискаємо на кнопку “Повторити розрахунок”. При цьому відкривається вкладка “Новий розрахунок” з заповненими полями значеннями цього запису.

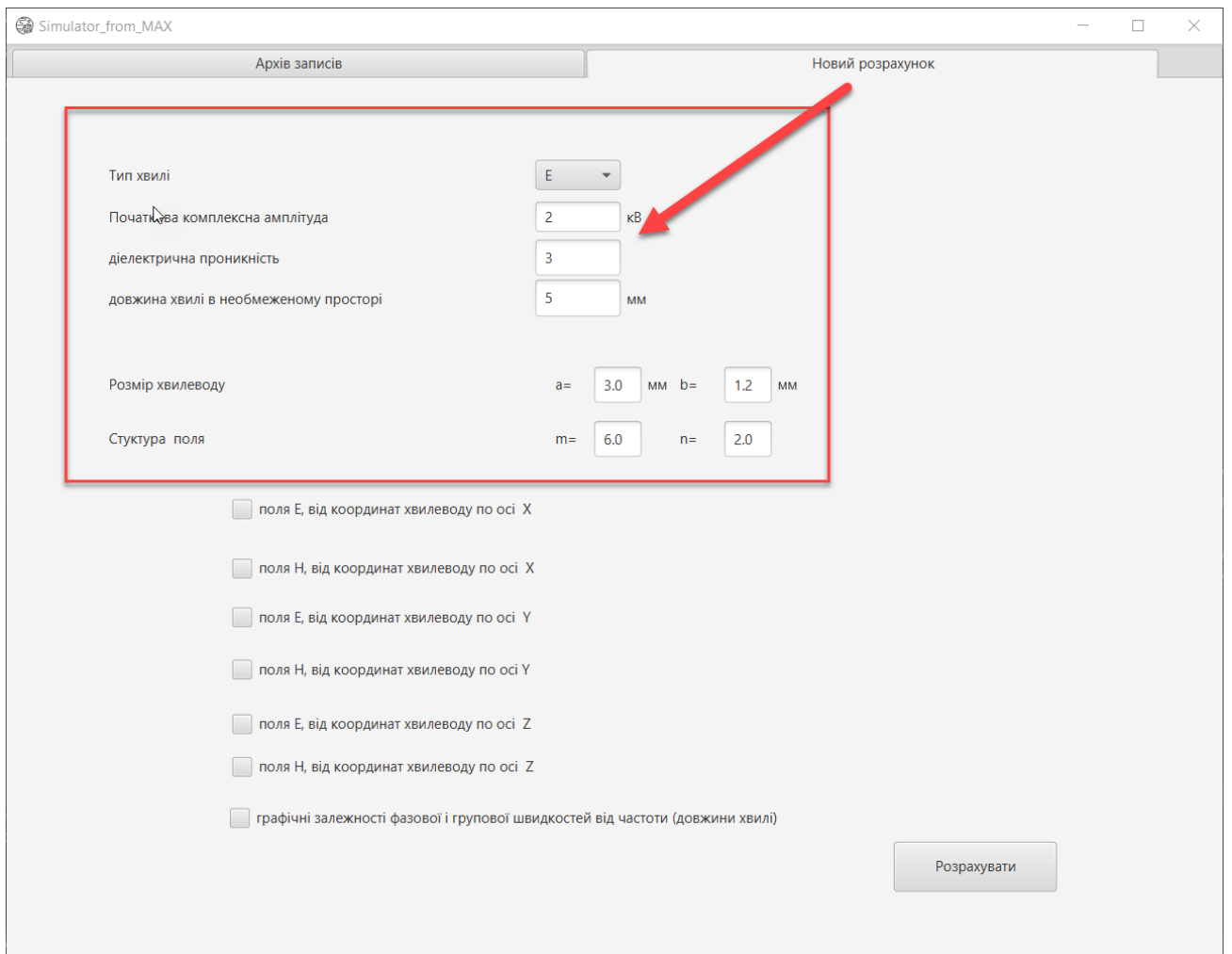


Рисунок 6.3 Вкладка “Новий розрахунок” з старими значеннями

Маємо змогу змінити будь-яке значення та розрахування з новими значеннями, натиснувши на кнопку “Розрахувати”.

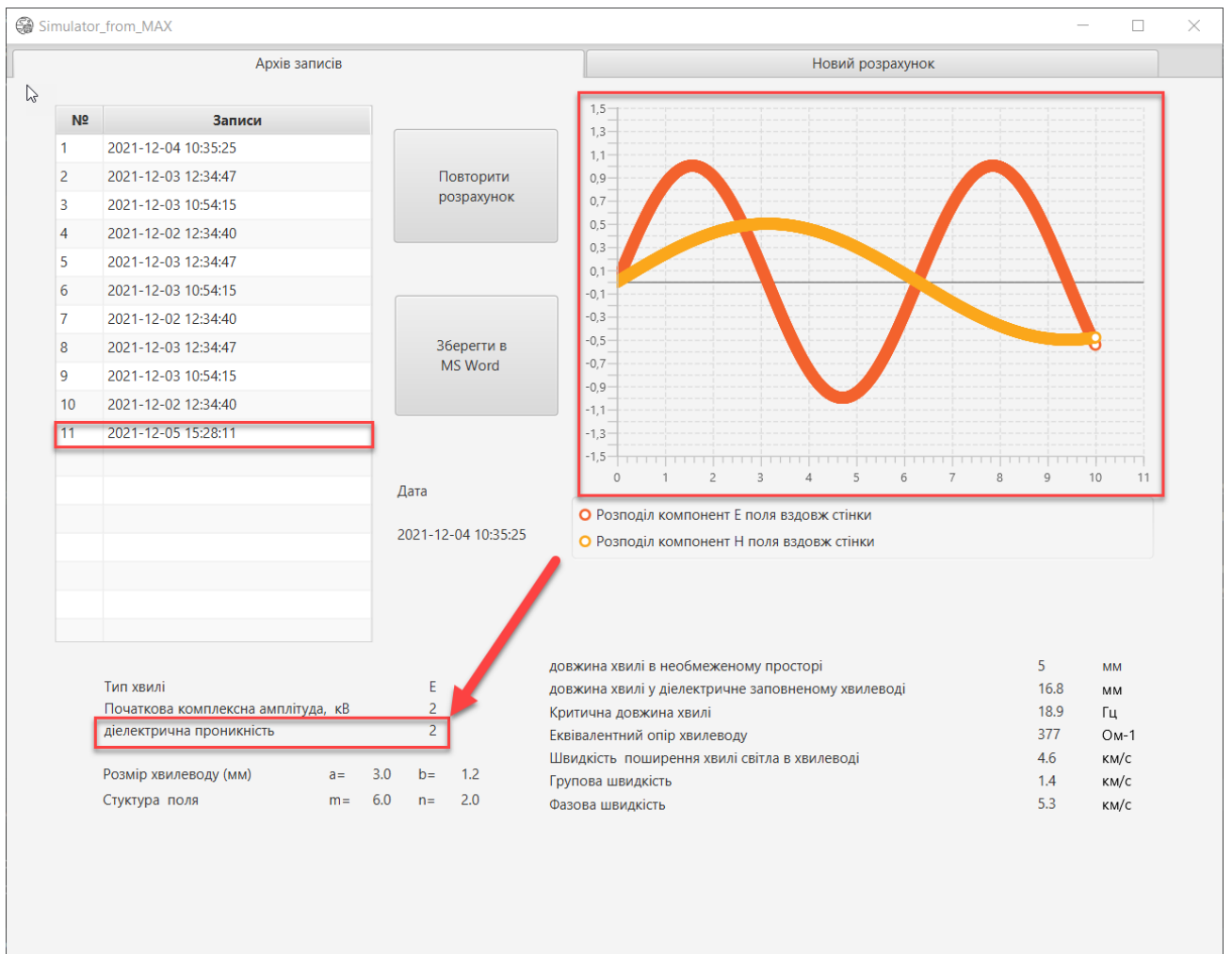


Рисунок 6.4 — Вкладка “Архів записів” з новим записом

Після розрахунку відкривається вкладка “Архів записів”. В таблиці записів додався новий запис і параметри цього запису автоматично відображуються в результатах. При розрахунку нового запису, переходимо на вкладку “Новий розрахунок” та вводимо значення для майбутнього розрахунку.

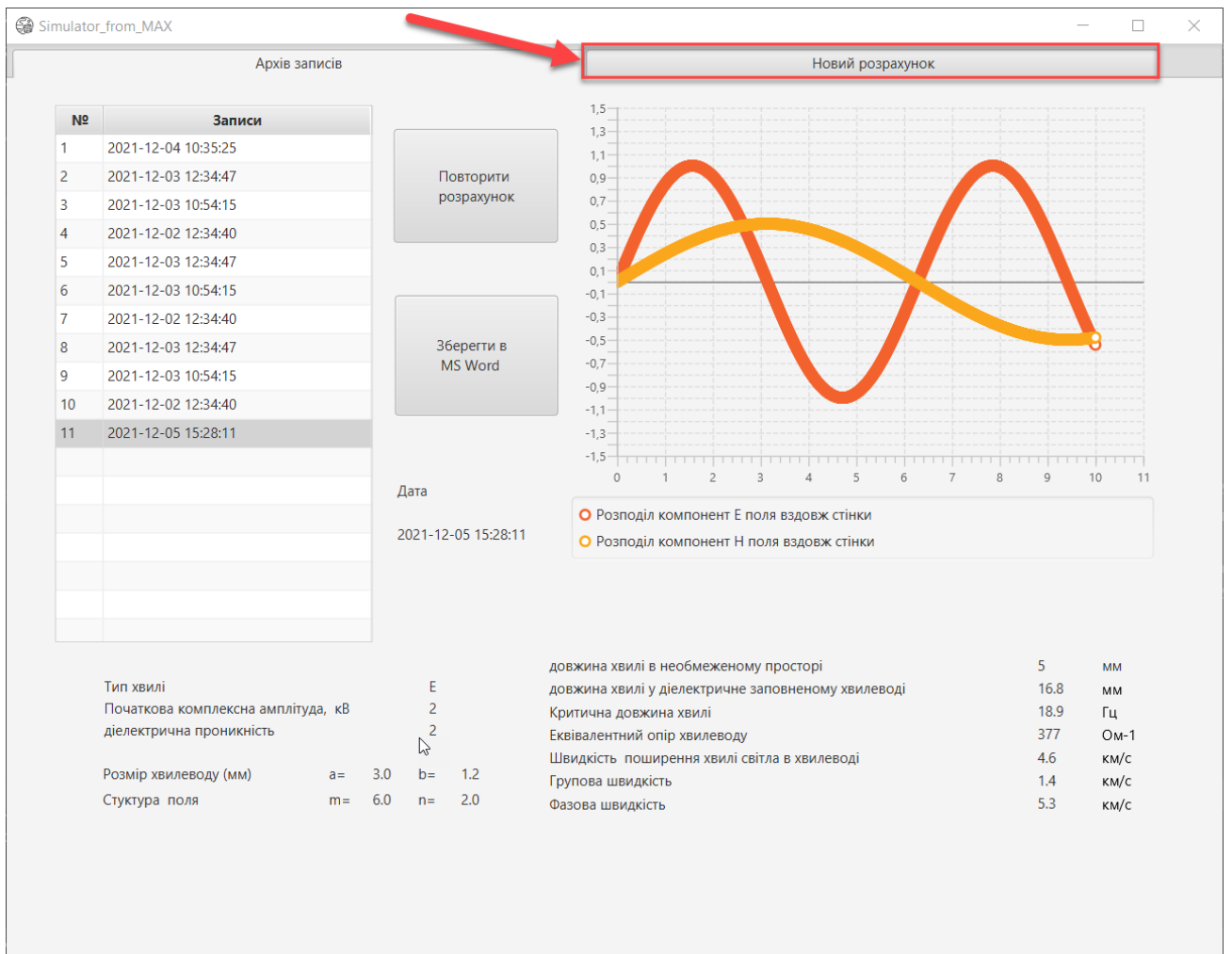


Рисунок 6.5 —Перехід з вкладки на вкладку

ВИСНОВКИ

Сьогодні інформаційні технології відіграють важливу роль в нашому житті. Сфера діяльності, яка пов'язана з математичними та фізичними розрахунками різних значень та проведення досліджень, в першу чергу потребує автоматизації цих робіт. На разі є велика кількість симуляторів здатних моделювати будь-які об'єкти, процесі та явища. На даний час студенти які вивчають властивості електричних і магнітних полів хвилеводів, їх поведінку у надвисоких частотах, проводять велику кількість розрахунків структури окремих видів електромагнітних полів. Такий підхід дозволяє на власні очі впевнитися у справедливості тих чи інших законів, зрозуміти їх фізичний зміст. Для пришвидшення виконання багато численних експериментів та розрахунків виникає потреба у спеціалізованому програмному продукті.

В дипломній роботі було виконано розробку інформаційного та програмного забезпечення системи розрахунків електромагнітних полів у прямокутному хвилеводі. В ході виконання роботи були сформульовані мета і задачі, проведено аналіз аналогів, в процесі якого були визначені переваги та недоліки аналогічних додатків. Були сформульовані функціональні вимоги: введення даних, розрахунок, перетворення на файл MS Word результатів, зберігання останніх 20 результатів. Також в ході роботи були обрані інструменти для реалізації. Враховуючи невеликий обсяг розрахунків серед засобів реалізації було обрано використання мови Java.

Було проведено моделювання контекстної діаграми в нотації IDEF0 та її декомпозицію на першому рівні. Були створені діаграми варіантів використання додатку, створена модель програмного продукту з побудовою моделі процесу, діаграма послідовності та діаграма діяльності. Було виконано побудова математичної моделі та проектування моделі бази даних. Визначено структуру вхідних даних та потреба в користувацькому інтерфейсі, що дало змогу

створити прототип майбутнього програмного продукту. Сформовані вимоги до реалізації програмного продукту.

Метою роботи було створення програмного продукту для пришвидшення розрахунків полів у прямокутному хвилеводі у надвисоких частотах. Після виконання всіх вимог, було розроблено додаток який в майбутньому буде слугувати для прискореного розрахунку електромагнітних полів.

Дане рішення покликане спростити процес розрахунків, та мінімізувати час виконання робіт студентами, що є актуальним питтям під час навчання. Для перевірки працездатності системи було проведено її закриті бета-тестування. Сам тренажер-симулятор показав себе з позитивної сторони.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Симуляція [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://uk.wikipedia.org/wiki/Симуляція> (дата звернення 08.10.2021)
2. Комп'ютерне моделювання об'єктів і процесів. Інформаційні моделі. [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://mozok.click/667-kompyuterne-modelyuvannya-obyektiv-procesv-nformacyn-model.html> 2017р. (дата звернення 12.10.2021)
3. Симулятор Tracker [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <http://physlets.org/tracker/>. (дата звернення 10.10.2021)
4. Симулятор Graph [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.padowan.dk/> (дата звернення 10.10.2021)
5. Симулятор GeoGebra [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.geogebra.org/> (дата звернення 10.10.2021)
6. Методичні вказівки та завдання до виконання курсової роботи «Розрахунок структури електромагнітних полів» із курсу «ТЕОРІЯ ПОЛЯ» Сумський державний університет, 03.01.2013
7. Крачтен Филипп. Введение в Rational Unified Process. - 2-е изд.: Пер. с англ. - М.: Издательский дом “Вильямс”, 2002. - 240 с.
8. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. — СПб.: Питер, 2002. — 496 с.
9. Java. Базовый курс [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://stepik.org/course/187/syllabus> (дата звернення 10.10.2021)
10. Основи програмування на Java [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу:

https://courses.prometheus.org.ua/courses/EPAM/JAVA101/2016_T2/about

(дата звернення 10.10.2021)

11. OMG UNIFIED MODELING LANGUAGE SPECIFICATION VERSION 2.5.1 [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.omg.org/spec/UML/2.5.1/> (дата звернення 13.10.2021)
12. [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: Онлайн – редактор Diagram.net <https://app.diagrams.net/> (дата звернення 13.10.2021)
13. 36. Скурихин В. И. Математическое моделирование / Скурихин В. И., Шифрин В. Б., Дубровский В. В. – К. : Техніка, 1983. – 270 с.
14. 37. Математические основі теории автоматического регулирования / [под ред. Б.К. Чемоданова]. – М. : Висшая школа, 1997. – 600с.
15. Інформаційні системи в економіці: підручник для академічного бакалавриата / В.Н. Волкова [та ін.]; ред. В.Н Волковой, В.Н. Юрьева. – М.: Видавництво Юрайт, 2016 – 402 с. – Серія: Бакалавр. Академічний курс
16. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес. Приемы объектноориентированного проектирования. Паттерны проектирования. Питер-ДМК, 2001.
17. Г. Буч, Дж. Рамбо, А. Джекобсон. Язык UML.Руководство пользователя. М., ДМК, 2000.
18. Lesson: JDBC Basics [Електронний ресурс] : [Веб-сайт]. – Електронні дані. –Режим доступу: <https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html> (дата звернення 03.11.2021).
19. What is MVC, and how is it like a sandwich shop? [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:

<https://www.freecodecamp.org/news/simplified-explanation-to-mvc-5d307796df30/> (дата звернення 11.11.2021).

ДОДАТОК А

Функції для обробки даних

```
//з'єднання з БД

private static Connection ConnectToBase() {

    Connection dbConnection = null;

    String= "jdbc:mysql://localhost:3306/BaseRecords" + "?"
+"useUnicode=true&useSSL=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimest
ode=false&serverTimezone=UTC";

    String user = "UserMax";

    String password = "MaxPass";

    try {

        Class.forName("com.mysql.cj.jdbc.Driver");

    } catch (ClassNotFoundException e) {

        System.out.println("Нема драйверу MySQL JDBC....");

        e.printStackTrace();

    }

    try {

        dbConnection = DriverManager.getConnection(connectionString,user,password);

        System.out.println ( "Підключення до бази даних успішне");

        return dbConnection;

    } catch (SQLException e) {

        System.out.println("Підключення до бази даних не вдалося");

    }

    return dbConnection;

}
```

```
//Завершення з'єднання з БД
public void shutdown() throws SQLException {
    if (dbConnection != null) {
        dbConnection.close();
    }
}

// видалення запису
public void DelRecords(int rec) {
    String deleteTableSQL = "DELETE DBRECORDS WHERE REC_ID =" + rec;
    try {
        dbConnection = getDBConnection();
        statement = dbConnection.createStatement();

        // виконуємо запит delete SQL
        statement.execute(deleteTableSQL);
        System.out.println("Запис видалено з таблиці DBRECORDS!");
    } catch (SQLException e) {
        System.out.println("Помилка видалення запису");
    }
}

//додавання запису
public void AddRecords(Recording reco) {
    String insertTableSQL = "INSERT INTO DBRECORDS"
```

```

        + "(REC_ID,
REC_DATA,TYPE_W,M,N,E,A,B,L0,D,L1,Z,FK,VE,VF,VG,CREATED_DATE) " + "VALUES"
        +
reco.Data[0]+"," +reco.Data[1]+"," +reco.Data[2]+"," +reco.Data[3]+"," +reco.Data[4]+"," +reco.Data[
5]+"," +reco.Data[6]+"," +reco.Data[7]+","
        +
reco.Data[8]+"," +reco.Data[9]+"," +reco.Data[10]+"," +reco.Data[11]+"," +reco.Data[12]+"," +reco.D
ata[13]+"," +reco.Data[14]+"," +reco.Data[15]+","
        + "to_date('" + getcurrentTimeStamp() + "', 'yyyy/mm/dd hh24:mi:ss'))";
try {
    dbConnection = getDBConnection();
    statement = dbConnection.createStatement();
    statement.executeUpdate(insertTableSQL);
    System.out.println("Запис створено!");
} catch (SQLException e) {
    System.out.println("Помилка запису");
}
}

private static String getcurrentTimeStamp() { Date today = new Date(); return
dateFormat.format(today.getTime()); }

//отримання запису

public String[] getRecordList() throws SQLException {
String selectTableSQL = "SELECT REC_ID, REC_DATA from DBRECORDS";
try {
    dbConnection = getDBConnection();
    statement = dbConnection.createStatement();

```

```

Recording record = new Recording();

// вибираем данні з БД

ResultSet rs = statement.executeQuery(selectTableSQL);

// Та якщо щось було отримане що цикл while спрацює
while (rs.next()) {

    Data[0] = rs.getString("REC_ID");    System.out.println("rec_id : " + recid);

    Data[1] = rs.getString("REC_DATA");    System.out.println("rec_date : " + recdate);

        Data[2] = rs.getString("TYPE_W");    Data[3] = rs.getString("M");

        Data[4] = rs.getString("N");        Data[5] = rs.getString("E");

        Data[6] = rs.getString("A");        Data[7] = rs.getString("B");

        Data[8] = rs.getString("L0");        Data[9] = rs.getString("D");

        Data[10] = rs.getString("L1");    Data[11] = rs.getString("Z");

    Data[12] = rs.getString("FK");    Data[13] = rs.getString("VE");

    Data[14] = rs.getString("VF");    Data[15] = rs.getString("VG");

    return Data;

}

} catch (SQLException e) {

    System.out.println("помилка считування даних з БД...");//e.getMessage()

}

}

//Створюємо Word файл

public void createWord(List<String> lines) throws IOException {

```

```

//Перевірка створення папки результатів. Якщо її немає, то створюємо.
if (!Paths.get("./result").toFile().exists()) Files.createDirectories(Paths.get("./result"));

//Створюємо Word документ.
for (String rec_data_line : lines) {

    //Бланк (шаблон) документа
    XWPFDocument document = new XWPFDocument();

    //Записуємо документ у файлоу систему
    FileOutputStream out = new FileOutputStream("result/" + "createdWord" + "_" +
rec_data_line + ".docx");

    //створюємо параграф
    XWPFParagraph paragraph = document.createParagraph();

    XWPFRun run = paragraph.createRun();

    run.setText("Віртуальний симулятор структури змінних електромагнітних полів
в хвилеводі\n"+

        +"запис № " + rec_data_line + "\n"+ DATA_RECORDS);

    document.write(out);

    //Закриваємо документ
    out.close();

    System.out.println("createdWord" + "_" + rec_data_line + ".docx" + " документ
створено");
}
}

```

ДОДАТОК Б

Програмний код макету(View)

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.chart.LineChart?>
<?import javafx.scene.chart.NumberAxis?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.text.*?>
<TabPane fx:id="tabPane" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
    prefHeight="768.0" prefWidth="1024.0" tabClosingPolicy="UNAVAILABLE"
    tabMinWidth="470.0"
    xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
    fx:controller="MyApp.Interface">
    <Tab text="Архів записів">
        <AnchorPane minHeight="-Infinity" minWidth="-Infinity">
            <GridPane layoutX="81.0" layoutY="576.0" prefHeight="43.0"
                prefWidth="325.0">
                <columnConstraints>
                    <ColumnConstraints hgrow="SOMETIMES" maxWidth="315.0"
                minWidth="10.0" prefWidth="191.5"/>
                    <ColumnConstraints hgrow="SOMETIMES" maxWidth="144.0"
                minWidth="9.5" prefWidth="38.5"/>
                    <ColumnConstraints hgrow="SOMETIMES" maxWidth="144.0"
                minWidth="10.0" prefWidth="40.0"/>
                    <ColumnConstraints hgrow="SOMETIMES" maxWidth="144.0"
                minWidth="10.0" prefWidth="38.0"/>
                    <ColumnConstraints hgrow="SOMETIMES" maxWidth="144.0"
                minWidth="10.0" prefWidth="27.0"/>
                </columnConstraints>
                <rowConstraints>
                    <RowConstraints minHeight="10.0" prefHeight="30.0"
                vgrow="SOMETIMES"/>
                    <RowConstraints minHeight="10.0" prefHeight="30.0"
                vgrow="SOMETIMES"/>
                </rowConstraints>
                <Label text="Структура поля" GridPane.rowIndex="1"/>
                <Label text="m=" GridPane.columnIndex="1"
                GridPane.rowIndex="1"/>
                <Label text="n=" GridPane.columnIndex="3"
                GridPane.rowIndex="1"/>
                <Label text="Розмір хвилеводу (мм)"/>
                <Label layoutX="277.0" layoutY="35.0" text="a="
                GridPane.columnIndex="1"/>
                <Label layoutX="330.0" layoutY="35.0" text="b="
                GridPane.columnIndex="3"/>
                <Label fx:id="lb5" GridPane.columnIndex="2"/>
                <Label fx:id="lb7" GridPane.columnIndex="2"
                GridPane.rowIndex="1"/>
                <Label fx:id="lb8" GridPane.columnIndex="4"
                GridPane.rowIndex="1"/>
                <Label fx:id="lb6" GridPane.columnIndex="4"/>
            </GridPane>
            <GridPane layoutX="457.0" layoutY="485.0" prefHeight="132.0"
```

```

prefWidth="529.0">
    <columnConstraints>
        <ColumnConstraints hgrow="SOMETIMES" maxWidth="411.0"
minWidth="10.0" prefWidth="406.5"/>
        <ColumnConstraints hgrow="SOMETIMES" maxWidth="158.0"
minWidth="10.0" prefWidth="41.5"/>
        <ColumnConstraints hgrow="SOMETIMES" maxWidth="158.0"
minWidth="10.0" prefWidth="50.5"/>
    </columnConstraints>
    <rowConstraints>
        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
    </rowConstraints>
    <Label text="довжина хвилі в необмеженому просторі"/>
    <Label text="довжина хвилі у діелектричне заповненому хвилеводі"
GridPane.rowIndex="1"/>
    <Label text="мм" GridPane.columnIndex="2"/>
    <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Гц"
GridPane.columnIndex="2"
    GridPane.rowIndex="2"/>
    <Label text="Критична довжина хвилі" GridPane.rowIndex="2"/>
    <Label text="Еквівалентний опір хвилеводу"
GridPane.rowIndex="3"/>
    <Label text="Швидкість поширення хвилі світла в хвилеводі"
GridPane.rowIndex="4"/>
    <Label text="Групова швидкість" GridPane.rowIndex="5"/>
    <Label text="Фазова швидкість" GridPane.rowIndex="6"/>
    <Text strokeType="OUTSIDE" strokeWidth="0.0" text="мм"
GridPane.columnIndex="2"
    GridPane.rowIndex="1"/>
    <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Ом-1"
GridPane.columnIndex="2"
    GridPane.rowIndex="3"/>
    <Text strokeType="OUTSIDE" strokeWidth="0.0" text="км/с"
GridPane.columnIndex="2"
    GridPane.rowIndex="4"/>
    <Text layoutX="476.0" layoutY="95.0" strokeType="OUTSIDE"
strokeWidth="0.0" text="км/с"
    GridPane.columnIndex="2" GridPane.rowIndex="5"/>
    <Text layoutX="476.0" layoutY="113.0" strokeType="OUTSIDE"
strokeWidth="0.0" text="км/с"
    GridPane.columnIndex="2" GridPane.rowIndex="6"/>
    <Label fx:id="lb9" prefHeight="21.0" prefWidth="54.0"
GridPane.columnIndex="1"/>
    <Label fx:id="lb10" prefHeight="21.0" prefWidth="55.0"
GridPane.columnIndex="1"
    GridPane.rowIndex="1"/>

```



```

        <Label fx:id="lb11" prefHeight="21.0" prefWidth="53.0"
GridPane.columnIndex="1"
        GridPane.rowIndex="2"/>
        <Label fx:id="lb12" prefHeight="21.0" prefWidth="51.0"
GridPane.columnIndex="1"
        GridPane.rowIndex="3"/>
        <Label fx:id="lb13" prefHeight="21.0" prefWidth="53.0"
GridPane.columnIndex="1"
        GridPane.rowIndex="4"/>
        <Label fx:id="lb14" prefHeight="21.0" prefWidth="53.0"
GridPane.columnIndex="1"
        GridPane.rowIndex="5"/>
        <Label fx:id="lb15" prefHeight="21.0" prefWidth="53.0"
GridPane.columnIndex="1"
        GridPane.rowIndex="6"/>
    </GridPane>
    <AnchorPane layoutX="471.0" layoutY="10.0" prefHeight="400.0"
prefWidth="500.0">
        <LineChart fx:id="lineGraph" prefHeight="400.0"
prefWidth="500.0">
            <xAxis>
                <NumberAxis side="BOTTOM" tickUnit="1" upperBound="10"/>
            </xAxis>
            <yAxis>
                <NumberAxis autoRanging="false" lowerBound="-1.5"
side="LEFT" tickUnit="0.1"
                upperBound="1.5"/>
            </yAxis>
        </LineChart>
    </AnchorPane>
    <TableView fx:id="table" layoutX="41.0" layoutY="23.0"
prefHeight="452.0" prefWidth="267.0">
        <columns>
            <TableColumn fx:id="idColumn" prefWidth="40.0" text="№"/>
            <TableColumn fx:id="name_data_Column" prefWidth="225.0"
text="Записи"/>
        </columns>
    </TableView>
    <Button fx:id="fxButton1" layoutX="326.0" layoutY="43.0"
mnemonicParsing="false" onAction="#click1"
prefHeight="96.0" prefWidth="138.0"
text="Повторити&#10;розрахунок"/>
    <Button fx:id="fxButton2" layoutX="327.0" layoutY="183.0"
mnemonicParsing="false" onAction="#click2"
prefHeight="101.0" prefWidth="137.0" text="Зберегти в &#10;
MS Word"/>
    <GridPane layoutX="82.0" layoutY="504.0" prefHeight="56.0"
prefWidth="343.0">
        <columnConstraints>
            <ColumnConstraints hgrow="SOMETIMES" maxWidth="323.0"
minWidth="10.0" prefWidth="273.5"/>
            <ColumnConstraints hgrow="SOMETIMES" maxWidth="187.0"
minWidth="10.0" prefWidth="69.5"/>
        </columnConstraints>
        <rowConstraints>
            <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
            <RowConstraints minHeight="10.0" prefHeight="30.0"

```

```

vgrow="SOMETIMES"/>
        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
        </rowConstraints>

        <Label text="Тип хвилі"/>
        <Label text="Початкова комплексна амплітуда, кВ"
GridPane.rowIndex="1"/>
        <Label text="діелектрична проникність " GridPane.rowIndex="2"/>
        <Label fx:id="lb2" text="E" GridPane.columnIndex="1"
GridPane.rowIndex="0"/>
        <Label fx:id="lb3" GridPane.columnIndex="1"
GridPane.rowIndex="1"/>
        <Label fx:id="lb4" GridPane.columnIndex="1"
GridPane.rowIndex="2"/>
        </GridPane>
        <Label fx:id="lb1" layoutX="329.0" layoutY="373.0" prefHeight="21.0"
prefWidth="135.0"/>
        <Label layoutX="329.0" layoutY="339.0" text="Дата"/>
        </AnchorPane>
    </Tab>
    <Tab text="Новий Розрахунок">
        <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0"
prefWidth="200.0">
            <GridPane layoutX="86.0" layoutY="65.0" prefHeight="136.0"
prefWidth="503.0">
                <columnConstraints>
                    <ColumnConstraints hgrow="SOMETIMES" maxWidth="395.0"
minWidth="10.0" prefWidth="395.0"/>
                    <ColumnConstraints hgrow="SOMETIMES" maxWidth="187.0"
minWidth="10.0" prefWidth="108.0"/>
                    <ColumnConstraints hgrow="SOMETIMES" maxWidth="187.0"
minWidth="10.0" prefWidth="108.0"/>
                </columnConstraints>
                <rowConstraints>
                    <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
                    <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
                    <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
                    <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
                </rowConstraints>
                <Label text="Тип хвилі"/>
                <Label text="Початкова комплексна амплітуда"
GridPane.rowIndex="1"/>
                <Label prefHeight="21.0" prefWidth="216.0" text="діелектрична
проникність "
                    GridPane.rowIndex="2"/>
                <Label text="довжина хвилі в необмеженому просторі"
GridPane.rowIndex="3"/>
                <TextField fx:id="lb23" prefHeight="17.0" prefWidth="55.0"
GridPane.columnIndex="1"
                    GridPane.rowIndex="1"/>
                <TextField fx:id="lb24" prefHeight="30.0" prefWidth="53.0"
GridPane.columnIndex="1"
                    GridPane.rowIndex="2"/>
                <TextField fx:id="lb29" prefHeight="31.0" prefWidth="57.0"

```

```

GridPane.columnIndex="1"
    GridPane.rowIndex="3"/>
    <Label text="мм" GridPane.columnIndex="2" GridPane.rowIndex="3">
    <GridPane.margin>
    <Insets bottom="5.0" left="5.0" right="5.0" top="5.0"/>
    </GridPane.margin>
    </Label>
    <Label text="кВ" GridPane.columnIndex="2" GridPane.rowIndex="1">
    <GridPane.margin>
    <Insets bottom="5.0" left="5.0" right="5.0" top="5.0"/>
    </GridPane.margin>
    </Label>
    <ComboBox fx:id="lb22" onAction="#Wave_setType"
prefWidth="150.0" GridPane.columnIndex="1"/>
    </GridPane>
    <GridPane layoutX="86.0" layoutY="236.0" prefHeight="91.0"
prefWidth="598.0">
    <columnConstraints>
    <ColumnConstraints hgrow="SOMETIMES"
maxWidth="375.33331298828125" minWidth="10.0"
    prefWidth="375.33331298828125"/>
    <ColumnConstraints hgrow="SOMETIMES"
maxWidth="193.33334350585938" minWidth="8.0"
    prefWidth="27.80000000000001"/>
    <ColumnConstraints hgrow="SOMETIMES"
maxWidth="205.66668701171875" minWidth="10.0"
    prefWidth="66.0"/>
    <ColumnConstraints hgrow="SOMETIMES" maxWidth="144.0"
minWidth="10.0"
    prefWidth="32.39999999999998"/>
    <ColumnConstraints hgrow="SOMETIMES" maxWidth="144.0"
minWidth="10.0"
    prefWidth="77.20000000000005"/>
    </columnConstraints>
    <rowConstraints>
    <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
    <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES"/>
    </rowConstraints>
    <Label text="Структура поля" GridPane.rowIndex="1"/>
    <Label text="m" GridPane.columnIndex="1"
GridPane.rowIndex="1"/>
    <Label text="n" GridPane.columnIndex="3"
GridPane.rowIndex="1"/>
    <Label text="Розмір хвилеводу" />
    <Label layoutX="277.0" layoutY="35.0" text="a="
GridPane.columnIndex="1"/>
    <Label layoutX="330.0" layoutY="35.0" text="b="
GridPane.columnIndex="3"/>
    <TextField fx:id="lb25" maxWidth="-Infinity" minHeight="0.0"
minWidth="0.0" prefHeight="30.0"
    prefWidth="40.0" GridPane.columnIndex="2"/>
    <TextField fx:id="lb27" layoutX="327.0" layoutY="18.0"
maxWidth="-Infinity" minHeight="0.0"
    minWidth="0.0" prefHeight="30.0" prefWidth="40.0"
GridPane.columnIndex="2"
    GridPane.rowIndex="1"/>
    <TextField fx:id="lb26" layoutX="327.0" layoutY="64.0"

```

```

maxWidth="-Infinity" minHeight="0.0"
        minWidth="0.0" prefHeight="30.0" prefWidth="40.0"
GridPane.columnIndex="4"/>
        <TextField fx:id="lb28" layoutX="488.0" layoutY="18.0"
maxWidth="-Infinity" minHeight="0.0"
        minWidth="0.0" prefHeight="30.0" prefWidth="40.0"
GridPane.columnIndex="4"
        GridPane.rowIndex="1"/>
        <Label prefHeight="18.0" prefWidth="29.0" text="мм"
GridPane.columnIndex="2">
        <GridPane.margin>
        <Insets left="45.0"/>
        </GridPane.margin>
        </Label>
        <Label layoutX="461.0" layoutY="24.0" prefHeight="18.0"
prefWidth="29.0" text="мм"
        GridPane.columnIndex="4">
        <GridPane.margin>
        <Insets left="45.0"/>
        </GridPane.margin>
        </Label>
</GridPane>
<CheckBox layoutX="190.0" layoutY="354.0" mnemonicParsing="false"
text="поля E, від координат хвилеводу по осі X"/>
<CheckBox layoutX="190.0" layoutY="404.0" mnemonicParsing="false"
text="поля H, від координат хвилеводу по осі X"/>
<CheckBox layoutX="190.0" layoutY="445.0" mnemonicParsing="false"
text="поля E, від координат хвилеводу по осі Y"/>
<CheckBox layoutX="190.0" layoutY="489.0" mnemonicParsing="false"
text="поля H, від координат хвилеводу по осі Y"/>
<CheckBox layoutX="190.0" layoutY="535.0" mnemonicParsing="false"
text="поля E, від координат хвилеводу по осі Z"/>
<CheckBox layoutX="190.0" layoutY="571.0" mnemonicParsing="false"
text="поля H, від координат хвилеводу по осі Z"/>
<CheckBox layoutX="188.0" layoutY="614.0" mnemonicParsing="false"
text="графічні залежності фазової і групової швидкостей
від частоти (довжини хвилі)"/>
        <Button fx:id="fxButton3" layoutX="747.0" layoutY="643.0"
mnemonicParsing="false" onAction="#click3"
        prefHeight="42.0" prefWidth="137.0" text="Розрахувати"/>
    </AnchorPane>
</Tab>
</TabPane>

```