

Вінницький національний технічний університет  
Факультет комп'ютерних систем і автоматики  
Кафедра системного аналізу, комп'ютерного моніторингу  
та інженерної графіки

## **ІНФОРМАЦІЙНА СИСТЕМА АВТОМАТИЗОВАНОГО ФОРМУВАННЯ РОЗКЛАДУ**

Пояснювальна записка до магістерської кваліфікаційної роботи

Виконав: студент 2 курсу, групи ІСТ-18м  
спеціальності 126 – «Інформаційні системи та  
технології»  
Федюк О. П.

Керівник: к.т.н., доц. Крижановський Є.М.  
Рецензент: \_\_\_\_\_

Вінниця ВНТУ – 2019 року

Вінницький національний технічний університет  
Факультет комп'ютерних систем і автоматики  
Кафедра системного аналізу, комп'ютерного моніторингу  
та інженерної графіки

Освітньо-кваліфікаційний рівень магістр  
Спеціальність 126 - Інформаційні системи та технології  
(шифр і назва)

ЗАТВЕРДЖУЮ  
Завідувач кафедри САКМІГ  
\_\_\_\_\_ д.т.н., проф. В. Б. Мокін  
(підпис)  
“ \_\_\_ ” \_\_\_\_\_ 201\_\_ р.

ЗАВДАННЯ  
на магістерську кваліфікаційну роботу студенту  
Федюку Олегу Петровичу

1. Тема роботи : «Інформаційна система автоматизованого формування розкладу»  
керівник роботи Крижановський Є.М., к.т.н., доц., каф. САКМІГ  
затверджені наказом вищого навчального закладу від “ \_\_\_ ” \_\_\_\_\_ 20\_\_ року №\_\_
2. Строк подання студентом роботи \_\_\_\_\_
3. Вихідні дані до роботи:
  - мова програмування – C#, SQL
  - дані про аудиторії, викладачів, студентів
  - дані про навчальний план
  - вимоги студентів до розкладу
  - вимоги викладачів до розкладу.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ; обґрунтування вибору методу складання розкладу занять та постановка задачі дослідження; розробка алгоритму роботи програми генерації розкладу на його оцінки на основі переваг; розробка плану тестування розкладу ; висновки; додатки.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): архітектура програми; діаграма класів проекту; діаграма залежності класів; зовнішній вигляд інтерфейсу програми.

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Крижановський Є.М., к.т.н., доц. каф. САКМІГ		
4	Глущенко Л.Д., к.е.н., доц. каф. ЕПВМ		

7. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування вибору методу розробки та постановка задачі дослідження	09.2019	
2	Розробка структури та алгоритмів програмного продукту	09.2019	
3	Розробка інформаційної системи	09.2019	
4	Економічна частина	10.2019	
5	Розробка інтерфейсу користувача додатку	10.2019	
6	Тестування роботи додатку	11.2019	
7	Розробка інструкції користувача	11.2019	
8	Оформлення матеріалів до захисту МКР	11.2019	

Студент \_\_\_\_\_ Федюк О.П.  
( підпис ) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_ Крижановський Є.М.  
( підпис ) (прізвище та ініціали)

## РЕФЕРАТ

Магістерська кваліфікаційна робота: 100 ст., 7 табл., 31 рис., 16 джерел.

Об'єкт досліджень – процес автоматизації планування графіку навчання.

Мета роботи – розробка та реалізація оптимального підходу до складання розкладу, що відповідає сучасним стандартам, включаючи вимоги дуальної освіти, та забезпечує високу швидкодію та стабільність роботи системи.

Здійснено огляд існуючих систем для автоматичного складання графіку навчального процесу.

Прогнозні припущення про розвиток об'єкта дослідження – розробка інформаційної моделі складання розкладу, яка на відміну від існуючих більш краще масштабується при більших даних.

Галузь застосування – освітні сервіси і підприємства з навчанням співробітників.

ІНФОРМАЦІЙНА СИСТЕМА АВТОМАТИЗОВАНОГО ФОРМУВАННЯ РОЗКЛАДУ, ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, ЗАДАЧА ПРО ПЛАНУВАННЯ, ГЕНЕТИЧНИЙ АЛГОРИТМ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, АВТОМАТИЗАЦІЯ СКЛАДАННЯ РОЗКЛАДУ.

## ABSTRACT

This project is Master's qualification work: 100 pages, 7 tables, 31 figures, 16 sources.

The object of research is the process of automating the scheduling of training.

The purpose of the work is to develop and implement an optimal approach to scheduling that meets modern standards, including the requirements of dual education, and provides high speed and stability of the system.

An overview of the existing systems for automatic scheduling of the educational process is carried out.

Forecast assumptions about the development of the object of study - the development of an information model of scheduling, which, unlike existing, is better scaled with larger data.

The scope of application - educational services and enterprises with employee training.

INFORMATION SYSTEM OF AUTOMATED, FORMAT, FORMATION, GENETIC, ALLIATIC, ALLIANCE, PLANET, GENETIC, ALLIATIC, ALLIANCE, PLANET, GENETIC, ALLIRE

## ЗМІСТ

ВСТУП.....	5
1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ.....	10
1.1 Основні поняття системи складання розкладу занять.....	10
1.2 Варіанти реалізації генерацій графіків навчального процесу .....	13
1.3 Організація роботи з базою даних.....	16
1.4 Висновки .....	18
2 ТЕХНОЛОГІЇ СКЛАДАННЯ СИСТЕМИ РОЗКЛАДУ ЗАНЯТЬ.....	19
2.1 Вибір IDE .....	19
2.2 Опис алгоритму .....	19
2.3 Цільова функція.....	20
2.4 Модель.....	23
2.5 Алгоритм роботи генерації розкладу та оцінки .....	26
2.5.1 Перша генерація .....	27
2.5.2 Змішування нової генерації.....	28
2.5.3 Селекція та струс.....	30
2.5.4 Вихід із циклу .....	31
2.6 Висновки .....	31
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ РОЗКЛАДУ ЗАНЯТЬ ТА ЇЇ ТЕСТУВАННЯ.....	32
3.1 Вибір середовища програмування.....	32
3.2 Архітектура бази даних .....	33
3.3 Розробка опису програми на мові C#.....	35
3.3.1 Клас Dictionary.....	35
3.3.2 Клас Rozklad .....	38
3.3.3 Клас Покоління - Generation .....	38
3.4 Розробка плану тестування розкладу .....	39
3.5 Розробка тестового додатку .....	40
3.6 Виконання тестування додатку на реальних даних.....	43
3.7 Висновки .....	47
4 ЕКОНОМІЧНА ЧАСТИНА.....	48
4.1 Оцінювання комерційного потенціалу розробки (технологічний аудит розробки).....	48
4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько-технологічної роботи.....	52
550.....	54
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки ...	56
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності.....	58
4.5 Висновки .....	62
ВИСНОВКИ.....	64
СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	66
ДОДАТКИ.....	68

Додаток А Технічне завдання .....	69
Додаток Б. Інтерфейс програми.....	71
Додаток В. Лістинг програми .....	72
Додаток Г. Графічна частина .....	76
Додаток Д. Таблиця критеріїв оціювання розробки.....	99

## ВСТУП

**Актуальність.** Планування є важливим завданням для університету. Планування є проблемою у багатьох галузях. Перш за все, у плануванні виробництва, у підприємствах пасажирських та вантажних перевезеннях, проектуванні та проведенні навчальних занять у середній, професійній та вищій школі. Він ґрунтується на необхідності забезпечення оптимального розподілу роботи між виконавцями різних послуг з урахуванням факторів простору та часу.

Звичайно, у кожному навчальному закладі є людина, відповідальна за організацію навчального процесу, але ця робота є складною для однієї людини. Величезна кількість інформації, яку потрібно зібрати, безліч різних умов, яких необхідно досягти, і відповідальність, яка покладається на відповідальну особу, є надто важким фактором для виконання однією людиною одночасно з її робочими обов'язками. А якщо взяти до уваги проблему великої кількості лабораторій, практичних кабінетів та лекційних залів, також предметів, викладачів, підприємств, та ще із розширенням і збільшенням навчальних предметів і інших факторів навчання це завдання стає складнішим. Ця задача є складною для людини, але її можна зробити набагато простіше і швидше за допомогою комп'ютера.

Сам навчальний графік залежить від об'єктивних (жорстких) та суб'єктивних (непостійних) параметрів. Жорсткі параметри - це база даних університету, яка зберігає інформацію про навчальні предмети, викладачів, начальних кімнат та аудиторій. Суб'єктивними є побажання учнів та викладачів та керівників підприємств, якщо для університету є актуальним принципи дуальної освіти.

Якщо взяти ретельно оцінити таку задачу, то є помітним, що завдання складання навчального графіку є важким для людини, тому що потрібно пам'ятати про всі кабінети, робочі кімнати, навчальні зали, лабораторії, підприємства, які підтримують університет і його програму по дуальній освіті, також всі побажання студентів та викладачів, яких означає необхідність збирання цієї інформації та частіше в особистій формі, а також необхідність гармонійного поєднання всіх цих



даних у навчальному графіку, щоб усі були задоволені, але, як ми знаємо, цього досягти майже ніколи не можливо.

Процес впровадження дуальної освіти додає до цієї задачі ще більше вимог, оскільки складений розклад має ще й погоджуватися з планами роботи підприємств, на яких студенти проходять практику під час навчання.

З другого погляду, комп'ютер може легко впоратися з цим завданням. Завдання зберігання інформації про навчальні кімнати, лабораторії, лекційні зали та підприємства в базі даних, застосування певних функцій і формул до бази даних, яка будує навчальний графік розкладу, та оцінювання цього графіку відповідно до побажань людей - це просто для комп'ютера і обчислювальних машин. Завдання планування викликає великий інтерес серед науковців та працівників університету та працівників підприємств також. Тому що вони особисто зацікавлені у його вирішенні та їхня кількість порівняно значна, у наш час розроблено потужний набір відповідних моделей та методів.

Оптимальність у цьому кейсові визначає ефективність процесів на виробництві (університеті, підприємстві). Для промислового виробництва - це максимальна кількість виробленої продукції та послуг за одиницю часу, економія матеріалів та енергоресурсів та інших ресурсів, зниження собівартості продукції та послуг, підвищення продуктивності і ефективності праці, для організації навчання - це деякий розподіл підгруп та груп студентів за навчальними кабінетами з урахуванням обмеження на спеціалізацію лабораторій та години роботи викладацького персоналу.

І якщо виробничі завдання забезпечення оптимальності графіків розкладу роботи виробництва відіграли вирішальну роль, то поняття оптимальності втратило своє значення при підготовці розкладу уроків, оскільки відповідне завдання є NP-завершеним, а відповідальні особи в його плануванні виходять лише з навчальних програм та об'єктивних обмежень та роботи. в режимі "real time", змінюючи графік на вимогу або на прохання викладацького персоналу. Очевидно, що отриманий таким чином розклад є далеко не оптимальним і викликає скарги як у студентів, так і викладацького персоналу і керівників підприємств, як представників дуальної

освіти. Висновок з цього, що оптимальність - це декомпозиція понять, неформалізованих, залежно від того, який критерій буде заснований на ній.

Також процеси трансформації, пов'язані з європейською змінною постійною інтеграцією, позитивні і негативні результати реформ, підвищення і пониження конкурентоспроможності економіки, це все передбачає створення відповідної національної системи освіти та навчання. Важливо створити умови для поєднання роботи з освітою у формі навчання вечірньої або заочної форми навчання, а також у форматі практичної підготовки вищих навчальних закладів денної форми навчання та студентів професійно-технічної (професійно-технічної) освіти (далі - освіта). Закон України "Про освіту" передбачає, що людина має право отримувати освіту в різних формах або поєднуючи їх, і визначає такі основні форми навчання: інституційна (денна (денна, вечірня), заочна, дистанційна, мережева); індивідуальний (зовнішній, сімейний (домашній), педагогічний патронат, робоче місце (на роботі); подвійний - дуальний).

Подвійна форма навчання (дуальна) - це спосіб здобуття освіти, який передбачає поєднання навчання осіб у навчальних закладах з навчанням на робочому місці на підприємствах, в установах та організаціях для набуття певної кваліфікації, як правило, на підставі контракту на здійснення подвійної освіти (далі - договір).

У такій Концепції термін "спеціаліст" означає здобувача професійної (професійно-технічної), професійної вищої та вищої освіти.

У період ринкової трансформації економіки та зростання попиту на освітні послуги традиції практичної підготовки отримувачів освіти були девальвовані. Багато організацій відмовилися працювати зі студентами, посиляючись на комерційні таємниці або складні конкурентні обставини. Внаслідок корпоратизації та приватизації держава втратила більшість важелів адміністративного впливу на підприємства, що дало можливість розглянути практичну підготовку студентів у неекономічній площині. Податкові та інші ринкові преференції не визріли і після їх впровадження швидко скасовуються з міркувань бюджетної консолідації та запобігання корупції.

**Актуальність розробки** полягає у можливості використання розробки системи в органах державного управління та громадських органах з використанням як ліцензійного так і офіційно безкоштовного програмного забезпечення.

**Зв'язок роботи з науковими програмами, планами, темами.** Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри системного аналізу, комп'ютерного моніторингу та інженерної графіки Вінницького національного технічного університету та плану наукової та навчально-методичної роботи.

**Метою дослідження** є розробка та реалізація оптимального підходу до складання розкладу, що відповідає сучасним стандартам, включаючи вимоги дуальної освіти, та забезпечує високу швидкодію та стабільність роботи системи.

Для досягнення вищевказаної мети необхідне розв'язання такі **задачі**:

- врахувати побажання студентів та викладачів щодо розкладу;
- забезпечення оптимального розподілу робіт серед викладачів, враховуючи просторові і часові обмеження;
- дослідити технологію створення серверної частини системи на основі патерну MVC 5;
- проаналізувати взаємодії із базою даних за допомогою технології Entity Framework 6;
- на основі розглянутих підходів та концепцій розробити приклад інтернет-системи розкладу занять.

Використані у науковій роботі **методи дослідження**:

- порівняння – для визначення переваг та недоліків існуючих аналогів, що будуть враховані при створенні серверної частини системи складання розкладу;
- узагальнення – для визначення загальних властивостей і ознак існуючих систем складання розкладу;
- моделювання – для представлення алгоритму складання розкладу;
- формалізація – для перетворення схем та діаграм у код мови

програмування;

- аналіз – для поділу об'єкту дослідження на складові частини з метою їх самостійного дослідження;
- експеримент – для тестування та визначення ефективності роботи створеного, аналізу доцільності його створення.

**Об'єктом дослідження** є процес складання розкладу занять.

**Предметом дослідження** є методи і інструменти планування оптимального навчального розкладу.

**Наукова новизна одержаних результатів.** Результати, які були отримані в процесі вирішення задач та складають наукову новизну дослідження, самі полягають у удосконаленні підходу до автоматизованого формування розкладу, який, на відміну від існуючих, передбачає можливість врахування більшої кількості суб'єктивних даних при складанні розкладу, а також передбачає його адаптацію до умов дуальної освіти.

**Практична цінність** даної роботи – запропоновано рішення для отримання оптимального розкладу, що на відміну від існуючих використовують ефективні алгоритми оптимізації з використанням цільової функції.

**Достовірність теоретичних положень** магістерської кваліфікаційної роботи підтверджується правильною постановкою завдань, коректним застосуванням формул і методів під час доведення наукових положень, порівнянням результатів з відомими існуючими.

**Особистий внесок здобувача.** Усі результати, наведені у магістерській кваліфікаційній роботі, отримані особисто автором програмної системи.

**Апробація результатів роботи.** Атрибуцію подальшого дослідження було здійснено на конференції «Інформаційні технології і автоматизація - 2019» (ГО «Клуб молодих вчених Одеської національної академії харчових технологій», 2019).

**Публікації.** За результатами магістерської кваліфікаційної роботи було опубліковано тези на науково-технічну конференцію: «Інформаційні технології і автоматизація - 2019» [1].

## 1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ

### 1.1 Основні поняття системи складання розкладу занять

Графік навчального процесу (розклад) - сам по собі є поняттям простим з точки зору сучасного життя, але проблему його формування важко назвати простою. За класичним визначенням графік навчального процесу (розкладу) - це документ підприємства, який регулює робочий ритм, визначає обмеження в часі всіх робочих процесів і формує оптимальний розподіл такого важливого ресурсу, як робочий час.

Планування - це складне завдання, оскільки воно вимагає від відповідальної людини врахувати багато факторів: жорсткі вимоги та нежорсткі. У роботі зосереджено велику увагу саме на графіку занять закладу, для якого жорсткими умовами будуть:

- а) години на предмет для відпрацювання;
- б) викладацький персонал для ведення предметів. Надалі все що актуальне для викладачів є актуальним для керівників підприємств;
- в) студенти;
- г) аудиторії або місця підприємств, де ці заняття/робочі заняття проходять.

Не таким і простим ресурсом буде час, який потрібно бути розділити між робочими днями та навчальними уроками.

До необ'єктивних (нежорстких) умов віднесено:

- вимоги та побажання викладацького персоналу/керівників підприємств;
- побажання та вимоги студентів.

Об'єктивні (жорсткі) умови повинні завжди дотримуватися, інакше розклад невірний і збитковий. Необ'єктивні (нежорсткі) умови можуть бути не виконані, а їх виконання безпосередньо впливає на ефективність розкладу з психологічного погляду.

Якщо розглянути навчальний процес для розуміння вимог до самого графіку, він складається з таких понять, як предмет, викладацький персонал, студенти, навчальні кімнати, лекційні зали і лабораторії.

Предмет – це наукова дисципліна, яку викладає певний викладач певній групі учнів чи студентів. Для спрощення проблеми ми припускаємо, що один вчитель може вести одну і ту ж дисципліну в одній групі, тобто немає спільного для декількох груп пар (уроків).

Предмети мають кілька форм занять. Класика включає лекції, практики, лабораторні роботи, семінари, самостійне навчання і самостійні роботи студентів, іспити, факультативи. У цій роботі ми обмежимося лекціями, практиками та лабораторними заняттями, оскільки вони є важливими та частішими [2].

Кожне заняття повинно займати академічну годину, або пара годин. Ми пропустимо варіанти, коли можливо бути пів пари і більше.

Вчителі (викладачі) – працівники навчального закладу, які проводять заняття та є рушійною силою навчального процесу. Викладачі мають посади, вчені звання та ступінь, які безпосередньо стосуються їх важливості та досягнень у кафедрі. Звичайно, чим вище посада, тим більше пріоритету надається вимогам цього вчителя.

Вчитель (викладач) – це жива біологічна людина, тому процес планування повинен бути суворим, оскільки один вчитель не може мати пару (урок) в двох групах одночасно. І один вчитель не може бути в двох навчальних кімнатах одночасно.

Кожен викладач, а також керівник підприємства, мають свої вимоги до навчального графіку та пріоритетів цих вимог, тому всі вони повинні бути враховані. І кожен викладач має своє власне значення, яке вираховується через його зарплату, для спрощення розрахунків.

Слід також зазначити, що не всі форми навчальних занять можуть вести всі викладачі. Лекції можуть читати лише викладачі-доценти, практики та лабораторні для студентів можуть проводити аспіранти або нижчі звання. Ці вимоги

регулюються у навчальній програмі та на розсуд Вченої ради і повинні розглядатися як жорсткі.

Студенти є такими ж важливими учасниками процесу, як і викладачі, але їх набагато більше, тому буде розглянуто лише групи учнів у розкладі, а не кожен окремо. Студенти діляться на потоки за терміном навчання - курси та на групи по 25 - 35 осіб. У кожній групі є свій код, який складається з назви, номера курсу та номера групи.

Жорсткі вимоги такі ж, як і до викладачів, адже студенти теж люди, але є одна різниця, яка викликана тим, що їх багато. А саме, якщо аудиторія невелика, то вся група не вміститься, тому це потрібно враховувати.

Для полегшення задачі будемо не враховувати те, що групи можуть поділятися на пів-групи на певних дисциплінах. Для полегшення завдання ми не будемо враховувати той факт, що групи в певних дисциплінах можна розділити на напівгрупи.

Вимоги формувати простіше, ніж з викладачами. Вимоги кожної групи будуть враховані залежно від голосування студентів. Тому кожна група відповідатиме одному списку вимог та одному списку пріоритетів. Голосування формуватиметься наступним чином: після того, як будуть зібрані голоси та всі вимоги кожного студента, усі вимоги збираються в один список. Суперечливі вимоги повністю закреслені. Наприклад, один учень хоче, щоб у вівторок не було пар, а інший хоче навпаки. Якщо двоє студентів не можуть досягти консенсусу, то вимоги обох студентів відкидаються.

Після цього формується матриця порівнянь кінцевих вимог (матриця Сааті) та знайдено власний вектор найбільшого власного значення. Цей вектор буде відображати пріоритети та бажання всієї групи та буде використовуватися в цільовій функції, про яку піде мова в наступних розділах.

Аудиторія вже детально висвітлювалася раніше в розділі, але багато деталей потрібно уточнити.

Кожна аудиторія – це фізична кімната в приміщенні освітньої організації, обладнана особливим чином для проведення занять. Кожна з кімнат обладнана для певного типу занять: лекції - для лекцій, лабораторні - для лабораторій.

Також навчальна кімната (аудиторія чи практичний кабінет) містить деяку кількість місць. Хоча у деяких навчальних кабінетах можна проводити декілька форм занять. Та такі навчальні кабінети будуть використовуватись тільки тоді, коли інші рівнозначні зайняті [3].

## **1.2 Варіанти реалізації генерацій графіків навчального процесу**

Це завдання є критичним для всіх навчальних закладів. І освітні організації намагалися вирішити проблему планування автоматично.

Однією з таких є система в Вінницькому університеті для планування магістерського розкладу та сесій, описаних у джерелах [4-5]. Схема побудована на трьох важливих таблицях: магістранти, викладачі та дисципліни (предмети). Також ця схема враховує всі вимоги Болонського процесу та регулює їх виконання.

Дисципліни в графіку навчального процесу також поділяються на лекційні, практичні, лабораторні та семінарські уроки. Кожен клас/група/підгрупа вимагає певної кількості часу і його можна навчати безліччю викладачів, які навчають певну базу студентів, до безлічі студентів. Тому планування розкладу - це також рівномірне та оптимальне розміщення предметів між викладачами та учнями/студентами. Також у цьому методі існує четвертий потік даних - індивідуальний план потоку магістра, який включає загальний потік, потоки практичних та семінарських занять, професійні потоки та комбіновані потоки студентів магістратури [4].



Цільова функція, яка оцінює графік навчального процесу у цій моделі, як у формулі 1.1:

$$\delta = \max \left( \sum_{k=1}^{m_t} w_k \sum_{j=1}^{m_x} \frac{v_{kj} \pi_{kj}}{\sum_{i=1}^{m_x} \pi_{ki}} + \sum_{j=1}^{m_h} n_k \cdot h_k \right) \quad (1.1)$$

де  $v_{kj}$  - індикаторна функція визначення врахування вимоги викладача/керівника підприємства для певної дисципліни/задачі,

$\pi_{ki}$  – пріоритет цієї вимоги/задачі,

$w_k$  – ваговий коефіцієнт рейтингу викладача/управляючого,

$\pi_{kj}$  – пріоритети комплексу умов викладача/управляючого.

Алгоритм планування графіку навчального процесу розроблений за такими принципами:

- а) набір інформації про дисципліни;
- б) набір та встановлення взаємозв'язків між викладачами (менеджерами підприємства) і заняттями (задачами);
- в) набір та встановлення взаємозв'язків між групами студентів і заняттями;
- г) введення обмежень та вимог;
- д) формування розкладу;
- е) перевірка графіку навчального процесу ( в разі неуспіху – формування розкладу знову);
- ж) оптимізація розкладу;
- з) вивід на екран, або у XLS файл.

Повний алгоритм можна побачити на рисунку 1.1.

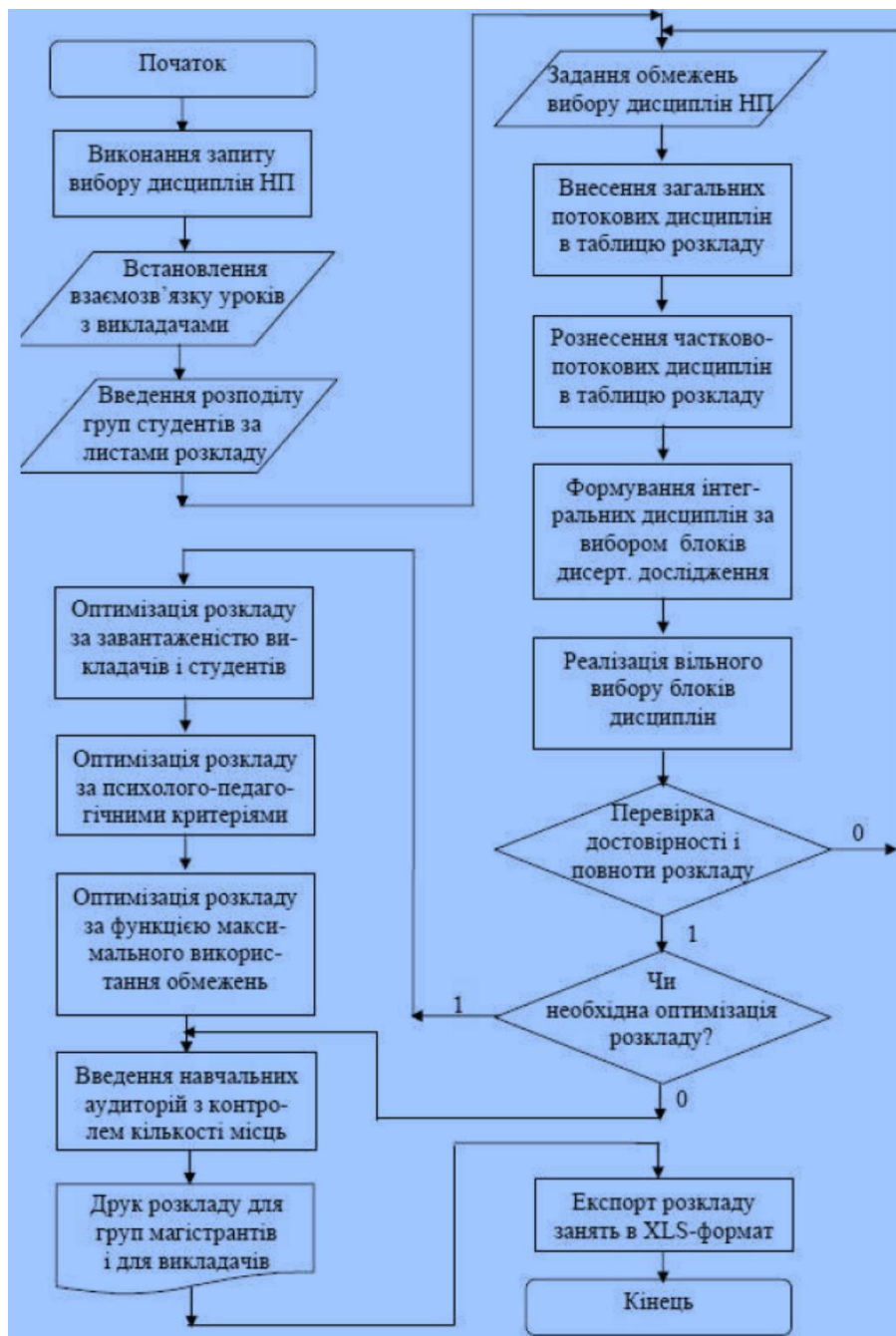


Рисунок 1.1 - Реалізація методу планування графіку навчального процесу магістратури

Цей спосіб реалізації схожий на той, що використовується в цій роботі, але оптимізація та дизайн будуть різними. Також алгоритм, описаний у цій роботі, враховуватиме не лише вимоги викладачів, але й учнів/студентів [5].

Беручи до уваги вищевказану інформацію та відомі методи реалізації завдання, було визначено такі вимоги до планування графіку навчального процесу:

- повинен відповідати навчальному/робочому процесу і адекватності;
- жорсткі умови мають виконуватись, адже вони визначають адекватність графіку навчального процесу;
- повинен відповідати побажанням студентів та викладачам/управлінців;
- пріоритет побажання викладача залежить від посади викладача.
- зручний інтерфейс користування.

Отже, можна зробити висновок, що графік навчального процесу буде відповідати плану, об'єднуючи дисципліни, групи, викладацького персоналу та навчальним кабінетам. Потім графік навчального процесу оцінюватиметься за цільовою функцією, щоб визначити, наскільки вигідний графік навчального процесу. Його адекватність буде гарантована тестуваннями та самим алгоритмом. Після того, як графік навчального процесу буде сформований та оцінений, його потрібно оптимізувати певним методом та оцінити ще раз. Він повинен бути оптимізований та оцінений до досягнення необхідної точності.

### **1.3 Організація роботи з базою даних**

База даних – це систематизована сукупність даних, призначених для довгого в часі зберігання (в зовнішній пам'яті ЕОМ) і по частоті постійного використання певних даних. База даних є не випадковою збіркою відомостей, а постійною основою для певної діяльності користувача. Для управління базою даних використовують діалогову мову запитів SQL [6].

SQL (Structured Query Language) – це стандартна мова запитів для роботи з реляційними базами даних. SQL в чистому (базовому) вигляді є інформаційно-логічною мовою, а не мовою програмування. Однак стандарт мови специфікацією SQL / PSM передбачає можливість її процедурних розширень, з урахуванням яких мова вже може розглядатися в якості мови програмування.

В архітектурі «клієнт-сервер» мова SQL займає дуже важливе місце. Саме вона використовується як мова спілкування клієнтського програмного забезпечення з серверної СУБД, розташованої на віддаленому комп'ютері. Розроблений стандарт мови SQL, який є спільною розробкою ANSI (American National Standards Institute) і ISO (International Organization for Standardization).

Основними перевагами SQL є [6]:

- незалежність від конкретних СУБД. Незважаючи на наявність діалектів і відмінності в синтаксисі, більшість текстів SQL запитів, що містять DDL і DML, можуть бути легко перенесені з однієї СУБД в іншу. Існують системи, на які розробники спочатку покладаються на використання принаймні декількох СУБД (наприклад: електронна система документообігу Documentum може працювати як з Oracle, так і з Microsoft SQL Server та IBM DB2);
- наявність стандартів. Наявність стандартів та тестового набору для виявлення сумісності та відповідності конкретній SQL реалізації загальноприйнятого стандарту, який лише сприяє «стабілізації» мови. Однак варто зазначити, що сам стандарт іноді занадто формалізований і завищений розмірами, наприклад, Основна частина стандарту SQL: 2003 включає понад 1300 сторінок тексту;
- декларативність. За допомогою SQL програміст описує лише ті дані, які потрібно витягнути або змінити. Як це зробити, СУБД вирішує безпосередньо під час обробки запиту SQL. Однак не слід думати, що це абсолютно універсальний принцип - програміст описує набір даних для вибірки чи модифікації, але йому корисно уявити, як СУБД буде розбирати текст свого запиту. Особливо критичними такі моменти стають при роботі з великими базами даних і складними запитами - чим складніший запит, тим більше він дозволяє написання, різні за швидкістю написання, але однакові в наборі даних [7]. За допомогою SQL програміст описує лише ті дані, які потрібно витягнути або змінити. Як це зробити, вирішує СУБД

безпосередньо під час обробки запиту SQL. Однак програмісту корисно знати, як СУБД буде розбирати текст свого запиту. Чим складніший запит, тим він допускає написання, які відрізняються швидкістю виконання, але однаковими в наборі результатів [8]

Недоліки SQL:

- неповна відповідність реляційної моделі даних (наявність дублікатів, необов'язковість первинного ключа);
- недостатньо продуманий механізм невизначених значень;
- складність формулювань і громіздкість запитів.

#### **1.4 Висновки**

Отже, як висновок, графік навчального процесу буде відповідати плану, він буде об'єднувати дисципліни (предмети), групи, викладачів (керівників підприємств) та навчальні кабінети. Потім графік навчального процесу оцінюватиметься за цільовою функцією для визначення ефективності графіку навчального процесу. Його адекватність буде гарантована тестуванням та алгоритмом. Після того, як графік навчального процесу буде сформований та оцінений, його потрібно оптимізувати певним методом та оцінити ще раз. Його слід оптимізувати.

## 2 ТЕХНОЛОГІЇ СКЛАДАННЯ СИСТЕМИ РОЗКЛАДУ ЗАНЯТЬ

### 2.1 Вибір IDE

Для виконання цього завдання нам потрібно вибрати ide (середовище розробки і програмування), яке буде відповідати нашим вимогам, а також мати низьку вартість. Щодо останнього, то можна зазначити, що більшість сучасних середовищ вже перебувають у відкритому коді, а ті, які не є у відкритому коді, вже стали безкоштовними, тому вартість не варто турбуватися.

Основними вимогами до середовища програмування цього проекту будуть:

- легкість;
- швидкість;
- ефективність роботи;
- наявність підтримки технічних спеціалістів;
- наявність бібліотек та додаткових інструментів;
- дружлюбний юзер інтерфейс та інструментарій для дебагінгу коду;
- підключення до БД (бази даних) і функціонал для роботи з БД.

Отже було вибрано пакет MS Visual Studio 2015 [9] Community з базою даних на основі MS SQL Server [10]. Мовою програмування буде C#.

### 2.2 Опис алгоритму

Основна мета цієї роботи - побудувати зручний та оптимальний графік навчання, який найкращим чином відповідатиме побажанням людей та полегшить завдання людини, відповідальної за розклад занять. Це буде зроблено за допомогою генетичного алгоритму та цільової функції, яка відповідає вашим побажанням. Сама цільова функція буде складатися з двох частин: переваги голосування для студентів та викладачів. Це буде записано у вигляді матриці переваг, яка буде побудована шляхом голосування. За допомогою цієї функції ми можемо оцінити графік.

Сам графік буде генерований за допомогою певного алгоритму, після чого він буде оцінюватися функцією, описаною вище, і, якщо необхідно, уточнений у наступному поколінні та знову оціниться. І так до тих пір, поки не знайдеться оптимальний варіант [11].

Повний опис алгоритму можна побачити на рисунку 2.1.

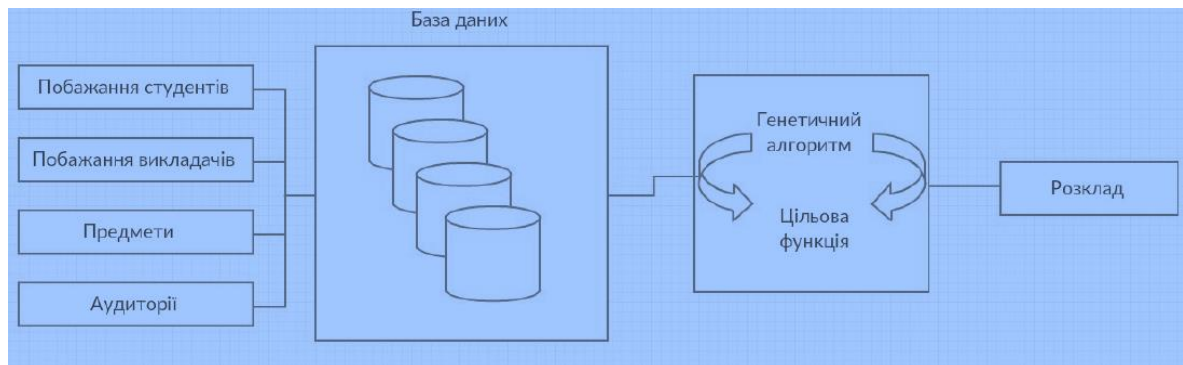


Рисунок 2.1 - Архітектура програми

Необхідні дані зберігаються в базі даних, де оператор буде отримувати дані в жорстких умовах і мати змогу додавати нову інформацію про установу. Вихід з розкладу буде виконано у файлі XLS для зручного використання.

### 2.3 Цільова функція

В цільовій функції тре врахувати багато факторів, що визначають розклад як ефективний, адекватний та оптимальний з точки зору навчального процесу. Основні вимоги:

- повинна відповідати вимогам навчального процесу;
- повинна давати кращі результати, коли її аргумент є вигіднішим і кращим за гірші аргументи (кращі вихідні дані – кращі вхідні дані);
- значення функції не мають бути менше нуля;
- функція має масштабуватись для більшої кількості аргументів.

Отже, цим вимогам відповідає формула (2.1), яка існує на множині (2.2). Така функція дискретна з немалою кількістю числових розривів. Залежить вона прямо

пропорційна від виконання вимог графіком навчального процесу. Індикаторні функції гарантують такий результат.

$$F(r) = a_s \sum_{j=1}^l x_j \chi\{Z_j^v\} + a_L \sum_{j=1}^K y_j \sum_{i=1}^M \chi\{L_i \in T_i\} \sum_{j=1}^K d_{ij}^j \chi\{Z_{ij}^{Tj}\} \rightarrow \max \quad (2.1)$$

$$r \in \Omega(P, S, L, A) \quad (2.2)$$

де  $r$  – графік навчального процесу,

$a_s, a_L$  - коефіцієнти вагових пріоритетів, що вказують на пріоритетність викладачів і керівників підприємств і студентів, як суб'єктів навчального і робочого процесу,

$x_j, y_j$  – пріоритети вимог всіх груп студентів і викладацького, управлінського персоналу,

$Z_j^v$  - вимоги груп учнів/студентів,

$L_i$  – викладачі і керівники підприємств,

$T_j$  – групи викладачів/керівників підприємств,

$Z_{ij}^{Tj}$  - переваги викладачів/керівників підприємств,

$d_{ij}^j$  - пріоритети вищевказаних побажань,

$l$  - кількість вимог студентів,

$K$  - кількість груп викладачів, які розподілені посадами, науковими ступенями та вченими званнями,

$M$  – кількість викладачів, і  $n$  - кількість викладачів в  $i$  -й групі,

$\Omega$  - область обмежень,

$P, L, A$  – множина навчальних дисциплін, викладацького персоналу і навчальних кабінетів.



Вектори пріоритету будуть оператором генеруватися вручну через обмеження ресурсів та часу під час виконання цієї роботи, але в майбутньому залишать можливість для розширення. Кожен вектор буде помножений на векторний індикатор, який у відповідних позиціях матиме «1», якщо вимогу виконано, і «0», якщо - ні, як у формулі (2.3). Це забезпечить простоту реалізації, що дозволить програмі просто перевірити певні умови та перенести обчислення на якийсь інший метод..

$$x\{Z_j^y\} = \begin{cases} 1, \text{ якщо } Z_j^y \text{ виконується} \\ 0, \text{ в іншому випадку} \end{cases} \quad (2.3)$$

Ця функція є чудовою і може збивати з пантелику на перший погляд, тому буде наведено спрощений приклад, щоб зрозуміти, як вона працює. Вихідними для неї є вектор вимог до групи учнів, вектори вимог кожного вчителя та вектор пріоритетів самих викладачів, який має відповідний коефіцієнт пріоритетності кожного вчителя. По-перше, відбувається перевірка вимог, при якій ці вектори множать на відповідні індикаторні функції.

Потім отримані значення для вчителів множать на вектор пріоритету і отримують кінцеве значення функції для вчителів. Зрештою, ми маємо визначити, які побажання важливіші: викладачі чи студенти, і тому ми множимо значення функції для студентів на коефіцієнт учнів, а вчителів на коефіцієнт викладачів. І ми пропускаємо отримані значення. Зрозуміло, що керуючи кожним вектором, кожним фактором, ми можемо оцінити графік з багатьох сторін, формуючи наші вимоги. Це дозволить нам мати простий і прямий контроль за процесом оптимізації.

Так як ця функція не є безперервною, неможливо застосувати до неї методи класичного математичного аналізу. Тому було запропоновано використовувати генетичний алгоритм. Але перед цим необхідно констатувати, як будуватиметься графік пам'яті [11].

## 2.4 Модель

На основі інформації, описаної в першому розділі, ми можемо створити математичну модель розкладу. Запишіть графік навчального процесу у форму, як показано в таблиці 1.1.

Таблиця 1.1 - Початкова модель

День	Пара	Курс	Група	Предмет	Викладач	Тип	Аудиторія
...	...	...	...	...	...	...	...

Очевидним фактом є те, що це первинний ключ у цій таблиці, а що ні, тому що безпосередньо залежить, з якої сторони підійти до завдання, чи з якої сторони подивитися. У цій роботі первинним ключем було вибір набору з Предмет, Тип класу та Групи, в якій буде проводитися заняття, оскільки ця інформація одночасно визначає процес навчання з точки зору студента. Тут можна зазначити, що Вчитель - це також важлива сфера, але один предмет може вести кілька вчителів, що робить його не унікальним ключем.

Можна також зазначити, що один предмет можна викладати в декількох підгрупах і групах. Саме для усунення таких проблем було вибрано таку модель.

Ця модель, описана в таблиці 1, потребує спрощення, оптимізації для вдосконалення. Введемо всю інформацію про графік навчального процесу паралелепіпедом.

Спочатку було розділено на чотири групи, як показано у формулах 2.4, 2.5, 2.6, 2.7.

$$X_1 = \langle \text{День} \rangle, \quad (2.4)$$

$$X_2 = \langle \text{Пара} \rangle, \quad (2.5)$$

$$X_3 = \langle \text{Аудиторія} \rangle, \quad (2.6)$$

$$Z = \langle \text{Викладач} - (\text{Предмет} - \text{Тип}) - (\text{Курс} - \text{Група}) \rangle, \quad (2.7)$$

де  $X_1, X_2, X_3$  – координати у паралелепіпеді, а  $Z$  – значення.

Значення Група, Курс (далі Група) об'єднано, бо різні. Тип і Предмет (далі Предмет) об'єднуються, бо їх суміщення однозначно визначаються одиницю навчання. Пара, День і Аудиторія не розділяються, бо вони відповідають фізичні жорсткі умови, які не порібно порушити і це буде гарантувати що в одній аудиторії в той самий час не буде проходити 2 заняття.

Викладач – предмет – група будуть міститись у вузлах паралелепіпеда. Отже запис у списку графіка навчального процесу у моделі виглядатиме як у формулі (2.8).

$$r_n = (D_i, T_j, R_k, N_l) \quad (2.8)$$

де  $D_i$  - день тижня (вівторок -2),

$T_j$  - номер пари/уроку,

$R_k$  – навчальний кабінет номер,

$N_l$  – ланка для з'єднання ключа <Викладач – предмет – група>,

$r_n$  – Розклад (графік навчального процесу).

З цього можна зробити висновок про розмірність куба, який матиме 3 розміри з четвертим у вузлах [12]. Розмір куба буде статичним, оскільки за тиждень всього 6 робочих днів, а в день може бути лише 6 пар, а кількість аудиторії буде взята з даних про відділ, але це вже в наступному розділ. Графік можна зобразити графічно, як показано на рисунку 2.2.

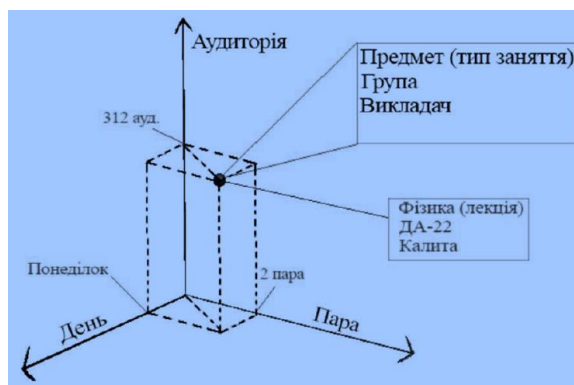


Рисунок 2.2 – Графічне представлення графіку навчального процесу

Для розуміння вищевказаного рисунка 2.2 розглянуто приклад. Нехай ключовим буде лекція з дисципліни фізика, яку в групі DA-22 проведе вчитель Віктар Киліта. У вівторок пара уроків відбудеться у 317 навчальному кабінеті для другої пари. Отже, як показано на рис. 2.2, у кубі за координатами (2, 2, 317), що відповідає 2 дню тижня понеділок, друга пара та відповідна навчальна кімната будуть посиланням <Фізика (лекція) - ТАК -22 - Киліта>.

Створюючи розклад таким чином, ми зможемо забезпечити жорсткі умови та мати зручний доступ до будь-якої вартості чи пункту в графіку. Слід зазначити, що обмеження, яке полягає в тому, що один вчитель і одна група можуть бути одночасно лише в одному посиланні, не можуть бути гарантовані цією схемою, тому це гарантуватиме алгоритм планування.

## 2.5 Алгоритм роботи генерації розкладу та оцінки

Вибравши модель розкладу та цільову функцію, нам потрібно поговорити про те, як оптимізувати цей графік. Це допоможе вдосконалити графік роботи та змусити його відповідати вимогам, визначеним у цій функції.

Для виконання такого завдання було обрано еволюційний генетичний алгоритм, який полягає в тому, що кожне наступне покоління графіків формуватиметься з минулих ітерацій. Тоді відбудеться відбір, під час якого всі слабкіші графіки будуть скинуті. Таким чином цільова функція буде рости без будь-якого втручання користувача, оскільки ця модель замкнена.

Загальні кроки такі є:

- а) створення першої генерації розкладів випадковим чином.
- б) оцінка цих розкладів цільовою функцією.
- в) генерація нового “потомства” з минулої ітерації.
- г) відкидання гірших розкладів.
- д) повторювати кроки 2-4 до тих пір, поки не буде отримана задана точність, або кількість кроків перебільшить допустиму.

Алгоритм на рисунку 2.3:

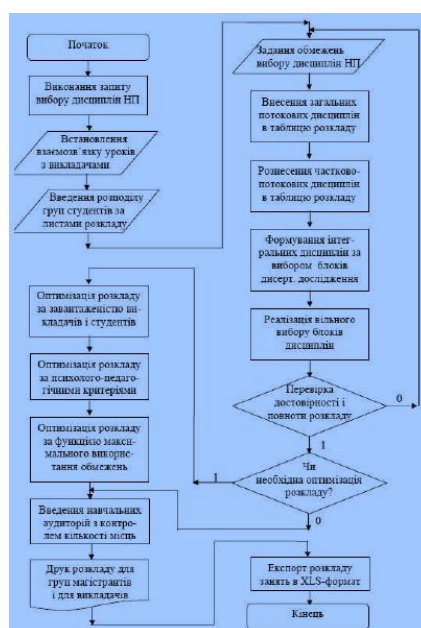


Рисунок 2.3 – Блок-схема алгоритму формування розкладу

Звичайно, це більш загальний і неспецифічний опис алгоритму, але він показує основну ідею, що графіки будуть генеруватися випадковим чином, зміщуватися випадковим чином, але не вибиратися випадковим чином. Це означає, що шанс вибрати вдалий графік збільшиться. І завдяки сучасним технологіям такі розрахунки відбудуться швидко, на відміну від старих машин [13].

### 2.5.1 Перша генерація

Перший список графіків навчальних процесів буде створений випадковим чином. Спочатку алгоритм вибирає випадковий день тижня та час, вибирає аудиторію, щоб було достатньо місця для групи. Потім ми перевіряємо, чи є в цей час якісь класи в цій аудиторії, якщо ні, то присвоюємо клас, який ми обрали раніше зі списку класів, якщо так, шукаємо інші координати.

І так повторюється для кожного уроку/пари. Коли графік буде готовий, ми перевіряємо його на дійсність і записуємо його в розклад. Коли список розкладу заповнений, виходимо з циклу. Діаграму цього кроку можна побачити на рисунку 2.4.

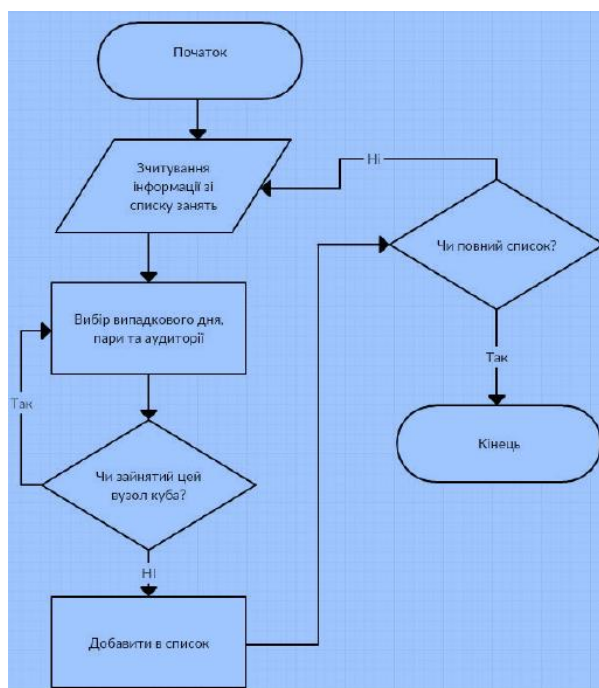


Рисунок 2.4 - Блок-схема формування першої генерації

## 2.5.2 Змішування нової генерації

Перший крок у створенні нового покоління - змішати два графіки навчального процесу, які минулі, щоб їхні нащадки мали риси обох батьків. Це буде реалізовано в програмі наступним чином. Спочатку формується сегмент, який ділиться на частини, розміри яких дорівнюють відсотку цільової функції кожного графіку до загальної величини всієї сукупності, як показано на рисунку 2.5. Це потрібно для того, щоб графік, який є більш значущим для функції, має більший шанс «відтворення» або більший шанс бути вибраним алгоритмом змішування.

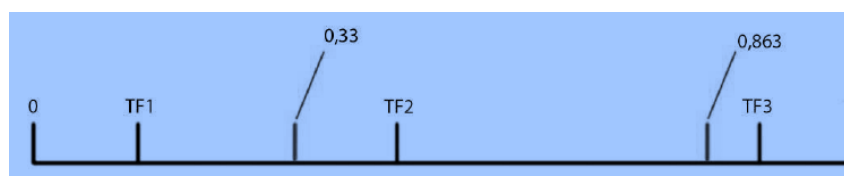


Рисунок 2.5 – Вибір «батьків»

Після минулого кроку дві координати вибираються випадковим чином на сегмент. Якщо координата потрапляє в інтервал, що відповідає конкретному графіку навчального процесу, то цей графік/розклад буде обраний для змішування.

Розміщується два графіки, які так само: спочатку або просто виберіть координати кубу ( $X_1, Y_1, Z_1$ ). Ми завжди перевіряємо, що у всіх позиціях немає дисципліни, але ми не дивимось далі. Як тільки ми знайшли дисципліну, шукаємо дисципліну в іншій «іншій» графіку навчального процесу з випадковими рівнями ( $X_2, Y_2, Z_2$ ).

Процес можна побачити на рисунку 2.6.



Рисунок 2.6 – Перший крок

Потім відбувається обмін, але з дисципліною і з координатами. Тобто дисципліну 1 з координатами  $(X1, Y1, Z1)$ , яку ми вибрали на першому кроці, переходимо до положення  $(X2, Y2, Z2)$ , а дисципліну 2 з координатами  $(X2, Y2, Z2)$ , рухаємося за координатами  $(X1, Y1, Z1)$ . Кінцевий результат змішування можна побачити на рисунку 2.7.

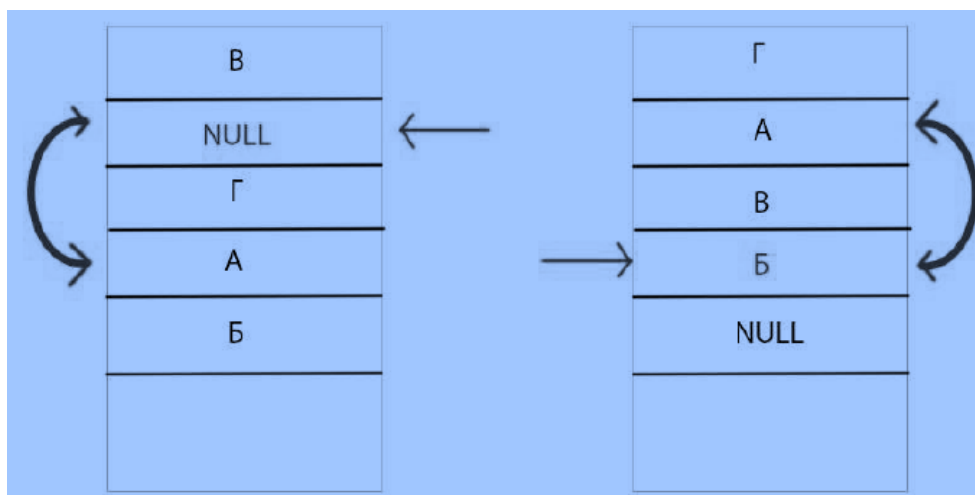


Рисунок 2 .7 – Результат

Як можна побачити на рисунку, багато дисциплін не змінилися. Це дуже важливий момент. Справа в тому, що якщо ми внесемо зміни до розкладу, не приймаючи ззовні нічого, тоді ми збережемо цілісність наукового плану, і суворі вимоги, що реалізуються через куб, будуть гарантовані.

Потім настає перевірка графіка навчального процесу, яка йде так. Спочатку ми робимо фрагмент розкладу для кожного вчителя та групи учнів. Потім ми перевіряємо цей розділ на наявність дублікатів, тобто, чи є один вчитель одночасно в одній аудиторії чи одна група має дві пари одночасно. Якщо обидва «дочірні» графіки перевірені, то записуємо їх у загальний список. Ми робимо це, поки кількість нових графіків не дорівнює кількості батьків [13].



Загальну схему створення нових потомків на рисунку 2.8.

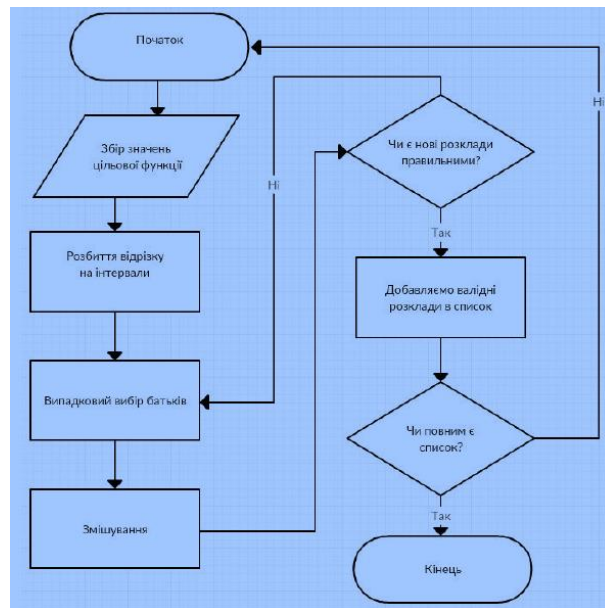


Рисунок 2.8 – Блок-схема формування потомства

### 2.5.3 Селекція та струс

Селекція – це процес, коли погані графіки скидаються, а хороші залишаються до наступного кроку алгоритму. Спочатку ми беремо всі графіки і сортуємо їх за значенням їх функцій. Після цього ми відхиляємо рівно половину графіків.

Струс – це процес штучного впровадження нового потомства для збагачення генофонду. Алгоритм повинен використовувати цей інструмент, коли функція не буде змінювати значення після багатьох кроків. Це відбувається подібно до першого покоління, після старих записуються лише нові графіки з подальшим змішуванням та відбором [13].

#### 2.5.4 Вихід із циклу

Після кожної ітерації різниця між найменшим значенням у списку генерації та найменшим буде обчислюватися, як показано у формулі 2.9.

$$\max_i \left| \max_j F_{ij} - \min_j F_{ij} \right| < \varepsilon \quad (2.9)$$

де  $F_{ij}$  –  $j$ -тий розклад на  $i$ -тій ітерації,

$\varepsilon$  – задана точність.

Точність алгоритму було встановлено на основі експериментальних досліджень. Але чисельна потужність системи, на якій працює програма, оскільки чим вища точність, тим довше буде рішення [14].

#### 2.6 Висновки

У цьому розділі розглядалось основні алгоритми та моделі, які будуть використані для виконання цього завдання. В якості моделі компонування обраний багатовимірний куб. Генетичним алгоритмом було обрано метод оптимізації. Також було розглянуто об'єктивну функцію та її основні компоненти, які враховуватимуть суб'єктивні вимоги. Розглянуто метод відбору та змішування потомства.

## 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ РОЗКЛАДУ ЗАНЯТЬ ТА ЇЇ ТЕСТУВАННЯ

### 3.1 Вибір середовища програмування

Для виконання даної задачі нам потрібно вибрати хороше середовище програмування, яке буде виконувати наші вимоги, і також мати невисоку ціну. Щодо останнього можна зауважити, що більшість сучасних середовищ уже є в open-source, а ті що не у open-source уже стали безплатні, тому про ціну не варто переживати.

Основні вимоги до середовища програмування для виконання даного проекту будуть:

- легкість написання коду;
- потужність мови програмування;
- швидкість компіляції;
- швидкість та ефективність роботи;
- наявність тех. підтримки;
- наявність бібліотек та інструментів, що можуть полегшити розробку;
- зручний інтерфейс та інструментарій для оптимізації на дебагінгу коду;
- легке підключення до бази даних;
- легкий функціонал для адміністрування бази.

Виходячи з цих вимог вибір був зупинений на пакеті MS Visual Studio 2015 Community з базою даних на основі MS SQL Server [10]. Мовою програмування буде C#. Причини такого вибору [6] :

- безплатна платформа;
- дуже потужний інструментарій завдяки бібліотекам та вбудованим класам;
- зручний редактор коду з виділенням синтаксису, перевірки синтаксису в реальному часі та підказкам, які допомагають виправляти помилки;
- зручний графічний редактор класів, який дозволяє автоматизовано генерувати потрібні зв'язки;

- можливість адмініструвати базу даних напряму з середовища;
- зручний конструктор інтерфейсу на основі WPF та XML;
- об'єктно-орієнтованість мови програмування допоможе реалізувати велику кількість проблем;
- автоматична система стирання зайвих даних допоможе рівномірно розділяти ресурси системи;
- легкість написання коду на мові C#;
- можливість компіляції для мультиплатформного користування.

Опис програмної реалізації та архітектура бази даних буде описана нижче. Вся реалізація була написана використовуючи лише інструменти MS VS 2015.

### **3.2 Архітектура бази даних**

Для того, щоб реалізувати схему, описану в розділі 2, нам потрібно створити базу даних, яка буде зберігати всі дані про установи, які нам потрібні для складання розкладу. На основі опису в таблиці 1 нам потрібно створити такі таблиці:

- а) таблиця аудиторій;
- б) таблиця викладачів;
- в) таблиця груп студентів;
- г) таблиця дисциплін.

Кожна з цих таблиць повинна містити інформацію про установу та встановлювати деякі зв'язки між собою. Таблиця аудиторії повинна вказувати номер аудиторії, її тип та кількість місць. Таблиця викладачів містить інформацію про ім'я вчителя, його науковий ступінь, яку дисципліну він викладає та які форми навчання він проводить.

У таблиці студентів зберігається основна інформація про групи, такі як номер групи, курс, кількість студентів та код групи. У таблиці курсу зберігаються назва предмета, група, в якій викладається дисципліна, та кількість лекцій, практик та лабораторних занять, які потрібні протягом тижня. На основі цього опису ми створимо таку базу даних на основі наступних таблиць: Дисципліна - таблиця

дисциплін, Вчитель - таблиця викладачів, Кімнати - таблиця аудиторій та Групи - таблиця студентів [15-16]

— Discipline

- Id\_disc – ID дисципліни;
- disc\_name = назва дисципліни;
- course – курс, на якому проводиться;
- group\_no – група, в якій проводиться;
- lectures – кількість лекцій в тиждень;
- practice – кількість пар в тиждень;
- labs – кількість лабораторних в тиждень.

— Teacher

- Pib – ім'я викладача;
- Degree – його ступінь;
- disc\_name – предмет, який він веде;
- lectures – дорівнює 1, якщо він проводить лекції;
- practice - дорівнює 1, якщо він проводить практичні;
- Labs - дорівнює 1, якщо він проводить лабораторні.

— Groups

- Id\_grp – ID групи;
- Group\_no – номер групи;
- Course – курс;
- Cnt – кількість студентів в групі.

— Rooms

- Id\_room – номер аудиторії;
- Type – тип аудиторії;
- Places – кількість місць.

Повне описання БД разом із посиланнями можна побачити на рисунку 3.1.

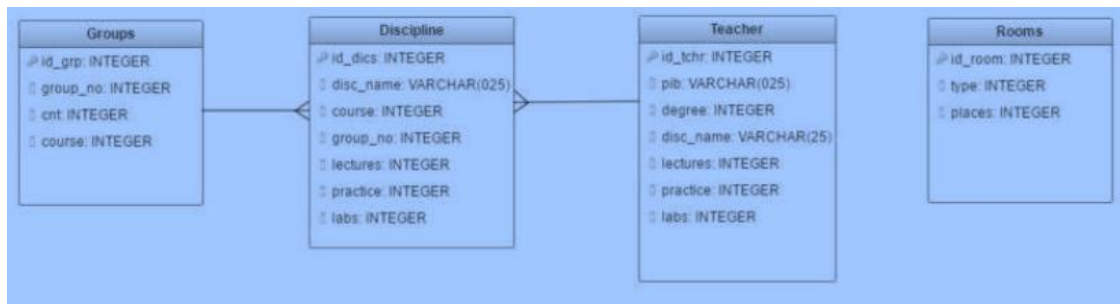


Рисунок 3.1 – Архітектура бази даних

Посилання між таблицями будуть наступними. Від вчителя до предмета один до багатьох, оскільки один викладач може керувати багатьма дисциплінами. А також один до багатьох від груп до предметів, тому що одна група має багато предметів.

### 3.3 Розробка опису програми на мові C#

Щоб реалізувати ці задачі ми створимо такі класи. В кожного будуть свої особливості і цілі.

#### 3.3.1 Клас Dictionary

Ці класи матимуть підкласи для виконання своїх завдань. Словник буде зберігати інформацію, яку буде отримано з бази даних. Тому кожне поле містить списки об'єктів, які зберігають інформацію про процес навчання та методи, які отримують його з бази даних.

Описання класу можна побачити на рисунку 3.2. Діаграму залежності можна побачити на рисунку 3.4. Ви можете побачити списки для кожної бази даних у полях атрибутів цього класу.

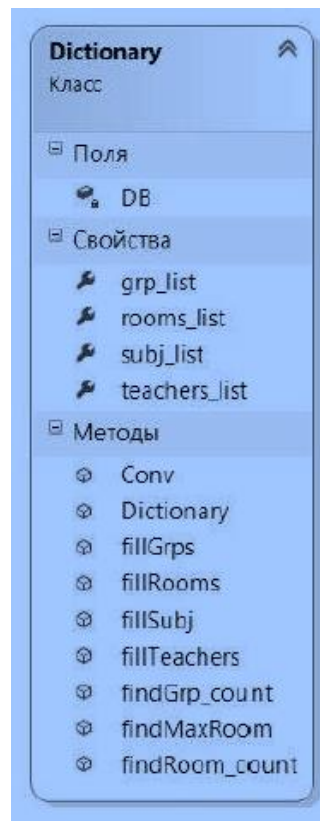


Рисунок 3.2 – Клас Dictionary

Діаграму класів проекту наведено на рисунку 3.3.

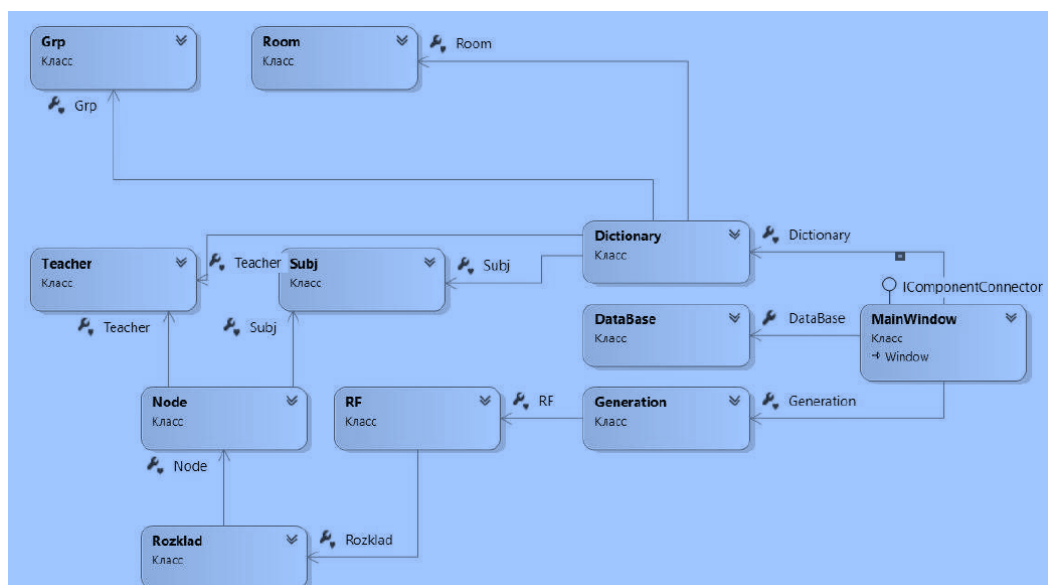


Рисунок 3.3 – Діаграма класів проекту

На рисунку 3.4 наведено діаграму залежностей класу Dictionary.

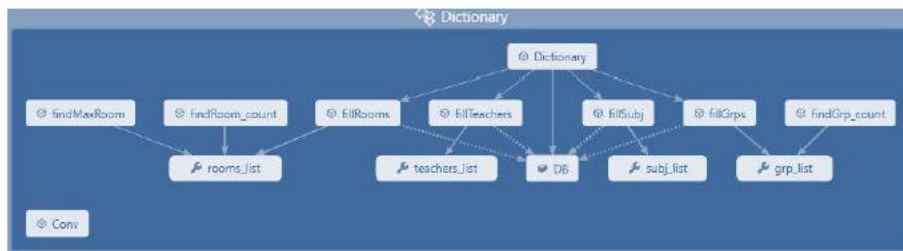


Рисунок 3.4 – Діаграма залежності класу Dictionary

Слід зазначити список елементів `subj_list`. Об'єкти в ньому не зберігаються, як у базі даних. Вони формуються в класі `Subj`, який зберігає кожну форму навчання для кожного предмета окремо і показаний на рисунку 3.5 як посилання в графіку. Наприклад, давайте предмет з математики, який слід робити в 21 групі. Має бути 2 лекції на тиждень та 1 практика, тому, виходячи з цього, у списку класів у нас буде 3 записи: 2 лекції та 1 практика з математики. Це полегшить нам подальше формування розкладу.

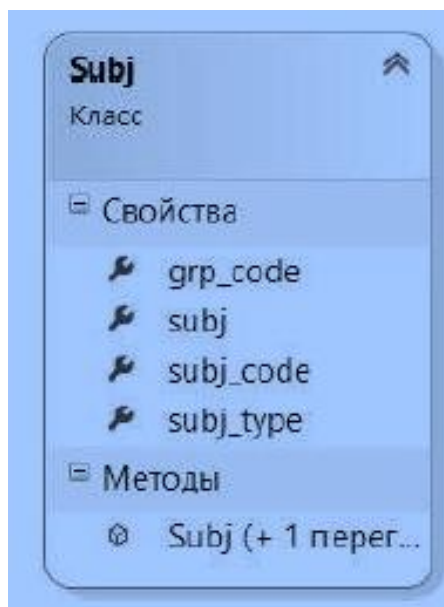


Рисунок 3.5 – Клас Subj



### 3.3.2 Клас Rozklad

Розклад зберігає куб, описаний у Розділі 2, який має розмір 6 на 6 для кількості аудиторій. Діаграма залежності показана на рисунку 3.6..

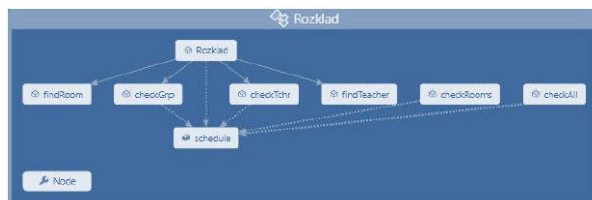


Рисунок 3.6 – Діаграма залежності класу Rozklad

Усі методи з префіксом check спочатку реалізують перевірку адекватності планування для кожного критерію окремо. І тоді вони викликаються з одного місця методом checkAll. Тому на закінчення можна сказати, що цей клас точно відображає суть завдання.

### 3.3.3 Клас Покоління - Generation

Цей клас зберігає всі списки поколінь, а також загальний список, який буде відсортований. Його опис можна побачити на рисунку 3.7.

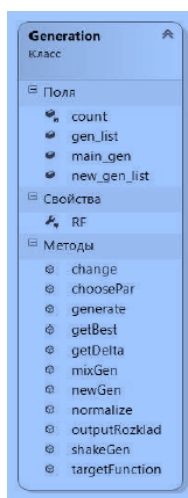


Рисунок 3.7 – Клас Generation

Опис його зв'язків з іншими класами можна на рисунку 3.8.

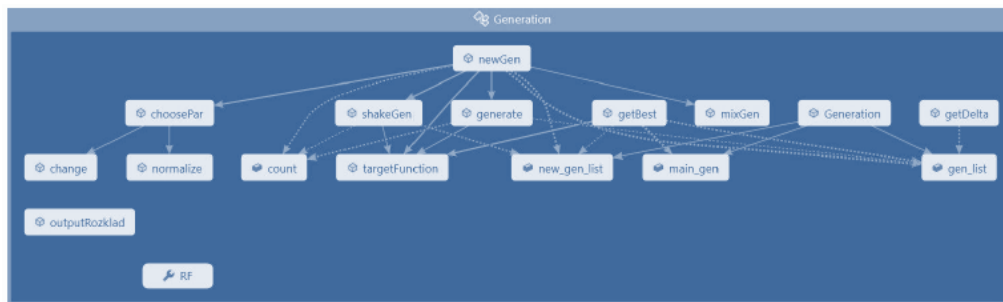


Рисунок 3.8 - Діаграма залежності класу Generation

У діаграмі чітко показані методи, необхідні для реалізації алгоритму. Поля `gen_list`, `new_gen_list` та `main_gen` відповідають списку первинного покоління, нового покоління та списку, який поєднує їх для сортування та відбору.

Методи `genege`, `mixGen`, `getBest`, `shakeGen` генерують, вибирають, еволюціонують та розхитують населення відповідно. Метод `targetFunction` підраховує цільову функцію. Методи зміни та нормалізації утворюють вектор цільової функції, необхідний для змішування.

Існує також клас обслуговування `DataBase`, який містить методи, що з'єднують базу даних з кодом. Він містить код запиту до бази даних. Кожен запит прописаний в окремому файлі та оптимізований перед його запуском у проекті.

Також варто відзначити клас `RF`. Його суть полягає у створенні унікальної структури, яка зберігатиме значення функції цільового планування разом із самим графіком, і буде перераховувати це значення щоразу, коли графік змінюється.

### 3.4 Розробка плану тестування розкладу

Для того щоб перевірити програму, нам потрібно проаналізувати цілі, яких хотіли досягти, будуючи графік. Виходячи з вимог, описаних раніше, слід сформулювати наступні вимоги до планування:

- зберігати всю необхідну інформацію;
- мати методи введення інформації та відображення результату обчислень;
- виконувати всі необхідні розрахунки;

– результатом є готовий графік занять для навчального закладу.

Тестування кожного елемента гарантує нам адекватність результату, його правильність та впевненість у правильній роботі алгоритму..

### 3.5 Розробка тестового додатку

Інтерфейс програми складається основного вікна, яке показано на рисунку 3.10.

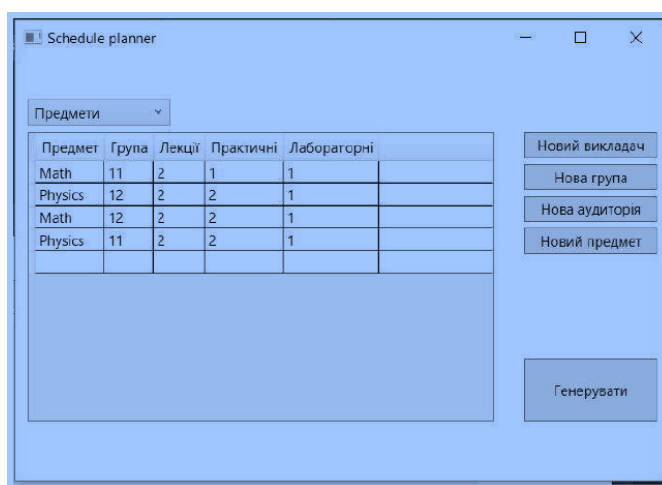


Рисунок 3.10 – Головне вікно

Вікно реалізує всі основні інструменти, необхідні для планування. Це вікно містить кнопки адміністрування баз даних, які дозволяють додавати нові дисципліни, викладачів, студентів та аудиторії, а також електронну таблицю, де всі ці дані можна переглянути до покоління.

При натисканні на всі кнопки введення з'являться окремі діалогові вікна, які показані на рисунку 3.11, де чітко пояснено, як користуватися цим інтерфейсом. При натисканні головна кнопка «Створити» запускає всі алгоритми, необхідні для обчислення, що створюють перше покоління, впроваджують еволюційний алгоритм та вибирають найкращі графіки.

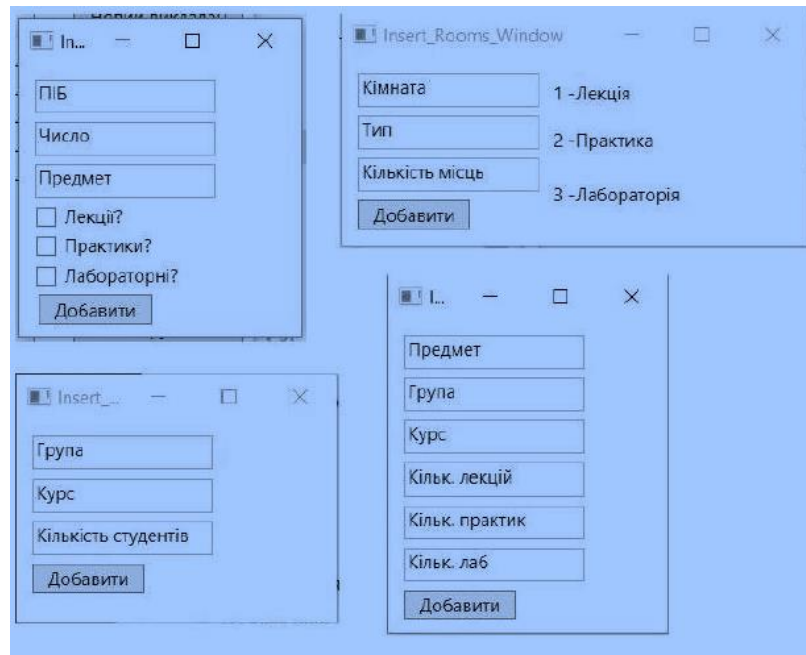


Рисунок 3.11 – Діалогові вікна введення даних

Потім, як графік буде успішно сформований, він буде завантажений у файл XLSX на диску в папці, що виконується. Це робиться для зручності при його публікації, а при необхідності редагування.

Папка, де будуть розміщені всі необхідні файли, виглядатиме як на рисунку 3.12. Він також показує файли запитів, файли баз даних та виконувані файли, а також вихідний файл розкладу.

Имя	Дата изменения	Тип	Размер
DB1.mdf	13.06.2016 0:20	Файл "MDF"	3 200 КБ
DB1_log.ldf	13.06.2016 0:20	Файл "LDF"	832 КБ
Diplom.exe	13.06.2016 0:13	Приложение	130 КБ
Diplom.exe.config	12.05.2016 19:05	XML Configuration...	1 КБ
Diplom.pdb	13.06.2016 0:13	Program Debug D...	252 КБ
Diplom.vshost.exe	13.06.2016 0:14	Приложение	23 КБ
Diplom.vshost.exe.config	12.05.2016 18:05	XML Configuration...	1 КБ
Rozklad.xlsx	13.06.2016 0:19	Лист Microsoft Ex...	9 КБ
SQL_groups.sql	25.05.2016 18:53	Файл "SQL"	1 КБ
SQL_rooms.sql	25.05.2016 21:18	Файл "SQL"	1 КБ
SQL_subjects.sql	25.05.2016 21:42	Файл "SQL"	1 КБ
SQL_teachers.sql	25.05.2016 18:49	Файл "SQL"	1 КБ

Рисунок 3.12 – Тека з файлами проекту

Файл XLSX виводиться у такому форматі: стовпці ділять графік на групи так, щоб кожен стовпець відповідав конкретній групі, рядки поділялися на 6 груп відповідно до кожного дня тижня та кожного дня тижня ділиться на 6 пар. Таким чином, єдиним результатом для кожної групи є окремий графік. Також можливо показати графік для кожного вчителя, щоб полегшити розповсюдження інформації та використання цього додатка. Тоді замість групи учнів прізвища викладачів відобразатимуться у стовпцях. Виведення графіку на основі даних випробувань показано на рисунку 3.13.

Слід також зазначити, що вимоги викладачів та учнів не можуть зберігатися в програмі після складання, і вони повинні бути записані перед складанням. Це пов'язано з тим, що для реалізації такої функціональності потрібно набагато більше часу та ресурсів, ніж було доступно під час цієї роботи.

На рисунку 3.14 підсумований графік.

	A	B	C
1		Група 11	Група 12
17	Середа 4 пара		Math практика Bohanov 102
18	Середа 5 пара		
19	Середа 6 пара		
20	Четвер 1 пара	Math лабораторна Statkevics 203	Physics практика Kalita 102
21	Четвер 2 пара		
22	Четвер 3 пара		
23	Четвер 4 пара	Math лекція Bohanov 101	
24	Четвер 5 пара		Math лекція Bohanov 101
25	Четвер 6 пара		Physics лекція Kalita 101
		Math практика	

Рисунок 3.13 – Excel файл з розкладом

### 3.6 Виконання тестування додатку на реальних даних

Для перевірки програми було внесено до бази даних додатків. Були представлені відомі вчителі, аудиторії та студентські колективи. Було розглянуто 2 курс семестру 4 потоку СІ-2Х.

Ви можете переглянути вихід через інтерфейс на рисунках 3.14-3.17.

На рисунку 3.14 узагальнені вихідні дані вчителя.

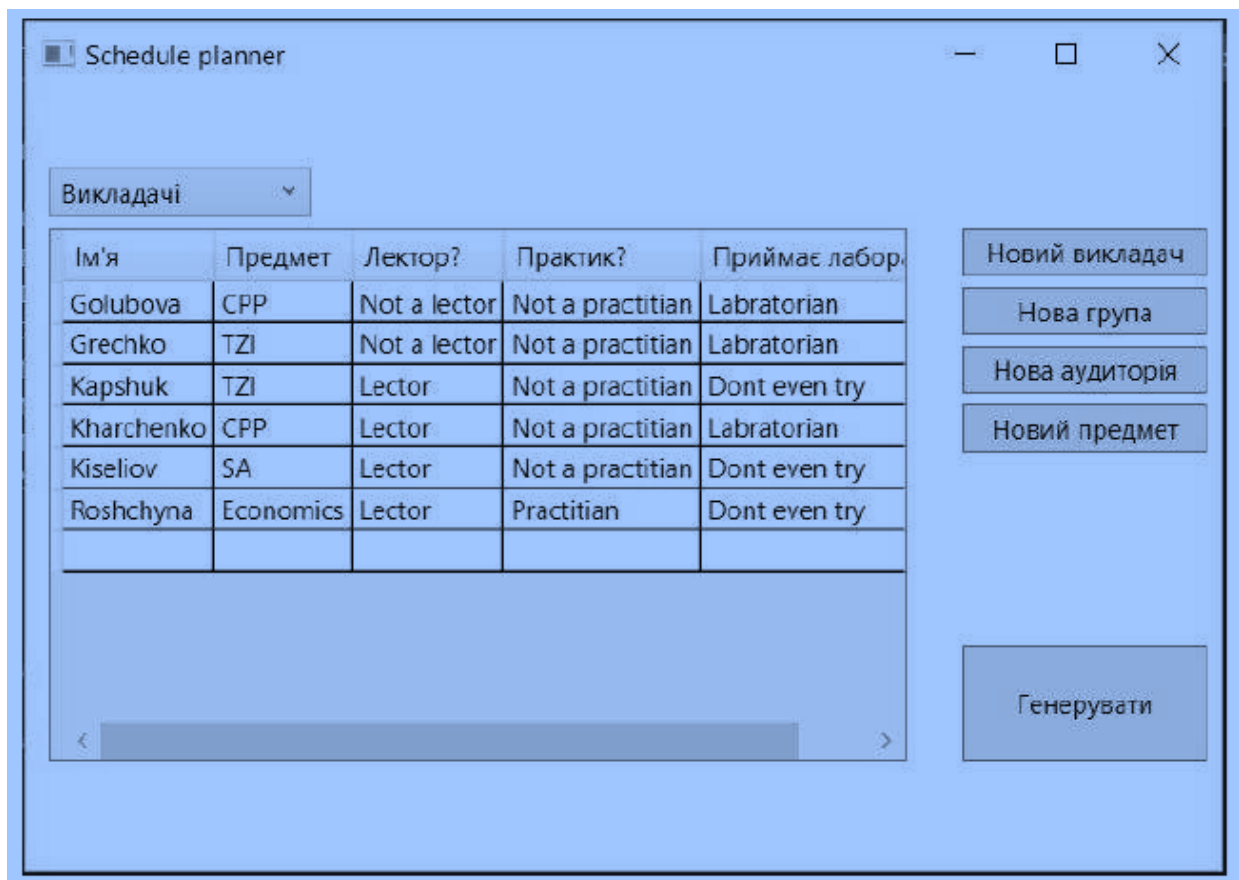


Рисунок 3.14 – Вихідні дані про викладачів

На рисунку 3.15 наведено вихідні дані про студентів.

Студенти

Курс	Група	Кількість	Код групи
4	1	29	41
4	2	30	42
0	2	58	2

Новий викладач  
Нова група  
Нова аудиторія  
Новий предмет

Генерувати

Рисунок 3.15 – Вихідні дані про студентів

На рисунку 3.16 наведено вихідні дані про аудиторії.

Аудиторії

Аудиторія	Тип	Кільсть місць
101	Lab	35
102	Lab	35
103	Practice	35
203	Lab	35
206	Lab	40
303	Lecture	60
304	Lecture	60
307	Lecture	60
309	Practice	60
310	Practice	60
312	Lecture	60

Новий викладач  
Нова група  
Нова аудиторія  
Новий предмет

Генерувати

Рисунок 3.16 – Вихідні дані про аудиторії

На рисунку 3.17 наведено вихідні дані про навчальний план на тиждень.

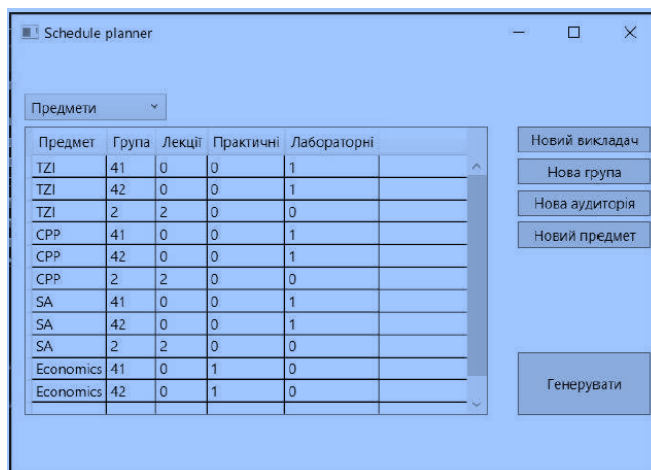


Рисунок 3.17 – Вихідні дані про навчальний план на тиждень

Згенерований таким чином розклад можна побачити на рисунку 3.18.

	Понеділок						Вівторок						
	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	
Група 21	SA класів Київської 204	Економіка Валентина Рибальська 181	TZ лабораторна Басілка 307	TZ класів Київської 204	OPP лабораторна Київська 206			OPP класів Київська 303	SA класів Київської 204	Економіка практична Київська 203			
Група 22							SA лабораторна Київська 206						
	Середа						Четвер						
	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	
Група 21	Воєнна підготовка								TZ лабораторна Басілка 307				
Група 22												OPP лабораторна Київська 102	
	П'ятниця						Субота						
	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	
Група 21		OPP класів Київська 303					SA лабораторна Київська 102						
Група 22								TZ класів Київська 307					

Рисунок 3.18 – Згенерований програмою графік навчального процесу

Цей розклад відповідає всім суворим вимогам: жодних дублікатів, жодного викладача чи групи учнів не знаходяться в двох місцях одночасно. І відповідні аудиторії вибираються залежно від кількості учнів та типу класу.

Час обчислення становить 2,356742412 секунд у 7 кроків.

З точки зору нежорстких вимог, графік відмінно відповідає цим потребам, наприклад, вихідний день протягом тижня, відсутність вікон, невелика кількість перших пар, відсутність вікон у вчителів, пари в основному зосереджені на 2-3 парах і мінімум суботніх пар.



Виконання цих вимог свідчить про те, що генерація розкладу пройшла успішно. Також розглянемо графік цільової функції проти кількості кроків, як показано на рисунку 3.19.

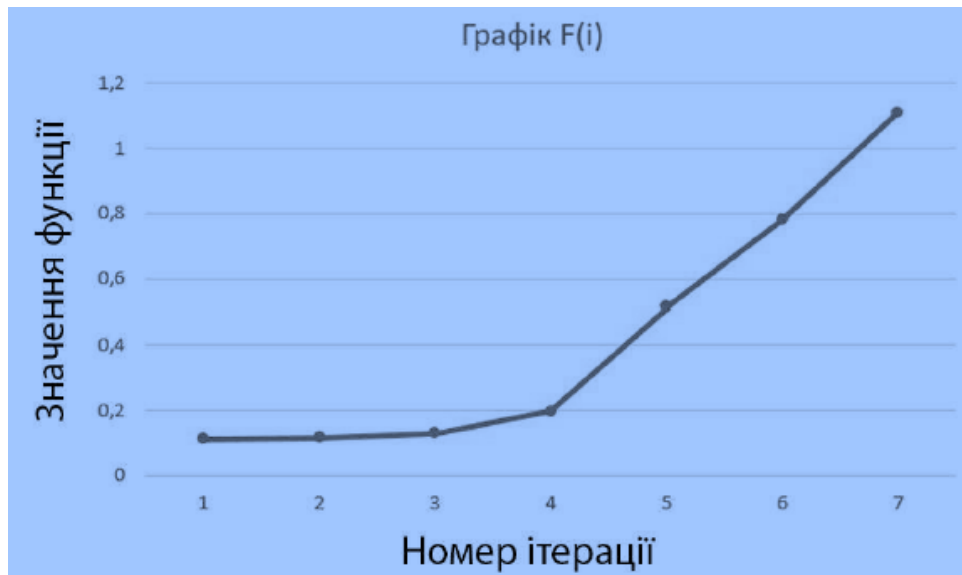


Рисунок 3.19 – Залежність середнього значення функції по всій генерації від ітерації

Нам також потрібно розглянути графік, що показує різницю між мінімумом та максимумом цільової функції на ітерації, як у формулі 10. Цей графік показаний на рисунку 3.20.



Рисунок 3.20 – Залежність різниці між мінімумом та максимумом цільової функції від ітерації

### 3.7 Висновки

Переглянувши отримані результати, можна зробити наступні висновки. Програма створює графік, який гарантовано відповідає суворим вимогам. Ніякі фізичні умови, ні логічні не порушуються під час покоління. Отриманий графік відповідає частково вимогам викладачів та учнів лише частково, але достатньо, щоб рівень ефективності був задовільним. Це важливий факт, тому що неможливо задовольнити всіх одночасно. Вимоги вчителів відповідали пріоритетно, а вимоги вчителів вищого рангу ще краще відповідали

З точки зору часу, алгоритму знадобилося дуже мало часу для створення гірського графіку роботи. Цей час компенсує дні та навіть тижні роботи оператора факультету, якщо він не враховує інформацію, що вводиться оператором лише один раз. Такі дані, як інформація про аудиторію, вчителів, групи, часто не можуть змінюватися. І плани тренувань змінюються лише незначно, але не сильно.

Середнє значення збільшується з кожною ітерацією, що є відмінним доказом алгоритму. Але значення різниці містить стрибок. Це пов'язано насамперед з тим, що в певний момент часу програма випадково сформувала графік, який мав дуже велике значення цільової функції, і це значення було набагато більше, ніж інші у своєму поколінні. Однак після пізніших ітерацій ця різниця впала через змішування цього «альфа» графіка з іншими в ітерації.

Результуючи все сказане вище, можна зробити висновок, що генерація пройшла успішно і розклад пройшов тестування.

## 4 ЕКОНОМІЧНА ЧАСТИНА

### 4.1 Оцінювання комерційного потенціалу розробки (технологічний аудит розробки)

Роль та цінність економічного обґрунтування в останні роки істотно зросла. Під час цієї стадії інноваційного процесу виконується перехід від економічної моделі з нечіткими вимогами до більш розвиненої структури з визначеними економічними мотивами проекту. Вказаний етап є дуже важливим, оскільки його недооцінка може призвести до таких негативних наслідків як неефективність розроблюваної системи або нестача ресурсів при її розробці. Саме для передбачення вказаних проблем застосовується економічне обґрунтування прийнятих технічних рішень, яке перевіряє чи є розробка інформаційної системи економічно доцільною.

Метою економічної частини магістерської кваліфікаційної роботи є оцінювання комерційного потенціалу розробки інформаційної системи автоматизованого формування розкладу, яка передбачає виконання наступних задач:

- визначити комерційний потенціал розроблюваної системи;
- передбачити витрати на виконання розробки та подальшу її реалізацію;
- визначити комерційний ефект від впровадження результатів наукової роботи;
- обчислити ефективність вкладених інвестицій та періоду їх окупності.

У даній роботі проводилась розробка інформаційної системи автоматизованого формування розкладу.

Для проведення технологічного аудиту було залучено двох незалежних експертів: керівник магістерської кваліфікаційної роботи – к.т.н., Крижановський Євген Миколайович та студент групи ІСТ-18м ВНТУ: Федюк О.П. Кожен з експертів був попередньо ознайомлений з розробкою інформаційної системи, та заповнив таблицю з відповідними критеріями для оцінки комерційного потенціалу наукової роботи за 5-ти бальною шкалою. Після цього було підраховано

середньоарифметичну суму балів та визначено рівень комерційного потенціалу вказаної розробки.

Оцінювання комерційного потенціалу розробки здійснено за 12-ма критеріями, наведеними в таблиці А.1 Додатку А.

Результати оцінювання комерційного потенціалу розробки занесені до таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розроблюваної інформаційної системи

Критерії	Прізвище та ініціали експерта	
	Крижановський Є. М.	Федюк О.П.
	Бали, виставлені експертами:	
1	4	4
2	3	3
3	4	3
4	3	3
5	3	3
6	4	3
7	3	3
8	4	3
9	3	3
10	3	4
11	3	3
12	4	4
Сума балів	$\overline{СБ} = 41$	$\overline{СБ} = 39$
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_i \overline{СБ}_i}{i} = \frac{41+39}{2} = 40$	

За даними, які представлені у таблиці 4.1, можна зробити висновок щодо рівня комерційного потенціалу розроблюваної інформаційної системи, використовуючи рекомендації, які наведені в таблиці 4.2.

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\bar{СБ}$ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Отже, за результатами, занесеними до таблиці 4.1, видно, що середньоарифметична сума балів дорівнює 40, тобто нова розробка має рівень комерційного потенціалу вище середнього.

Характеризуємо рівень комерційного потенціалу розробки:

- можливий шлях реалізації розробки: інформаційну систему впроваджено в департамент освіти;

- загальна якість розробки: головні результати, які були одержані в процесі вирішення встановлених задач становлять наукову новизну дослідження. Результати полягають у наступному: удосконалено інформаційну технологію розробки інформаційної системи створення розкладу з можливістю дуальної освіти, яка, на відміну від існуючих, передбачає можливість роботи з даними як за допомогою ліцензійного, так і за допомогою безкоштовного програмного забезпечення;

- перспективність розробки: на даний час в багатьох сферах управління використовуються інформаційні технології. Для управління даними, які містять також дані вчителів і керівників підприємств, на сучасному етапі використовують геоінформаційні системи. Управління навчальною і виробничою діяльністю в університетах є одним з напрямків освітньої політики;

- конкурентоспроможність розробки: на даний момент чимала кількість інформації знаходиться на паперових носіях або в електронних таблицях, що вимагає чималих затрат сил і часу на пошук та аналіз необхідної інформації. Для створення зручного навчального процесу необхідно використовувати сучасні технології.

Актуальність розробки полягає у можливості використання розробки системи в органах державного управління та громадських органах з використанням як ліцензійного так і офіційно безкоштовного програмного забезпечення, так і на підприємствах.

У таблиці 4.3 визначені основні технічні показники аналога і нової інформаційної системи автоматизованого формування розкладу.

Таблиця 4.3 - Основні технічні показники аналога і нової інформаційної системи автоматизованого формування розкладу

Функції	Аналог	Нова розробка
Інформаційна система	Відсутня	Наявна
База даних	Відсутня	Наявна
Архітектура системи	Відсутня	Наявна
Інтерфейс користувача	Відсутня	Наявна
Веб-ресурс	Відсутня	Відсутня
Реалізація електронного розкладу	Наявна	Наявна

Успішна реалізація вказаної інформаційної системи вимагає розробку регламентних документів та отримання значної кількості дозвільних документів на створення і впровадження нового продукту у систему освіти.

Вказана інформаційна система може бути реалізована на сайті вищого навчального закладу.

## 4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько-технологічної роботи

Щоб спрогнозувати витрати на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи потрібно виконати наступні задачі:

- 1-й етап: розрахувати витрати на виконавців даного розділу роботи.
- 2-й етап: визначити загальні витрати на виконання даної роботи.
- 3-й етап: спрогнозувати загальних витрат на реалізацію та впровадження результатів даної роботи.

Розрахунок витрат на розробку методики дослідження здійснюється за наступними статтями калькуляції:

- основна заробітна плата виконавців;
- додаткова заробітна плата виконавців;
- нарахування на заробітну плату виконавців;
- амортизація персонального комп'ютера;
- витрати на матеріали, що були задіяні під час розробки інформаційної системи;
- витрати на електроенергію;
- інші витрати.

У розробці даного програмного продукту брав участь один спеціаліст та його науковий керівник.

Основна заробітна плата для спеціаліста визначається за формулою (4.1):

$$Z_o = \frac{M}{T_p} \cdot t \text{ [грн]}, \quad (4.1)$$

де  $M$  – місячний посадовий оклад конкретного розробника, наукового керівника, грн.;

$T_p$  – кількість робочих днів в місяці;  $T_p = 22$  дні;

$t$  – число днів роботи: розробника – 3 місяці по 22 дні ( $t = 66$  днів); наукового керівника – 3 дні ( $t = 3$  дні).

Розрахунки основної заробітної плати зводимо до таблиці 4.4.

Таблиця 4.4 – Розрахунки основної заробітної плати

Найменування посади виконавця	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на оплату праці, грн.
Розробник	7500	340,91	22	22500
Науковий керівник	9000	500	3	1500
Всього				24000

Додаткова заробітна плата  $Z_d$  розробника становить 10% від основної заробітної плати:

$$Z_d = 0,10 \cdot 24000 = 2400 \text{ (грн).}$$

Нарахування на заробітну плату  $H_{зп}$  розробника розраховується за наступною формулою:

$$H_{зп} = (Z_o + Z_d) \cdot \frac{\beta}{100} \text{ [грн]}, \quad (4.2)$$

де  $Z_o$  – основна заробітна плата розробника;

$Z_d$  – додаткова заробітна плата розробника;

$\beta$  – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування становить 22%.

$$H_{зп} = (24000 + 2400) \cdot 0,22 = 5808 \text{ (грн).}$$



Амортизація обладнання, комп'ютерів та приміщень, які були використані для виконання даного етапу роботи розраховуємо за формулою:

$$A = \frac{Ц \cdot T}{12 \cdot T_B} \text{ [грн]}, \quad (4.3)$$

де Ц – загальна балансова вартість обладнання, приміщення тощо, грн;

T – фактична тривалість використання, міс;

T<sub>B</sub> – термін використання обладнання, приміщень тощо, роки.

Розробка вказаної інформаційної системи проводилася протягом 3 місяців.

Відповідні розрахунки були зведені до таблиці 4.5.

Таблиця 4.5 – Амортизаційні відрахування

Найменування	Балансова вартість, грн	Норма амортизації, %	Термін використання, міс	Величина амортизаційних відрахувань, грн.
Офісне приміщення	15000	10	3	375
Ноутбук	7000	10	3	175
Всього				550

Витрати на канцтовари - 200 грн.

Під час розробки програмного продукту використовувались лише безкоштовні програмні засоби.

Витрати на енергію визначаємо за наступною формулою:

$$V_e = V \cdot П \cdot \Phi \cdot K_{II} \text{ [грн]}, \quad (4.4)$$

де V – вартість 1кВт електроенергії;

П – установлена потужність обладнання, кВт;

Φ – фактична кількість годин роботи комп'ютера при створенні програмного продукту, годин;

$K_{\Pi}$  – коефіцієнт використання потужності .

$$V_e = 2.00 \cdot 0,3 \cdot 660 \cdot 0,3 = 118,80 \text{ (грн)}.$$

Інші витрати  $V_{\text{ін}}$  складаються з витрат на управління організацією, оплати службових відряджень, витрат на утримання, ремонт та експлуатацію основних засобів, витрат на опалення, освітлення, водопостачання, охорону праці тощо. Інші витрати будуть становити 100% від суми основної заробітної плати розробника:

$$V_{\text{ін}} = 24000 \cdot 1 = 24000 \text{ (грн)}.$$

Отже, сума усіх вказаних вище витрат ( $V$ ) на реалізацію даного етапу роботи становить:

$$V = Z_o + Z_d + H_{\text{зп}} + A + V_{\text{мат}} + V_e + V_{\text{ін}} \text{ [грн]},$$

$$V = 24000 + 2400 + 5808 + 550 + 200 + 118,80 + 24000 = 57076,8 \text{ (грн)}.$$

2-й етап. Розрахунок загальних витрат на виконання даної роботи. Загальну вартість всієї наукової роботи ( $V_{\text{заг}}$ ) можна знайти за наступною формулою:

$$V_{\text{заг}} = \frac{V}{\alpha} \text{ [грн]}, \quad (4.5)$$

де  $\alpha$  – частка витрат, які здійснюються самим виконавцем даного етапу роботи, у відносних одиницях.

Так, як робота здійснюється однією людиною, то частка витрат рівна 1.

$$V_{\text{заг}} = 57076,8 \text{ (грн)}.$$

3-й етап. Прогнозування загальних витрат на реалізацію та впровадження результатів виконаної роботи. Вказане прогнозування загальних витрат здійснюється за наступною формулою:

$$ЗВ = \frac{V_{\text{заг}}}{\beta} \text{ [грн]}, \quad (4.6)$$

де  $\beta$  – коефіцієнт, який характеризує стадію реалізації даної роботи. Так, якщо розробка знаходиться:

- на стадії науково-дослідних робіт, то  $\beta \approx 0,1$ ;
- на стадії технічного проектування, то  $\beta \approx 0,2$ ;
- на стадії розробки конструкторської документації, то  $\beta \approx 0,3$ ;
- на стадії розробки технологій, то  $\beta \approx 0,4$ ;
- на стадії розробки дослідного зразка, то  $\beta \approx 0,5$ ;
- на стадії розробки промислового зразка,  $\beta \approx 0,7$ ;
- на стадії впровадження, то  $\beta \approx 0,9$ .

Отже, для отримання показника загальних витрат підставимо дані в формулу:

$$ЗВ = \frac{57076,8}{0,9} = 63418,67 \text{ (грн).}$$

#### **4.3 Прогнозування комерційних ефектів від реалізації результатів розробки**

На виконання усіх необхідних робіт необхідно затратити 66 робочих днів. Дана розробка вважається економічно вигідною, тому термін її окупності становитиме один рік.

В умовах ринку єдиним позитивним результатом для підприємства вважається збільшення чистого прибутку після впровадження вказаної розробки. Зростання чистого прибутку означає для організації надходження додаткових коштів, які можуть бути використані для покращення фінансових результатів діяльності. Для даного засобу його можна оцінити у сьогоденній вартості грошей.

Збільшення чистого прибутку підприємства  $\Delta\Pi_i$  для кожного із років, від яких очікують отримання позитивних результатів від впровадження розробки, обчислюється за наступною формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta C_0 \cdot N + C_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right) [\text{грн}], \quad (4.7)$$

де  $\Delta C_0$  – покращення головного оціночного показника внаслідок впровадження результатів розробки у поточному році;

$N$  – основний кількісний показник, що визначає діяльність підприємства у поточному році до впровадження результатів розробки;

$\Delta N$  – покращення головного кількісного показника діяльності підприємства внаслідок впровадження результатів розробки;

$C_0$  – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

$n$  – кількість років, протягом яких очікується отримання позитивних результатів внаслідок впровадження розробки;

$\lambda$  – коефіцієнт, що враховує сплату податку на додану вартість. З 2014 року коефіцієнт  $\lambda = 0,8547$ ;

$\rho$  – коефіцієнт, що враховує рентабельність продукції. Знаходиться у межах від 0,2 до 0,3;

$v$  – ставка податку на прибуток. У 2019 року становить 18%.

Допустимо, що під час впровадження результатів наукової розробки стає кращою якість певного продукту, що дозволяє збільшити ціну його реалізації на 2000 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 5 шт., протягом другого року – ще на 10 шт., протягом третього року – ще 15 шт.

Реалізація продукції до впровадження результатів наукової розробки становила 2 шт., а її ціна – 12000 грн.

Визначимо можливе підвищення чистого прибутку після впровадження результатів наукової розробки у кожному наступному році після базового.

Таким чином, підвищення чистого прибутку  $\Delta\Pi_i$  протягом першого року становитиме:

$$\Delta\Pi_1 = (2000 \cdot 2 + (12000 + 2000) \cdot 5) \cdot 0,8547 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 15558,96 \text{ (грн).}$$

Протягом другого року:

$$\Delta\Pi_2 = (2000 \cdot 2 + (12000 + 2000) \cdot (5 + 10)) \cdot 0,8547 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 44994,83 \text{ (грн).}$$

Протягом третього року:

$$\Delta\Pi_3 = (2000 \cdot 2 + (12000 + 2000) \cdot (5 + 10 + 15)) \cdot 0,8547 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 89148,63 \text{ (грн).}$$

Таким чином, вказані вище обчислення показують, що комерційний ефект від впровадження розробки виражається у значному підвищенні чистого прибутку організації.

#### **4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності**

Основними Комерційний ефект від можливого впровадження розробки ще не означає, що вона реально буде впроваджена. Якщо збільшення можливого прибутку від впровадження результатів наукової розробки є вигідним для організації, то це ще

не означає, що інвестор погодиться фінансувати дану розробку, не виставляючи жодних своїх вимог.

Головними показниками для визначення доцільності фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та період їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає виконання наступних задач:

1-й крок. Розрахунок теперішньої вартості інвестицій  $PV$ , що вкладаються в наукову розробку. Такою вартістю можна вважати прогнозовану величину загальних витрат  $ZB$  на реалізацію та впровадження результатів НДДКР, яка була розрахована раніше. Тобто,  $ZB = PV = 63418,67$  грн.

2-й крок. Розрахунок очікуваного збільшення прибутку  $\Delta\Pi_i$ , який отримає організація внаслідок впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку було визначене у попередньому розділі і становить:  $\Delta\Pi_1=15558,96$  грн;  $\Delta\Pi_2=44994,83$  грн;  $\Delta\Pi_3=89148,63$  грн

3-й крок. Для спрощення подальших розрахунків необхідно побудувати вісь часу, яка буде містити дані про всі платежі, а саме про інвестиції та прибутки, що будуть отримані під час виконання науково-дослідної роботи та впровадження її результатів.

Рисунок, який відображає рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рисунку 4.1.



Рисунок 4.1 – Вісь часу з зазначенням платежів, що мають місце в процесі розробки та впровадження результатів НДДКР

4-й крок. Розрахунок абсолютної ефективності вкладених інвестицій  $E_{абс}$  за відповідною формулою:

$$E_{абс} = (ПП - PV), \quad (4.8)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає організація від впровадження результатів наукової розробки, грн;

PV – теперішня вартість інвестицій  $PV = ЗВ$ , грн.

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^{\tau} \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.9)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

$t$  – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

$t$  – період часу (в роках) від моменту отримання чистого прибутку до точки „0”;

$$\text{ПП} = \frac{15558,96}{(1 + 0,1)^1} + \frac{44994,83}{(1 + 0,1)^2} + \frac{89148,63}{(1 + 0,1)^3} = 107553,63 \text{ (грн)}.$$

$$E_{\text{abc}} = 107553,63 - 63418,67 = 44134,96 \text{ (грн)}.$$

Оскільки  $E_{\text{abc}} > 0$ , результат від проведення наукових досліджень щодо розробки інформаційної системи та її впровадження принесе прибуток, тобто є доцільним, але це ще не означає те, що інвестор буде зацікавлений у вкладання власних коштів у реалізацію даної наукової розробки.

5-й крок. Розрахунок відносної (щорічної) ефективності вкладених в наукову розробку інвестицій  $E_{\text{в}}$  за формулою:

$$E_{\text{в}} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{abc}}}{PV}} - 1, \quad (4.10)$$

де  $E_{\text{abc}}$  – абсолютна ефективність вкладених інвестицій, грн;

$PV$  – теперішня вартість інвестицій  $PV = 3B$ , грн;

$T_{\text{ж}}$  – життєвий цикл наукової розробки, роки.

$$E_{\text{в}} = \sqrt[3]{1 + \frac{44134,96}{63418,67}} - 1 = 0,193 \text{ або } 19.3\%$$

Порівняємо  $E_{\text{в}}$  з мінімальною (бар'єрною) ставкою дисконтування  $\tau_{\text{мін}}$ , яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть.

Спрогнозуємо величину  $\tau_{\text{мін}}$ . У загальному вигляді мінімальна (бар'єрна) ставка дисконтування  $\tau_{\text{мін}}$  визначається за формулою:



$$\tau = d + f, \quad (4.11)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках;  $d = 0,136$ ;

$f$  – показник, що характеризує ризикованість вкладень; зазвичай, величина  $f = 0,05$ .

$$\tau = 0,136 + 0,05 = 0,141$$

Оскільки  $E_b = 19.3\% > \tau_{\min} = 14.1\%$ , то у інвестора може бути потенційна зацікавленість у фінансуванні даної наукової розробки..

*б-й крок.* Обчислення періоду окупності вкладених у реалізацію наукового проекту інвестицій  $T_{ок}$  за формулою:

$$T_{ок} = \frac{1}{E_g} [\text{року}]. \quad (4.12)$$

$$T_{ок} = \frac{1}{0,193} = 5,18 \text{ (року)}.$$

Оскільки термін окупності вкладених у реалізацію наукового проекту інвестицій менше трьох років ( $T_{ок} \leq 5$  років), то фінансування даної нової розробки в принципі є доцільним, але необхідно провести додаткові розрахунки.

#### 4.5 Висновки

В даному розділі було виконано оцінювання комерційного потенціалу розробки .

Проведено технологічний аудит з залученням двох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки є високим.

Аналіз комерційного потенціалу розробки показав, що інформаційна система автоматизованого формування розкладу за своїми характеристиками не має аналогів і є перспективною розробкою. Вказана система має кращі функціональні показники, підтверджуючи конкурентоспроможність товару на ринку. Існуючі переваги нової розробки дозволять швидко її поширити та популяризувати.

Згідно із розрахунками всіх показників витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальні витрати на розробку становлять 63418,67 грн.

Розрахована абсолютна ефективність вкладених інвестицій в сумі 44134,96 грн свідчить про отримання прибутку інвестором від комерціалізації програмного продукту.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 19.1 %, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 14.1%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності вкладених у реалізацію проекту інвестицій становить приблизно 5 років, що свідчить про доцільність фінансування нової розробки, але необхідність проведення додаткових розрахунків.

## ВИСНОВКИ

В ході виконання даної роботи було досліджено предметну область складання розкладів та розроблено алгоритм формування розкладу навчального і часткового робочого процесу. Розглянуто проблематику, актуальність даної теми і її використання в наш час.

Було розглянуто різні варіанти вирішення поставленої задачі і вибрано остаточний алгоритм, за яким буде формуватися розклад, а саме еволюційний алгоритм, на основі цільової функції, в якості фітнес функції оцінки якості розкладу. Було розроблено саму цільову функцію, яка відображає побажання як і викладачів так і студентів, враховує пріоритети та побажання. Також було розроблено кроки генетичного алгоритму, які повинні вирішити задачу. Було розроблено та реалізовано математичну модель розкладу, яка близько відображає суть фізичного розпорядку роботи навчального закладу.

Було вибрано середовище програмування MS VS 2015 та мова C# для реалізації цього завдання. Після чого було написано готовий додаток з графічним інтерфейсом, який дозволяє адміністратору зі зручністю користуватись базою даних та адмініструвати її. База даних була організована завдяки MS SQL Server, який зберігає в собі всі важливі для обрахунків дані. Архітектура бази даних добре відображає зв'язки між різними даними про навчальний процес і зручно зберігає інформацію в розроблених таблицях. Було реалізовано всі розроблені методи на сові програмування C# і SQL. Повна архітектура повністю задовольняє всім вимогам алгоритму, реалізовує всі структури та гарантує виконання поставлених задач. Також ця реалізація є оптимізовано та швидкою. Надійність гарантується методами, які перевіряються розклад на адекватність. А збіжність цільової функції гарантує сам алгоритм. І зібрано їх всіх в один додаток

Було перевірено розроблену програму на реальних даних та отримані хороші результати. Виконання жорстких умов було виконано, оптимізація розкладу та приближення до виконання всіх нежорстких умов було досягнуто. Звісно ж, отриманий з першого експерименту, розклад не відповідає всім вимогам здорового

глузду, проте час потрачений на його побудову цей недолік компенсує. А той факт, що після його генерації оператор, може вручну змінити невідповідності призводить до швидкого і якісного результату.

Судячи з результатів дана реалізація має сильний потенціал до розширення. Кількість дисциплін, викладачів, груп, аудиторів можна розширити з масштабу одного потоку до масштабу університету. Вимоги можна формувати для різних інстанцій. Якщо в дану реалізацію імплементувати систему розпізнавання тексту, додаток зможе сам формувати вимоги без участі людини.

Отримані дані про цільову функцію показують, що алгоритм працює правильно і якісно, проте з одною проблемою. Вона полягає у випадковій природі даного алгоритму. Практично всі методи містять якусь генерація псевдовипадкових чисел, що залишає можливість до зациклення, чи помилок. Проте дана проблемам була вирішена введенням струсу, який розбавляє генофонд та запобігає застою.

Під час експерименту було вибрано розмір генерації 8 розкладів. При збільшенні цього числа, час виконання буде рости, проте якість буде зростати, тому при правильному виборі цього числа та конфігурації машини, на якій буде запуснений даний додаток, можна досягнути оптимального часу генерації хорошого розкладу, а при подальшому розширенні проекту, можна зменшити кількість зусиль з боку людини до мінімуму, швидкість обрахунку до моментального.

Тому виконання даного завдання можна вважати успішним, а отриманий розклад лише трохи, але не принципово, відрізнявся від того, що був сформований людиною за довгий час. Швидкість і зручність алгоритму та врахування вимог дуального підходу в освіті є його суттєвою перевагою.

## СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Використання технологій аналізу та синтезу даних для розробки оптимального розкладу занять [Електроний ресурс] – Режим доступу до ресурсу: <https://card-file.onaft.edu.ua/handle/123456789/10175>
2. Снитюк В.Є. Про особливості формування цільової функції та обмежень в задачі складання розкладу занять / Снитюк В.Є., Сіпко Є.Н. // Математичні машини і системи – 2014 - №3 – С. 67-76.
3. Снитюк В.Є. Аспекти формування цільової функції в задачі складання розкладу занять у вищих навчальних закладах на основі суб'єктивних переваг / Снитюк В.Є., Сіпко Є.Н. // Автоматика. Автоматизація. Електротехнічні комплекси і системи - 2013 – №2 – С.98-104.
4. Бевз С. В. Розробка автоматизованої системи формування розкладу магістратури / Бевз С. В., Войтко В. В., Бурбело С. М., Шоботенко А. М. // Інформаційні технології та комп'ютерна техніка – 2009 - №4 – С. 30-65.
5. Бевз С. В. Автоматизація процесу формування розкладу сесії. / Бевз С.В., Войтко В.В., Бурбело С.М., Куба Т.О., Сухоносів О.О.// Принципові концепції та структурування різних рівнів освіти з оптико-електронних інформаційно-енергетичних технологій – 2009 - №4 – С.25-36.
6. Глушков С. В. Базы данных / С. В. Глушков, Д. В. Ломотько. – Харьков: Фолио, 2000. – 504с. – ISBN ISBN 5-8459-0384-X.
7. Горев А. Эффективная работа с СУБД. / А. Горев, С. Макашарипов, Р. Ахаян. – Питер: СПб, 2003. – 445 с. – ISBN: 5-88782-132-9.
8. DB-Engines Ranking [Електронний ресурс] // DB-engines. – 2017. – Режим доступу до ресурсу: <https://db-engines.com/en/ranking>.
9. Байдачный С. SQL Server 2012. Навчальний посібник / Байдачный. С. – М.: «Солон-Пресс», 2005 – 347 с. – ISBN: 5-98003-244-4.
10. Офіційний сайт компанії Microsoft – Режим доступу : [https://www.microsoftstore.com/store/msusa/en\\_US/pdp/Visual-Studio-Professional-2015/productID.323825200](https://www.microsoftstore.com/store/msusa/en_US/pdp/Visual-Studio-Professional-2015/productID.323825200) - Дата доступу 13.06.2016.

11. Microsoft SQL Server [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: [https://ru.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server).
12. Астахова І.Ф. Створення розкладу навчальних занять на основі генетичного алгоритму / Астахова І.Ф., Фірас А.М. // Вісник воронежского державного університету, серія: «Системний аналіз і інформаційні технології». – 2013. – № 2. – С 93-99.
13. Деканова М.В. Математична модель и алгоритм побудови розкладу навчальних занять університету / Деканова М.В. // Вісник Полоцького державного університету. Серія С. – 2013. – №12. – С. 24-33.
14. Верьовкін В.И. Автоматизоване створення розкладів навчальних занять вишу с урахуванням складності дисциплін і втомленості студентів / Верьовкін В.И., Ісмагілова О.М., Атавін Т.А. // Доповіді ТУСУР. – 2009. – №1 (19), частина 1. – С. 221-225.
15. Бабкіна Т.С. Задача складання розкладу: рішення на основ багатоагентного підходу / Бабкіна Т.С. // Бізнес-інформатика. – 2008. –№1. – С.23-28.
16. Database [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Database>.

## **ДОДАТКИ**

**Додаток А****Технічне завдання**

Вінницький національний технічний університет  
Факультет комп'ютерних систем і автоматики

ЗАТВЕРДЖУЮ

Завідувач кафедри САКМІГ

\_\_\_\_\_ д. т. н., проф. В.Б. Мокін

(підпис)

“ ” \_\_\_\_\_ 2019

**ТЕХНІЧНЕ ЗАВДАННЯ**

на магістерську кваліфікаційну роботу

**ІНФОРМАЦІЙНА СИСТЕМА АВТОМАТИЗОВАНОГО ФОРМУВАННЯ**

**РОЗКЛАДУ**

08-53.МКР.011.01.000 ТЗ

Керівник магістерської  
кваліфікаційної роботи

к.т.н., доц.

\_\_\_\_\_ Є.М. Крижановський

(підпис)

“ ” \_\_\_\_\_ 2019 р.

Розробив студент гр. ІСТ-18м

\_\_\_\_\_ О.П. Федюк

(підпис)

“ ” \_\_\_\_\_ 2019 р.

Вінниця 2019



### 1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № \_\_ по ВНТУ від «\_\_» \_\_\_\_\_ 201\_\_ р., та індивідуальне завдання на МКР, затверджене протоколом № \_\_ засідання кафедри САКМІГ від «\_\_» \_\_\_\_\_ 201\_\_ р.

### 2. Джерела розробки

- 1) Снитюк В.Є. Про особливості формування цільової функції та обмежень в задачі складання розкладу занять
- 2) Снитюк В.Є. Аспекти формування цільової функції в задачі складання розкладу занять у вищих навчальних закладах на основі суб'єктивних переваг / Снитюк В.Є., Сіпко Є.Н.
- 3) Бевз С. В. Розробка автоматизованої системи формування розкладу магістратури / Бевз С. В., Войтко В. В., Бурбело С. М., Шоботенко А. М. // Інформаційні технології та комп'ютерна техніка – 2009 - №4 – С. 30-65.
- 3) Бевз С. В. Автоматизація процесу формування розкладу сесії. / Бевз С.В., Войтко В.В., Бурбело С.М., Куба Т.О., Сухоносів О.О

### 3. Мета і призначення роботи

Розробка інформаційної системи автоматизованого формування розкладу.

### 4. Вихідні дані для проведення робіт

- 1) Масив робітників СТО та кваліфікація, швидкість роботи та оплата.
- 2) Масив задач СТО та їх складність.

### 5. Методи дослідження

- 1) Ручний перебор.
- 2) Генетичний алгоритм вирішення задачі про планування.

### 6. Етапи роботи і терміни їх виконання

- 1) Огляд існуючих методів.....\_\_ - \_\_
- 2) Технології складання системи розкладу занять .....\_\_ - \_\_
- 3) Розробка програмного забезпечення системи розкладу занять та її тестування .....\_\_ - \_\_

### 7. Очікувані результати та порядок реалізації

Розробка інформаційної моделі складання розкладу, яка на відміну від існуючих більш краще масштабується при більших даних..

### 8. Вимоги до розробленої документації

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 – «Інформаційні системи та технології» денної форми навчання».

### 9. Порядок приймання роботи

Публічний захист.....\_\_\_.201\_\_ р.

Початок розробки «\_\_» \_\_\_\_\_ 201\_\_ р.

Граничні терміни виконання МКР «\_\_» \_\_\_\_\_ 201\_\_ р.

Розробив студент групи ІСТ-18м \_\_\_\_\_ Федюк О. П.

## Додаток Б. Інтерфейс програми

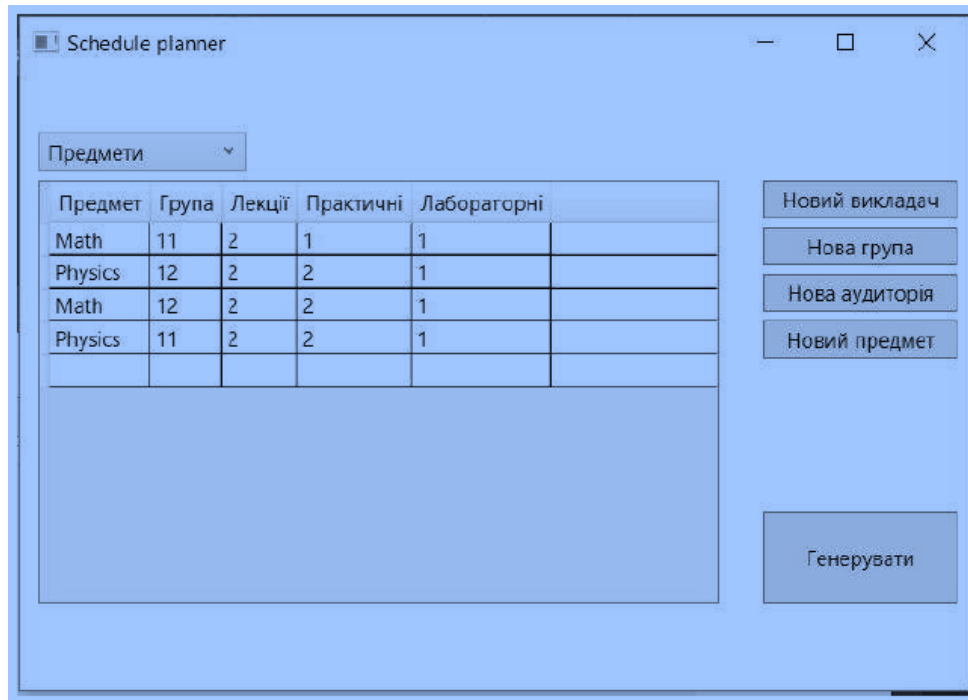


Рисунок Б.1 – Головне вікно

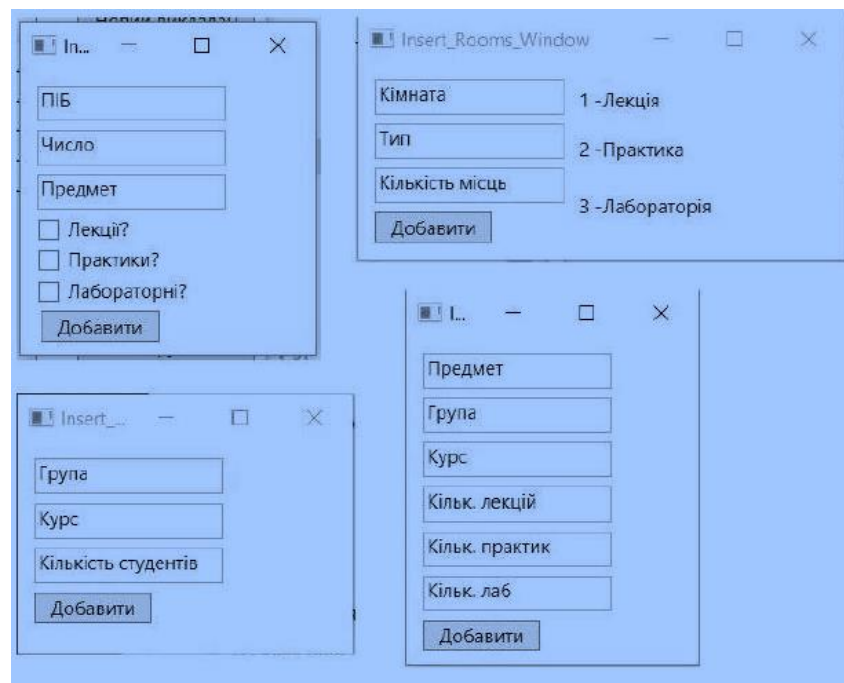


Рисунок Б.2 – Діалогові вікна введення даних

## Додаток В.

### Лістинг програми

```

class Subj
{
    public int subj_code { get; set; }
    public string subj { get; set; }
    public int subj_type { get; set; }
    public int grp_code { get; set; }
    public Subj() { }
    public Subj(string subj_new, int subj_type_new, int course, int group)
    {
        subj = subj_new;
        subj_type = subj_type_new;
        subj_code = Dictionary.Conv(subj) + subj_type ;
        grp_code = course * 10 + group;
    }
}

class Teacher
{
    public string subj { get; set; }
    public int labs { get; set; }
    public Teacher(string name_n, string subj_n, int l, int p, int lab )
    {
        name = name_n;
        subj = subj_n;
        lectures = l;
        practice = p;
        labs = lab;
    }
}

private bool checkTchr(int x, int y, Teacher chsn_teacher)
{
    for (int i = 0; i < schedule.GetLength(2); i++)
    {
        if (schedule[x, y, i] != null)
        {
            if ((schedule[x, y, i].teacher == chsn_teacher))
                return true;
        }
    }
    return false;
}

public Teacher findTeacher(Subj subj, Dictionary DC)
{
    int[] help = new int[3];
    int choose;
    List<Teacher> subj_oriented_teachers = new List<Teacher>();
    foreach (Teacher teacher in DC.teachers_list)
    {
        if (subj.subj == teacher.subj)
        {
            help[0] = teacher.lectures;
            help[1] = teacher.practice;
            help[2] = teacher.labs;
        }
    }
}

```

```

        if (help[subj.subj_type - 1] == 1)
        {
            subj_oriented_teachers.Add(teacher);
        }
    }
}
choose = MainWindow.rand.Next(0, subj_oriented_teachers.Count());
return subj_oriented_teachers[choose];
}

class Generation
{
    public List<RF> main_gen = new List<RF>();
    public List<RF> gen_list = new List<RF>();
    public List<RF> new_gen_list = new List<RF>();
    ///....
    public void generate(Dictionary DC, int cnt)
    {
        count = cnt;
        Rozklad new_roz;
        for (int i = 0; i < count; i++)
        {
            new_roz = new Rozklad(DC);
            gen_list.Add(new RF(new_roz, targetFunction(new_roz, DC)));
        }
        /*for (int i = 0; i < count; i++)
        {
            gen_list[i].value = targetFunction(gen_list[i].rozklad, DC);
        }*/
    }
    /.....
    public double targetFunction(Rozklad schedule, Dictionary DC)
    {
        Dictionary<int, int[]> dict = new Dictionary<int, int[]>();
        Dictionary<int, int> count_d = new Dictionary<int, int>();
        double res_1 = 0;
        foreach (Grp grp in DC.grp_list)
        {
            count_d.Add(grp.grp_code, 0);
        }
        foreach (Grp grp in DC.grp_list)
        {
            dict.Add(grp.grp_code, new int[6] { 0, 0, 0, 0, 0, 0 });
        }
        for (int i = 0; i < schedule.schedule.GetLength(0); i++)
            for (int j = 0; j < schedule.schedule.GetLength(1); j++)
                for (int k = 0; k < schedule.schedule.GetLength(2); k++)
                {
                    if (schedule.schedule[i, j, k] != null)
                    {
                        dict[schedule.schedule[i, j, k].subject.grp_code][i]++;
                    }
                }
        foreach (Grp grp in DC.grp_list)
            for (int j = 0; j < dict[grp.grp_code].GetLength(0); j++)
                if (dict[grp.grp_code][j] == 0)
                {
                    count_d[grp.grp_code]++;
                }
        foreach (Grp grp in DC.grp_list)
        {

```

```

        res_1 += count_d[grp.grp_code];
    }
    res_1 = 1 / (res_1 + 1);
    int saturday = 0;
    for (int j = 0; j < schedule.schedule.GetLength(1); j++)
        for (int k = 0; k < schedule.schedule.GetLength(2); k++)
        {
            if (schedule.schedule[5, j, k] != null)
            {
                saturday++;
            }
        }
    double res_2 = 0;
    if (saturday == 0)
        res_2 = 1;
    return (res_1 + res_2);
}

/...

public void outputRozklad(Rozklad roz, Dictionary DC)
{
    string file = @"C:\Users\oleg_000\Documents\Visual Studio
2015\Projects\Diplom\Diplom\bin\Debug\Rozklad.xlsx";
    List<string> type_list = new List<string>();
    type_list.Add("лекція");
    type_list.Add("практика");
    type_list.Add("лабораторна");
    Excel.Application MyApp = new Excel.Application();
    //MyApp.Visible = false;
    Excel.Workbook MyBook = MyApp.Workbooks.Open(file);
    Excel.Worksheet MySheet = (Excel.Worksheet)MyBook.Sheets[1];
    MySheet.Cells.ClearContents();
    MySheet.Columns.AutoFit();
    foreach (Grp grp in DC.grp_list)
    {
        MySheet.Cells[1, 1 + grp.grp] = "Група " + (grp.grp_code);
    }
    for (int i = 0; i < roz.schedule.GetLength(0); i++)
        for (int j = 0; j < roz.schedule.GetLength(1); j++)
        {
            switch (i)
            {
                case 0:
                    MySheet.Cells[2 + 6 * i + j, 1] = "Понеділок " + (j + 1) + " пара";
                    break;
                case 1:
                    MySheet.Cells[2 + 6 * i + j, 1] = "Вівторок " + (j + 1) + " пара";
                    break;
                case 2:
                    MySheet.Cells[2 + 6 * i + j, 1] = "Середа " + (j + 1) + " пара";
                    break;
                case 3:
                    MySheet.Cells[2 + 6 * i + j, 1] = "Четвер " + (j + 1) + " пара";
                    break;
                case 4:
                    MySheet.Cells[2 + 6 * i + j, 1] = "П'ятниця " + (j + 1) + " пара";
                    break;
                case 5:
                    MySheet.Cells[2 + 6 * i + j, 1] = "Субота " + (j + 1) + " пара";
                    break;
            }
        }
}

```

```

    }
    for (int i = 0; i < roz.schedule.GetLength(0); i++)
        for (int j = 0; j < roz.schedule.GetLength(1); j++)
            for (int k = 0; k < roz.schedule.GetLength(2); k++)
                {
                    if (roz.schedule[i, j, k] != null)
                        {
                            if (roz.schedule[i, j, k].subject.grp_code == 41)
                                MySheet.Cells[2 + 6 * i + j, 1 + 1] = roz.schedule[i, j, k].subject.subj
                                + " " +
                                type_list[roz.schedule[i, j, k].subject.subj_type - 1] + "\n" + roz.schedule[i, j,
                                k].teacher.name + "\n" +
DC.rooms_list[k].room;

                            if (roz.schedule[i, j, k].subject.grp_code == 42)
                                MySheet.Cells[2 + 6 * i + j, 1 + 2] = roz.schedule[i, j, k].subject.subj
                                + " " +
                                type_list[roz.schedule[i, j, k].subject.subj_type - 1] + "\n" + roz.schedule[i, j,
                                k].teacher.name + "\n" +
DC.rooms_list[k].room;

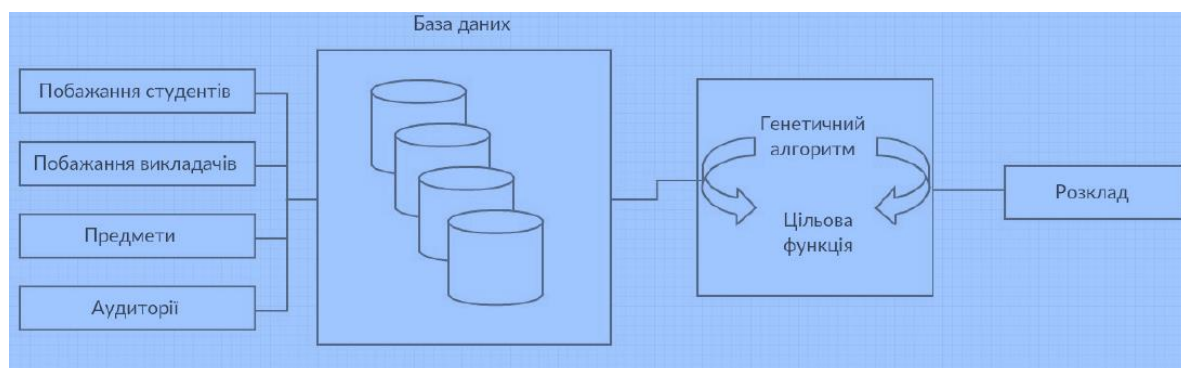
                            if (roz.schedule[i, j, k].subject.grp_code == 2)
                                MySheet.Cells[2 + 6 * i + j, 1 + 3] = roz.schedule[i, j, k].subject.subj
                                + " " +
                                type_list[roz.schedule[i, j, k].subject.subj_type - 1] + "\n" + roz.schedule[i, j,
                                k].teacher.name + "\n" +
DC.rooms_list[k].room;

                        }
                    //MySheet.Cells[2 + 6 * i + j, 1 + roz.schedule[i, j, k].subject.grp_code - 10] =
                    roz.schedule[i, j,
                    k].subject.subj + " " + type_list[roz.schedule[i, j, k].subject.subj_type - 1] + "\n" +
                    roz.schedule[i, j, k].teacher.name +
                    "\n" + DC.rooms_list[k].room;
                }
    MySheet.Columns.AutoFit();
    MySheet.Rows.AutoFit();
    MyBook.Save();
    MyBook.Close();
    MyApp.Quit();
}

```

**Додаток Г**  
**Графічна частина**

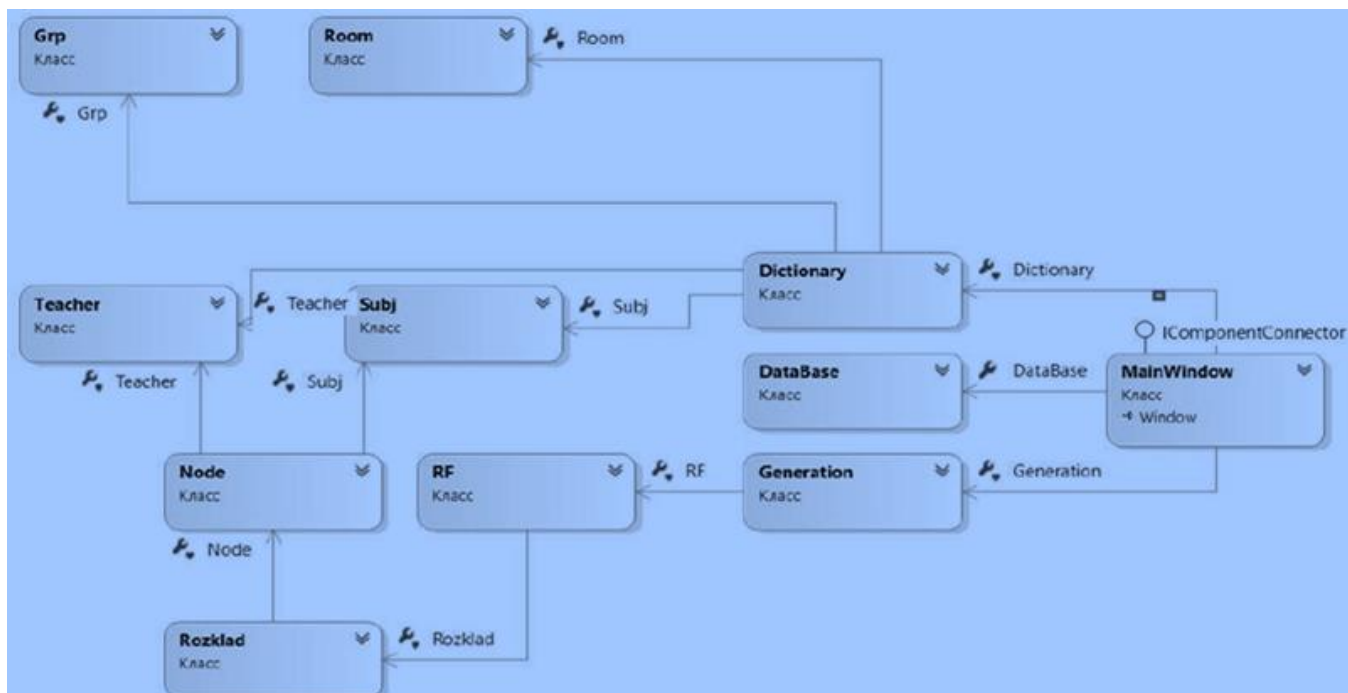
## Архітектура програми





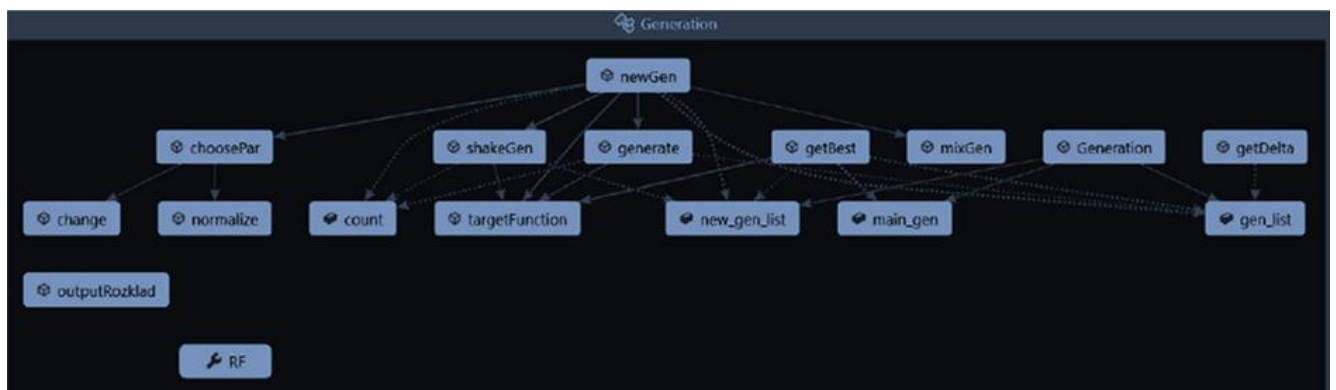
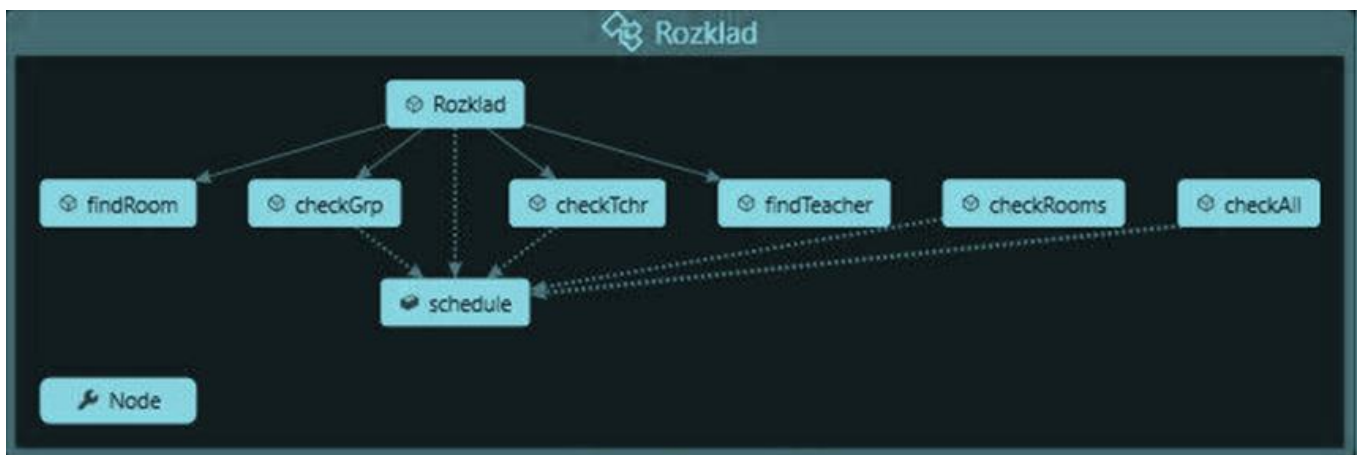
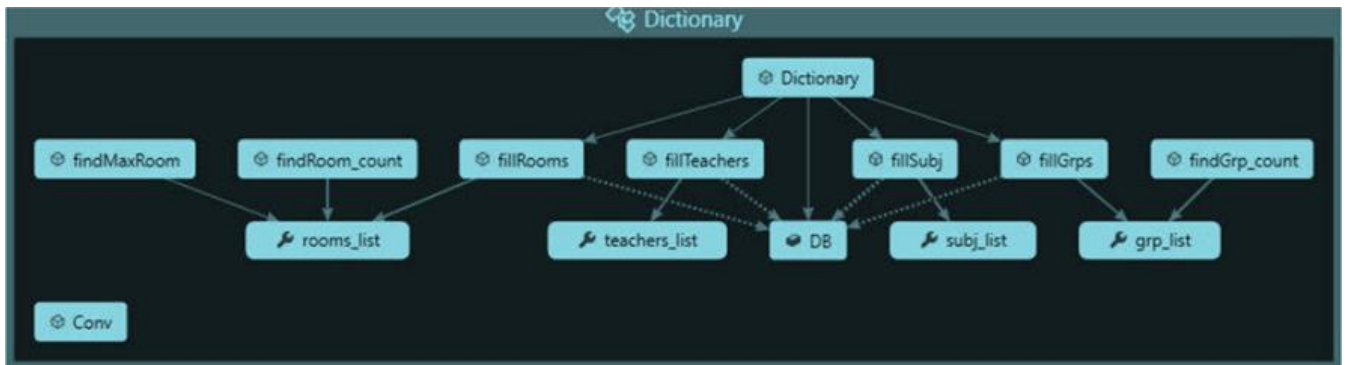
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка інформаційної системи автоматизованого формування розкладу</i>	<i>Літ.</i>	<i>Маса</i>	<i>Масштаб</i>
<i>Розроб.</i>		<i>Федюк О.П.</i>						
<i>Перевір.</i>		<i>Крижановський Є.М.</i>						
<i>Т. Контр.</i>						<i>Арк.</i>	<i>1</i>	<i>Аркушів</i>
<i>Реценз.</i>		<i>Паламачук Є.А.</i>			<i>Архітектура програми</i>	<i>ВНТУ, ICT-18М</i>		
<i>Н. Контр.</i>		<i>Жуков С.О.</i>						
<i>Затверд.</i>		<i>Мокін В.Б.</i>						

## Діаграма класів проекту



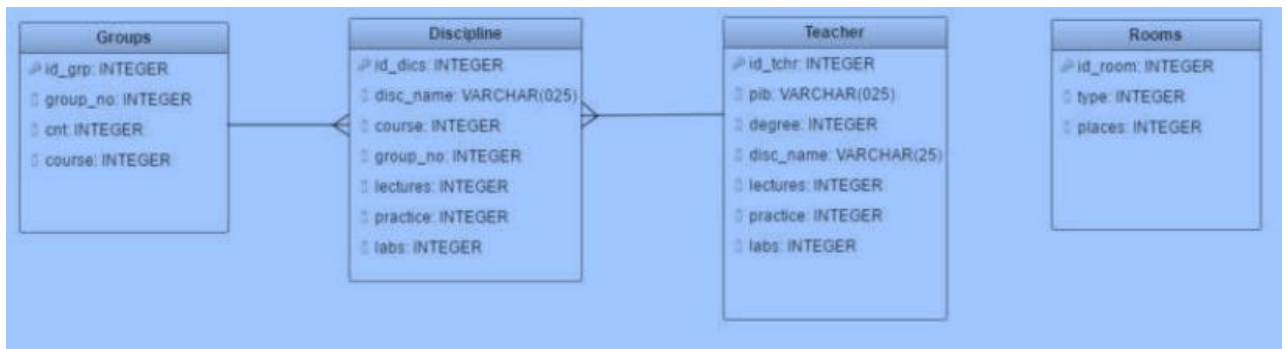
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка інформаційної системи автоматизованого формування розкладу</i>	<i>Літ.</i>	<i>Маса</i>	<i>Масштаб</i>
<i>Розроб.</i>		<i>Федюк О.П.</i>						
<i>Перевір.</i>		<i>Крижановський Є.М.</i>						
<i>Т. Контр.</i>						<i>Арк.</i> 2	<i>Аркушів</i> 11	
<i>Реценз.</i>		<i>Паламачук Є.А.</i>			<i>Діаграма класів проекту</i>	<i>ВНТУ, ICT-18М</i>		
<i>Н. Контр.</i>		<i>Жуков С.О.</i>						
<i>Затверд.</i>		<i>Мокін В.Б.</i>						

## Діаграма залежності класів



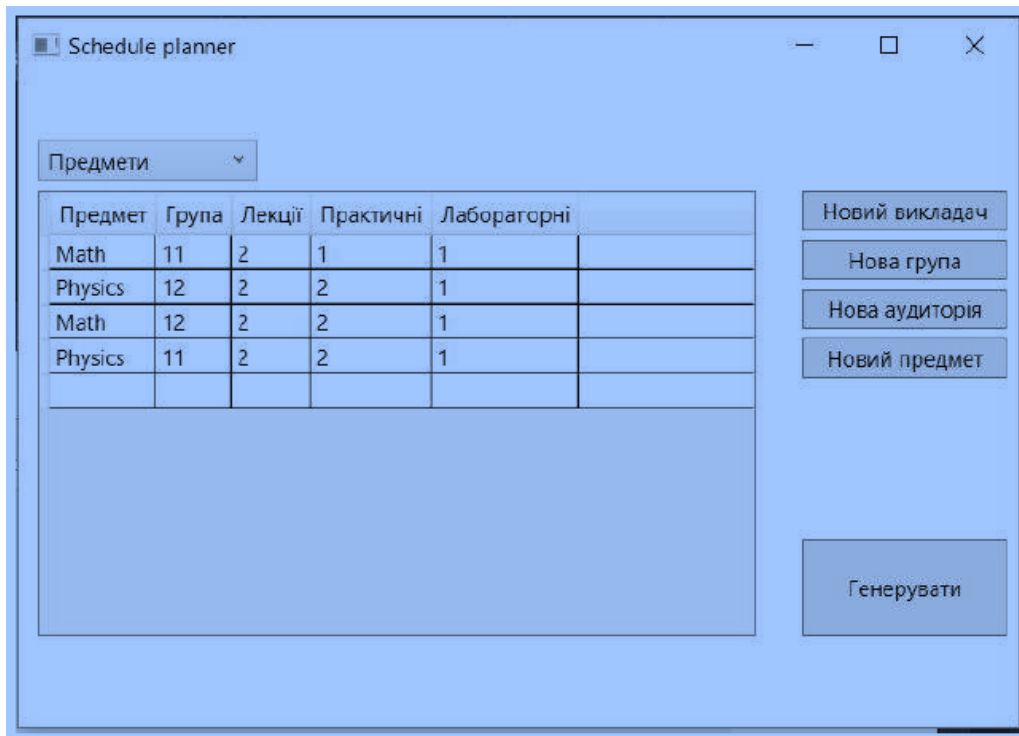
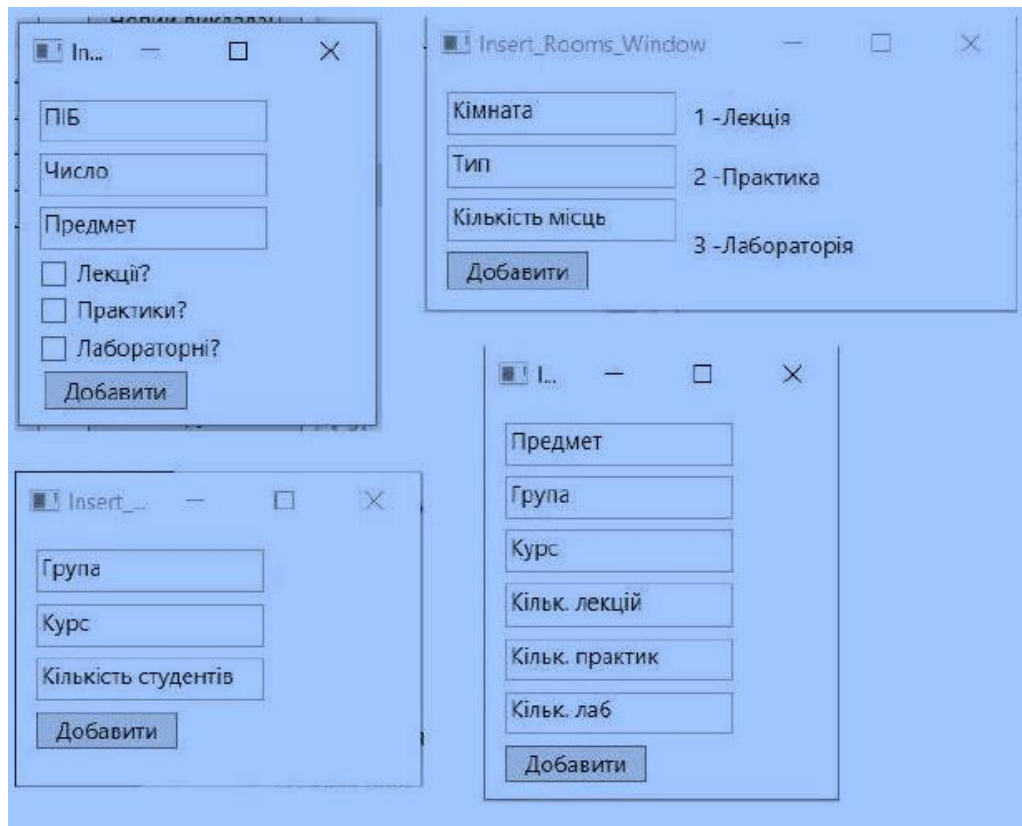
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка інформаційної системи автоматизованого формування розкладу</i>	<i>Літ.</i>	<i>Маса</i>	<i>Масштаб</i>
<i>Розроб.</i>		<i>Федюк О.П.</i>						
<i>Перевір.</i>		<i>Крижановський Є.М.</i>						
<i>Т. Контр.</i>						<i>Арк.</i>	<i>3</i>	<i>Аркушів</i>
<i>Реценз.</i>		<i>Паламачук Є.А.</i>			<i>Діаграма залежності класів</i>	<i>ВНТУ, ICT-18М</i>		
<i>Н. Контр.</i>		<i>Жуков С.О.</i>						
<i>Затверд.</i>		<i>Мокін В.Б.</i>						

## Архітектура Бази Даних





## Зовнішній вигляд інтерфейсу програми







## Згенерований розклад в форматі XLSX

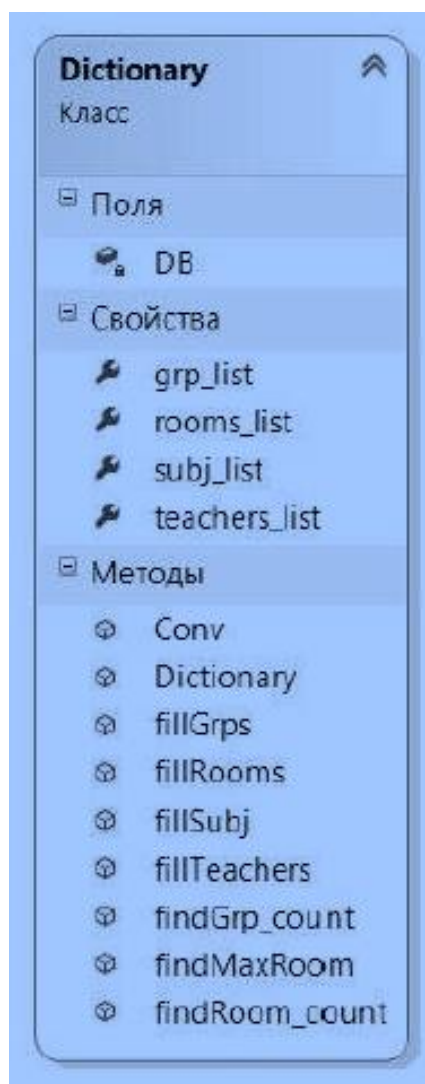
	A	B	C
1		Група 11	Група 12
17	Середа 4 пара		Math практика Bohanov 102
18	Середа 5 пара		
19	Середа 6 пара		
20	Четвер 1 пара	Math лабораторна Statkevici 203	Physics практика Kalita 102
21	Четвер 2 пара		
22	Четвер 3 пара		
23	Четвер 4 пара	Math лекція Bohanov 101	
24	Четвер 5 пара		Math лекція Bohanov 101
25	Четвер 6 пара		Physics лекція Kalita 101
		Math практика	

Лист 1

готово

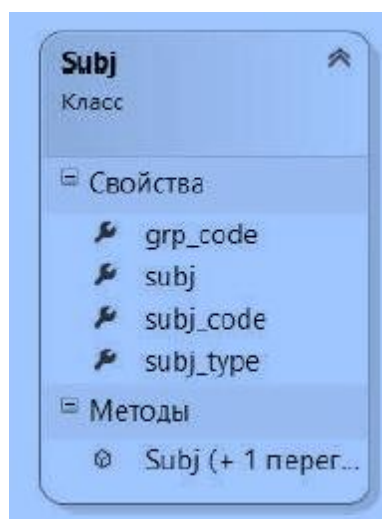


## Клас Dictionary



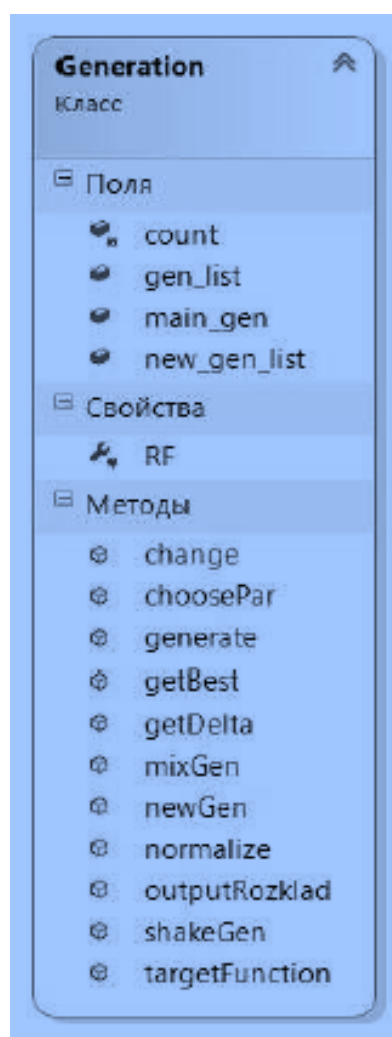


## Клас Subj





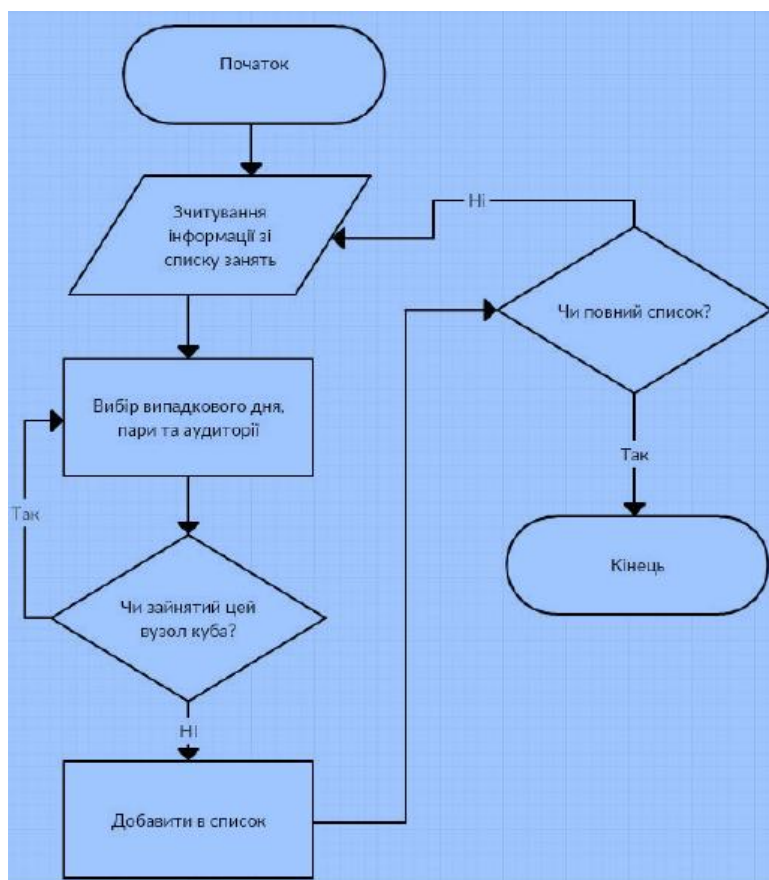
## Клас Generation





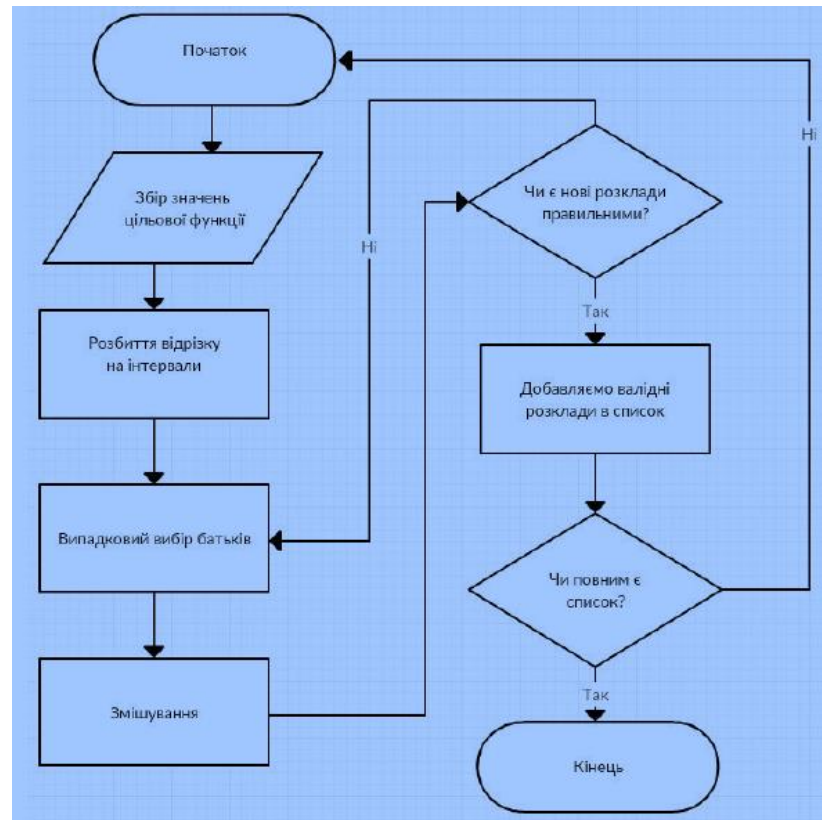


## Блок-схема генерування першої генерації





## Блок-схема формування потомства





## Додаток Д

## Таблиця критеріїв оцінювання розробки

Таблиця Д – Таблиця критеріїв оцінювання розробки

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри- те- рій	0	1	2	3	4
<b>Технічна здійсненність концепції:</b>					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
<b>Ринкові переваги (недоліки):</b>					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно до рівня аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

## Продовження таблиці Д

6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
<b>Практична здійсненність</b>					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту