

Мулява Ігор Ярославович

студент,

Національний технічний університет України «Київський політехнічний інститут»

Куц Михайло Сергійович

студент,

Національний технічний університет України «Київський політехнічний інститут»

Мулява Игорь Ярославович

студент,

Национальный технический университет Украины «Киевский политехнический институт»

Куц Михаил Сергеевич

студент,

Национальный технический университет Украины «Киевский политехнический институт»

Mulyava I.

Student of IASA

National Technical University of Ukraine «KPI»

Kiev, Ukraine

Kuts M.

Student of IASA

National Technical University of Ukraine «KPI»

Kiev, Ukraine

**ПРОГРАМНА МОДЕЛЬ ФОРМУВАННЯ РОЗКЛАДУ
НАВЧАЛЬНИХ ЗАНЯТЬ
ПРОГРАММНАЯ МОДЕЛЬ ФОРМИРОВАНИЯ РАСПИСАНИЯ
УЧЕБНЫХ ЗАНЯТИЙ
PROGRAM MODEL OF THE EDUCATION SCHEDULE
FORMATION SYSTEM**

Анотація. Дана стаття присвячена розробці програмної моделі реалізації алгоритму формування розкладу навчальних занять.

Ключові слова: формування розкладу занять, програмна модель, архітектура додатку.

Аннотация. Данная статья посвящена разработке программной модели реализации алгоритма формирования расписания учебных занятий.

Ключевые слова: формирование расписания занятий, программная модель, архитектура приложения.

Annotation. This article is devoted to developing software model algorithm of implementation schedule of training sessions.

Key words: formation of schedule, programming model, architecture of the application.

Вступ

Складання розкладу є важливим завданням для університету. Задача складання розкладу розв'язується у багатьох галузях. Насамперед, при плануванні дискретного виробництва, організації пасажирських та товарних перевезень, проектуванні та проведенні

навчальних занять у середній, професійно-технічній та вищій школі. В її основу покладено необхідність забезпечення оптимального розподілу робіт серед виконавців, враховуючи просторові та часові обмеження. В даній статті розглянуто програмну модель та модель бази даних, які реалізують цю задачу.

Архітектура бази даних

Для того щоб реалізувати схему описану в [1] нам треба створити базу даних яка буде зберігати всі дані про навчальний заклад, які нам потрібні для побудови розкладу. Ми повинні створити такі таблиці:

1. Таблиця аудиторій
2. Таблиця викладачів
3. Таблиця груп студентів
4. Таблиця дисциплін.

Кожна з цих таблиць повинна містити інформацію про заклад, а також реалізувати певні зв'язки між одна одною. Таблиця аудиторій повинна відображати номер аудиторії, її тип та кількість місць. Таблиця викладачів несе в собі інформацію про ім'я викладача, його наукову ступінь, яку дисципліну він веде, а також які саме форми навчання він проводить.

Таблиця студентів зберігає основні відомості про групи, а саме номер групи, курс, кількість студентів і код групи. Таблиця предметів зберігає назву предмету, групу, у якої дана дисципліна ведеться та кількість лекцій, практик та лабораторних занять з цієї дисципліни, які повинні бути проведені протягом тижня. Виходячи з даного опису ми створимо таку базу даних побудовану на таких таблицях: Discipline — таблиця дисциплін, Teacher — таблиця викладачів, Rooms — таблиця аудиторій і Groups — таблиця студентів [2, ст. 69–72].

- 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
- Id_tchr — ID викладача
- 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.

Повний опис бази даних разом з зв'язками можна побачити на рис. 1. Зв'язки між таблицями будуть такі. Один до багатьох від викладача до предметів, бо один викладач може вести багато дисциплін. А також один до багатьох від груп до предметів, бо одна група має багато предметів.

Розробка програмної моделі на мові C#

Щоб реалізувати ці задачі ми створимо такі класи.

Клас Dictionary

Кожен з цих класів буде мати підкласи, які будуть реалізовувати свої завдання. Dictionary буде зберігати інформацію, яка буде завантажена з бази даних. Тому кожне його поле містить списки об'єктів, які зберігають інформацію про навчальний процес та методи, які ці дані завантажують з бази даних. Опис класу можна побачити на рис. 2. Діаграму залежностей можна розглянути на рис 3. У полях атрибутів цього класу можна побачити списки для кожної бази даних.

Треба зауважити список предметів subj_list. Предмети в ньому зберігаються не так як в базі даних. Вони формуються в класі Subj, який зберігає кожну форму навчання з кожного предмету окремо та показаний на рис 5, як ланку розкладу. Наприклад, нехай у нас предмет Математика, який повинен проводитись в 21 групі. Повинно бути 2 лекції на тиждень та 1 практика, тому, виходячи з цього, у нас буде 3 записи в списку занять: 2 лекції та 1 практика з математики. Це полегшить нам подальшу генерацію розкладу.

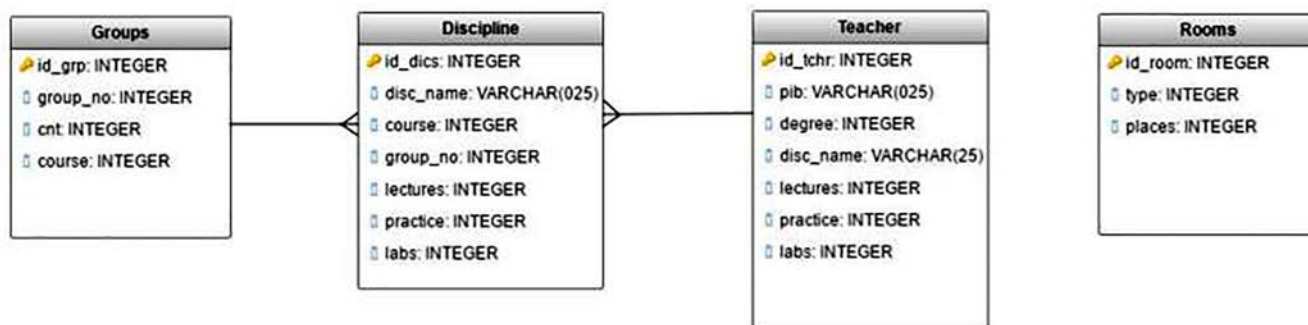


Рисунок 1. Архітектура бази даних

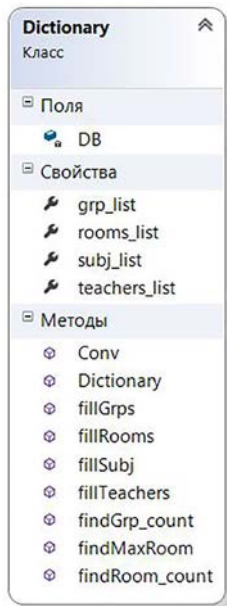


Рисунок 2. Класс Dictionary

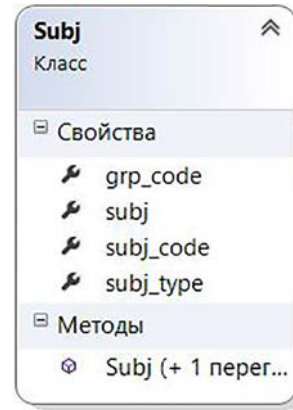


Рисунок 5. Класс Subj

Клас Rozklad

Наступним класом, який ми розглянемо буде Rozklad. Цей клас зберігає куб, який має розмірність 6 на 6 на кількість аудиторій. Діаграма залежностей показана на рис 6.

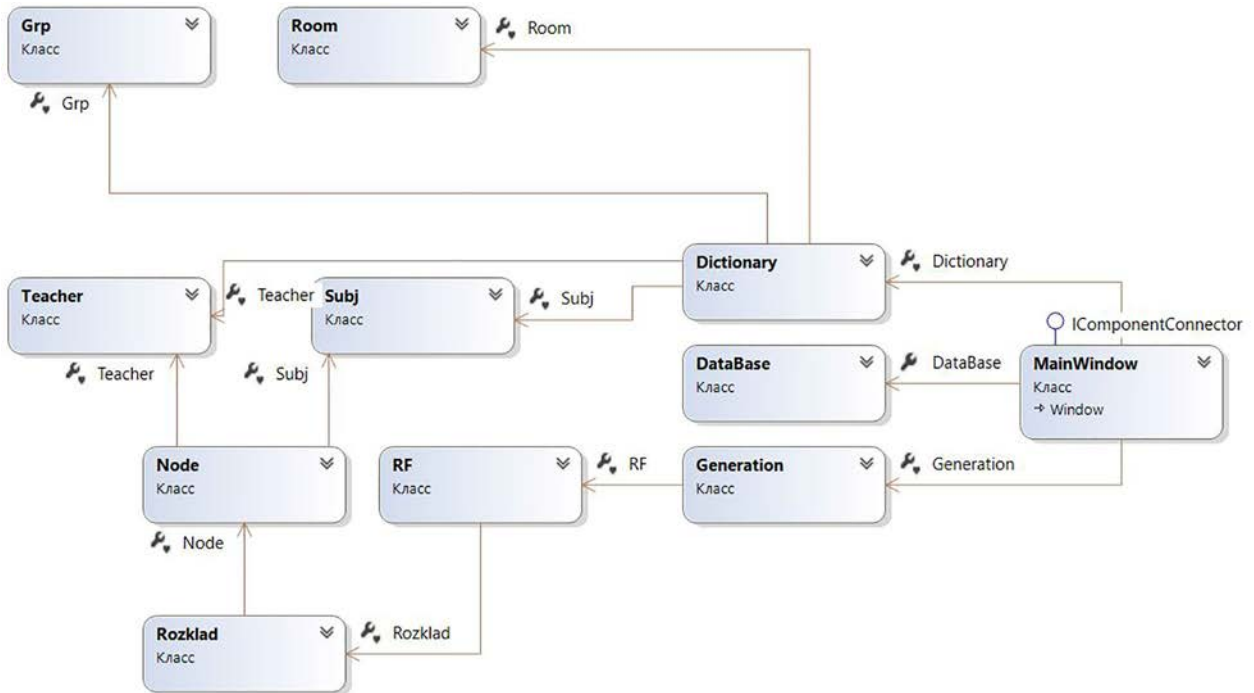


Рисунок 3. Діаграма класів проекту

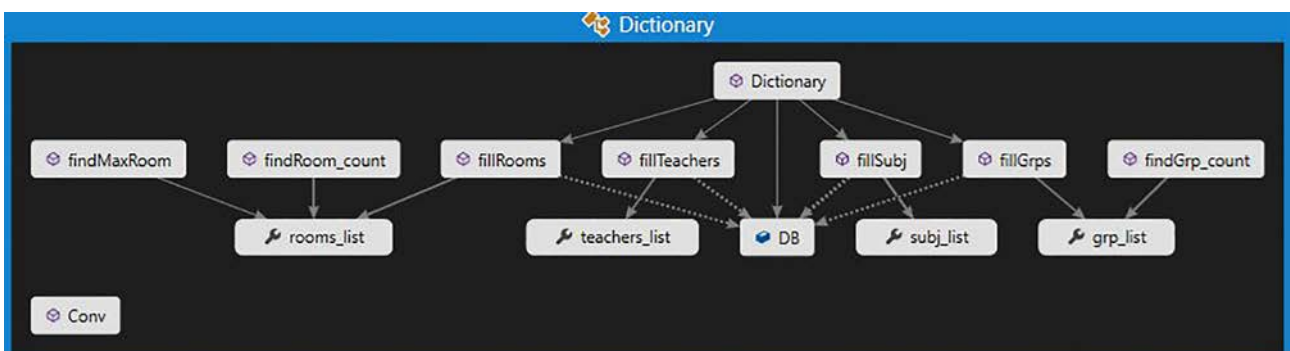


Рисунок 4. Діаграма залежності класу Dictionary

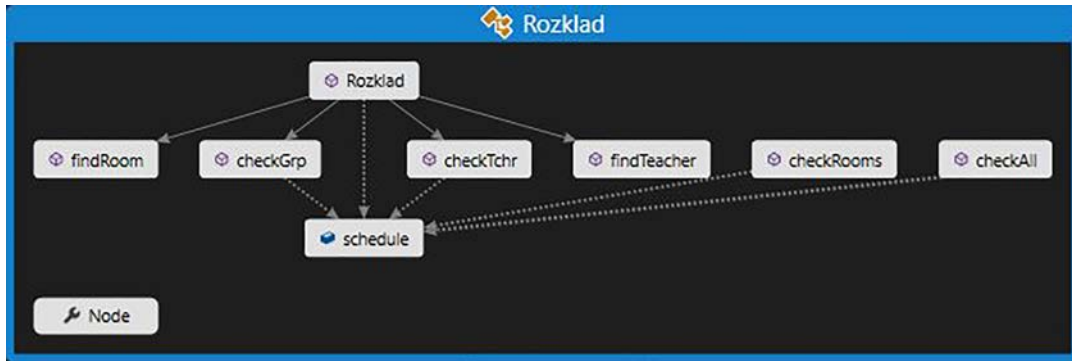


Рисунок 6. Диаграма залежності класу Rozklad

Всі методи з приставкою check на початку реалізують перевірку адекватності розкладу по кожному критерію окремо. А потім викликаються з одного місця в методі checkAll. Тому, роблячи висновок, можна сказати, що цей клас точно відображає суть поставленого завдання.

Клас Generation

Даний клас зберігає в собі всі списки генерацій, а також загальний список який буде сортуватись. Він рахує цільову функцію, а також реалізує генерацію, селекцію, еволюцію та струс популяції. Це основний клас, який реалізує поставлені нами задачі. Його опис можна побачити на рис. 7.

Опис його зв'язків з іншими класами можна на рис. 8.

На діаграмі чітко видно методи необхідні для реалізації алгоритму. Поля gen_list, new_gen_list та main_gen відповідають списку основної генерації, новій генерації та списку, який їх об'єднує для реалізації сортування та селекції.

Методи generate, mixGen, getBest, shakeGen здійснюють генерацію, селекцію, еволюцію та струс популяції відповідно. Метод targetFunction рахує цільову функцію. Методи change і normalize формують вектор цільової функції, необхідний для змішування.

Generation
Класс

Поля

- count
- gen_list
- main_gen
- new_gen_list

Свойства

- RF

Методы

- change
- choosePar
- generate
- getBest
- getDelta
- mixGen
- newGen
- normalize
- outputRozklad
- shakeGen
- targetFunction

Рисунок 7. Клас Rozklad

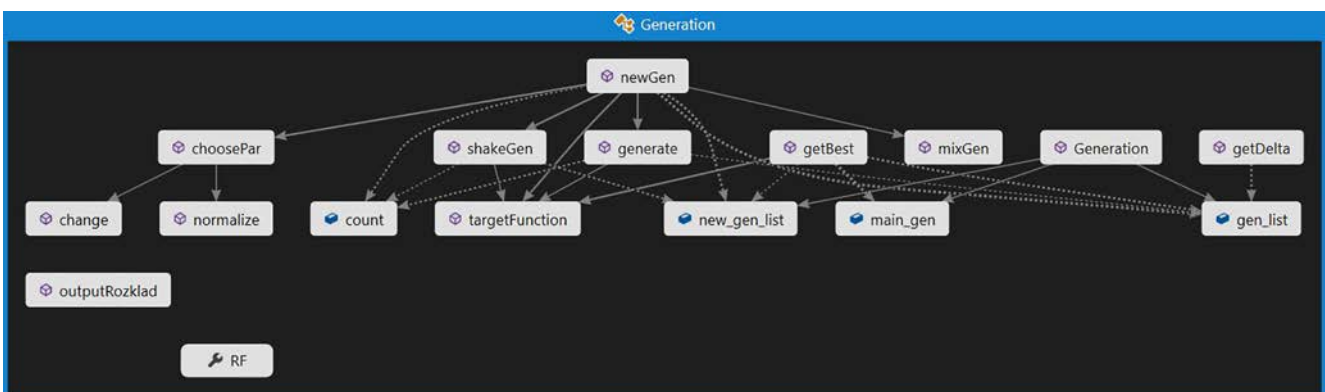


Рисунок 8. Диаграма залежності класу Generation

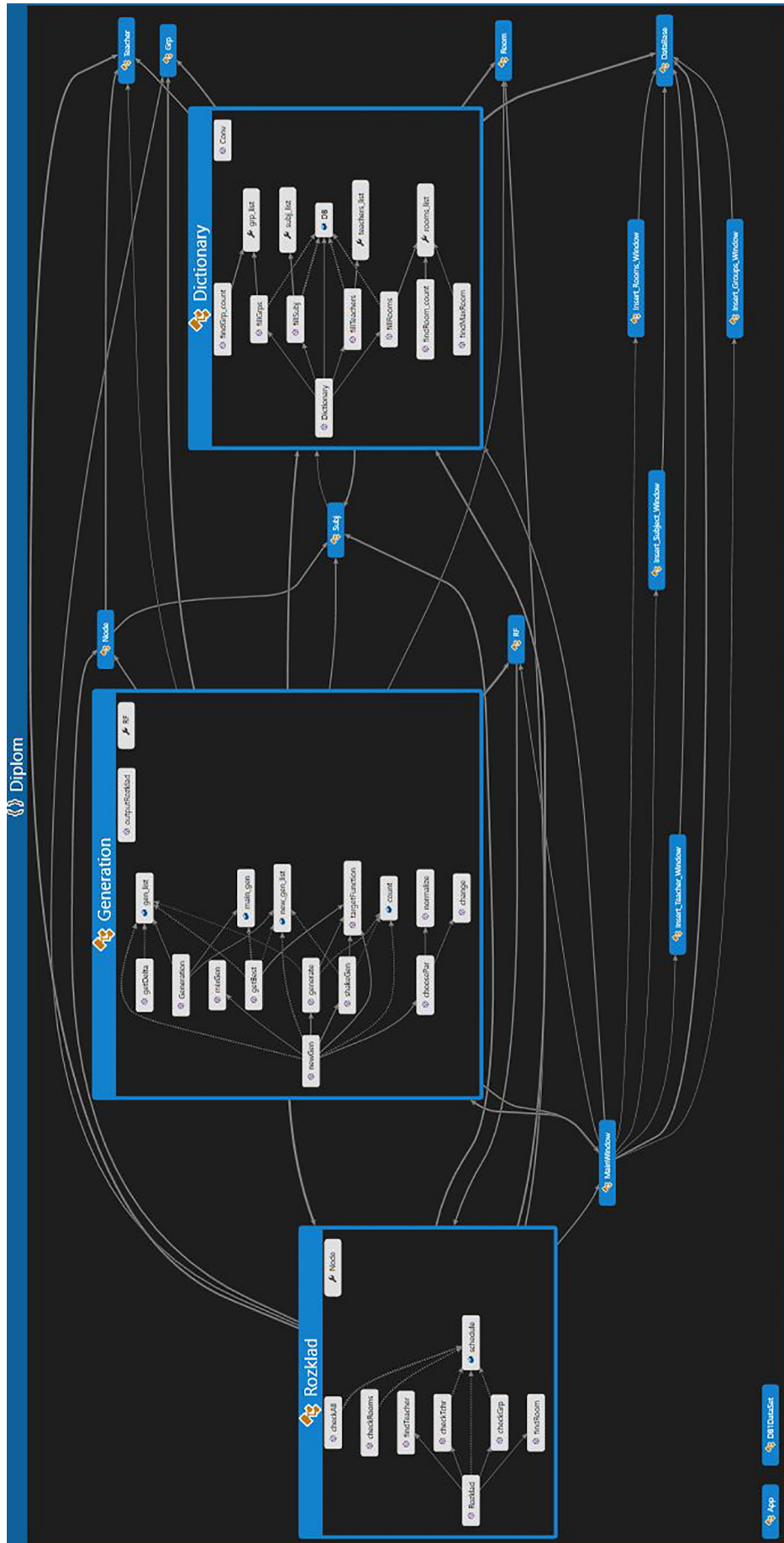


Рисунок 9. Загальна діаграма залежності класів

Також є службовий клас DataBase який містить в собі методи, які з'єднують базу даних з програмним кодом. Він містить код для створення запитів до бази. Кожен запит є прописаний в окремому файлі та оптимізований перед його запуском в проєкті.

Також вартує зауважити клас RE. Його суть полягає в тому, щоб створити однозначну структуру, яка буде зберігати значення цільовою функції розкладу разом з самим розкладом і буде перераховувати це значення кожен раз, коли розклад буде мінятись. Загальна діаграма зв'язків між класами зображена на рис. 9.

Висновок

Запропонована в даній статті модель достатньо близько описує реальну задачу та допомагає спростити не тільки обрахунки, а і збереження даних. Формує класи та структури, необхідні для оптимальної роботи алгоритму. Також запропоноване рішення має хороший рівень масштабування, що дозволить розширювати модель та використовувати її на будь-яких даних.

Література

1. Мулява І. Я. Система формування розкладу навчальних занять з використанням суб'єктивних переваг / Мулява І. Я. // Міжнародний науковий журнал — 2016 — № 8.
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. Астахова І. Ф. Створення розкладу навчальних занять на основі генетичного алгоритму / Астахова І. Ф., Фірас А. М. // Вісник воронежського державного університету, серія: «Системний аналіз і інформаційні технології». — 2013. — № 2. — С 93–99.
5. 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.
6. Вєрьовкін В. И. Автоматизоване створення розкладів навчальних занять вищу с урахуванням складності дисциплін і втомленості студентів / Вєрьовкін В. И., Ісмагілова О. М., Атавін Т. А. // Доповіді ТУСУР. — 2009. — № 1 (19), частина 1. — С. 221–225.