

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

# **КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА**

**на тему:**

**«Інформаційна технологія автоматизованого формування  
розкладу закладу освіти»**

**Завідувач**

**випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Берест О.Б.**

**Студент групи ІН.м-02**

**Яковлев М.М.**

**СУМИ 2021**

Сумський державний університет

(назва вузу)

Факультет ЕЛІП Кафедра Комп'ютерних наук

Спеціальність «122 -Комп'ютерні науки»

Затверджую:

зав.кафедрою

“ \_\_\_\_\_ ”

20\_\_ р.

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Яковлеву Максиміліану Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інформаційна технологія автоматизованого формування розкладу закладу освіти

затверджую наказом по інституту від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін здачі студентом закінченого проекту (роботи)

3. Вхідні данні до проекту (роботи)



7. Дата видачі завдання

---

Керівник

---

(підпис)

Завдання прийняв до виконання

---

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	<i>Аналіз проблеми. Постановка задачі дослідження</i>		
2.	<i>Вибір методу розв'язку задачі</i>		
3.	<i>Побудова математичної моделі</i>		
4.	<i>Розробка інформаційного та програмного забезпечення системи формування розкладу навчального закладу</i>		
5.	<i>Оформлення пояснювальної записки до дипломної роботи</i>		

Студент – дипломник

---

(підпис)

Керівник проекту

---

(підпис)

## РЕФЕРАТ

**Записка:** 57 стор., 13 рис., 4 додатки, 12 джерел.

**Об'єкт дослідження** — інформаційна технологія формування розкладу.

**Мета роботи** — розробка інформаційної системи автоматичного формування оптимального розкладу для навчальних закладів

**Результати** — розроблено кросплатформну інформаційну систему з зручним користувацьким інтерфейсом, засновану на вирішенні задачі нелінійного програмування. Результатом роботи програми є CSV таблиця з розкладом.

## ЗМІСТ

ВСТУП .....	7
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ .....	9
1.1 Відомі алгоритми розв’язку задачі .....	9
1.1.1 Засновані на теорії графів .....	9
1.1.2 Ймовірнісний алгоритм .....	10
1.1.3 Побудова розкладу за нижньою оцінкою .....	10
1.1.4 Генетичний алгоритм.....	11
1.2 Існуючі програми для розв’язку задачі формування розкладу .....	12
2 ПОСТАНОВКА ЗАДАЧІ.....	15
3 МАТЕМАТИЧНА МОДЕЛЬ ФОРМУВАННЯ РОЗКЛАДУ ЗАНЯТЬ .....	17
4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	22
4.1 Розробка моделей автоматизованої системи формування розкладу .....	22
4.2 Розробка програмного забезпечення системи.....	23
4.3 Приклад використання програмного продукту .....	25
ВИСНОВОК.....	32
ДОДАТКИ.....	35
Додаток А. Приклади вхідних даних з CSV файлу .....	35
Додаток Б. Приклад вихідних даних.....	36
Додаток В. Приклад GUI класу.....	39
Додаток Г. Підключення та ініціалізація бази даних .....	54

## ВСТУП

Якість підготовки спеціалістів у вищих навчальних закладах та ефективність використання науково-педагогічного потенціалу залежать певною мірою від організації навчального процесу. Одна з основних складових цього процесу - складання розкладу - один з найважливіших та завершальний етап планування навчального процесу в навчальному закладі. Розклад буде оптимальним та якісним, якщо воно повністю реалізує можливості структурологічних схем дисциплін, що дозволяють оптимізувати склад навчальних дисциплін та їх взаємозв'язки. Якщо при складанні розкладу навчальних занять не враховано в повному обсязі міжпредметні зв'язки або не виконано певних вимог, то якість планування навчального процесу знижується, що істотно впливає на якість підготовки фахівців.

Розклад навчальних занять – це документ, що регламентує роботу викладачів, студентів, усього навчального закладу, який розподіляє склад навчального плану та робочих програм по календарним дням навчального року, та що забезпечує їх реалізацію. Оптимально складений розклад не повинен змінюватися протягом семестру або навчального циклу, щоб не порушити міжпредметні зв'язки та виконані умови.

Документи, на базі яких формується розклад[1]:

- структурно-логічні схеми курсів,
- спеціальності,
- сітки годин по курсах та спеціальностям,
- аудиторний фонд кафедр та ВНЗ,
- штатний розклад викладацького складу,
- перелік вимог до розкладу.

Вимоги до розкладу навчальних закладів можна умовно розділити на три види[2]:

- методичні,
- організаційні,
- вимоги щодо забезпечення самостійної роботи студентів.

Вони переважно охоплюють всі сторони навчального процесу. Деякі з них є безумовними, без яких навчальний процес не може нормально функціонувати, інші – визначають певною мірою якість навчального процесу та якість передачі інформації. Тому кожен вид вимог можна розділити за ступенем їх впливу на якість планування навчальних занять[6]:

- головні,
- першорядні,
- другорядні.

Головні методичні умови до розкладу мають забезпечити послідовності проведення навчальних занять відповідно до структурно-логічних схем дисциплін на спеціальності. Першорядні впливають на рівномірність навчального навантаження, чергування лекційних та практичних занять, планування занять з використанням технічного обладнання в спеціалізованих аудиторіях. Другорядні – на відсутність повторення занять з однієї дисципліни в один день, на планування занять зі складних дисциплін на дні посеред неділі.



# 1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Задача формування оптимального розкладу виникла багато століть назад, за часів заснування перших навчальних закладів. Існує багато готових розв'язків цієї задачі, але через її складність та комплексність вони мають багато обмежень та результат їх роботи часто не повною мірою вирішує завдання.

## 1.1 Відомі алгоритми розв'язку задачі

Серед алгоритмів вирішення задачі формування розкладу можна виділити такі[3]:

- засновані на теорії графів,
- ймовірнісний алгоритм,
- побудова розкладу за нижньою оцінкою,
- генетичний алгоритм

### 1.1.1 Засновані на теорії графів

Основна ідея даного типу алгоритмів є зіставлення графів великих розмірностей та знаходження найкоротшого шляху на графі.

Побудову розкладу можна розбити на три етапи[5]:

- 1) розподіл клієнтів між агентами (розбиття множини  $S$  на  $m$  непересічних підмножин);
- 2) вибір порядку обслуговування клієнтів кожному за агенту;
- 3) оптимізація маршруту кожного агента при фіксованому порядку обслуговування клієнтів.

Таким чином, сумарні обмеження за характеристиками для кожного агента

перевіряються на першому етапі, обмеження на вершини - на перших двох, а на третьому етапі перевіряються тимчасові обмеження.

### 1.1.2 Ймовірнісний алгоритм

Даний алгоритм заснований на методі вирішення задачі пакування, викладений в [11]. Напишемо поставлену задачу у вигляді булевого лінійного програмування

$$\begin{aligned} \sum_{j=1}^m x_{ji} &= 1, \quad i = 1, \dots, n, \\ \sum_{j=1}^n x_{ji} \tau_{ij} &\leq B, \quad j = 1, \dots, m, \\ x_{ji} &\in \{0, 1\}, \\ Z = B &\longrightarrow \min. \end{aligned}$$

При цьому робота  $T_i$  виконується на процесорі  $P_j$  тільки тоді, коли  $x_{ji} = 1$ . Кожна робота буде виконана на одому процесорі, зсилаючись на умови. Також накладені обмеження на довжину розкладу, та цільовою функцією буде знайдемо розклад мінімальної довжини.

### 1.1.3 Побудова розкладу за нижньою оцінкою

Даний метод зазвичай застосовується до завдань, у яких цільова функція –  $C_{\max}$ . Складання розкладу по за нижньою оцінкою відбувається у 2 етапи[4]:

1. Побудова набору нижніх обмежень на довільний розклад для задачі S.
2. Побудова допустимого розкладу, досягаючого максимального обмеження побудованого набору.

За допомогою цього методу вирішуються наступні задачі:

- $P \mid pmtn \mid C_{\max}$
- $R \mid pmtn \mid C_{\max}$
- $O \mid p_{ij} = 1 \mid C_{\max}$

- $Q \mid \text{pmtn} \mid C_{\max}$

#### 1.1.4 Генетичний алгоритм

Для підвищення якості розкладу занять, основний алгоритм складання розкладу може бути вдосконалено. Зважаючи на наявність набору коефіцієнтів алгоритму, які можна налаштувати, а також критеріїв оцінки якості розкладу, можливо використання генетичного алгоритму[7].

Пропонується використовувати в якості генів коефіцієнти  $\omega_j$ . У результаті мутацій будуть тримані нові набори коефіцієнтів, та якість роботи алгоритму буде підвищена[8].

Хромосоми такого генетичного алгоритму виглядають як набір дійсних чисел:

$$\{ \omega_1, \omega_2, \dots, \omega_m \},$$

де  $\omega_1, \dots, \omega_m$  – гени хромосоми,  $m$  – кількість коефіцієнтів алгоритму.

Ціллю роботи генетичного алгоритму є досягнення максимуму функціонала:

$$R_{il} = \sum_{j=1}^m w_j k_{jl}$$

$$R_i = \max_l (R_{il}), l = 1..h$$

$$R = \sum_{i=1}^n R_i \rightarrow \max,$$

де  $R_{ij}$  – якість розміщення  $i$ -го заняття на  $l$ -й позиції в розкладі,

$k_{il}$  – значення, отримане по  $j$ -му критерію оцінки якості розміщення заняття,

$\omega_j$  – ваговий коефіцієнт  $j$ -го критерія оцінки якості,

$m$  – кількість критеріїв оцінки якості,

$l$  – можлива позиція  $i$ -го заняття в розкладі,

$h$  – кількість можливих варіантів розташування занять в розкладі,

R – якість розкладу.

## 1.2 Існуючі програми для розв'язку задачі формування розкладу

1С: Автоматизоване складання розкладу[12]:

Програма для складання розкладу призначена для вирішення завдань автоматизованого складання навчальних розкладів та оперативного управління приміщеннями у ВНЗ. З його допомогою скласти розклад можна в автоматичному, ручному та змішаному режимах з урахуванням багатьох обмежень та умов. При цьому можна побудувати як допустимий розклад, так і оптимізований, в якому скорочено кількість вікон або кількість приміщень, що використовуються.

Можливості продукту:

- оптимізація навчальних розкладів за одним із критеріїв:
  - кількість використовуваних приміщень,
  - мінімізувати кількість вікон тощо;
- порівняння на допустимість при складанні розкладу у будь-якому режимі:
  - тип приміщення / тип заняття,
  - місткість приміщення/кількість студентів у групі;
- вибір довільної періодичності розкладу (тиждень, два тижні, семестр, фіксований період тощо);
- складання розкладу сесії;
- облік паралельних занять, розбиття на підгрупи та потокових лекцій при складанні розкладу;
- облік максимальної допустимої кількості занять на день для групи студентів або викладача під час складання розкладу;

- побудова розкладу для 2-х і більше змін;

Інтерфейс 1С для розкладу (Рис. 1)

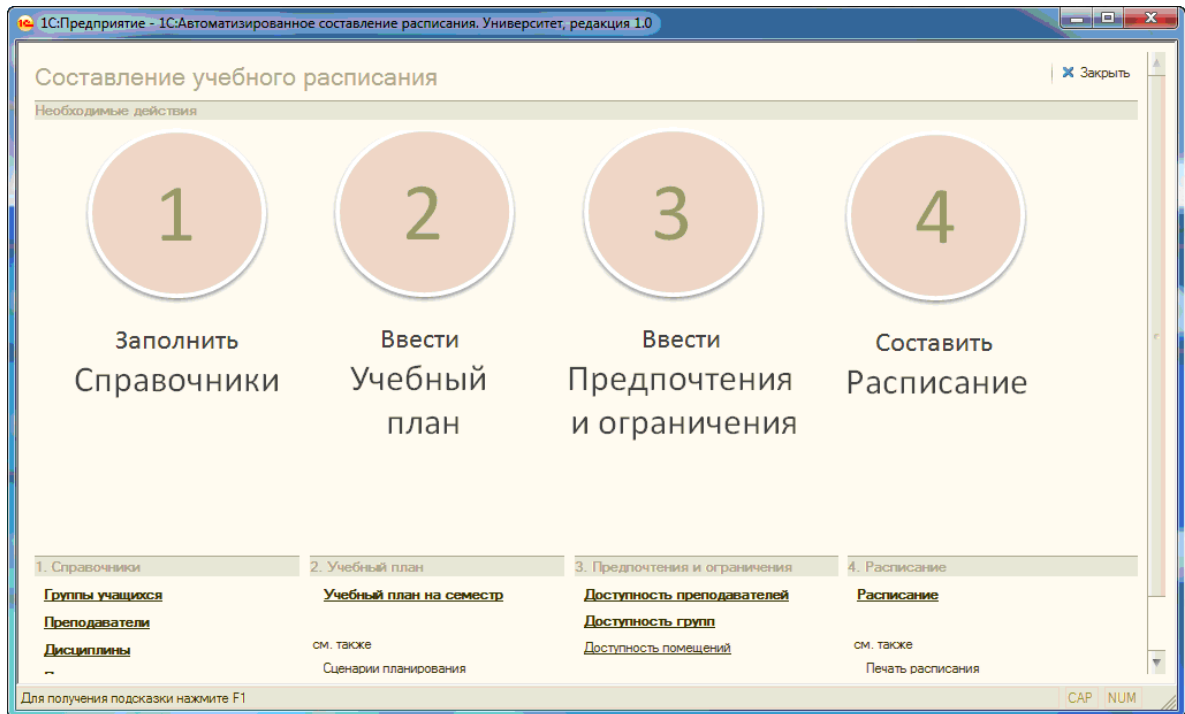


Рис. 1. Етапи формування розкладу в програмі 1С

Готовий розклад є доволі інформативним, але інтерфейс дуже застарілий(Рис. 2)

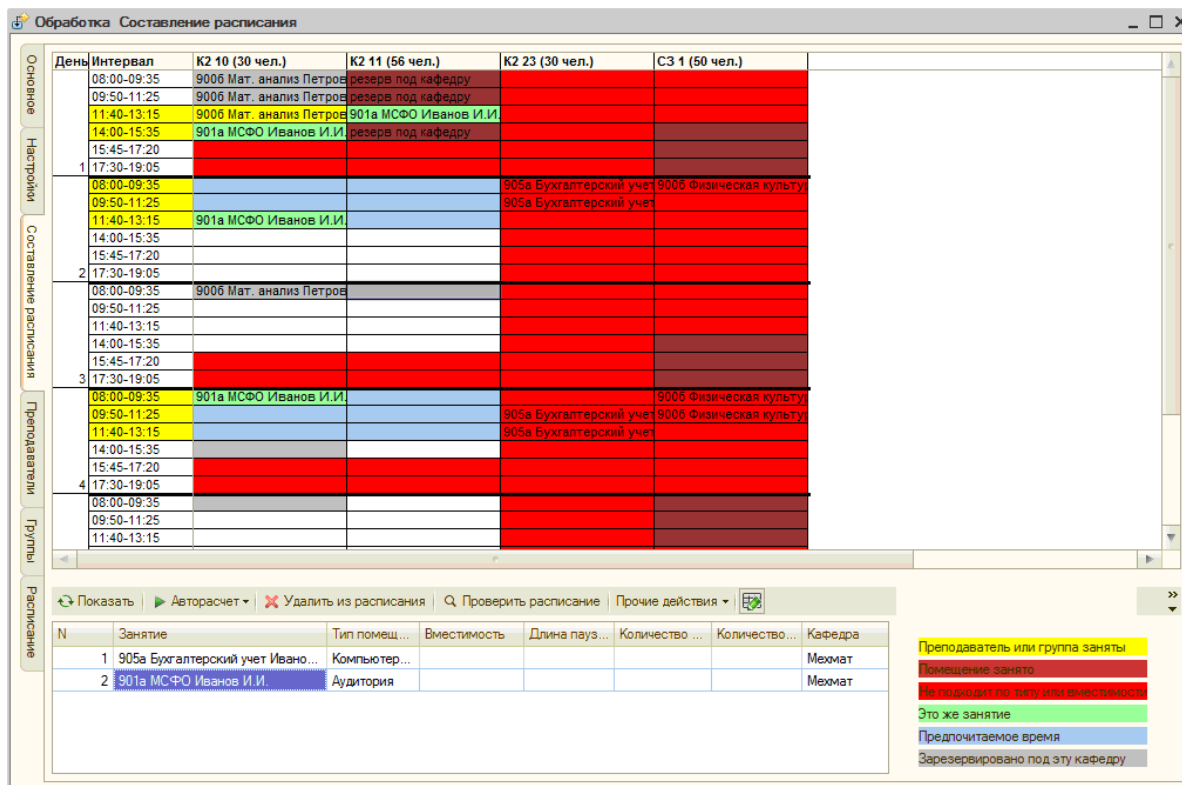


Рис. 2. Приклад сформованого розкладу в програмі 1С

#### Преваги:

- готовий продукт для використання,
- непогана точність,
- можливість вручну корегувати розклад,
- можливість вибору критерію оптимізації,
- можливість будувати розклад для 2 змін.

#### Недоліки:

- платний продукт з високою вартістю,
- застарілий інтерфейс,
- не може працювати в повністю автоматичному режимі.

## 2 ПОСТАНОВКА ЗАДАЧІ

Мета науково-дослідної роботи полягає в розробленні інформаційної системи автоматичного формування розкладу навчального закладу.

При складанні розкладу виникає проблема оптимального управління ресурсами: викладацьким складом та аудиторним фондом[9]. У процесі вирішення

завдання необхідно враховувати обов'язкові обмеження, а також додаткові вимоги, які можуть порушуватись у деяких випадках.

До обов'язкових обмежень відносяться[10]:

- місткість аудиторій має бути достатньою для груп, які в ній займаються, при цьому можливий варіант, коли в одній аудиторії проводяться заняття одночасно для кількох груп студентів;
- повинні виконуватись вимоги занять до обладнання аудиторій, у яких вони проводяться;
- викладачі з інших вузів можуть проводити заняття лише у певні дні та години.

До додаткових вимог належать:

- лекції мають проводитися на початку дня, практики – наприкінці;
- навантаження кожної групи має бути рівномірним, щоб уникнути перевтоми студентів, тобто в ті дні, коли проводиться лекція з складного предмета, решта занять має проводитися з відносно простим;
- у заняттях студентів не повинно з'являтися вікон, водночас можлива наявність вікна у розкладі викладача;
- по можливості викладачам повинні надаватися дні, вільні від проведення занять у ВНЗ;

- у п'ятницю кількість занять має бути меншою, ніж у решту днів тижня;
- першим заняттям у понеділок має йти відносно простий предмет, інакше успішність студентів може суттєво знизитися.



### 3 МАТЕМАТИЧНА МОДЕЛЬ ФОРМУВАННЯ РОЗКЛАДУ ЗАНЯТЬ

Спочатку визначимо базові множини для формування розкладу:

$T = \{T_1, T_2, \dots, T_{st}\}$  – множина викладачів;

$S = \{S_1, S_2, \dots, S_{ss}\}$  – множина студентів;

$D = \{D_1, D_2, \dots, D_s\}$  – множина дисциплін;

$D_i = \{L_{i1}, L_{i2}, L_{i3}, L_{i4}, L_{i5}, L_{i6}\}$  – множина типів дисциплін.

Були визначені такі типи занять:

$L_{i1}$  – лекційні заняття;

$L_{i2}$  – лабораторні заняття;

$L_{i3}$  -практичні заняття;

$L_{i4}$  – семінарські заняття;

$L_{i5}$  – самостійна робота студентів;

$L_{i6}$  - курсові роботи.

Значення параметру  $ss$  визначається кількістю студентів, у яких є обрана дисципліна в певний момент часу, та лежить у межах від 1 до  $ss$ .

Кожне заняття  $L_{ij}$  має певну кількість годин  $p_{ij}$  та має викладача з множини  $\tau \subset T$  ( $i$  – дисципліна,  $j$  – вид заняття) студентам з множини  $\omega \subset S$  – множина студентів, які обрали дисципліну  $D_i$ .

Таким чином, пріоритетними напрямками розв'язання задачі є концептуальний аналіз основних характеристик процесу створення розкладів для занять студентів, як:  $\alpha$  – характеристики викладачів;  $\beta$  –

характеристики студентів;  $\gamma$  – характеристики дисциплін,  $\delta$  – реляційне формування цільового функціоналу задачі.

Основаючись на дослідженнях [1] можна стверджувати, що поле  $\alpha$  складається з двох частин  $\alpha = \alpha_1 \cap \alpha_2$ , де  $\alpha_1 = P$  – завантаженість викладачів дисциплінами,  $\alpha_2 = W$  – відносний коефіцієнт рейтингу викладача.

Загальний час навантаження складає:

$$p_k = \sum_{i=1}^n \sum_{j=1}^6 q_{ij} \begin{cases} q_{ij} = p_{ij} & \left( \tau_{ij} = T_k \right) \cap \left( \sum_{i=1}^n \sum_{l=1}^{s_s} \omega_{il} > 0 \right); \\ q_{ij} = 0 & \left( \tau_{ij} \neq T_k \right) \cup \left( \sum_{i=1}^n \sum_{l=1}^{s_s} \omega_{il} = 0 \right). \end{cases}$$

Нехай  $\lambda_{du}$  – множина уроків.  $TD \in \{1, 2, 3, 4, 5, 6\}$  – дні, під час яких проводиться навчання,  $TU \in \{1, 2, 4, 5, 6, 7\}$  – номери уроків розкладу занять. Отримаємо множину часових інтервалів  $TR = TD * TU$ .

Приклад обмежень для викладача  $k$ -го викладача:

$$x_{k1} = \lambda_{du} | (d \in \{5, 6\}) \cap (u \in \{1, 2, 3, 4, 5\});$$

$$x_{k2} = \lambda_{du} | ((d \neq 2)(d \neq 4)) \cap (u \in \{4, 5\});$$

$$x_{k3} = \lambda_{du} | (d = 2) \cap (u \in \{1, 2, 6\});$$

За даними обмеженнями можна зіставити таблицю, в якій буде відображено дні та номери занять викладача, в які йому можна поставити заняття:

№Уроку\день неділі	Понеділок	Вівторок	Середа	Четвер	П'ятниця	Субота
1	-	+	-	-	+	+

2	-	+	-	-	+	+
3	-	-	-	-	+	+
4	+	-	+	+	+	+
5	+	-	+	+	+	+
6	-	+	-	-	-	-
7	-	-	-	-	-	-
8	-	-	-	-	-	-

Подібні обмеження накладаються на кожного викладача.

Характеристичні параметри студентів у узагальненій системі створення розкладу подаються у вигляді наступної контамінації:  $\beta = \beta_1 \cap \beta_2$ , де  $\beta_1 = G_i$  – студенти складають групи, які відрізняють спеціальністю базової підготовки та кафедрою професійного напрямку.

$B_2 = F_i$  – студенти формують потоки згідно індивідуального плану підготовки студентів:  $F = \{F_1, F_2, \dots, F_{11}\}$

- $F_1$  - загальний потік студентів;
- $F_2$  – потоки студентів для практичних та семінарських занять із циклу гуманітарних та соціально-економічних дисциплін;
- $F_3$  – професійні потоки студентів на викладання дисциплін теоретичних основ;
- $F_4$  – об'єднані потоки студентів для викладання дисципліни "Теорія та практика наукових досліджень";
- $F_5$  – потоки студентів для викладання дисциплін наукового напрямку;

- $F_6$  – потоки студентів окремих спеціальностей чи кафедр для викладання прикладних аспектів дисциплін підготовки;
- $F_7$  – потоки суміжних та окремих спеціальностей для викладання інтегральних дисциплін;
- $F_8$  – потоки студентів для викладання дисциплін з іноземної мови;
- $F_9$  – потоки студентів для викладання дисциплін окремих спеціалізацій;
- $F_{10}$  – потоки студентів для викладання обраних дисциплін з циклу гуманітарних та соціально-економічних дисциплін;
- $F_{11}$  – потік студентів індивідуального вибору дисциплін.

Характеристика дисциплін  $\gamma$  направлена на наступне:  $\gamma_1 = f_{ij}$  – дисципліна належить плану спеціальності та читається в довільному порядку.

Час занять для студента  $k$  за індивідуальним розкладом визначається з виразу:

$$r_k = \sum_{i=1}^n \sum_{j=1}^6 q_{ij} \begin{cases} q_{ij} = p_{ij} & | (\omega_{ik} > 0); \\ q_{ij} = 0 & | (\omega_{ik} = 0). \end{cases}$$

$\gamma_2 = R$  – дисципліни належать к множині дисциплін, обраних студентом.

Множина належності дисципліни до індивідуального плану студента:

$$R \subseteq D \left| p_i = \sum_{k=1}^{s_s} q_{ik} > 0 \begin{cases} q_{ik} = p_{ik} & | \omega_{ik} > 0 \\ q_{ik} = 0 & | \omega_{ik} = 0 \end{cases} \text{ для } i = \overline{1, s_d}.$$

$\gamma_3 = J$  – певний вид дисципліни, який викладається одним або декількома викладачами. Множина належності дисципліни до індивідуального плану роботи викладача визначається:

$$\tau_{ijk} \in \{0, 1\}, \text{ де } i = \overline{1, s_d} \quad j = \overline{1, 7} \quad k = \overline{1, s_t} : J \subseteq D \left| \sum_{k=1}^{s_t} \tau_{ijk} > 0 \right| p_{ij} > 0$$

Отже, при існуючому аудиторному навантаженню  $p_{ij}$  будь-якого типу занять для всіх дисциплін визначаються відповідно до розподілу одного або декількох викладачів для проведення  $ij$ -того заняття.

$\gamma_4 = B = B_1 \cup B_2 \cup B_3$  – поділ дисциплін за видом занять відповідно до кратності неділь де  $B_1$  – множина щотижневих занять,  $B_2$  – множина занять, які відбуваються раз на дві неділі,  $B_3$  – множина занять дисциплін, кількість яких некратно половинній кількості тижнів.

$\gamma_5 = H$  – ділення типів занять дисциплін, враховуючи обмеження, які накладаються психолого-педагогічними особливостями викладача і сприйняттям дисциплін студентами  $h_k \in \{-20, \dots, 20\}$ ,  $k = \overline{1, s_h}$ , де  $s_h$  – кількість таких обмежень.

Цільова функція автоматизованого формування розкладу  $\delta$  з урахуванням необхідних умов з максимальною кількістю балів:

$$\delta = \max \left( \sum_{k=1}^{s_t} w_k \sum_{j=1}^{s_x} \frac{v_{kj} \pi_{kj}}{\sum_{i=1}^{s_x} \pi_{ki}} + \sum_{k=1}^{s_h} n_k h_k \right)$$

де  $v_{kj} \in \{0, 1\}$ ,  $j = \overline{1, s_x}$ ,  $k = \overline{1, s_x}$  – Ознаки врахування додаткової умови для викладача відповідної дисципліни,  $\pi_{kj}$  – пріоритет реалізованої умови,  $w_k$  – коефіцієнт рейтингу викладача,  $\pi_{ki}$  – пріоритети комплексу різних умов для відповідного викладача.

## 4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 4.1 Розробка моделей автоматизованої системи формування розкладу

Модель автоматизованої системи (рис. 3) є структурним підпорядкуванням загальної автоматизованої системи ведення документообігу відображає функціональний взаємозв'язок та можливість узгодженого використання загальних ресурсів.

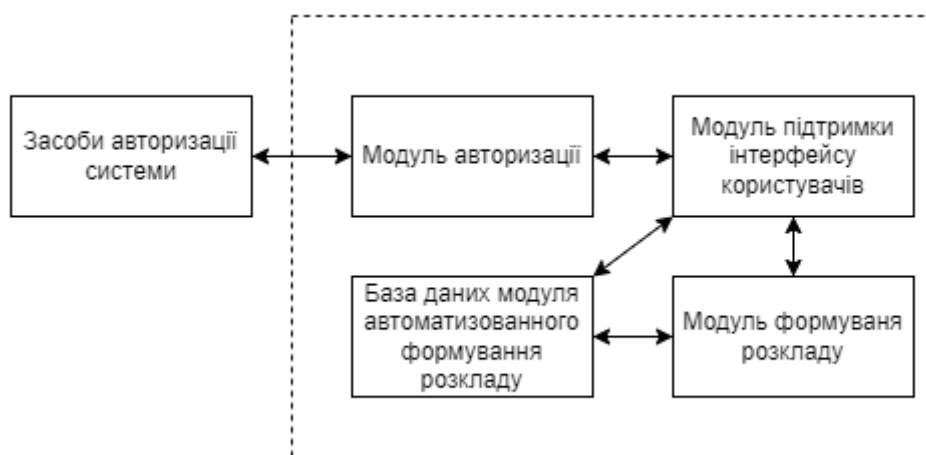


Рис. 3. Загальна модель автоматизованої системи формування розкладу

Результатом аналізу предметної області є складання структурної схеми даних, яка відображає основні зв'язки між інформативними таблицями.

Програму можна легко описати як систему, яка приймає набори даних та в результаті роботи алгоритму виводить структурований результат (Рис.4).

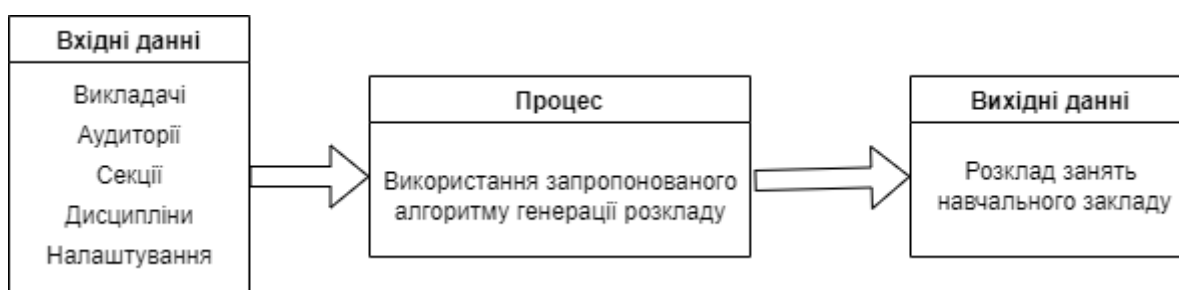


Рис. 4. Спрощення IPO-модель (Input-Process-Model) програми

Для розробки ПО була використана спрощена математична модель, для якої були зроблені такі припущення:

$$D_i = \{L_1, L_2\}$$

$L_{i1}$  – лекційні заняття;

$L_{i2}$  – лабораторні заняття;

$$S = \emptyset$$

Додаткові ознаки та рейтинг викладача не враховуються. Коефіцієнти набувають константних значень для усіх дисциплін, предметів та викладачів.

#### 4.2 Розробка програмного забезпечення системи

Для розробки інформаційної системи було використано мову програмування Python версії 3.8. Додатково використовувалися бібліотеки NumPy, PyQt5, psutil, sqlite3. Для демонстрації працездатності алгоритму була використана спрощена модель, в якій враховувалися не всі фактори.

Вхідні данні, які враховуються при складенні розкладу:

- Викладачі (навантаження викладача, список предметів які він може викладати, доступні робочі години)
- Аудиторії (тип аудиторії: лекційна/лабораторна)
- Секції (предмети, які до неї відносяться та викладачі)
- Дисципліни (перелік викладачів, які можуть вести дисципліну, кількість годин, потреба в лабораторних або лекційних аудиторіях)
- Налаштування (години роботи університету)

PyQt5 - це набір бібліотек Python для створення графічного інтерфейсу на базі платформ Qt5 від компанії Digia. Він доступний для Python 2.x і 3.x. Бібліотека Qt є однією з найпотужніших бібліотек GUI (графічний інтерфейс користувача).

Функціонал даної бібліотеки дуже широкий, але нам він знадобиться для створення користувацького інтерфейсу. Як приклад використання бібліотеки головний клас GUI додатку (Додаток В).

Для зберігання введених користувачем даних використовується бібліотека sqlite3 (Рис.5). Головна перевага даної СУБД (Система Управління Базами Даних) є її інтегрування в python. SQLite не використовує парадигму клієнт-сервер, тобто не є окремим процесом, з якою взаємодіє програма, а є бібліотекою, з якою програма компонується і стає складовою частиною програми. Тобто в якості протоколу обміну використовуються виклики функцій. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму. Дані зберігаються в файлі файлової системи, на якій виконується програма. Головний клас підключається до бази даних та ініціалізує її (Додаток Г).

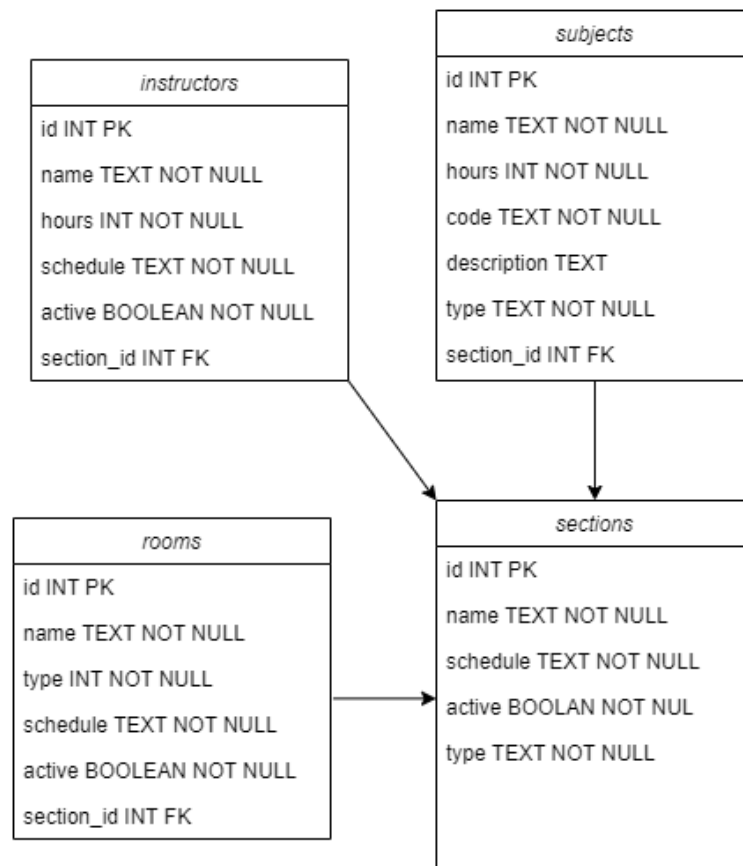


Рис. 5. Схема бази даних SQLite



### 4.3 Приклад використання програмного продукту

Для прикладу використаємо обмежену вибірку з кафедри Інформаційних наук Сумського Державного Університету. Для початку необхідно заповнити викладацький склад. На рис.6 зображено початкову сторінку з заповненими даними.

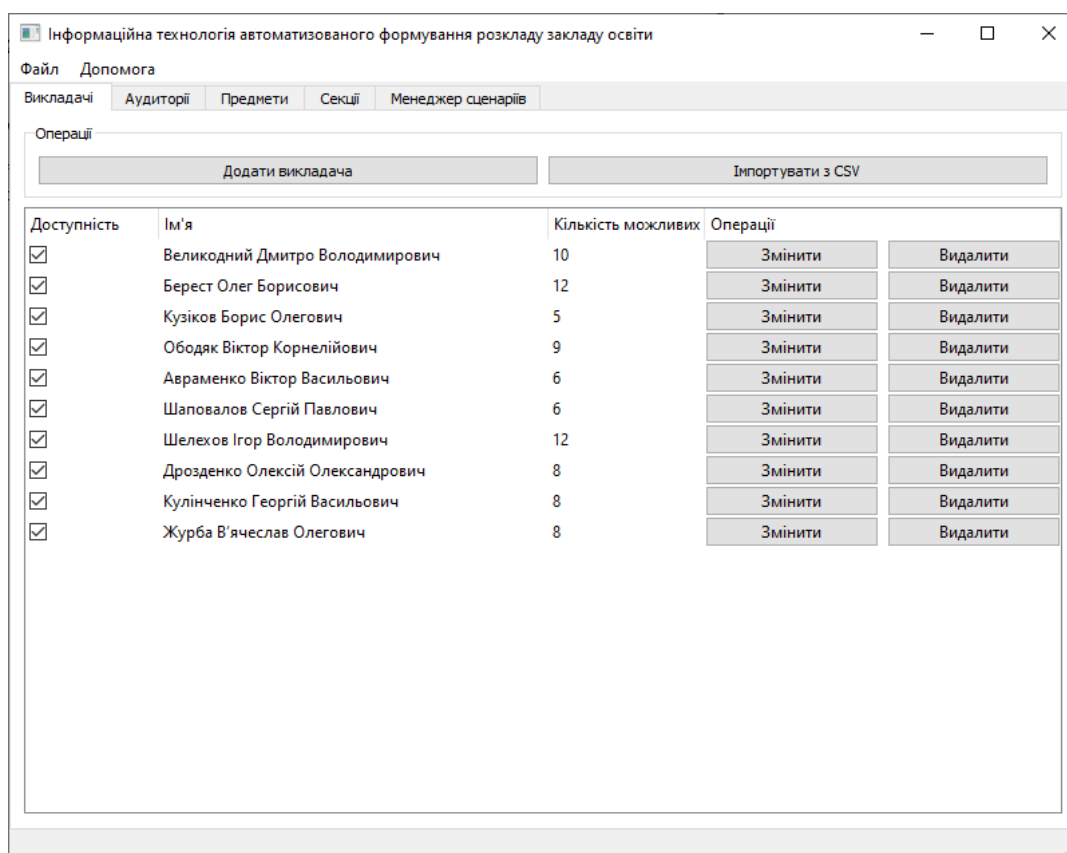


Рис. 6. Вкладка «Викладачі» системи автоматизованого формування розкладу

При додаванні нового викладача необхідно вказати ППІ викладача, кількість доступних годин в неділю та зазначити в які дні та години викладач може проводити заняття. Інтерфейс додавання нового викладача зображено на рис. 7.

Викладач

Ім'я:  Кількість доступних занять:

	Понеділок	Вівторок	Середа	Четвер	П'ятниця	Суббота
8:30-9:50	Доступно	Доступно	Доступно	Доступно	Недоступно	Недоступно
10:05-11:25	Доступно	Доступно	Доступно	Недоступно	Недоступно	Недоступно
11:40-13:00	Недоступно	Доступно	Доступно	Доступно	Недоступно	Недоступно
14:00-15:20	Доступно	Доступно	Доступно	Доступно	Недоступно	Недоступно
15:35-16:55	Доступно	Доступно	Доступно	Доступно	Недоступно	Недоступно
17:10-18:30	Недоступно	Доступно	Недоступно	Доступно	Недоступно	Недоступно

Рис. 7. Вікно додавання/редагування викладача

Далі необхідно додати аудиторії, вказати в які години там можуть поводитися пари та обрати тип аудиторії. На рис. 8 зображена вкладка «Аудиторії» з вибіркою аудиторій, в якій можна змінити данні про аудиторію, видалити її або зробити тимчасово недоступною. На рис. 9 зображено вікно додавання/редагування аудиторії.

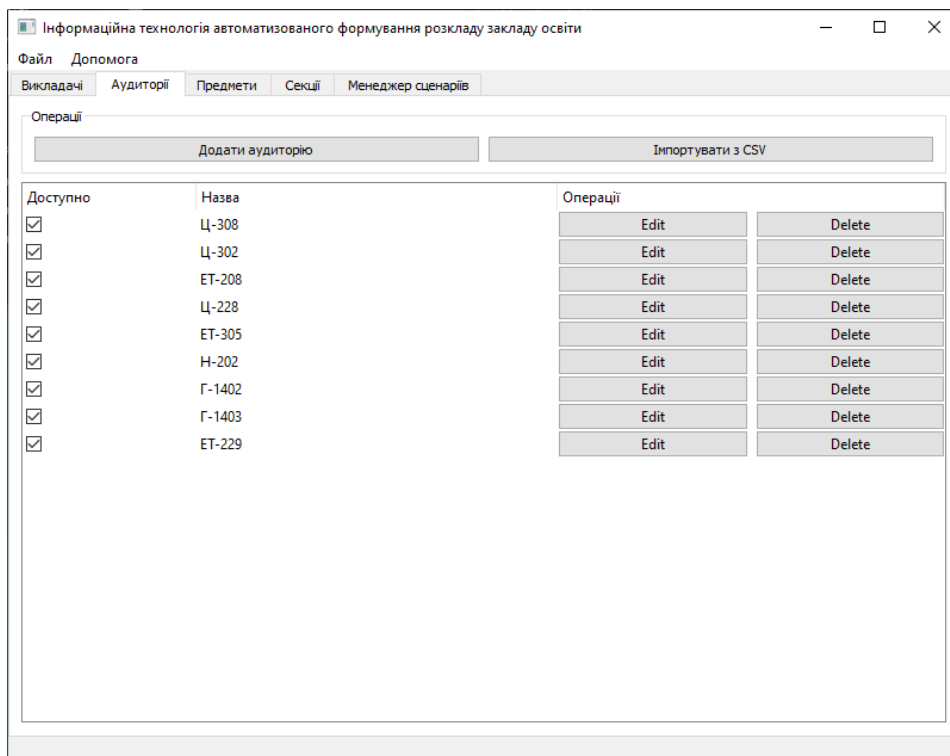


Рис. 8 Вкладка «Аудиторії» системи автоматизованого формування розкладу

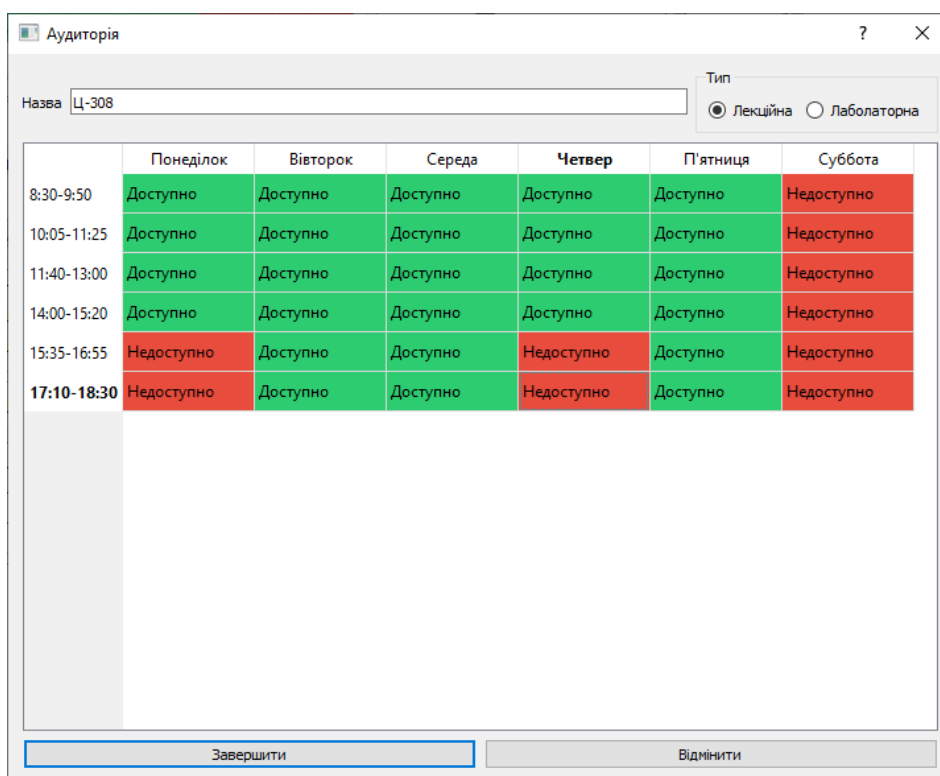


Рис. 9. Вікно додавання/редагування аудиторії

Наступним кроком необхідно додати список дисциплін у вкладці «Предмети», вказати їх назву, код, кількість годин, обрати тип заняття та, опціонально, вказати короткий опис дисципліни. На рис. 10 зображена вкладка «Предмети», в якій відображається список доданих дисциплін з можливістю редагувати та видаляти їх. На рис. 11 зображено вікно додавання/редагування дисципліни.

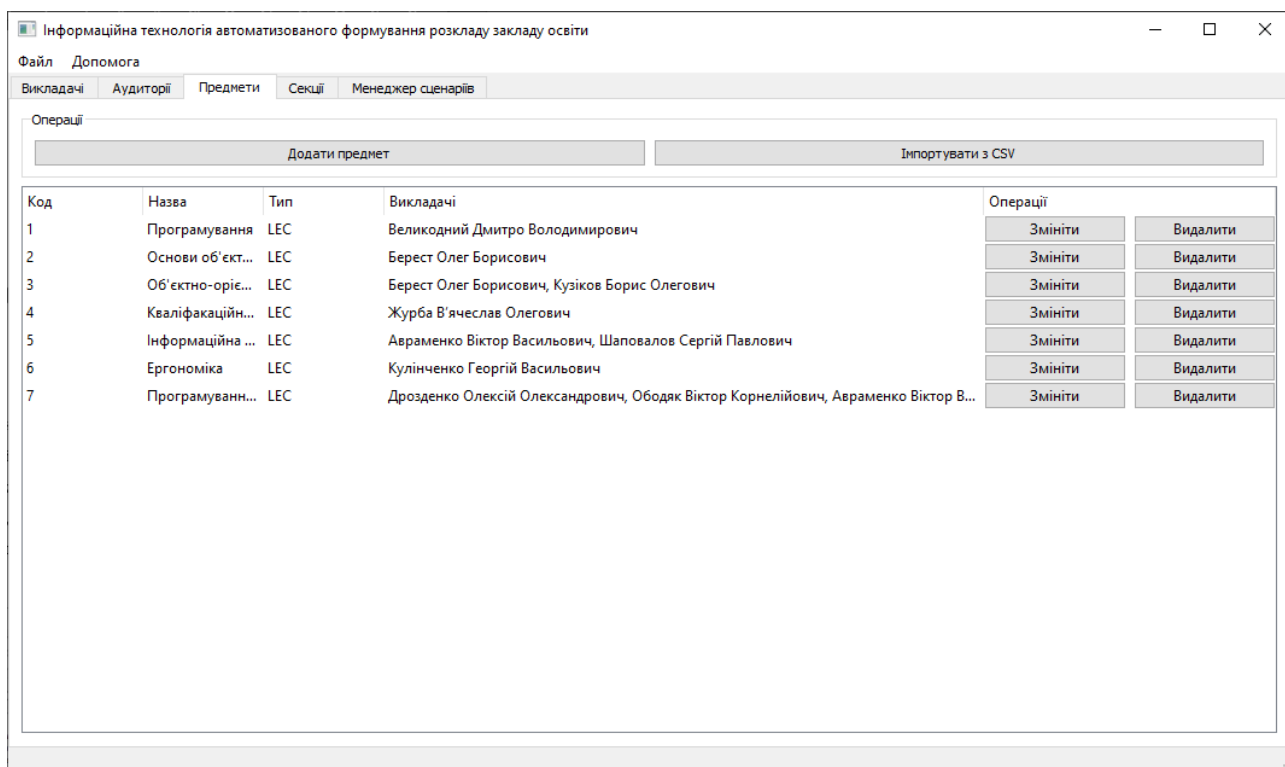


Рис. 10 Вкладка «Предмети» системи автоматизованого формування розкладу

Предмет

Назва:  Кількість занять в неділю:

Код:  Опис:

Поділити розклад:  Так  Ні

Тип:  Будь-який  Лекція  Лабораторне заняття

Available	Name
<input checked="" type="checkbox"/>	Великодний Дмитро Володимирович
<input checked="" type="checkbox"/>	Берест Олег Борисович
<input type="checkbox"/>	Кузіков Борис Олегович
<input type="checkbox"/>	Ободяк Віктор Корнелійович
<input checked="" type="checkbox"/>	Авраменко Віктор Васильович
<input type="checkbox"/>	Шаповалов Сергій Павлович
<input type="checkbox"/>	Шелехов Ігор Володимирович
<input type="checkbox"/>	Дрозденко Олексій Олександрович
<input type="checkbox"/>	Кулінченко Георгій Васильович
<input type="checkbox"/>	Журба В'ячеслав Олегович

Закрити

Рис. 11 Вікно додавання/редагування дисципліни

Далі можна додати секції (опціонально), на які будуть ділитися дисципліни. Існує можливість перевести всю секцію на дистанційне навчання замість того, щоб вказувати окремо для кожної дисципліни. На рис. 10 зображена вкладка «Секції», в якій відображається список існуючих секцій, їх доступність, чи є секція дистанційною та можливість редагувати та видалити існуючі секції. На рис. 13 зображено вікно додавання/редагування секції.

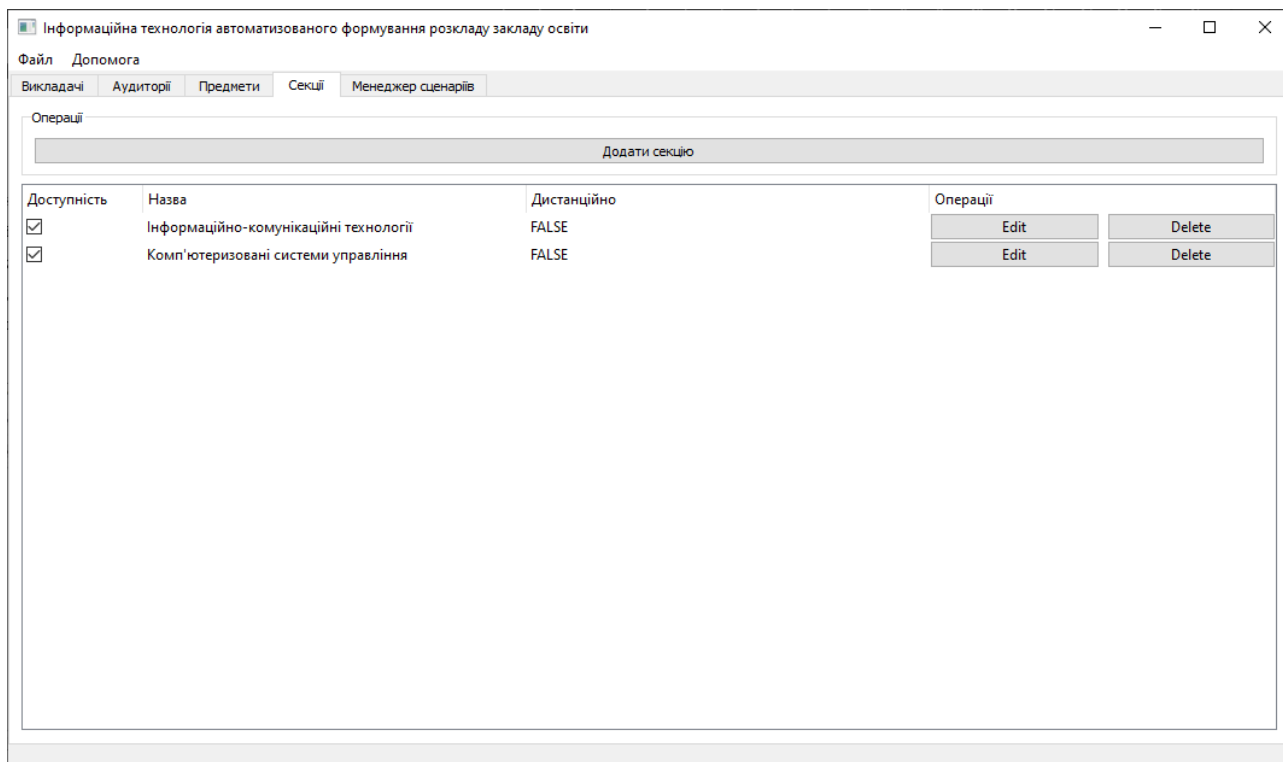


Рис. 12 Вкладка «Секції» системи автоматизованого формування розкладу

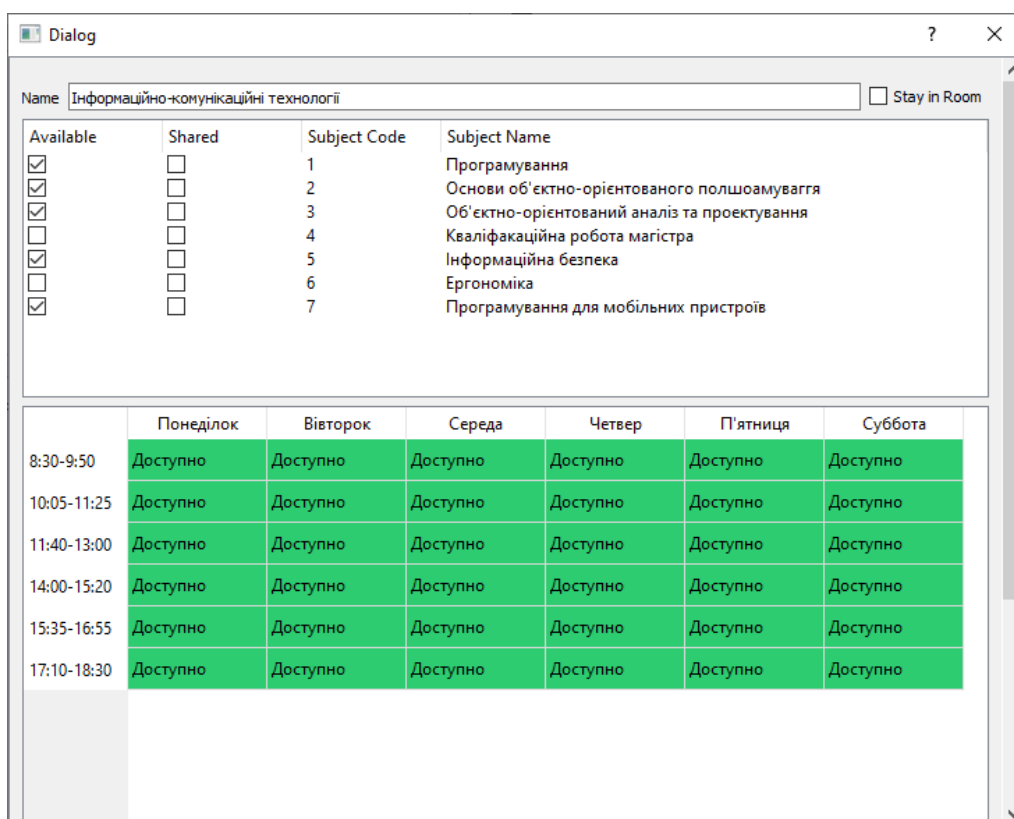


Рис. 13 Вікно додавання/редагування секцій

Після заповнення додатку усіма необхідними даними потрібно перейти у вкладку «Менеджер сценаріїв», в якій з'явиться можливість сгенерувати та переглянути новий розклад. Також можна (опціонально) вказати час початку та кінця навчання. На рис. 14 зображена вкладка «Менеджер сценаріїв» з можливими налаштуваннями програми.

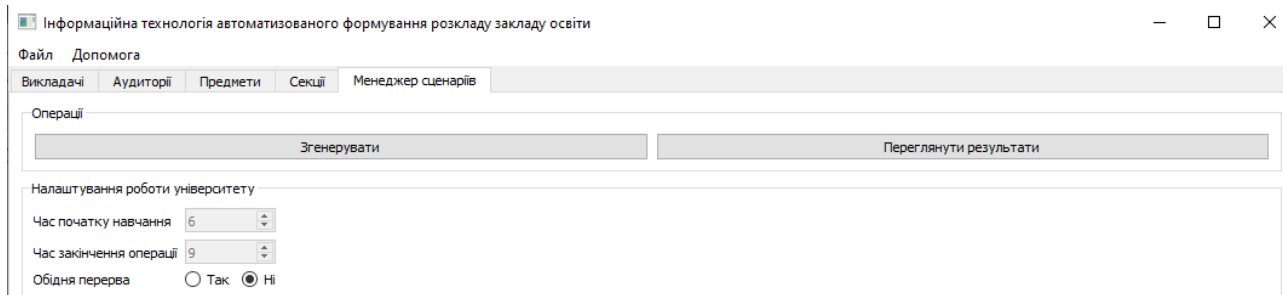


Рис. 14 Вкладка «Менеджер сценаріїв» системи автоматизованого формування розкладу

Існує 2 способи заповнення програми даними: вручну або імпортувати з CSV файлу. В додатку А наведено приклад даних з правильним форматуванням, які можна імпортувати.

## ВИСНОВОК

За результатами аналізу наявних програмних продуктів, що надають можливість створити розклад навчального закладу, було виявлено ряд проблем у їх функціональності та комфорти роботи з ними.

Було вирішено створити програмний продукт, який в автоматизованому режимі буде створювати зручний розклад для навчального закладу.

Проаналізовано існуючі алгоритми вирішення поставленої задачі та обрано алгоритм, заснований на вирішенні задачі нелінійного програмування.

Вирішено розробити десктоп додаток, основними вимогами якої буде кросплатформність та зручність використання.

Інформаційну вирішено написати на мові програмування Python 3 через ряд переваг:

- простота та швидкість написання коду,
- якісні бібліотеки, які є зручними в використанні,
- динамічна типізація,
- кросплатформність,
- якісна документація,
- велика кількість інтернет-ресурсів з прикладами використання бібліотек.

Розроблений додаток задовольняє всім вимогам, поставленим на етапі постановки задачі.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розклад ПРО // DigSee Ltd [Електронний ресурс]. Режим доступу: <http://www.digsee.com/rus/timetable/>
2. Мельников А. Ю., Сусяк Н. М. Комп'ютерне моделювання в освіті: Матеріали Всеукраїнського науково-методичного семінару, Кривий Ріг, 29 березня 2005 р. – Кривий Ріг: КДПУ, 2005. – С. 52-53.
3. Мокін В. Б., Бевз С.В., Мацко Л. А. Застосування інноваційних та інформаційних технологій у навчанні магістрантів у ВНТУ (з досвіду роботи) / Зб. матеріалів VIII Міжнародної науково-практичної конференції “Гуманізм та освіта”. – Вінниця: УНІВЕРСУМ-Вінниця, 2006. — С. 22-25.
4. Баришніков А. В. Ректор – програма для складання розкладу уроків // Комп'ютерні інструменти в освіті. – Санкт-Петербург. – № 5. – 1998. – ([www.aec.neva.ru/journal](http://www.aec.neva.ru/journal)).
5. Е.В.Панкратьев, А.М. Чеповський, Е.А.Черепанов, С.В.Чернишев - Алгоритми та методи вирішення задач складання розкладу та других екстремальних задач на графах великих розмірностей.
6. Лазарев А. А. - Теорія розкладів. Оцінки абсолютної похибки та схема наближеного розв'язання задач теорії розкладів. 2008. — 222 с.
7. S. Zheng, Y. Liang, S. Wang, R. Chen, K. Sheng - FlexTensor: An Automatic Schedule Exploration and Optimization Framework for Tensor Computation on Heterogeneous System, 2020
8. J. Wang and H. Yuan - System Dynamics Approach for Investigating the Risk Effects on Schedule Delay in Infrastructure Projects - Journal of Management in Engineering, 2017
9. HS. Moon, N. Dawood, LS. Kang - Development of workspace conflict visualization system using 4D object of work schedule - Advanced Engineering Informatics, 2014

10. HS. Moon, HS. Kim, CH. Kim, LS. Kang - Development of a schedule-workspace interference management system simultaneously considering the overlap level of parallel schedules and workspaces - Automation in Construction, 2014
11. Raghavan R., Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs
12. solutions.1c.ru [Электронный ресурс] : [Интернет-портал]. – Режим доступа: [https://solutions.1c.ru/catalog/asp\\_univer/features](https://solutions.1c.ru/catalog/asp_univer/features) [ (дата звернення 01.11.2021). – Назва з екрана.

## ДОДАТКИ

### Додаток А. Приклади вхідних даних з CSV файлу

Викладачі:

Ім'я	Кількість можливих занять	пн	вт	ср	чт	пт	сб
Великодний Дмитро Володимирович	10		1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
Берест Олег Борисович	12	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3	1,2,3,4,5,6	
Кузіков Борис Олегович	5	4,5,6	1,2,3,4,5,6		1,2,4	1,2,3,4,5,6	1,2,3,4,5,6
Ободяк Віктор Корнелійович	9	1,2,3,4,5,6			1,2,3,4,5,6		1,2,3,4,5,6
Авраменко Віктор Васильович	6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6		1,2,3,4,5,6
Шаповалов Сергій Павлович	6	1,2,3,4,5,6	4,5,6	1,2,3,4,5,6		1,2,3,4,5,6	1,2,3,4,5,6
Шелехов Ігор Володимирович	12	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3
Дрозденко Олексій Олександрович	8	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6		1,2,3,4,5,6	1,2,3,4,5,6
Кулінченко Георгій Васильович	8			1,2,3,4,5,6	1,2,3,4,5,6	4,5,6	1,2,3,4,5,6
Журба В'ячеслав Олегович	8		1,2,3,4,5,6	1,2,3,4,5,6		1,2,3,4,5,6	

Аудиторії:

Назва	Тип аудиторії	пн	вт	ср	чт	пт	сб
Ц-308	Лекційна		1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
Ц-302	Лекційна	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3	1,2,3,4,5,6	
ЕТ-208	Лаборатона	4,5,6	1,2,3,4,5,6		1,2,4	1,2,3,4,5,6	1,2,3,4,5,6
Ц-228	Лаборатона	1,2,3,4,5,6			1,2,3,4,5,6		1,2,3,4,5,6
ЕТ-305	Лекційна	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6		1,2,3,4,5,6
Н-202	Лекційна	1,2,3,4,5,6	4,5,6	1,2,3,4,5,6		1,2,3,4,5,6	1,2,3,4,5,6
Г-1402	Лаборатона	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3
Г-1403	Лекційна	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6		1,2,3,4,5,6	1,2,3,4,5,6
ЕТ-229	Лекційна			1,2,3,4,5,6	1,2,3,4,5,6	4,5,6	1,2,3,4,5,6







	Великодний Дмитро Володимирович	Берест Олег Борисович	Кузіков Борис Олегович	Ободяк Віктор Корнелійович	Авраменко Віктор Васильович	Шаповалов Сергій Павлович	Шелехов Ігор Володимирович	Дрозденко Олексій Олександрович	Кулінченко Георгій Васильович	Журба В'ячеслав Олегович
Субота 1 урок (8:30-9:50)			Інформаційна безпека Н-202					Мови програмування інтелектуальних систем ET-208		
Субота 2 урок (10:05-11:25)			Інформаційна безпека Н-202					Мови програмування інтелектуальних систем ET-208		
Субота 3 урок (11:40-13:00)										
Субота 4 урок (14:00-15:20)										
Субота 5 урок (15:35-16:55)										
Субота 6 урок (17:10-18:30)										

## Дотаток В. Приклад GUI класу

```
from PyQt5 import QtCore, QtGui, QtWidgets
```

```
class Ui_MainWindow(object):
```

```
    def setupUi(self, MainWindow):
```

```
        MainWindow.setObjectName("MainWindow")
```

```
        MainWindow.setWindowModality(QtCore.Qt.NonModal)
```

```
        MainWindow.resize(800, 600)
```

```
        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Maximum,
QtWidgets.QSizePolicy.Maximum)
```

```
        sizePolicy.setHorizontalStretch(0)
```

```
        sizePolicy.setVerticalStretch(0)
```

```
        sizePolicy.setHeightForWidth(MainWindow.sizePolicy().hasHeightForWidth())
```

```
        MainWindow.setSizePolicy(sizePolicy)
```

```
        MainWindow.setMinimumSize(QtCore.QSize(800, 600))
```

```
        MainWindow.setMaximumSize(QtCore.QSize(16777215, 16777215))
```

```
        self.centralWidget = QtWidgets.QWidget(MainWindow)
```

```
        self.centralWidget.setMinimumSize(QtCore.QSize(711, 526))
```

```
        self.centralWidget.setObjectName("centralWidget")
```

```
        self.verticalLayout = QtWidgets.QVBoxLayout(self.centralWidget)
```

```
        self.verticalLayout.setContentsMargins(0, 0, 0, 0)
```

```
self.verticalLayout.setSpacing(0)
self.verticalLayout.setObjectName("verticalLayout")
self.tabVidger = QtWidgets.QTabVidger(self.centralVidger)
font = QtGui.QFont()
font.setBold(False)
font.setWeight(50)
self.tabVidger.setFont(font)
self.tabVidger.setStyleSheet("")
self.tabVidger.setTabPosition(QtWidgets.QTabVidger.North)
self.tabVidger.setTabShape(QtWidgets.QTabVidger.Rounded)
self.tabVidger.setObjectName("tabVidger")
self.tabInstructors = QtWidgets.QVidger()
self.tabInstructors.setObjectName("tabInstructors")
self.verticalLayout_2 = QtWidgets.QVBoxLayout(self.tabInstructors)
self.verticalLayout_2.setContentsMargins(9, 9, 9, 9)
self.verticalLayout_2.setObjectName("verticalLayout_2")
self.groupBox = QtWidgets.QGroupBox(self.tabInstructors)
self.groupBox.setObjectName("groupBox")
self.horizontalLayout = QtWidgets.QHBoxLayout(self.groupBox)
self.horizontalLayout.setObjectName("horizontalLayout")
self.btnInstrAdd = QtWidgets.QPushButton(self.groupBox)
self.btnInstrAdd.setObjectName("btnInstrAdd")
self.horizontalLayout.addVidger(self.btnInstrAdd)
self.btnInstrImport = QtWidgets.QPushButton(self.groupBox)
self.btnInstrImport.setObjectName("btnInstrImport")
self.horizontalLayout.addVidger(self.btnInstrImport)
self.verticalLayout_2.addVidger(self.groupBox)
self.treeInstr = QtWidgets.QTreeView(self.tabInstructors)
self.treeInstr.setObjectName("treeInstr")
self.verticalLayout_2.addVidger(self.treeInstr)
self.tabVidger.addTab(self.tabInstructors, "")
self.tabRooms = QtWidgets.QVidger()
self.tabRooms.setObjectName("tabRooms")
self.verticalLayout_3 = QtWidgets.QVBoxLayout(self.tabRooms)
self.verticalLayout_3.setContentsMargins(9, 9, 9, 9)
self.verticalLayout_3.setObjectName("verticalLayout_3")
```



```
self.groupBox_2 = QtWidgets.QGroupBox(self.tabRooms)
self.groupBox_2.setObjectName("groupBox_2")
self.horizontalLayout_7 = QtWidgets.QHBoxLayout(self.groupBox_2)
self.horizontalLayout_7.setObjectName("horizontalLayout_7")
self.btnRoomAdd = QtWidgets.QPushButton(self.groupBox_2)
self.btnRoomAdd.setObjectName("btnRoomAdd")
self.horizontalLayout_7.addWidget(self.btnRoomAdd)
self.btnRoomImport = QtWidgets.QPushButton(self.groupBox_2)
self.btnRoomImport.setObjectName("btnRoomImport")
self.horizontalLayout_7.addWidget(self.btnRoomImport)
self.verticalLayout_3.addWidget(self.groupBox_2)
self.treeRoom = QtWidgets.QTreeView(self.tabRooms)
self.treeRoom.setObjectName("treeRoom")
self.verticalLayout_3.addWidget(self.treeRoom)
self.tabWidget.addTab(self.tabRooms, "")
self.tabSubjects = QtWidgets.QWidget()
self.tabSubjects.setObjectName("tabSubjects")
self.verticalLayout_4 = QtWidgets.QVBoxLayout(self.tabSubjects)
self.verticalLayout_4.setContentsMargins(9, 9, 9, 9)
self.verticalLayout_4.setObjectName("verticalLayout_4")
self.groupBox_3 = QtWidgets.QGroupBox(self.tabSubjects)
self.groupBox_3.setObjectName("groupBox_3")
self.horizontalLayout_2 = QtWidgets.QHBoxLayout(self.groupBox_3)
self.horizontalLayout_2.setObjectName("horizontalLayout_2")
self.btnSubjAdd = QtWidgets.QPushButton(self.groupBox_3)
self.btnSubjAdd.setObjectName("btnSubjAdd")
self.horizontalLayout_2.addWidget(self.btnSubjAdd)
self.btnSubjImport = QtWidgets.QPushButton(self.groupBox_3)
self.btnSubjImport.setObjectName("btnSubjImport")
self.horizontalLayout_2.addWidget(self.btnSubjImport)
self.verticalLayout_4.addWidget(self.groupBox_3)
self.treeSubj = QtWidgets.QTreeView(self.tabSubjects)
self.treeSubj.setObjectName("treeSubj")
self.verticalLayout_4.addWidget(self.treeSubj)
self.tabWidget.addTab(self.tabSubjects, "")
self.tabSections = QtWidgets.QWidget()
```

```

self.tabSections.setObjectName("tabSections")
self.verticalLayout_5 = QtWidgets.QVBoxLayout(self.tabSections)
self.verticalLayout_5.setContentsMargins(9, 9, 9, 9)
self.verticalLayout_5.setObjectName("verticalLayout_5")
self.groupBox_4 = QtWidgets.QGroupBox(self.tabSections)
self.groupBox_4.setObjectName("groupBox_4")
self.horizontalLayout_3 = QtWidgets.QHBoxLayout(self.groupBox_4)
self.horizontalLayout_3.setObjectName("horizontalLayout_3")
self.btnSecAdd = QtWidgets.QPushButton(self.groupBox_4)
self.btnSecAdd.setObjectName("btnSecAdd")
self.horizontalLayout_3.addWidget(self.btnSecAdd)
self.verticalLayout_5.addWidget(self.groupBox_4)
self.treeSec = QtWidgets.QTreeView(self.tabSections)
self.treeSec.setObjectName("treeSec")
self.verticalLayout_5.addWidget(self.treeSec)
self.tabWidget.addTab(self.tabSections, "")
self.tabScenario = QtWidgets.QWidget()
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred,
QtWidgets.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.tabScenario.sizePolicy().hasHeightForWidth())
self.tabScenario.setSizePolicy(sizePolicy)
self.tabScenario.setObjectName("tabScenario")
self.verticalLayout_6 = QtWidgets.QVBoxLayout(self.tabScenario)
self.verticalLayout_6.setSizeConstraint(QtWidgets.QLayout.SetDefaultConstraint)
self.verticalLayout_6.setContentsMargins(9, 9, 9, 9)
self.verticalLayout_6.setObjectName("verticalLayout_6")
self.groupBox_5 = QtWidgets.QGroupBox(self.tabScenario)
self.groupBox_5.setObjectName("groupBox_5")
self.horizontalLayout_4 = QtWidgets.QHBoxLayout(self.groupBox_5)
self.horizontalLayout_4.setObjectName("horizontalLayout_4")
self.btnScenGenerate = QtWidgets.QPushButton(self.groupBox_5)
self.btnScenGenerate.setObjectName("btnScenGenerate")
self.horizontalLayout_4.addWidget(self.btnScenGenerate)
self.btnScenResult = QtWidgets.QPushButton(self.groupBox_5)

```

```

self.btnScenResult.setObjectName("btnScenResult")
self.horizontalLayout_4.addWidget(self.btnScenResult)
self.verticalLayout_6.addWidget(self.groupBox_5)
self.groupBox_6 = QtWidgets.QGroupBox(self.tabScenario)
self.groupBox_6.setObjectName("groupBox_6")
self.gridLayout_2 = QtWidgets.QGridLayout(self.groupBox_6)
self.gridLayout_2.setObjectName("gridLayout_2")
self.groupBox_7 = QtWidgets.QGroupBox(self.groupBox_6)
self.groupBox_7.setStyleSheet("border: none")
self.groupBox_7.setTitle("")
self.groupBox_7.setFlat(False)
self.groupBox_7.setCheckable(False)
self.groupBox_7.setChecked(False)
self.groupBox_7.setObjectName("groupBox_7")
self.horizontalLayout_5 = QtWidgets.QHBoxLayout(self.groupBox_7)
self.horizontalLayout_5.setContentsMargins(0, 0, 0, 0)
self.horizontalLayout_5.setSpacing(10)
self.horizontalLayout_5.setObjectName("horizontalLayout_5")
self.radioLunchYes = QtWidgets.QRadioButton(self.groupBox_7)
self.radioLunchYes.setChecked(True)
self.radioLunchYes.setObjectName("radioLunchYes")
self.horizontalLayout_5.addWidget(self.radioLunchYes)
self.radioLunchNo = QtWidgets.QRadioButton(self.groupBox_7)
self.radioLunchNo.setObjectName("radioLunchNo")
self.horizontalLayout_5.addWidget(self.radioLunchNo)
self.gridLayout_2.addWidget(self.groupBox_7, 2, 1, 1, 1)
self.label_6 = QtWidgets.QLabel(self.groupBox_6)
self.label_6.setObjectName("label_6")
self.gridLayout_2.addWidget(self.label_6, 2, 0, 1, 1)
self.label_4 = QtWidgets.QLabel(self.groupBox_6)
self.label_4.setObjectName("label_4")
self.gridLayout_2.addWidget(self.label_4, 0, 0, 1, 1)
self.timeStarting = QtWidgets.QTimeEdit(self.groupBox_6)
self.timeStarting.setCurrentSection(QtWidgets.QDateTimeEdit.HourSection)
self.timeStarting.setTime(QtCore.QTime(8, 0, 0))
self.timeStarting.setObjectName("timeStarting")

```

```

self.gridLayout_2.addWidget(self.timeStarting, 0, 1, 1, 1)
spacerItem = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Minimum)
self.gridLayout_2.addItem(spacerItem, 0, 2, 1, 1)
self.label_5 = QtWidgets.QLabel(self.groupBox_6)
self.label_5.setObjectName("label_5")
self.gridLayout_2.addWidget(self.label_5, 1, 0, 1, 1)
self.timeEnding = QtWidgets.QTimeEdit(self.groupBox_6)
self.timeEnding.setCalendarPopup(True)
self.timeEnding.setTime(QtCore.QTime(18, 0, 0))
self.timeEnding.setObjectName("timeEnding")
self.gridLayout_2.addWidget(self.timeEnding, 1, 1, 1, 1)
self.verticalLayout_6.addWidget(self.groupBox_6)
self.groupBox_8 = QtWidgets.QGroupBox(self.tabScenario)
self.groupBox_8.setObjectName("groupBox_8")
self.gridLayout_3 = QtWidgets.QGridLayout(self.groupBox_8)
self.gridLayout_3.setObjectName("gridLayout_3")
self.label_20 = QtWidgets.QLabel(self.groupBox_8)
self.label_20.setObjectName("label_20")
self.gridLayout_3.addWidget(self.label_20, 2, 3, 1, 1)
self.label_21 = QtWidgets.QLabel(self.groupBox_8)
self.label_21.setObjectName("label_21")
self.gridLayout_3.addWidget(self.label_21, 3, 3, 1, 1)
self.editDev = QtWidgets.QSpinBox(self.groupBox_8)
self.editDev.setMaximum(100)
self.editDev.setProperty("value", 55)
self.editDev.setObjectName("editDev")
self.gridLayout_3.addWidget(self.editDev, 3, 5, 1, 1)
self.editMaxPop = QtWidgets.QSpinBox(self.groupBox_8)
self.editMaxPop.setMinimum(50)
self.editMaxPop.setMaximum(10000)
self.editMaxPop.setProperty("value", 100)
self.editMaxPop.setObjectName("editMaxPop")
self.gridLayout_3.addWidget(self.editMaxPop, 1, 1, 1, 1)
self.label_12 = QtWidgets.QLabel(self.groupBox_8)
self.label_12.setObjectName("label_12")

```

```
self.gridLayout_3.addWidget(self.label_12, 1, 3, 1, 1)
self.editElite = QtWidgets.QSpinBox(self.groupBox_8)
self.editElite.setMaximum(100)
self.editElite.setProperty("value", 5)
self.editElite.setObjectName("editElite")
self.gridLayout_3.addWidget(self.editElite, 2, 5, 1, 1)
spacerItem1 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Minimum)
self.gridLayout_3.addItem(spacerItem1, 0, 7, 1, 1)
self.editMaxGen = QtWidgets.QSpinBox(self.groupBox_8)
self.editMaxGen.setMinimum(50)
self.editMaxGen.setMaximum(10000)
self.editMaxGen.setObjectName("editMaxGen")
self.gridLayout_3.addWidget(self.editMaxGen, 2, 1, 1, 1)
self.editMaxFit = QtWidgets.QSpinBox(self.groupBox_8)
self.editMaxFit.setMaximum(100)
self.editMaxFit.setProperty("value", 90)
self.editMaxFit.setObjectName("editMaxFit")
self.gridLayout_3.addWidget(self.editMaxFit, 1, 5, 1, 1)
self.label_11 = QtWidgets.QLabel(self.groupBox_8)
self.label_11.setObjectName("label_11")
self.gridLayout_3.addWidget(self.label_11, 0, 3, 1, 1)
self.label_10 = QtWidgets.QLabel(self.groupBox_8)
self.label_10.setObjectName("label_10")
self.gridLayout_3.addWidget(self.label_10, 3, 0, 1, 1)
self.label_9 = QtWidgets.QLabel(self.groupBox_8)
self.label_9.setObjectName("label_9")
self.gridLayout_3.addWidget(self.label_9, 2, 0, 1, 1)
self.editMaxCreation = QtWidgets.QSpinBox(self.groupBox_8)
self.editMaxCreation.setMinimum(1500)
self.editMaxCreation.setMaximum(30000)
self.editMaxCreation.setObjectName("editMaxCreation")
self.gridLayout_3.addWidget(self.editMaxCreation, 3, 1, 1, 1)
self.label_7 = QtWidgets.QLabel(self.groupBox_8)
self.label_7.setObjectName("label_7")
self.gridLayout_3.addWidget(self.label_7, 0, 0, 1, 1)
```

```

self.label_8 = QtWidgets.QLabel(self.groupBox_8)
self.label_8.setObjectName("label_8")
self.gridLayout_3.addWidget(self.label_8, 1, 0, 1, 1)
self.editMinPop = QtWidgets.QSpinBox(self.groupBox_8)
self.editMinPop.setMinimum(50)
self.editMinPop.setMaximum(10000)
self.editMinPop.setObjectName("editMinPop")
self.gridLayout_3.addWidget(self.editMinPop, 0, 1, 1, 1)
spacerItem2 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Minimum)
self.gridLayout_3.addItem(spacerItem2, 0, 11, 1, 1)
spacerItem3 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Minimum)
self.gridLayout_3.addItem(spacerItem3, 0, 10, 1, 1)
spacerItem4 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Minimum)
self.gridLayout_3.addItem(spacerItem4, 0, 2, 1, 1)
spacerItem5 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Minimum)
self.gridLayout_3.addItem(spacerItem5, 0, 9, 1, 1)
spacerItem6 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Minimum)
self.gridLayout_3.addItem(spacerItem6, 0, 8, 1, 1)
self.editMut = QtWidgets.QDoubleSpinBox(self.groupBox_8)
self.editMut.setMaximum(100.0)
self.editMut.setSingleStep(0.01)
self.editMut.setProperty("value", 0.08)
self.editMut.setObjectName("editMut")
self.gridLayout_3.addWidget(self.editMut, 0, 5, 1, 1)
self.verticalLayout_6.addWidget(self.groupBox_8)
self.groupBox_9 = QtWidgets.QGroupBox(self.tabScenario)
self.groupBox_9.setObjectName("groupBox_9")
self.gridLayout_4 = QtWidgets.QGridLayout(self.groupBox_9)
self.gridLayout_4.setObjectName("gridLayout_4")
self.label_15 = QtWidgets.QLabel(self.groupBox_9)
self.label_15.setObjectName("label_15")

```

```

self.gridLayout_4.addWidget(self.label_15, 2, 0, 1, 1)
self.label_18 = QtWidgets.QLabel(self.groupBox_9)
self.label_18.setObjectName("label_18")
self.gridLayout_4.addWidget(self.label_18, 3, 0, 1, 1)
self.label_19 = QtWidgets.QLabel(self.groupBox_9)
self.label_19.setObjectName("label_19")
self.gridLayout_4.addWidget(self.label_19, 5, 0, 1, 1)
self.label_17 = QtWidgets.QLabel(self.groupBox_9)
self.label_17.setObjectName("label_17")
self.gridLayout_4.addWidget(self.label_17, 7, 0, 1, 1)
self.label_16 = QtWidgets.QLabel(self.groupBox_9)
self.label_16.setObjectName("label_16")
self.gridLayout_4.addWidget(self.label_16, 4, 0, 1, 1)
self.editSbj = QtWidgets.QSpinBox(self.groupBox_9)
self.editSbj.setMaximum(100)
self.editSbj.setProperty("value", 70)
self.editSbj.setObjectName("editSbj")
self.gridLayout_4.addWidget(self.editSbj, 0, 1, 1, 1)
self.label_13 = QtWidgets.QLabel(self.groupBox_9)
self.label_13.setObjectName("label_13")
self.gridLayout_4.addWidget(self.label_13, 0, 0, 1, 1)
self.label_14 = QtWidgets.QLabel(self.groupBox_9)
self.label_14.setObjectName("label_14")
self.gridLayout_4.addWidget(self.label_14, 1, 0, 1, 1)
spacerItem7 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Minimum)
self.gridLayout_4.addItem(spacerItem7, 0, 2, 1, 1)
self.editLun = QtWidgets.QSpinBox(self.groupBox_9)
self.editLun.setMaximum(100)
self.editLun.setProperty("value", 5)
self.editLun.setObjectName("editLun")
self.gridLayout_4.addWidget(self.editLun, 1, 1, 1, 1)
self.editSec = QtWidgets.QSpinBox(self.groupBox_9)
self.editSec.setMaximum(100)
self.editSec.setProperty("value", 5)
self.editSec.setObjectName("editSec")

```

```

self.gridLayout_4.addWidget(self.editSec, 2, 1, 1, 1)
self.editInstrRest = QtWidgets.QSpinBox(self.groupBox_9)
self.editInstrRest.setMaximum(100)
self.editInstrRest.setProperty("value", 5)
self.editInstrRest.setObjectName("editInstrRest")
self.gridLayout_4.addWidget(self.editInstrRest, 4, 1, 1, 1)
self.editInstrLoad = QtWidgets.QSpinBox(self.groupBox_9)
self.editInstrLoad.setMaximum(100)
self.editInstrLoad.setProperty("value", 5)
self.editInstrLoad.setObjectName("editInstrLoad")
self.gridLayout_4.addWidget(self.editInstrLoad, 5, 1, 1, 1)
self.editIdle = QtWidgets.QSpinBox(self.groupBox_9)
self.editIdle.setMaximum(100)
self.editIdle.setProperty("value", 5)
self.editIdle.setObjectName("editIdle")
self.gridLayout_4.addWidget(self.editIdle, 3, 1, 1, 1)
self.editMeet = QtWidgets.QSpinBox(self.groupBox_9)
self.editMeet.setMaximum(100)
self.editMeet.setProperty("value", 5)
self.editMeet.setObjectName("editMeet")
self.gridLayout_4.addWidget(self.editMeet, 7, 1, 1, 1)
self.lblTotal = QtWidgets.QLabel(self.groupBox_9)
self.lblTotal.setObjectName("lblTotal")
self.gridLayout_4.addWidget(self.lblTotal, 8, 0, 1, 1)
self.verticalLayout_6.addWidget(self.groupBox_9)
spacerItem8 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.Expanding)
self.verticalLayout_6.addItem(spacerItem8)
self.tabWidget.addTab(self.tabScenario, "")
self.verticalLayout.addWidget(self.tabWidget)
MainWindow.setCentralWidget(self.centralWidget)
self.menubar = QtWidgets.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 21))
self.menubar.setObjectName("menubar")
self.menuFile = QtWidgets.QMenu(self.menubar)
self.menuFile.setObjectName("menuFile")

```



```
self.menuHelp = QtWidgets.QMenu(self.menubar)
self.menuHelp.setObjectName("menuHelp")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(MainWindow)
self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)
self.actionNew = QtWidgets.QAction(MainWindow)
self.actionNew.setObjectName("actionNew")
self.actionOpen = QtWidgets.QAction(MainWindow)
self.actionOpen.setObjectName("actionOpen")
self.actionSave = QtWidgets.QAction(MainWindow)
self.actionSave.setObjectName("actionSave")
self.actionSave_As = QtWidgets.QAction(MainWindow)
self.actionSave_As.setObjectName("actionSave_As")
self.actionSettings = QtWidgets.QAction(MainWindow)
self.actionSettings.setObjectName("actionSettings")
self.actionExit = QtWidgets.QAction(MainWindow)
self.actionExit.setObjectName("actionExit")
self.actionAdd_Instructor = QtWidgets.QAction(MainWindow)
self.actionAdd_Instructor.setObjectName("actionAdd_Instructor")
self.actionView_Instructors = QtWidgets.QAction(MainWindow)
self.actionView_Instructors.setObjectName("actionView_Instructors")
self.actionAdd_Room = QtWidgets.QAction(MainWindow)
self.actionAdd_Room.setObjectName("actionAdd_Room")
self.actionView_Rooms = QtWidgets.QAction(MainWindow)
self.actionView_Rooms.setObjectName("actionView_Rooms")
self.actionAdd_Subject = QtWidgets.QAction(MainWindow)
self.actionAdd_Subject.setObjectName("actionAdd_Subject")
self.actionView_Subjects = QtWidgets.QAction(MainWindow)
self.actionView_Subjects.setObjectName("actionView_Subjects")
self.actionAdd_Sections = QtWidgets.QAction(MainWindow)
self.actionAdd_Sections.setObjectName("actionAdd_Sections")
self.actionView_Sections = QtWidgets.QAction(MainWindow)
self.actionView_Sections.setObjectName("actionView_Sections")
self.actionImport = QtWidgets.QAction(MainWindow)
self.actionImport.setObjectName("actionImport")
```

```
self.actionExport = QtWidgets.QAction(MainWindow)
self.actionExport.setObjectName("actionExport")
self.actionView_Results = QtWidgets.QAction(MainWindow)
self.actionView_Results.setObjectName("actionView_Results")
self.actionGenerate = QtWidgets.QAction(MainWindow)
self.actionGenerate.setObjectName("actionGenerate")
self.actionImport_2 = QtWidgets.QAction(MainWindow)
self.actionImport_2.setObjectName("actionImport_2")
self.actionExport_2 = QtWidgets.QAction(MainWindow)
self.actionExport_2.setObjectName("actionExport_2")
self.actionScenario_Summary = QtWidgets.QAction(MainWindow)
self.actionScenario_Summary.setObjectName("actionScenario_Summary")
self.actionGenerate_2 = QtWidgets.QAction(MainWindow)
self.actionGenerate_2.setObjectName("actionGenerate_2")
self.actionView_Results_2 = QtWidgets.QAction(MainWindow)
self.actionView_Results_2.setObjectName("actionView_Results_2")
self.actionInstructions = QtWidgets.QAction(MainWindow)
self.actionInstructions.setObjectName("actionInstructions")
self.actionAbout = QtWidgets.QAction(MainWindow)
self.actionAbout.setObjectName("actionAbout")
self.menuFile.addAction(self.actionNew)
self.menuFile.addAction(self.actionOpen)
self.menuFile.addAction(self.actionSave_As)
self.menuFile.addSeparator()
self.menuFile.addAction(self.actionSettings)
self.menuFile.addSeparator()
self.menuFile.addAction(self.actionExit)
self.menuHelp.addAction(self.actionInstructions)
self.menuHelp.addAction(self.actionAbout)
self.menubar.addAction(self.menuFile.menuAction())
self.menubar.addAction(self.menuHelp.menuAction())

self.retranslateUi(MainWindow)
self.tabVidjet.setCurrentIndex(4)
QtCore.QMetaObject.connectSlotsByName(MainWindow)
```

```

def retranslateUi(self, MainWindow):
    _transfer = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_transfer("MainWindow", "Інформаційна технологія
автоматизованого формування розкладу закладу освіти"))
    self.tabInstructors.setWhatsThis(_transfer("MainWindow",
"<html><head><body><p><br/></p></body></html>"))
    self.groupBox.setTitle(_transfer("MainWindow", "Операції"))
    self.btnInstrAdd.setText(_transfer("MainWindow", "Додати викладача"))
    self.btnInstrImport.setText(_transfer("MainWindow", "Імпортувати з CSV"))
    self.tabVidget.setTabText(self.tabVidget.indexOf(self.tabInstructors), _transfer("MainWindow",
"Викладачі"))
    self.groupBox_2.setTitle(_transfer("MainWindow", "Операції"))
    self.btnRoomAdd.setText(_transfer("MainWindow", "Додати аудиторію"))
    self.btnRoomImport.setText(_transfer("MainWindow", "Імпортувати з CSV"))
    self.tabVidget.setTabText(self.tabVidget.indexOf(self.tabRooms), _transfer("MainWindow",
"Аудиторії"))
    self.groupBox_3.setTitle(_transfer("MainWindow", "Операції"))
    self.btnSubjAdd.setText(_transfer("MainWindow", "Додати предмет"))
    self.btnSubjImport.setText(_transfer("MainWindow", "Імпортувати з CSV"))
    self.tabVidget.setTabText(self.tabVidget.indexOf(self.tabSubjects), _transfer("MainWindow",
"Предмети"))
    self.groupBox_4.setTitle(_transfer("MainWindow", "Операції"))
    self.btnSecAdd.setText(_transfer("MainWindow", "Додати секцію"))
    self.tabVidget.setTabText(self.tabVidget.indexOf(self.tabSections), _transfer("MainWindow",
"Секції"))
    self.groupBox_5.setTitle(_transfer("MainWindow", "Операції"))
    self.btnScenGenerate.setText(_transfer("MainWindow", "Згенерувати"))
    self.btnScenResult.setText(_transfer("MainWindow", "Переглянути результати"))
    self.groupBox_6.setTitle(_transfer("MainWindow", "Налаштування роботи університету"))
    self.radioLunchYes.setText(_transfer("MainWindow", "Так"))
    self.radioLunchNo.setText(_transfer("MainWindow", "Ні"))
    self.label_6.setToolTip(_transfer("MainWindow", "Доступ до обідньої перерви."))
    self.label_6.setText(_transfer("MainWindow", "Обідня перерва"))
    self.label_4.setToolTip(_transfer("MainWindow", "Час відкриття університету."))
    self.label_4.setText(_transfer("MainWindow", "Час початку навчання"))
    self.timeStarting.setDisplayFormat(_transfer("MainWindow", "h AP"))

```

```

self.label_5.setToolTip(_transfer("MainWindow", "Час закінчення навчання"))
self.label_5.setText(_transfer("MainWindow", "Час закінчення операції"))
self.timeEnding.setDisplayFormat(_transfer("MainWindow", "h AP"))
self.groupBox_8.setTitle(_transfer("MainWindow", "Налаштування"))
self.label_20.setToolTip(_transfer("MainWindow", "The percent of population that would belong
to elite.))
self.label_20.setText(_transfer("MainWindow", "Elite Population"))
self.label_21.setToolTip(_transfer("MainWindow", "The maximum control of a sigma.))
self.label_21.setText(_transfer("MainWindow", "Deviation Tolerance"))
self.label_12.setToolTip(_transfer("MainWindow", "Stops the generation when a chromosome
meets this.))
self.label_12.setText(_transfer("MainWindow", "Maximum Fitness"))
self.label_11.setToolTip(_transfer("MainWindow",
                                "<html><head><body><p>Triggers mutation rate change when the
difference of average fitness falls to the specified level.</p></body></html>"))
self.label_11.setText(_transfer("MainWindow", "Mutation Rate Adjustment Trigger"))
self.label_10.setToolTip(_transfer("MainWindow", "Maximum attempts for creating a valid
chromosome.))
self.label_10.setText(_transfer("MainWindow", "Maximum Creation Attempts"))
self.label_9.setToolTip(
    _transfer("MainWindow", "Maximum amount of generations to be performed on solution
generation.))
self.label_9.setText(_transfer("MainWindow", "Maximum Generations"))
self.label_7.setToolTip(
    _transfer("MainWindow", "Starting point and lowest population count of the genetic
algorithm.))
self.label_7.setText(_transfer("MainWindow", "Minimum Population Count"))
self.label_8.setToolTip(_transfer("MainWindow", "Highest population count of the genetic
algorithm.))
self.label_8.setText(_transfer("MainWindow", "Maximum Population Count"))
self.groupBox_9.setTitle(_transfer("MainWindow", "Evaluation Matrix"))
self.label_15.setToolTip(_transfer("MainWindow",
                                "The weight of section rest (There must be a rest for every consecutive 3
hours of session))
self.label_15.setText(_transfer("MainWindow", "Section Rest"))
self.label_18.setToolTip(_transfer("MainWindow", "The weight of sections having less idle time.))

```

```

self.label_18.setText(_transfer("MainWindow", "Section Idle Time"))
self.label_19.setToolTip(
    _transfer("MainWindow", "The weight of having all instructors have normalized teaching
load.))
self.label_19.setText(_transfer("MainWindow", "Instructor Load Balance"))
self.label_17.setToolTip(
    _transfer("MainWindow", "The weight of having correct meeting patterns for subject
placement.))
self.label_17.setText(_transfer("MainWindow", "Meeting Pattern"))
self.label_16.setToolTip(_transfer("MainWindow",
    "The weight of instructor rest (There must be a rest for every consecutive 3
hours of session)"))
self.label_16.setText(_transfer("MainWindow", "Instructor Rest"))
self.label_13.setToolTip(_transfer("MainWindow", "The weight of having all subjects placed.))
self.label_13.setText(_transfer("MainWindow", "Subject Placement"))
self.label_14.setToolTip(_transfer("MainWindow", "The weight of having a lunch break.))
self.label_14.setText(_transfer("MainWindow", "Обідня перерва"))
self.lblTotal.setText(_transfer("MainWindow", "Сумарно: "))
self.tabVidget.setTabText(self.tabVidget.indexOf(self.tabScenario),
    _transfer("MainWindow", "Менеджер сценаріїв"))
self.menuFile.setTitle(_transfer("MainWindow", "Файл"))
self.menuHelp.setTitle(_transfer("MainWindow", "Допомога"))
self.actionNew.setText(_transfer("MainWindow", "Новий"))
self.actionOpen.setText(_transfer("MainWindow", "Відкрити"))
self.actionSave.setText(_transfer("MainWindow", "Зберегти"))
self.actionSave_As.setText(_transfer("MainWindow", "Зберегти як"))
self.actionSettings.setText(_transfer("MainWindow", "Налаштування"))
self.actionExit.setText(_transfer("MainWindow", "Вихід"))
self.actionAdd_Instructor.setText(_transfer("MainWindow", "Додати викладача"))
self.actionView_Instructors.setText(_transfer("MainWindow", "Переглянути викладачів"))
self.actionAdd_Room.setText(_transfer("MainWindow", "Додати аудиторію"))
self.actionView_Rooms.setText(_transfer("MainWindow", "Переглянути аудиторії"))
self.actionAdd_Subject.setText(_transfer("MainWindow", "Додати предмет"))
self.actionView_Subjects.setText(_transfer("MainWindow", "Переглянути предмети"))
self.actionAdd_Sections.setText(_transfer("MainWindow", "Додати секцію"))
self.actionView_Sections.setText(_transfer("MainWindow", "Переглянути секції"))

```

```

self.actionImport.setText(_transfer("MainWindow", "Імпортувати"))
self.actionExport.setText(_transfer("MainWindow", "Експортувати"))
self.actionView_Results.setText(_transfer("MainWindow", "Переглянути результати"))
self.actionGenerate.setText(_transfer("MainWindow", "Згенерувати"))
self.actionImport_2.setText(_transfer("MainWindow", "Імпортувати"))
self.actionExport_2.setText(_transfer("MainWindow", "Експортувати"))
self.actionScenario_Summary.setText(_transfer("MainWindow", "Підсумок сценарію"))
self.actionGenerate_2.setText(_transfer("MainWindow", "Згенерувати"))
self.actionView_Results_2.setText(_transfer("MainWindow", "Переглянути результати"))
self.actionInstructions.setText(_transfer("MainWindow", "Інструкції"))
self.actionAbout.setText(_transfer("MainWindow", "Інформація про продукт"))

```

## Додаток Г. Підключення та ініціалізація бази даних

```
import sqlite3
```

```
def checkSetup():
```

```
    conn = sqlite3.connect('gas.db')
```

```
    cursor = conn.cursor()
```

```
    cursor.execute("SELECT name FROM sqlite_master WHERE type='table' AND
name='instructors'")
```

```
    result = cursor.fetchone()
```

```
    conn.close()
```

```
    if result is None:
```

```
        return False
```

```
    return True
```

```
def setup():
```

```
    conn = sqlite3.connect('gas.db')
```

```
    cursor = conn.cursor()
```

```
    create_instructors_table = """
```

```
        CREATE TABLE IF NOT EXISTS instructors (
```

```
            id INTEGER PRIMARY KEY,
```

```
            name TEXT NOT NULL,
```

```
            hours INTEGER NOT NULL,
```

```

    schedule TEXT NOT NULL,
    active BOOLEAN NOT NULL DEFAULT 1 CHECK (
        active IN (0, 1)
    )
);
''''

create_rooms_table = ''''
CREATE TABLE IF NOT EXISTS rooms (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    type TEXT NOT NULL,
    schedule TEXT NOT NULL,
    active BOOLEAN NOT NULL DEFAULT 1 CHECK (
        active IN (0, 1)
    )
);
''''

create_subjects_table = ''''
CREATE TABLE IF NOT EXISTS subjects (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    hours REAL NOT NULL,
    code TEXT NOT NULL,
    description TEXT NOT NULL,
    instructors TEXT NOT NULL,
    divisible BOOLEAN NOT NULL DEFAULT 1 CHECK (
        divisible IN (0, 1)
    ),
    type TEXT NOT NULL
);
''''

create_sections_table = ''''
CREATE TABLE IF NOT EXISTS sections (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    schedule TEXT NOT NULL,

```

```

subjects TEXT NOT NULL,
active BOOLEAN NOT NULL DEFAULT 1 CHECK (
    active IN (0, 1)
),
stay BOOLEAN NOT NULL DEFAULT 0 CHECK (
    active IN (0, 1)
)
);
''''''

create_sharing_table = ''''''
CREATE TABLE IF NOT EXISTS sharings (
    id INTEGER PRIMARY KEY,
    subjectId INTEGER NOT NULL,
    sections TEXT NOT NULL,
    final BOOLEAN NOT NULL DEFAULT 0 CHECK (
        final IN (0, 1)
    )
);
''''''

create_results_table = ''''''
CREATE TABLE IF NOT EXISTS results (
    id INTEGER PRIMARY KEY,
    content BLOB NOT NULL,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
);
''''''

cursor.execute(create_instructors_table)
cursor.execute(create_rooms_table)
cursor.execute(create_subjects_table)
cursor.execute(create_sections_table)
cursor.execute(create_sharing_table)
cursor.execute(create_results_table)

conn.commit()

conn.close()

```



```
def getConnection():  
    return sqlite3.connect('gas.db')
```