

УДК 004.4

Т. А. Чупілко, О. С. Чупілко, М. Ф. Мормуль

## ПРОЕКТУВАННЯ І ПРОГРАМНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВІДВІДУВАНОСТІ ТА АНАЛІЗУ ДАНИХ В ЗАКЛАДАХ ОСВІТИ

Університет митної справи та фінансів, Дніпро

**Анотація.** Робота присвячена дослідженню оптимального способу контролю відвідуваності закладів освіти учнями як викладачами, так і батьками: проектуванню і реалізації автоматизованої системи, яка дозволяє спростити цей процес, розробити необхідні звіти, включаючи оповіщення батьків інформацією про відвідуваність їх дітей; створення модулів для аналізу отриманої інформації з використанням методів математичного моделювання та мови програмування Python і бібліотек для обробки даних. Принцип роботи системи полягає в тому, що інформація про учня, який прийшов до закладу освіти, зчитується сканером та пересилається на локальний комп'ютер, де вона і зберігається. На цьому ж комп'ютері проводиться аналіз збереженої інформації та формуються необхідні звіти. Також в іншому підході до розробки системи використовується хмарний сервер. Аналізуються проблеми і недоліки, що виникають при різних можливостях організації автоматизованої системи. Для збереження інформації на комп'ютері, до якого приєднаний сканер, розглядаються підходи, засновані на використанні мови Python, що дозволяє легко переносити дані в різні бази даних, включаючи Access та MySQL, та інший, де сканована інформація одразу зберігається у бази даних Access, яка розташована на комп'ютері, до якого приєднаний сканер. Розроблено і реалізовано структури бази даних для збереження і обробки інформації. Система генерує різні звіти з необхідною інформацією, що може бути підставою для прийняття певних рішень: наприклад, звіт для столової про кількість учнів по класам, про загальну кількість дітей в школі, загальну кількість присутніх та відсутніх в абсолютних значеннях та відносних і т.ін. Для надсилання повідомлень було використано сучасний месенджер Telegram. Для реалізації ідеї було використано BotFather та функцію даної платформи, що дозволяє створювати автоматичних роботів для виконання завдань розсилки її користувачам. Логіка боту контролюється за допомогою HTTPS запитів до API системи для роботів. За допомогою SQL-запитів вилучається потрібна інформація, яка передається у розроблений програмний модуль для проведення аналізу по окремій групі учнів або в цілому по закладу. Використовуються регресійні методи аналізу в програмному режимі за допомогою інструментів Python. Роботу зі створення реалізації автоматизованої інтелектуальної системи було виконано на замовлення закладу середньої освіти.

**Ключові слова:** автоматизована система, база даних, Python, аналіз даних, моделювання.

**Abstract.** The work is devoted to the research of the optimal method of monitoring the attendance of educational institutions by students, both teachers and parents: the design and implementation of an automated system that allows to simplify this process, to develop the necessary reports, including notifying parents with information about the attendance of their children; creating modules for analyzing the received information using mathematical modeling methods and the Python programming language and libraries for data processing. The principle of operation of the system is that the information about the student who came to the educational institution is read by a scanner and sent to the local computer, where it is stored. On the same computer, the stored information is analyzed and the necessary reports are generated. Another approach to system development also uses a cloud server. Problems and shortcomings arising from various possibilities of organizing an automated system are analyzed. To save information on the computer to which the scanner is connected, approaches based on the use of the Python language are considered, which allows easy transfer of data to various databases, including Access and MySQL, and another where the scanned information is immediately stored in the Access database, which is located on the computer to which the scanner is connected. Database structures for storing and processing information have been developed and implemented. The system generates various reports with the necessary information, which can be the basis for making certain decisions: for example, a report for the dining hall on the number of students by class, on the total number of children in the school, the total number of people present and absent in absolute and relative terms, etc. The modern Telegram messenger was used to send messages. To implement the idea, Bot Father and the function of this platform were used, which allows you to create automatic robots to perform mailing tasks to its users. The bot logic is controlled using HTTPS requests to the bot system API. With the help of SQL queries, the necessary information is extracted, which is transferred to the developed software module for analysis on a separate group of students or on the institution as a whole. Regression analysis methods are used in software mode using Python tools. The work on the creation and implementation of an automated intellectual system was carried out on the order of a secondary education institution.

**Key words:** automated system, databases, Python, data analysis, modeling.

**DOI:** <https://doi.org/10.31649/1999-9941-2023-56-1-35-43>.

### Вступ

В освіті завжди є важливим питання надання якісних освітніх послуг. Це стосується закладів і середньої, і вищої освіти. Наразі в освітньому процесі найбільш розповсюдженою є модель змішаної форми навчання: заняття проводиться в оф-лайн і он-лайн режимах. Успішність засвоєння навчального матеріалу залежить від багатьох чинників. Важливу роль відіграють заняття в аудиторії. Особливо важливим стає облік присутніх студентів під час епідемії або воєнних дій в країні. Інформаційні технології використовуються в багатьох сферах нашого життя. Але в закладах освіти залишається багато рутинної паперової роботи, що пов'язана з навчальним процесом. Зокрема, облік і аналіз відвідуваності, пов'язані з цим якісні показники успішності проводяться деканатами і викладачами зазвичай вручну, в найкращому випадку – з допомогою електронних таблиць. Серед нагальних потреб в удосконаленні навчального процесу є проблеми, пов'язані з різноманітною звітністю, де використовуються одні й ті самі відомості по студентам, викладачам, дисциплінам і т. ін. Як і 20 років тому, всі відомості збираються кожного разу заново, враховуючи необхідну форму звітності. Давно вже настав час впровадити автоматизацію в навчальний процес для різних завдань, що виникають в закладах освіти.

Найкраще було б вирішити цю проблему на рівні країни, щоб заклади освіти мали готові автоматизовані інформаційні системи, розроблені фахівцями і запроваджені МОН.

Тож, однією з багатьох проблем є автоматичний облік відвідувань студентами аудиторних занять, участі в конференціях, різного типу засіданнях та інших заходах і аналіз даних, отриманих на основі зібраних відомостей. Також аналогічні питання постають в закладах середньої освіти, але ускладнюється тим, що потрібен зв'язок з батьками неповнолітніх учнів. Є відомі приклади застосування обмежених аналогів в школах деяких країн, зокрема, в Фінляндії система «Robo» та в Китаї «Розумна форма». В 2000-х р.р. була спроба в Україні створити студентський квиток з чипом, який міг використовуватися не тільки як документ, що посвідчує особу, а і в якості документа для електронної ідентифікації, що використовується для проведення фінансових операцій, оплати проїзду в транспорті і т. ін. З іншого погляду, автоматизована система може знайти застосунок і в школах, і в інших закладах, де здобувачі освіти не мають електронних документів з чипами. В нашому побуті наразі присутні технології ідентифікації особи за розпізнаванням обличчя, скануванням відбитку пальців або сітківки ока. Однак, зазначені технології авторизації не можуть використовуватися для публічної інформації. Найбільш прийнятний спосіб для зазначеної проблеми – використання електронної картки як підтвердження особи.

В цій роботі запропоновано проект і програмну реалізацію автоматизованої системи збору інформації щодо відвідувань, а також інтелектуальну обробку отриманих даних. Програмний проект зорієнтовано на потреби середньої освіти, тобто закладів освіти, де потрібен зв'язок з батьками учнів. Збір інформації проводиться при вході учнів до школи за допомогою сканерів магнітних карток, смарт-карт або QR-кодів, які підключені до комп'ютера через usb-порт. Кожен учасник навчального процесу має картку (брелок) з унікальним ідентифікатором, який зчитується сканером при вході у навчальний заклад. Зчитаний код разом з поточним часом і датою передається в базу даних для збереження, після чого батькам студента (учня) відправляється на Telegram повідомлення про те, що їх дитина прийшла до закладу. В повідомленні вказується ім'я та прізвище дитини, клас, час приходу до школи. Крім того, у визначений час автоматично генеруються звіти, необхідні для прийняття рішень за результатами аналізу даних з відвідування учнями школи. Серед них дані для столової щодо кількості учнів по класам, що снідають у школі в поточний день, аналіз даних про відсоток відвідуваності в період епідемій та інше. Також в системі присутній модуль, що дозволяє моделювати, прогнозувати дані і графічно представляти результати досліджень в програмному режимі за допомогою Python.

### **Актуальність**

Актуальність обраної теми обумовлена тим, що в навчальних закладах для організації навчального процесу та пов'язаного з цим документообігу, упорядкуванню і аналізу різного типу даних практично відсутня автоматизація та інтелектуальна обробка за допомогою комп'ютерних технологій. В цій роботі представлено проект і реалізацію частини такої автоматизованої системи в навчальному закладі, яка може бути доповнена і розширена на інші види обробки даних в деканатах, бібліотеках, кафедрах або фахових комісіях, особистих застосунках викладачів і т. ін. для створення потрібної звітності і аналізу досліджуваних показників. Відвідуваність, під якою ми розуміємо систему присутності студентів або учнів на заняттях з метою засвоєння освітньої програми, є важливою складовою освітнього процесу. Навчальна відвідуваність тісно пов'язана з вирішенням низки організаційних завдань навчального процесу, моральних та соціальних проблем сім'ї та освітньої установи, істотно впливає на їх зв'язок між собою, а також на створення іміджу освітньої установи та визнання високої кваліфікації педагогічних працівників. Відвідування навчальних занять – один із обов'язків учнів. Несумлінне її виконання пов'язане з низкою проблем самих учнів, освітньої установи та суспільства. Моніторинг відвідуваності занять і різних заходів в установах освіти є важливим засобом керування освітнім процесом.

Новітні технології дають можливість оптимізувати, зокрема, способи контролю відвідуваності учнів та студентів як викладачами, так і батьками, а також реалізувати автоматизовану систему, яка дозволить спростити цей процес. Окрім того, існують сучасні підходи до аналізу показників. Технології опрацювання даних визначаються характером даних та метою дослідження. Наразі мова програмування Python та велика кількість бібліотек є популярним і потужним інструментом, що дозволяють ефективно працювати з різними даними, використовуючи сумісну можливість написання коду та готові рішення.

Моделювання охоплює різні галузі, і наразі існують інструменти для високоефективної та швидкої обробки даних, що дають змогу аналізувати певні тенденції, прогнозувати, оцінювати якість побудованих моделей. Аналітика даних сприяє покращенню процесу прийняття рішень у будь-якій сфері діяльності і, зокрема, в закладах освіти. Тож, поєднання в одній системі інструментів автоматизації і моделювання показників є досить актуальною інтелектуальною проблемою за відсутності централізованого підходу до розробки подібних систем.

### **Мета**

Метою роботи є вивчення оптимального способу контролю відвідуваності студентів (учнів) як викладачами (деканатами), так і батьками; проектування і реалізація автоматизованої системи, яка

дозволить спростити цей процес, розробити необхідні звіти, включаючи оповіщення батьків інформацією про відвідуваність їх дітей; створення модулів для аналізу отриманої інформації з використанням методів математичного моделювання та мови програмування Python і бібліотек для обробки даних.

### Задачі

1. Виявити аспекти у порядку ведення контролю відвідуваності, які можна автоматизувати та систематизувати.
2. Розробити інформаційну систему та створити додатки для автоматизації ведення відвідуваності.
3. Спроекувати систему аналізу даних, на підставі яких розробити необхідні звіти для навчального закладу, а також оповіщення батьків інформацією про відвідуваність їх дітей.
4. Розробити програмний застосунок для моделювання показників відвідуваності.

### Розв'язання задач

#### СХЕМИ ПРОЕКТУВАННЯ СИСТЕМИ

Найпростіший варіант інформаційної системи обліку і аналізу відвідування навчальних занять (надалі ІС) складається з одного комп'ютера, до якого через USB-порт приєднаний зчитувач магнітних карток, смарт-карт або QR-кодів.

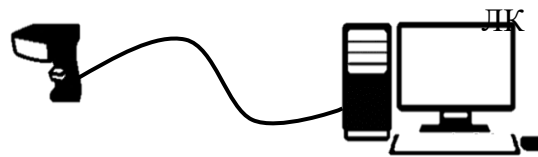


Рисунок 1 – Найпростіша схема

Принцип роботи вказаної системи простий: інформація про учня, що прийшов до закладу освіти, зчитується сканером та пересилається на локальний комп'ютер (ЛК), де вона і зберігається. На цьому ж комп'ютері проводиться аналіз збереженої інформації та формуються необхідні звіти.

Але ця проста система має суттєві недоліки:

- доступ до інформації про відвідування занять може бути тільки з комп'ютера, де зберігаються дані;
- розташування комп'ютера, який є інструментом для аналізу даних по відвідуванню учнями школи, не найкраще – біля вхідних дверей,
- труднощі аналізу даних у випадку, якщо учні входять до школи через два-три входи, кожен з яких повинен реєструвати учнів, які зайшли до школи.

Вказані недоліки усуваються, якщо локальні комп'ютери, до яких приєднані сканери, включити до шкільної мережі. В цьому випадку вся інформація зі сканерів через локальні комп'ютери (ЛК) може одразу передаватися на шкільний сервер даних (СД), який буде зберігати інформацію з декілька сканерів і який забезпечить доступ до інформації з робочих місць (РМ), де треба мати необхідні звіти.

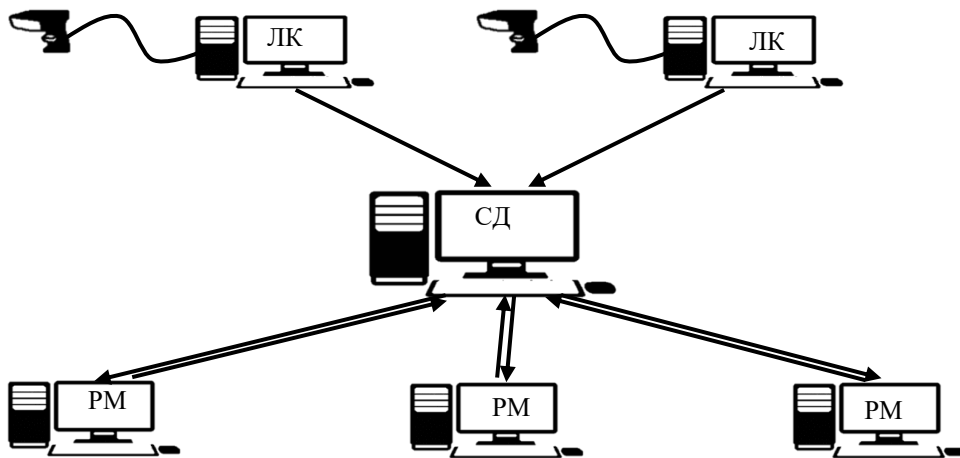


Рисунок 2 – Інформаційна система з сервером закладу освіти

Але, якщо шкільний сервер працює не цілодобово, а включається на початку робочого дня, дана система має суттєвий недолік, який пов'язаний з тим, що більшість учнів приходять до школи до включення сервера даних, і тому їх приход зареєструвати в системі буде неможливо.

Якщо сервер даних розташувати у хмарному середовищі, то проблеми з часом роботи сервера існувати не буде.

Дані зі сканерів будуть одразу пересилатися у хмарне сховище і там зберігатися. Доступ до них буде постійним. Крім того, доступ до них можна організувати з будь-яких пристроїв: комп'ютерів, планшетів, мобільних телефонів.

Але, ця система має недолік: при поганому інтернет-з'єднанні будуть проблеми зі збереженням вхідної інформації на сервері, що може привести до повної непрацездатності програми збереження вхідної інформації.

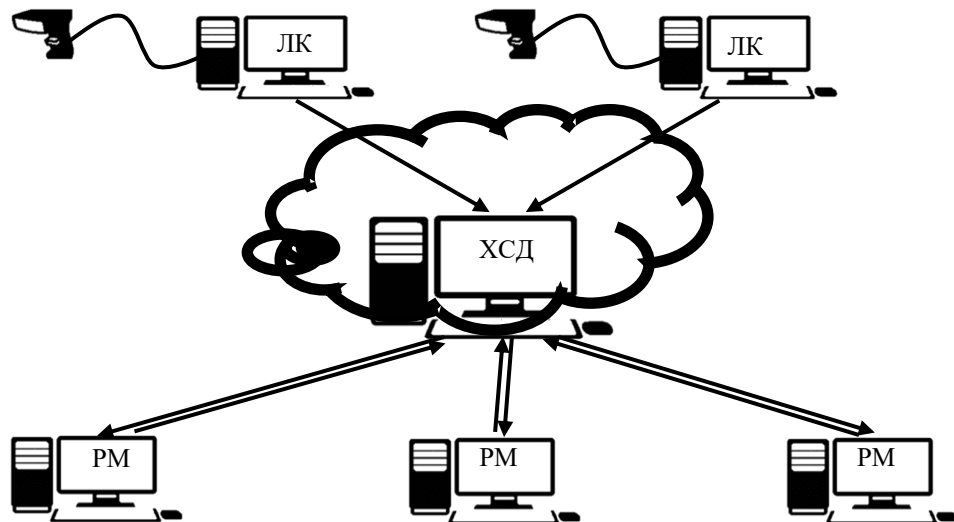


Рисунок 3 – Інформаційна система з хмарним сервером даних

Створена інформаційна система мінімізує проблеми збереження вхідних даних наступним чином:

- дані, які скануються, зберігаються в локальній базі даних на комп'ютері, до якого безпосередньо відключено сканер;
- в якості комп'ютера використовується ноутбук, який може працювати автономно не менше 4 годин;
- на одному з робочих комп'ютерів (або на локальному сервері) встановлюється програма, яка пересилає дані з локальної бази даних в базу даних, розташовану на сервері;
- всі робочі комп'ютери для обробки даних використовують дані сервера.

Вхідні дані при такому рішенні обробляються з деякою затримкою, але така схема вирішує проблему поганого інтернет-зв'язку та відсутність електропостачання.

#### *ОСНОВНІ ЕТАПИ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ*

Для збереження інформації на комп'ютері, до якого приєднаний сканер, було розглянуто два незалежних підходи.

У першому підході використовувалась мова Python. Дані зберігалися у файлі формату .csv. Цей формат підтримується багатьма мовами програмування і дозволяє легко переносити дані в різні бази даних, включаючи Access та MySQL. Головна доцільність використання такого підходу в тому, що на вході замість відносно дорогих ноутбуків можна використовувати недорогі мікрокомп'ютери Raspberry PI або Arduino. В них використовується Linux і вони добре підтримують мову Python, а також формат файлів .csv.

Другий підхід використовує СУБД Access [1], [2]: сканована інформація одразу зберігається у базі даних Access, яка розташована на комп'ютері, до якого приєднаний сканер. Якщо в якості комп'ютера використовувати ноутбук, то система збору інформації буде повністю автономною. Вона не буде залежати від інтернету, від електрики. Це значно підвищує її надійність.

Для збереження даних про відвідуваність на локальному комп'ютері створюється база даних, яка містить всього три об'єкти:

- таблицю Signal, яка зберігає ID-коди учнів, які відвідали школу,
- таблицю UserLocal, яка містить відомості про учнів,

- форму для відображення інформації про учня, який на даний момент проходить реєстрацію:
  - прізвище учня,
  - час і дату реєстрації у системі.

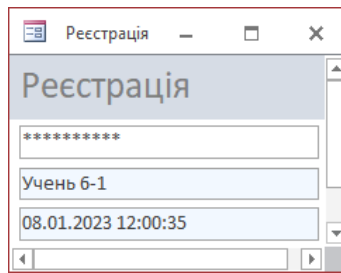


Рисунок 4 – Форма відображення інформації в момент реєстрації

Якщо сервер даних знаходиться у хмарному середовищі, то їх потрібно туди скопіювати з локальних баз даних, що знаходяться на локальних комп'ютерах. Це можна зробити за допомогою запитів, які можуть бути розташовані на локальних комп'ютерах, або на одному з робочих комп'ютерів (РМ), які використовуються персоналом для аналізу даних про відвідування. У першому випадку запити передають тільки дані на локальних комп'ютерах. У другому – дані спочатку збираються зі всіх локальних комп'ютерів, а потім передаються в базу даних, яка розташована у хмарному середовищі.

Якщо сервер даних знаходиться у закладі, то можна не переносити дані з локальних комп'ютерів на сервер, на якому можна розмістити тільки програми для створення необхідних звітів. При цьому програму спочатку збирає дані зі всіх локальних комп'ютерів, а потім обробляє їх разом, генеруючи необхідні звіти.

Але, на наш погляд, краще всі дані зберігати в одному місці. В цьому випадку перенесення даних на сервер даних закладу освіти проводиться аналогічно перенесенню даних на сервер у хмарному середовищі.

У створеній ІС всі дані знаходяться на шкільному сервері даних. Їх обробка може проводитися на комп'ютерах, які підключені до шкільної мережі і мають доступ до бази даних. Ці комп'ютери краще розташовувати на робочих місцях, де необхідна інформація про відвідування. Наприклад, на комп'ютерах директора, декана, куратора і т. ін., тобто тих працівників, кого стосується питання відвідування занять.

Крім цього, програма генерує деякі спеціальні звіти, які використовують для рішення інших задач. Наприклад, система генерує звіт для столової про кількість учнів по класам, які будуть «на сьогодні» приймати їжу. Іншим прикладом є звіт про загальну кількість дітей в школі, загальну кількість присутніх та відсутніх в абсолютних значеннях та відносних (відсотках). Ця інформація може бути підставою для прийняття певних рішень: наприклад, про об'явлення карантину в закладі чи групі та інше.

#### *НАДСИЛАННЯ ПОВІДОМЛЕНЬ БАТЬКАМ*

Про те, що діти прийшли до закладу освіти, батькам бажано знати вже за декілька хвилин після їх приходу. Ця опція реалізується в інформаційній системі через Telegram.

Після того, як учень прийшов до школи і зареєструвався в системі, за його ідентифікаційним номером визначаються батьки, яким висилається інформація про те, що їх дитина прийшла до закладу, вказується дата та час її реєстрації в системі.

Для надсилання повідомлень було використано сучасний популярний месенджер Telegram. Для реалізації ідеї було використано функцію даної платформи, що дозволяє створювати автоматичних роботів для виконання завдань розсилки її користувачам.

Боти – це спеціальні акаунти у Telegram, створені для того, щоб автоматично обробляти та надсилати повідомлення. Користувачі можуть взаємодіяти з ботами за допомогою повідомлень, що надсилаються через звичайні або групові чати. Логіка боту контролюється за допомогою HTTPS запитів до API системи для роботів.

Створюються роботи за допомогою BotFather. Це офіційний інструмент для створення ботів та управління ними. З його допомогою було створено бота з певною назвою та нікнеймом. Далі BotFather надає посилання на створений бот і токен для звернення до нього.

Для взаємодії з роботом Telegram було використано мову програмування Python. Створений додаток має модульну структуру. За відправку повідомлень відповідає клас telegram, який забезпечує взаємодію з платформою за допомогою API. Дані для надсилання повідомлень наша програма отримує з бази даних mysql, розташованої на віддаленому сервері.

За виконання запитів на сервер відповідає бібліотека requests:

```
import requests
```

Метод setChat класу telegram задає чат одержувача і зберігає його ID у властивість класу:

```
def setChat(self, chat_id):
    self.chat_id = chat_id
pass
```

Метод sendMessage відповідає за відправлення повідомлень: формує посилання нашого бота з методом надсилання повідомлень і передає дані методом POST.

```
def sendMessage(self, text):
    if self.chat_id == 0:
        raise Exception('chat_enter_error')
        link='https://api.telegram.org/bot'+ self.bot_id+'!'+self.token+'/sendMessage'
    answer = requests.post(link, data = {
        'chat_id': self.chat_id,
        'text': text
    })
    if answer.status_code != 200:
        raise Exception('post_text_error')
    return answer
```

За взаємодію з базою даних відповідає клас database, який реалізовано за допомогою стандартної бібліотеки pymysql мови програмування Python:

```
import pymysql
```

Метод connect відповідає за з'єднання з базою даних SQL:

```
def connect(self, host, user, password, db):
    connection = pymysql.connect(
        host = host,
        user = user,
        password = password,
        db = db
    )
    print ('connect successful!!')
    return connection
```

Метод getRows відповідає за отримання ID чатів одержувачів та текст, який потрібно їм відправити:

```
def getRows(self, connection):
    try:
        with connection.cursor() as cursor
            sql = 'SELECT `chat_id`, `text` FROM `signal`'
            cursor.execute(sql)
            rows = []
            for row in cursor:
                rows.append(row)
            return rows
    except:
        print('Database Connection Error')
    finally:
        print('end')
```

Сам код створеної програми виглядає так:

```
from classes.database import database
from classes.telegram import telegram
db = database()
messenger = telegram()
connection = db.connect('localhost', 'root', 'root', 'signal')
rows = db.getRows(connection)
for row in rows:
```

```

messenger.setChat(row[0])
result = messenger.sendMessage(row[1])
print(result)

```

Створюється об'єкт класів, викликаються методи, задаються потрібні параметри та виводиться результат.

#### *ПРОГРАМНІ ІНСТРУМЕНТИ ДЛЯ АНАЛІЗУ ВІДВІДУВАНOSTІ І МОДЕЛЮВАННЯ ПОКАЗНИКІВ*

Як було вище зазначено, в розробленій системі створено базу даних, в яку вносяться автоматично всі дані по відвідуваності. Структура бази даних і приклад моделювання будуть представлені в наступній роботі. За допомогою SQL-запитів можна вилучити потрібну інформацію, яка передається у розроблений програмний модуль для проведення аналізу по окремій групі учнів, факультету або в цілому по закладу. Дослідження в програмному застосунку проводиться регресійними методами математичної статистики. Для оцінювання якості моделей і прогнозування використовуються економетричні методи і критерії Фішера і Стьюдента. Можна розширити функціонал, додавши багатофакторний, кластерний аналіз, що буде напрямом подальшої розробки.

Функціонал застосунку для аналізу реалізується за допомогою мови програмування Python і відповідних бібліотек. Коротко наведемо можливості основних програмних модулів, що використовуються при аналізі даних [3].

NumPy – основний пакет для виконання наукових розрахунків на Python. Поверх NumPy побудовано інші бібліотеки.

Основні можливості пакету:

- швидкий і ефективний об'єкт багатовимірного масиву ndarray;
- функції для виконання обчислень з елементами одного масиву або математичних операцій з декількома масивами;
- засоби для читання і запису на диски наборів даних, представлених у вигляді масивів;
- операції лінійної алгебри, перетворення Фур'є і генератор випадкових чисел;
- засоби для інтеграції з кодом, що написаний на C, C++ або Fortran.

Крім прискорення роботи з масивами, однією з основних цілей NumPy стосовно аналізу даних є організація контейнера для передачі даних поміж алгоритмами. Як засіб зберігання і маніпуляції даними масиви NumPy значно ефективніші вбудованих в Python структур даних.

Багато засобів обчислень орієнтовано на Python та використовують масиви NumPy як основні структури даних.

Бібліотека pandas надає структури даних і функції для роботи зі структурованими даними. Основні об'єкти pandas – DataFrame – двовимірна таблиця, в якій рядки і стовпці мають мітки, і Series – об'єкт одновимірного масиву з мітками.

У бібліотеці pandas поєднуються висока продуктивність засобів роботи з масивами, притаманна NumPy, і гнучкі можливості маніпулювання даними, властиві електронним таблицям і реляційним базам даних (наприклад, на основі SQL). Оскільки маніпулювання даними, їх підготовка і очищення грають дуже велику роль в аналізі даних [3], [4], pandas є одним з основних інструментів.

Основні можливості бібліотеки:

- розвинені засоби індексування, що дозволяють просто змінювати форму наборів даних, формувати зрізи, виконувати агрегування і вибирати підмножини;
- структури даних з позначеними осями підтримують автоматичне або явне вирівнювання даних, що виключає появу типових помилок при роботі з невіривняні даними і даними з різних джерел, які по-різному індексовані;
- вбудована функціональність часових рядів;
- одні і ті ж структури даних підтримують як тимчасові ряди, так і дані інших видів;
- арифметичні операції;
- гнучка обробка відсутніх даних;
- інтеграція даних;
- підтримка з'єднання і інших реляційних операцій, наявних в популярних базах даних (наприклад, на основі SQL).

Сама назва pandas утворена від panel data (панельні дані), що застосовуються в економетричних дослідженнях для позначення багатовимірних структурованих наборів даних.

Бібліотека matplotlib – інструмент Python для створення графіків і інших способів візуалізації двовимірних даних. Вона підходить для створення графіків, придатних для публікації. Існують також інші бібліотеки візуалізації, але matplotlib використовується найчастіше і тому добре інтегрована з іншими частинами екосистеми.

SciPy – збір пакетів, призначених для вирішення різних стандартних обчислювальних задач:

- scipy.integrate – підпрограми чисельного інтегрування і розв'язання диференціальних рівнянь;

- `scipy.linalg` – підпрограми лінійної алгебри і розкладання матриць, доповнюють ті, що включені в `numpy.linalg`;
- `scipy.optimize` – алгоритми оптимізації функцій (знаходження екстремумів) і пошуку коренів;
- `scipy.signal` – засоби обробки сигналів;
- `scipy.sparse` – алгоритми роботи з розрідженими матрицями і розв'язання розріджених систем лінійних рівнянь;
- `scipy.special` – обгортка навколо SPECFUN, написаної на Fortran-бібліотеці, що містить реалізації багатьох стандартних математичних функцій, в тому числі гамма-функції;
- `scipy.stats` – стандартні безперервні і дискретні розподіли ймовірностей (функції щільності ймовірності, формування вибірки, функції безперервного розподілу ймовірності), різні статистичні критерії і додаткові описові статистики.

Бібліотека `scikit-learn` – основний інструментарій машинного навчання на Python. Є підмодулі для наступних моделей:

- класифікація: метод опорних векторів, метод найближчих сусідів, випадкові ліси, логістична регресія і т. ін.;
- регресія: Lasso, гребнева регресія і т. ін.;
- кластеризація: метод k середніх, спектральна кластеризація і т. ін.;
- зниження розмірності: метод головних компонент, відбір ознак, матрична факторизація і т. ін.;
- вибір моделі: пошук на сітці, перехресний контроль, метрики;
- попередня обробка: виділення ознак, нормування.
- `scikit-learn` орієнтований головним чином на передбачення.

`Statsmodels` – пакет статистичного аналізу, містить алгоритми класичної статистики та економетрики. У нього входять наступні підмодулі:

- регресійні моделі: лінійна регресія, узагальнені лінійні моделі, лінійні моделі зі змішаними ефектами і т. ін.;
- дисперсійний аналіз (ANOVA);
- аналіз часових рядів: AR, ARMA, ARIMA, VAR і інші моделі;
- непараметричні методи: ядерна оцінка щільності, ядерна регресія;
- візуалізація результатів статистичного моделювання.

Пакет `statsmodels` орієнтований більшою мірою на статистичне виведення, він дає оцінки невизначеності і р-значення параметрів.

Використовується разом з NumPy і pandas.

Всі зазначені інструменти використовуються для створення масивів інформації та різного роду аналізу, прогнозування і візуалізації показників. В попередніх роботах [5], [6], [7] було наведено приклад аналізу і прогнозування даних з використанням описаних програмних застосунків.

Розроблена інформаційна система на даний момент знаходиться на тестуванні, в результаті якого у програму вносяться додаткові опції і функції. Після успішного проходження тестування програму планується використовувати у закладі середньої освіти, за замовленням якого проведено представлено роботу.

### Висновки

1. Виявлено аспекти у порядку ведення контролю відвідуваності, які можна автоматизувати та систематизувати.
2. Створено інформаційну систему обліку, що дозволяє автоматизувати збір інформації про відвідування навчальних занять, аналізувати її з метою отримання необхідних звітів.
3. Результат використання роботи представлено у вигляді файлу БД Access, який містить структуру спроектованих таблиць, схему даних зі зв'язками між таблицями, екранні форми для занесення вхідних даних та екранні форми для перегляду результатів їх обробки, запити для аналізу даних, звіти. В систему інтегрована програма, яка створена на Python, для надсилання повідомлень батькам про прихід їх дитини до закладу освіти.
4. В систему інтегровано програмний застосунок для моделювання та прогнозування даних.
5. Створена система може бути основою для подальшого розвитку і додавання нових функцій і можливостей.
6. Подібні автоматизовані системи у поєднанні з інструментами штучного інтелекту для різнопланового аналізу даних можуть використовуватися закладами освіти і моделюватися у відповідності до їх потреб.

### Список літератури

- [1] О. Шпортько, *Розробка баз даних в СУБД Microsoft Access*. К.: Видавничий дім "Кондор", 2018, 184 с.



- [2] І. Бардус, М.І. Лазарев, А. Ніценко, *Бази даних у схемах*. Харків: Вид-во "Діса плюс", 2017, 133 с.
- [3] Christian Hill, *Learning Scientific Programming with Python. Second Edition*. Cambridge University Press, 2020, 646 p.
- [4] William McKinney, *Python for Data Analysis. Data Wrangling with Pandas, NumPy, and IPython. Second Edition*. O'REILLY, 2018, 365 p.
- [5] Т. Чупілко, Ю. Ульяновська, М. Мормуль і А. Лагода, "Python для обробки даних і моделювання фінансово-економічних показників", ІТКІ, т. 51, вип. 2, с. 68-77, 2021. DOI: <http://doi.org/10.31649/1999-9941-2021-51-2-68-77>
- [6] Т. Чупілко, "Актуальні проблеми високоефективної обробки даних. Моделювання показників за допомогою мови програмування Python" у *Актуальні напрями розвитку технічного та виробничого потенціалу національної економіки*. Дніпро, Україна: Пороги, 2021, с.151-163
- [7] Tetjana Chupilko, Yulia Ulianova, Mykola Mormul and Olexandr Shchitov, "Python for Data Processing and Modeling Indicators of Economic Security of the Country" in International conference "Challenges and realities of the IT space: software engineering and cyber security", Online Scientific Publishers OWN, ISBN 978-83-7712-049-1, № 9156, October 2022. Url: <https://easychair.org/publications/preprint/WW2F>.

Стаття надійшла: 21.02.2023.

#### References

- [1] O. Shportko, *Rozrobka baz danyx v SUBD Microsoft Access*. K.: Vydavnychii dim "Kondor", 2018, 184 s. [in Ukrainian].
- [2] I. Bardus, M. Lazarev, A. Nicenko, *Bazy danyx u schemax*. Kharkiv: Vydavnytctvo «Disa plus», 2017, 133 s. [in Ukrainian].
- [3] Christian Hill, *Learning Scientific Programming with Python. Second Edition*. Cambridge University Press, 2020, 646 p.
- [4] William McKinney, *Python for Data Analysis. Data Wrangling with Pandas, NumPy, and IPython. Second Edition*. O'REILLY, 2018, 365 p.
- [5] T. Chupilko, Yu. Ulianova, M. Mormul, A. Lagoda, "Python dlya obrobky danyx i modeluvannya finansovo-ekonomichnyx pokaznykiv", ІТКІ, т. 51, v. 2, s. 68-77, 2021. DOI: <http://doi.org/10.31649/1999-9941-2021-51-2-68-77> [in Ukrainian].
- [6] T. Chupilko, "Aktualni problemy vysokoeffektyvnoi obrobky danykh. Modeliuvannya pokaznykiv za dopomohoiu movy prohramuвання Python" u *Aktualni napriamy rozvytku tekhnichnoho ta vyrobnychoho potentsialu natsionalnoi ekonomiky*. Dnipro: Porohy, 2021, s.151-163 [in Ukrainian].
- [7] Tetjana Chupilko, Yulia Ulianova, Mykola Mormul and Olexandr Shchitov, "Python for Data Processing and Modeling Indicators of Economic Security of the Country" in *International conference "Challenges and realities of the IT space: software engineering and cyber security"*, Online Scientific Publishers OWN, ISBN 978-83-7712-049-1, № 9156, October 2022. Url: <https://easychair.org/publications/preprint/WW2F>.

#### Відомості про авторів

**Чупілко Тетяна Анатоліївна** – кандидат технічних наук, доцент.

**Чупілко Олександр Сергійович** – здобувач ступеня магістра.

**Мормуль Микола Федорович** – кандидат технічних наук, доцент.

T. A. Chupilko, O. S. Chupilko, M. F. Mormul

## DESIGN AND SOFTWARE IMPLEMENTATION OF THE AUTOMATED ATTENDANCE SYSTEM AND DATA ANALYSIS IN EDUCATIONAL INSTITUTIONS

University of Customs and Finance, Dnipro