

Горбань Г.В.

Чорноморський національний університет імені Петра Могили

Козачок Ю.А.

Чорноморський національний університет імені Петра Могили

ВИКОРИСТАННЯ ІНСТРУМЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ В РОЗРОБЦІ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ БІБЛІОТЕКОЮ

Штучний інтелект відкриває нові можливості та перспективи вдосконалення технік та методів розробки програмних продуктів. Адже він може підвищити ефективність та якість розробки програмного забезпечення за рахунок оптимізації процесів. У статті подано дослідження найбільш перспективних напрямків в галузі штучного інтелекту, як інструменту для оптимізації розробки програмного забезпечення. В ході роботи досліджено сучасні роботи, присвячені висвітленню теми використання штучного інтелекту в розробці програмного забезпечення. Також проведено аналіз можливості використання інструменту чат-бот ChatGPT при написанні програмного коду. Окрім того, проаналізовано та висвітлено основні можливості ChatGPT, які дають змогу автоматизувати та оптимізувати багато етапів розробки програмного продукту. В результаті проведеного аналізу було виконано практичне дослідження використання ChatGPT, а саме виконана розробка частини інформаційної системи управління бібліотекою. Була побудована архітектура інформаційної системи з трьох рівнів: DAO (data access object), service layer, API (REST, gRPC). Кожен рівень було реалізовано з використанням штучного інтелекту, а саме за допомогою інструменту ChatGPT. Перший рівень DAO було написано на мові програмування Java з використанням бази даних Postgres в середовищі програмування IntelliJ IDEA. Наступним кроком було згенеровано декілька Java entities на основі sql-схеми, яка також побудована за допомогою ChatGPT. Кінцевим етапом побудови інформаційної системи була генерація REST API для інформаційної системи управління бібліотекою з використанням ChatGPT для демонстрації можливостей штучного інтелекту. В результаті практичного дослідження було проаналізовано якість програмного коду та оптимізовано програмний код за допомогою ChatGPT. Виділено основні переваги застосування ChatGPT розробниками для написання програмного коду та визначено основні проблеми, які потребують подальших досліджень.

Ключові слова: штучний інтелект, chatGPT, розробка програмного забезпечення, технології, генерація програмного коду, автоматизація.

Постановка проблеми. Використання штучного інтелекту та машинного навчання може виявитися важливим напрямком для підвищення ефективності створення програмних продуктів. Дослідження в цьому напрямку стають все більш актуальними, оскільки з збільшенням попиту на використання штучного інтелекту в створенні програмних продуктів, розробники знаходять нові можливості використання штучного інтелекту для спрощення та прискорення процесу створення програмного продукту.

Отже, велика увага в дослідженні може бути приділена визначенню оптимальних стратегій використання штучного інтелекту під час розробки програмного забезпечення.

Аналіз останніх досліджень і публікацій. Сучасні дослідники та молоді науковці, такі як

Д. Пчелянський, С. Воїнова [8, с. 60], Т. Савчук [6, с. 30], С. Шаров, С. Хрустальов [10, с. 109] висвітлили тему використання штучного інтелекту в розробці програмного забезпечення. В роботах досліджено використання штучного інтелекту для прогнозування часу на виправлення дефектів програмного коду, також проведено аналіз інструментів для використання при написанні програмного коду. В результаті аналізу було встановлено, що інструменти ВІТО та ChatGPT можуть бути корисними при розробці програмного забезпечення, проте в роботах не були розглянуті існуючі рішення для їх практичного впровадження у процес написання програмного коду.

Виконаний аналіз демонструє, що використання штучного інтелекту в розробці програмних продуктів є актуальним напрямом досліджень,

але інструменти, які можна використовувати в цьому напрямку, потребують подальшого аналізу на застосування у використанні.

Постановка завдання. Метою цього дослідження є створення інформаційної системи управління бібліотекою з використанням штучного інтелекту, зокрема інструменту ChatGPT. Використання інформаційної системи, заснованої на використанні асистенту ChatGPT, має потенціал прискорити розробку та ітеративно оптимізувати бізнес-процеси.

Основні задачі, які вирішуються для досягнення заявленої мети:

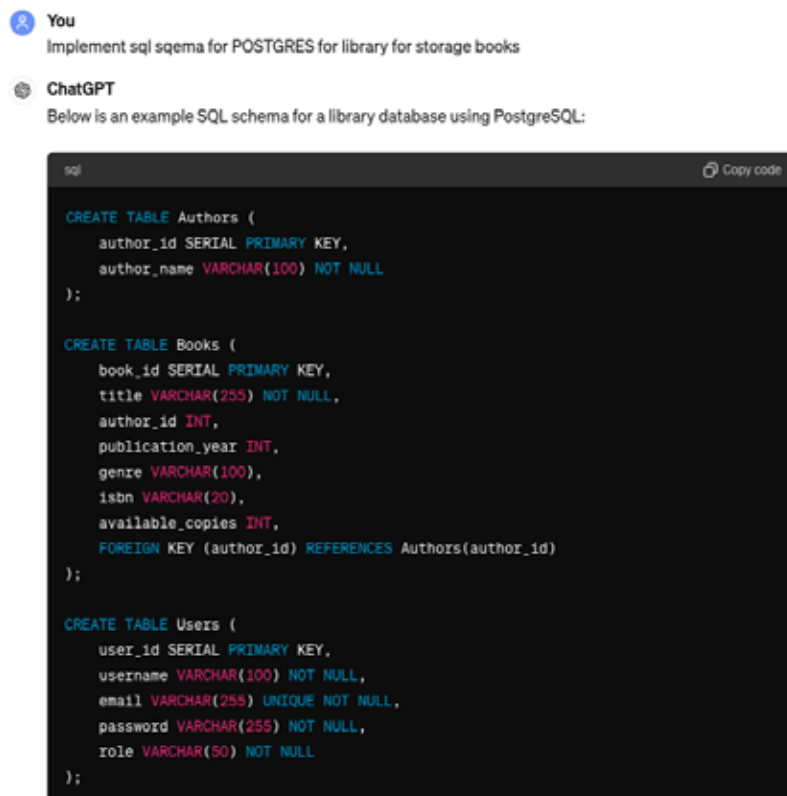
- 1) провести дослідження як треба вести спілкування з чат ботом для правильної генерації коду;
- 2) розробити частину програмного забезпечення для демонстрації можливостей штучного інтелекту;
- 3) проаналізувати якість написаного коду.

Виклад основного матеріалу. Штучний інтелект відкрив нові перспективи у сфері програмної інженерії, вносячи нові методики у процес розробки програмного забезпечення. Це дозволяє автоматизувати та оптимізувати багато етапів розробки, включаючи виявлення помилок, вибір алгоритмів та побудову архітектурних рішень. Одним із дієвих інструментів для розробки програмного коду є чат бот ChatGPT.

Під час дослідження було встановлено, що ChatGPT є ефективним інструментом для програмістів [13]. Він використовує нейронні мережі та алгоритми машинного навчання для автоматизації рутинних завдань, таких як виявлення помилок у коді, оптимізація алгоритмів та прискорення вирішення задач. Розробники можуть використовувати ChatGPT в браузері або інтегрувати з IDE, що також дозволяє отримувати пропозиції та рішення у режимі реального часу під час написання коду. До основних можливостей ChatGPT відносяться: написання програмного коду, підтримка з синтаксису, оптимізація та роз'яснення програмного коду, перевірка безпеки.

Розглянемо використання чат боту ChatGPT в розробці інформаційної системи (ІС) управління бібліотекою. Будь-яка enterprise ІС складається з трьох рівнів: DAO (data access object), service layer, API (REST, gRPC). Пояснивши штучному інтелекту, що саме необхідно згенерувати, можна отримати допомогу в генерації DAO структури даних під будь-яку мову програмування, включаючи Python, JavaScript, Java, C++, PHP, Ruby та інші.

Розглянемо створення DAO на мові Java з використанням бази даних Postgres в середовищі програмування IntelliJ IDEA. Штучний інтелект генерує sql-код (рис. 1), який готовий до використання та напише пояснення до даної схеми (рис. 2).



```
sql Copy code
CREATE TABLE Authors (
  author_id SERIAL PRIMARY KEY,
  author_name VARCHAR(100) NOT NULL
);

CREATE TABLE Books (
  book_id SERIAL PRIMARY KEY,
  title VARCHAR(255) NOT NULL,
  author_id INT,
  publication_year INT,
  genre VARCHAR(100),
  isbn VARCHAR(20),
  available_copies INT,
  FOREIGN KEY (author_id) REFERENCES Authors(author_id)
);

CREATE TABLE Users (
  user_id SERIAL PRIMARY KEY,
  username VARCHAR(100) NOT NULL,
  email VARCHAR(255) UNIQUE NOT NULL,
  password VARCHAR(255) NOT NULL,
  role VARCHAR(50) NOT NULL
);
```

Рис. 1. Генерація sql-коду за допомогою ChatGPT

This schema includes the following tables:

- **Authors:** Stores information about authors.
- **Books:** Stores information about books including title, author, publication year, genre, ISBN, and available copies. It has a foreign key relationship with the Authors table.
- **Users:** Stores information about library users including username, email, password, and role.
- **Loans:** Tracks the loans of books to users, including issue date, return date, and status. It has foreign key relationships with the Users and Books tables.
- **Reviews:** Stores user reviews of books including rating and comments. It has foreign key relationships with the Users and Books tables.

Рис. 2. Пояснення до sql-коду, згенероване за допомогою ChatGPT

Отримавши згенеровану схему, ми можемо вручну вносити корективи або надавати чату вказівки щодо того, які зміни необхідно зробити. Чат в свою чергу може згенерувати нову схему на основі цих вказівок. Таким чином, ми забезпечу-

ємо зручний ітераційний процес, де можливість ручного втручання поєднується з автоматизованим створенням схеми.

Наступним кроком згенеруємо Java entity Author та Book на основі sql-схеми (рис. 3, 4).

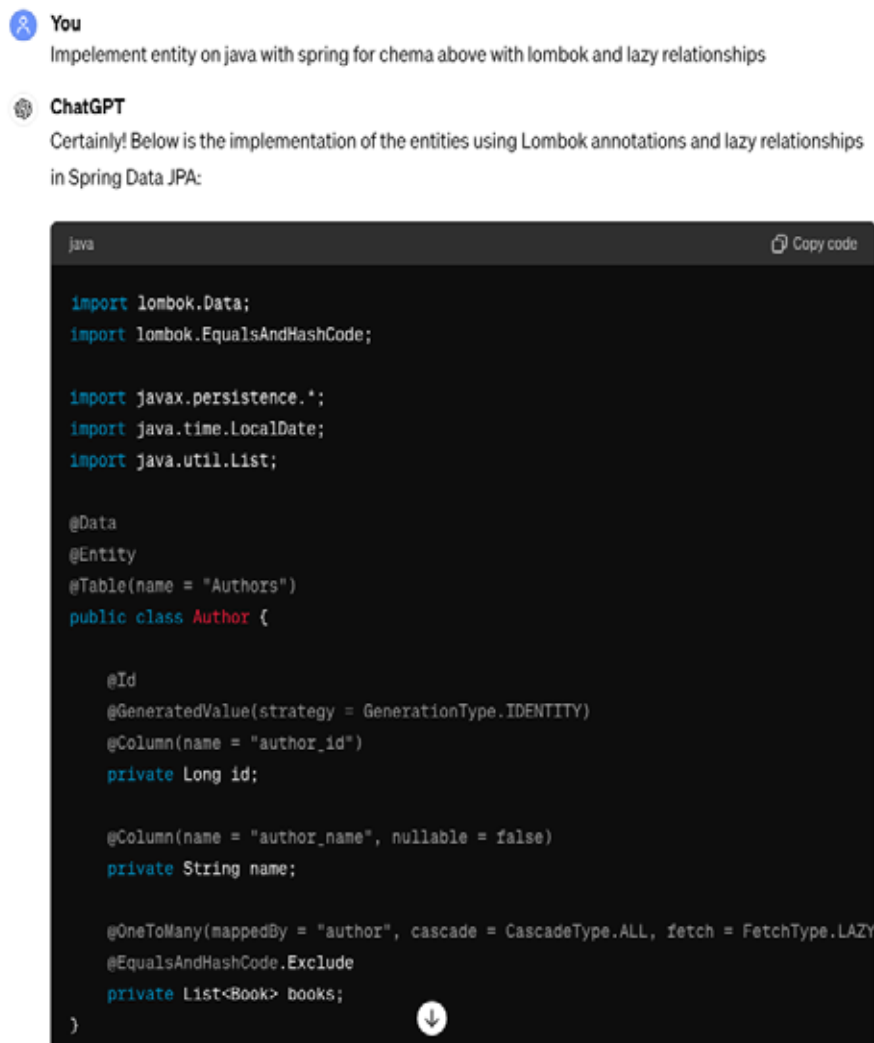


Рис. 3. Генерація Java entity Author на основі sql-схеми

```

@Data
@Entity
@Table(name = "Books")
public class Book {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "book_id")
    private Long id;

    @Column(name = "title", nullable = false)
    private String title;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "author_id")
    private Author author;

    @Column(name = "publication_year")
    private Integer publicationYear;

    @Column(name = "genre")
    private String genre;

    @Column(name = "isbn")
    private String isbn;

    @Column(name = "available_copies")
    private Integer availableCopies;
}

```

Рис. 4. Генерація Java entity Book на основі sql-схеми

У сфері реалізації бізнес-логіки використання штучного інтелекту відкриває широкі можливості для оптимізації та рефакторингу коду з метою підвищення продуктивності та ефективності системи. Цей підхід охоплює не лише виправлення недоліків, а й оптимізацію алгоритмів та застосування інших методів для покращення якості коду.

Розглянемо використання цього підходу на прикладі Java методу оцінки рівня цікавості книги (рис. 5). Можна зауважити, що оптимізація даного Java методу за допомогою штучного інтелекту передбачає чітку вказівку користувача що саме треба зробити. Штучний інтелект не зможе визначити це самостійно. Це може бути аналіз різноманітних параметрів, що впливають на цікавість книги для користувача або ж рефакторинг коду для покращення підтримки даної логіки. Серед таких параметрів можуть бути:

- 1. Аналіз вмісту книги.** Використання методів обробки природної мови (Natural Language Processing, NLP) для аналізу тексту книги з метою виявлення ключових тематик, стилю письма, а також визначення рівня складності тексту.

- 2. Історія читання користувача.** Збір та аналіз історії читання кожного конкретного користувача для визначення його індивідуальних уподобань та відповідності цікавості певній книзі.

- 3. Відгуки та рейтинги.** Аналіз відгуків користувачів та їх рейтингів книги для визначення загального сприйняття та оцінки цікавості.

- 4. Зовнішні дані.** Використання зовнішніх даних, таких як популярність книги, рейтинги на платформах, соціальні мережі тощо, для уточнення оцінки цікавості.

Рисунок 6 демонструє, як даний програмний код можна оптимізувати з використанням чат боту штучного інтелекту.

Фінішним кроком є генерація REST API за допомогою штучного інтелекту на основі вже згенерованих DAO (data access object) та service layer. Згенеруємо REST API для вже оптимізованого Java методу оцінки рівня цікавості книги (рис. 7).

Отриманий програмний код можна використовувати в подальшій побудові архітектури ІС управління бібліотекою та використовувати ChatGPT для генерації інших DAO, service layers та необхідних REST API для покриття бізнес-цілей ІС.

Висновки. В рамках даної роботи було:

1. Досліджено процес спілкування з чат ботом ChatGPT для правильної генерації коду. Наведено опис можливостей чат боту ChatGPT при написанні програмного коду.

2. Розроблено DAO, service layer та REST API для ІС управління бібліотекою з використанням

```
@Service
public class BookReviewService {

    @Autowired
    private ReviewRepository reviewRepository;

    @Transactional
    public boolean isBookInteresting(Book book) {
        List<Review> reviews = reviewRepository.findByBook(book);
        if (reviews.isEmpty()) {
            return false;
        }

        int totalRating = 0;
        for (Review review : reviews) {
            totalRating += review.getRating();
        }

        double averageRating = (double) totalRating / reviews.size();
        return averageRating >= 4.0;
    }
}
```

Рис. 5. Функція оцінки рівня цікавості книги

- You**
optimise the class above. The book will be interesting if score more then 4.2. Please, use constants for score and Use constructor injection
- ChatGPT**
Certainly! Here's the optimized version of the class using constructor injection and constants for the score threshold:

```
java Copy code

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

@Service
public class BookReviewService {


    private final ReviewRepository reviewRepository;
    private static final double INTERESTING_SCORE_THRESHOLD = 4.2;

    @Autowired
    public BookReviewService(ReviewRepository reviewRepository) {
        this.reviewRepository = reviewRepository;
    }

    @Transactional(readOnly = true)
    public boolean isBookInteresting(Book book) {
        Double averageRating = reviewRepository.getAverageRatingByBookId(book.getId());
        return averageRating != null && averageRating >= INTERESTING_SCORE_THRESHOLD;
    }
}
```

Рис. 6. Оптимізований програмний код за допомогою ChatGPT

 **You**
generate Spring REST endpoint for the method above

 **ChatGPT**
To generate a Spring REST endpoint for the `isBookInteresting` method, you can create a controller class that maps a HTTP GET request to a specific URL. Here's how you can implement it:

```

java Copy code

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/api/books")
public class BookController {

    private final BookReviewService bookReviewService;

    @Autowired
    public BookController(BookReviewService bookReviewService) {
        this.bookReviewService = bookReviewService;
    }

    @GetMapping("/{bookId}/interesting")
    public boolean isBookInteresting(@PathVariable Long bookId) {
        Book book = bookRepository.findById(bookId).orElseThrow(() -> new BookNotFoundException());
        return bookReviewService.isBookInteresting(book);
    }
}

```

Рис. 7. Згенерований REST API за допомогою ChatGPT

ChatGPT для демонстрації можливостей штучного інтелекту.

3. Проаналізовано якість написаного програмного коду. В рамках цього завдання було оптимізовано програмний код за допомогою ChatGPT.

Загалом, ChatGPT може стати важливим інструментом для розробників програмного коду, вносячи новаторські можливості у галузь штучного інтелекту. Спілкуючись с ChatGPT

можна казати свої побажання та оптимізувати бізнес-логіку до потреб замовника, але для високого рівня оптимізації необхідно мати експертизу в бізнес-процесі та мові програмування. Для досягнення значних покращень у якості та ефективності розробки програмного забезпечення необхідна взаємодія між фахівцями з бізнес-аналізу, розробниками та експертами у сфері штучного інтелекту.

Список літератури:

1. Амосов Н. М. Алгоритми розуму. Київ : Наук. думка, 1979. 223 с.
2. Бурдаев В. П. Системи навчання з елементами штучного інтелекту : монографія. Харків : ХНЕУ, 2009. 392 с.
3. Jain A., Mao J., Mohiuddin K. Artificial Neural Networks : *A Tutorial*. *IEEE*. 1996. Vol. 29, no 3. P. 31–44.
4. Bito – дозволяє легко писати код – AI Guru. AI Guru. URL: <https://aiguru.com.ua/pysaty-kod-za-dopomohoyu-shtuchnoho-intelektu/> (дата звернення: 07.05.2024).
5. Bito AI – Become a 10X Dev with Bito. Bito. URL: <https://bito.ai/> (date of access: 07.05.2024).
6. Савчук Т., Приймак Н. Використання методів штучного інтелекту для прогнозування часу для виправлення дефектів розробки програмного забезпечення. Вінниця : Інтернет-Освіта-Наука, 2016. С. 30–31.

7. Weizenbaum J. Computer power and human reason: From judgment to calculation. Freeman, 1976
8. Пчелянський Д. П., Воїнова С. А. Штучний інтелект: перспективи та тенденції розвитку. *Automation of technological and business processes*. 2019. Т. 11. № 3. С. 59–64.
9. Шаров С., Хрустальов С. Інструментальні засоби та етапи розробки експертних систем. *Збірник наукових праць Харківського національного університету Повітряних Сил*. 2016. Т. 49. № 4. С. 105–109.
10. Mougouei D., Powers D. M. W., Mougouei E. A fuzzy framework for prioritization and partial selection of security requirements in software projects. *Journal of Intelligent & Fuzzy Systems*. 2019. Vol. 37. № 2. P. 2671–2686.
11. Сівіцький В. ВІТО – ШТУЧНИЙ ІНТЕЛЕКТ ДЛЯ ДОПОМОГИ РОЗРОБНИКАМ ПРОГРАМНОГО КОДУ. *Збірник наукових праць Харківського національного економічного університету імені Семена Кузнеця* 2024. Т. 15. С. 95–97.
12. Використання ChatGPT для сценаріїв розробки програмного забезпечення – рішення Visure. *Visure Solutions*. URL: <https://visuresolutions.com/uk/blog/leveraging-ai-for-use-cases-in-software-development/> (дата звернення: 13.05.2024).

Horban H.V., Kozachok Yu.A. ARTIFICIAL INTELLIGENCE TOOLS USAGE IN THE DEVELOPMENT OF THE LIBRARY MANAGEMENT INFORMATION SYSTEM

Artificial intelligence opens up new opportunities and perspectives for improving techniques and methods of software development. It can enhance the efficiency and quality of software development by optimizing processes. The article presents research on the most promising directions in the field of artificial intelligence as a tool for optimizing software development. Modern works dedicated to the use of artificial intelligence in software development are examined. An analysis of the possibility of using the ChatGPT chatbot tool in writing software code is also conducted. Additionally, the main capabilities of ChatGPT are analyzed and highlighted, allowing for the automation and optimization of many stages of software product development. As a result of the analysis, a practical study of the use of ChatGPT was conducted, namely the development of a part of the library management information system. The architecture of the information system was built on three levels: DAO (data access object), service layer, API (REST, gRPC). Each level was implemented using artificial intelligence, specifically with the help of the ChatGPT tool. The first DAO level was written in Java using the Postgres database in the IntelliJ IDEA programming environment. The next step involved generating several Java entities based on the SQL schema, which was also built using ChatGPT. The final stage of building the information system was the generation of a REST API for the library management information system using ChatGPT to demonstrate the capabilities of artificial intelligence. As a result of the practical study, the quality of the software code was analyzed and the software code was optimized using ChatGPT. The main advantages of using ChatGPT by developers to write software code were identified, and the main problems requiring further research were highlighted.

Key words: artificial intelligence, chatGPT, software development, technology, code generation, automation.