

УДК 004.4

[https://doi.org/10.52058/2786-6025-2025-2\(43\)-1479-1490](https://doi.org/10.52058/2786-6025-2025-2(43)-1479-1490)

**Резанова Вікторія Георгіївна** кандидат технічних наук, доцент кафедри комп'ютерних наук, Київський національний університет технологій та дизайну, м. Київ, <http://orcid.org/0000-0003-4630-2701>

**Чупринка Наталія Вікторівна** кандидат технічних наук, доцент кафедри комп'ютерних наук, Київський національний університет технологій та дизайну, м. Київ, <http://orcid.org/0000-0002-8952-7567>

## ДОСЛІДЖЕННЯ ТА ВИКОРИСТАННЯ CASE-ТЕХНОЛОГІЙ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

**Анотація.** Швидке зростання комунікаційних інструментів та Інтернету суттєво змінило можливості інформаційних систем і те, як бізнес їх використовує. Інформаційні системи стають звичайним явищем і відіграють ключову роль в діяльності, управлінні та в процесах прийняття рішень на всіх рівнях організації. Інформаційні системи швидко еволюціонували від скромного ІТ-інструменту до важливого стратегічного елемента успішної організації. Еволюція таких систем призвела до того, що лідери галузі та компанії, що займаються програмним забезпеченням (ПЗ), повинні збалансувати рівень впровадження, продуктивність та вплив системи на процес прийняття рішень. З іншого боку, компанії, що займаються програмним забезпеченням, мають на меті розробляти продукти, які є достатньо гнучкими, щоб відповідати різноманітним очікуванням. Розробники ПЗ стикаються з багатьма проблемами при розробці таких систем. Оскільки технології, методології, управлінське бачення та бізнес-середовище продовжують швидко змінюватися, компанії-розробники змушені витрачати значну кількість ресурсів на дослідження та інновації, щоб забезпечити гнучкість кінцевого продукту, який можна легко інтегрувати в будь-який тип бізнес-моделі.

Автоматизоване програмне забезпечення (CASE) описує широкий набір трудозберігаючих інструментів і методів, які використовуються в розробці програмного забезпечення та моделюванні бізнес-процесів. Інструменти CASE створюють стандартизовану структуру для багаторазового використання для керування проектами програмного забезпечення та призначені для того, щоб допомогти розробникам заощадити час, підвищити продуктивність і залишатися організованими протягом усього проекту. Переваги від використання CASE-засобів: економія вартості, підвищення якості, уніфікація документації, спрощення роботи, зменшення витрат на обслуговування коду,

структурований підхід до роботи, швидші цикли розробки, зменшення помилок, покращена командна співпраця. Таким чином, інструменти CASE стають все більш важливими по мірі розвитку технологій, щоб задовольнити вимоги складних проєктів розробки програмного забезпечення.

**Ключові слова:** інформаційна система, CASE-технології, діаграма прецедентів, діаграма дій, діаграма класів.

**Rezanova Viktoriia Georhiivna** PhD in Engineering, Associate Professor of the Department of Computer Science, Kyiv National University of Technologies and Design, Kyiv, <http://orcid.org/0000-0003-4630-2701>

**Chuprynka Natalia Viktorivna** PhD in Engineering, Associate Professor, Department of Computer Science, Kyiv National University of Technologies and Design, Kyiv, <https://orcid.org/0000-0002-8952-7567>

## RESEARCH AND USE OF CASE-TECHNOLOGIES IN INFORMATION SYSTEMS DESIGN

**Abstract.** The rapid growth of communication tools and the Internet has significantly changed the capabilities of information systems and how businesses use them. Information systems are becoming commonplace and play a key role in operations, management, and decision-making processes at all levels of an organization. Information systems have rapidly evolved from a humble IT tool to a critical strategic element of a successful organization. The evolution of such systems has led industry leaders and software companies to balance the level of implementation, productivity, and impact of the system on decision-making. On the other hand, software companies aim to develop products that are flexible enough to meet diverse expectations. Software developers face many challenges in developing such systems. As technology, methodologies, management vision, and business environments continue to change rapidly, development companies are forced to spend significant resources on research and innovation to ensure the flexibility of the final product that can be easily integrated into any type of business model.

Computer-Aided Software Engineering (CASE) describes a broad set of labor-saving tools and techniques used in software development and business process modeling. CASE tools create a standardized, reusable framework for managing software projects and are designed to help developers save time, increase productivity, and stay organized throughout the project. Benefits of using CASE tools: cost savings, improved quality, documentation unification, work simplification, reduced code maintenance costs, structured approach to work, faster development cycles, reduced errors, improved team collaboration. Thus, CASE tools are becoming increasingly important as technology advances to meet the demands of complex software development projects.



**Keywords:** information system, CASE-technologies, use case diagram, activity diagram, class diagram.

**Постановка проблеми.** Тенденції розвитку сучасних інформаційних технологій приводять до постійного зростання складності інформаційних систем (ІС), що створюються в різних областях економіки. Сучасні крупні проекти ІС характеризуються, як правило, наступними особливостями:

- складність опису (чимала кількість функцій, процесів, елементів даних і складні взаємозв'язки між ними), що вимагає ретельного моделювання і аналізу даних і процесів;
- наявність сукупності тісно взаємодіючих компонентів (підсистем), що мають свої локальні завдання і цілі функціонування;
- відсутність прямих аналогів, що обмежує можливість використання яких-небудь типових проектних рішень і прикладних систем;
- необхідність інтеграції існуючих і новостворених додатків;
- функціонування в неоднорідному середовищі на декількох апаратних платформах;
- роз'єднаність і різноманітність окремих груп розробників по рівню кваліфікації і традиціям використання тих або інших інструментальних засобів;
- істотна часова протяжність проекту, обумовлена, з одного боку, обмеженими можливостями колективу розробників, і, з іншого боку, масштабами організації-замовника і різною мірою готовності окремих її підрозділів до впровадження ІС.

Для успішної реалізації проекту об'єкт проектування має бути перш за все адекватно описаний, мають бути побудовані повні і не суперечливі функціональні і інформаційні моделі ІС. Накопичений до теперішнього часу досвід проектування ІС показує, що це логічно складна, трудомістка і тривала за часом робота, що вимагає високої кваліфікації фахівців.

Ще нещодавно при розробці ІС досить широко застосовувалася методологія, що надає в розпорядження розробників строгі формалізовані методи опису ІС і технічних рішень, що приймаються. Вона заснована на наочній графічній техніці: для опису різного роду моделей ІС використовуються схеми і діаграми. Наочність і строгість засобів аналізу дозволяла розробникам і майбутнім користувачам системи із самого початку неформально брати участь в її створенні, обговорювати і закріплювати розуміння основних технічних рішень. Проте, широке застосування цієї методології і дотримання її рекомендаціям при розробці конкретних ІС зустрічалося досить рідко, оскільки при неавтоматизованій (ручній) розробці це практично неможливо.

Перераховані чинники сприяли появі програмно-технологічних засобів спеціального класу - CASE-засобів, що реалізують CASE-технологію

створення і супроводу ІС. Термін CASE (Computer Aided Software Engineering) використовується в даний час у вельми широкому сенсі. Первинне значення терміну CASE, обмежене питаннями автоматизації розробки тільки програмного забезпечення (ПЗ), в даний час набуло нового сенсу, що охоплює процес розробки складних ІС в цілому. Тепер під терміном CASE-засоби розуміються програмні засоби, що підтримують процеси створення і супроводу ІС, включаючи аналіз і формулювання вимог, проектування прикладного програмного забезпечення і баз даних, генерацію коду, тестування, документування, забезпечення якості, конфігураційне управління і управління проектом, а також інші процеси. CASE-засоби разом з системним ПЗ і технічними засобами утворюють повне середовище розробки ІС.

**Аналіз останніх досліджень і публікацій.** Комп'ютеризована інженерія програмного забезпечення (Computer-Aided Software Engineering - CASE) — це реалізація комп'ютерних засобів і методів у розробці програмного забезпечення. CASE використовується для забезпечення високоякісного та бездефектного програмного забезпечення. CASE забезпечує чіткий і дисциплінований підхід і допомагає дизайнерам, розробникам, тестувальникам, менеджерам та іншим бачити віхи проекту під час розробки. [1-2]. Однією з головних переваг використання CASE є виготовлення кінцевого продукту, який відповідатиме вимогам реального світу, оскільки сам процес розробки гарантує, що клієнти залишаються частиною процесу.

CASE-засоби ілюструють широкий набір трудомістких інструментів, які використовуються при розробці програмного забезпечення [4].

Різноманітні інструменти, включені в CASE, використовуються для підтримки різних етапів життєвого циклу розробки програмного забезпечення [5-9]. Інструменти діаграм допомагають у графічному представленні даних і системних процесів. Вони представляє елементи системи, потік керування та потік даних між різними програмними компонентами та системними структурами в графічній формі. Інструменти CASE спрощують створення та підтримку комплексної документації. Генерація коду та тестування — трудомісткі і повторювані операції, які виконують інструменти CASE. Завдяки такій автоматизації інженери можуть зосередитися на більш складних і творчих аспектах розробки програмного забезпечення, що підвищує продуктивність. Досягнення уніфікованості та стандартизації: домовленості про кодування, формати документації та шаблони проектування — області розробки програмного забезпечення, де інструменти CASE забезпечують уніфікованість і стандарти. Це гарантує послідовну та придатну для обслуговування розробку програмного забезпечення.

Розробка інформаційних систем з використання новітніх технологій є актуальною і для України. У роботі [10] досліджено особливості розвитку інформаційних систем і технологій, проведено аналіз їх впливу на формування



інформаційного суспільства та розглянуто сучасний рівень розвитку України за різними напрямками інформаційної індустрії. Визначено роль інформаційних систем у розвитку економіки, яка базується на знаннях та інформації, розглянуто види інформаційних систем та їх функціональні можливості. Це створює умови для врахування всіх витрат у процесі впровадження інформаційних технологій.

**Мета статті:** дослідження та аналіз CASE-засобів розробки інформаційних систем та впровадження їх для побудови інформаційної системи кафедри вищого навчального закладу .

**Виклад основного матеріалу.** CASE-технології є методологією проектування ІС, а також набір інструментальних засобів, що дозволяють в наочній формі моделювати предметну область, аналізувати цю модель на всіх етапах розробки і супроводу ІС і розробляти додаток відповідно до інформаційних потреб користувачів. Більшість існуючих CASE-засобів заснована на методологіях структурного або об'єктно-орієнтованого аналізу і проектування, що використовують специфікації у вигляді діаграм або текстів для опису зовнішніх вимог, зв'язків між моделями системи, динаміки поведінки системи і архітектури програмних засобів.

Для успішного впровадження CASE-засобів організація повинна володіти наступними якостями:

- *Технологія.* Розуміння обмеженості існуючих можливостей і здатність прийняти нову технологію;
- *Культура.* Готовність до впровадження нових процесів і взаємин між розробниками і користувачами;
- *Управління.* Чітке керівництво і організованість по відношенню до найбільш важливих етапів і процесів впровадження.

Візуальне моделювання - це спосіб представлення ідей і проблем реального світу за допомогою моделей. Модель допомагає зрозуміти проблему всім учасникам, задіяним у реалізації проекту на різних етапах: замовнику, експерту, проектувальнику, аналітику, автору документації, програмісту та ін. Моделювання забезпечує більш точну оцінку необхідних ресурсів, чітку проробку планів і ефективне функціонування створюваних систем.

Модель – це абстракція, що описує суть складної проблеми чи структури без акценту на несуттєві деталі, що робить її більш зрозумілою. Моделі допомагають нам організовувати, відображати, розуміти і створювати складні речі. Вони покликані допомогти у розв'язанні складних задач при розробці програм сьогодні та у майбутньому.

Для підтримки управління життєвим циклом розробки використовується методика, за допомогою якої можна детально описати технічні і організаційні аспекти створення програмного забезпечення на стадіях визначення вимог до системи, аналізу та проектування.

Робота над проектом складається з наступних етапів:

- задумка – визначення загальної ідеї проекту;

- проробка – планування необхідних робіт та ресурсів, створення архітектури;
- створення – побудова продукту за допомогою серії послідовних версій;
- поставка продукту користувачам та навчання персоналу.

Основним прикладом, на якому будуть демонструватися всі можливості, буде проект інформаційної системи кафедри вищого навчального закладу. До системи повинні мати доступ користувачі - викладачі, студенти та інженери з різними правами, які надає адміністратор. Користувачі мають доступ до матеріалів системи, виходячи зі своїх ролей і прав. При порушенні правил користувачам адміністратор має право їх заблокувати. Адміністратор повинен мати змогу переглянути дані про те, хто, на якому комп'ютері і коли заходив у систему. Адміністратор має змогу створити обліковий запис для нового користувача. Реалізацію розробки здійснюємо у середовищі об'єктно-орієнтованого CASE-засобу – Rational Rose [11].

Поведінка системи, що розробляється (тобто функціональність, що забезпечується системою), описується за допомогою функціональної моделі, яка відображує системні можливості (прецеденти, use cases), системне оточення (дійових осіб або акторів - actors) і зв'язки між можливостями і акторами (діаграми можливостей - use cases diagrams). Основне завдання моделі прецедентів - бути єдиним засобом, що дає змогу замовникові, кінцевому користувачеві і розробникові спільно обговорювати функціональність і поведінку системи.

Розробка моделі прецедентів починається на стадії задумки з вибору акторів і визначення спільних принципів функціонування системи. Потім на етапі опрацювання модель доповнюється детальною інформацією до існуючих можливостей, а при необхідності додаються нові.

Актори не є частиною системи - вони є кимось або чимось, що повинне взаємодіяти з системою. Актори можуть:

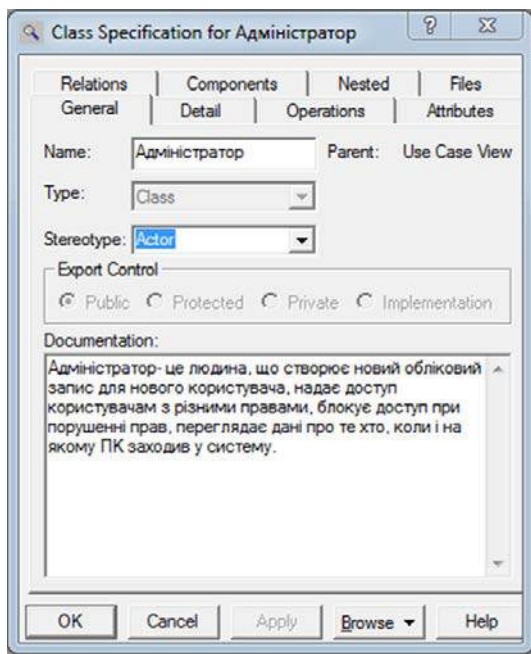
- лише забезпечувати інформацією систему;
- лише отримувати інформацію з системи;
- забезпечувати інформацією і отримувати інформацію з системи.

Зазвичай актори визначаються з опису завдання або шляхом переговорів із замовниками і експертами. Необхідно уважно підійти до питання визначення акторів для системи. Таке визначення зазвичай відбувається ітеративним чином - перший із затверджених списків акторів часто далекий від кінцевого. Для наведеного прикладу визначимо у системі двох акторів: *адміністратор* та *користувач*. Опис акторів наведено на рис. 1.

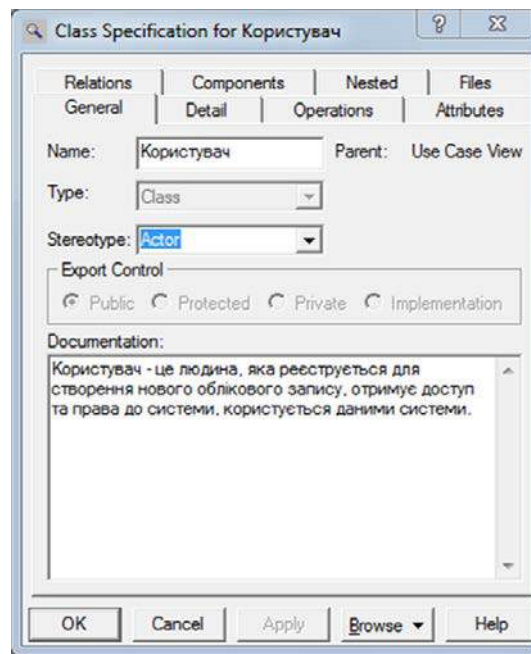
За допомогою прецедентів (use cases) моделюється діалог між актором і системою. Іншими словами, вони визначають можливості, що забезпечуються системою для актора. Набор всіх прецедентів системи визначає способи її використання. Для визначення правильного прецедента зручно використувати наступне правило: «Прецедент зазвичай визначає основний елемент



функціональності і здійснюється від початку до кінця. Він повинен приносити щось значиме для актора».



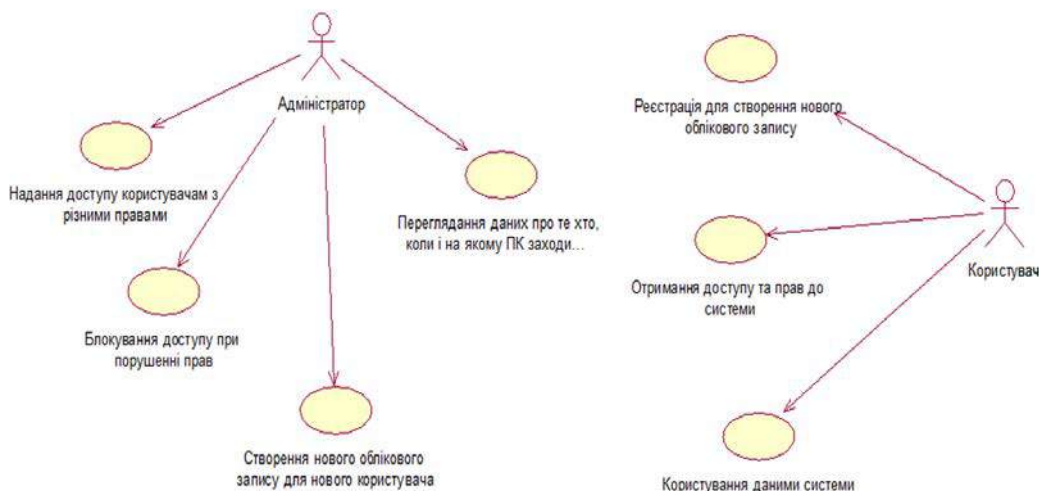
а)



б)

**Рис. 1** Опис акторів системи: а – адміністратор ; б - користувач

Діаграма прецедентів (use case diagram) - це графічне представлення всіх або частини акторів, прецедентів і їх взаємодій в системі. У кожній системі зазвичай є головна діаграма прецедентів, яка відображує кордони системи (акторів) і основну функціональну поведінку системи (прецеденти). Для наведеного прикладу діаграму можливостей наведено на рис. 2.



**Рис. 2** Діаграма прецедентів

Діаграми дій відображають динаміку проекту і являються схемами потоків управління в системі від дії до дії, а також паралельних дій і альтернативних потоків. В конкретній точці життєвого циклу діаграми дій можуть представляти потоки між функціями або всередині окремої функції. На різних етапах життєвого циклу вони створюються для відображення послідовності виконання операцій. Для розглядуваного прикладу діаграму можливостей наведено на рис. 3.

Об'єкт - це якась сутність реального світу або концептуальна сутність. Об'єкт може бути чимось конкретним, наприклад вантажівка або комп'ютер, або концептуальним, як, наприклад, хімічний процес, банківська операція, торговельне замовлення, кредитна історія або ставка прибутку.

Клас - це опис групи об'єктів із спільними властивостями (атрибутами), поведінкою (операціями), стосунками з іншими об'єктами і семантикою. Таким чином, класом є шаблон для створення об'єкту.

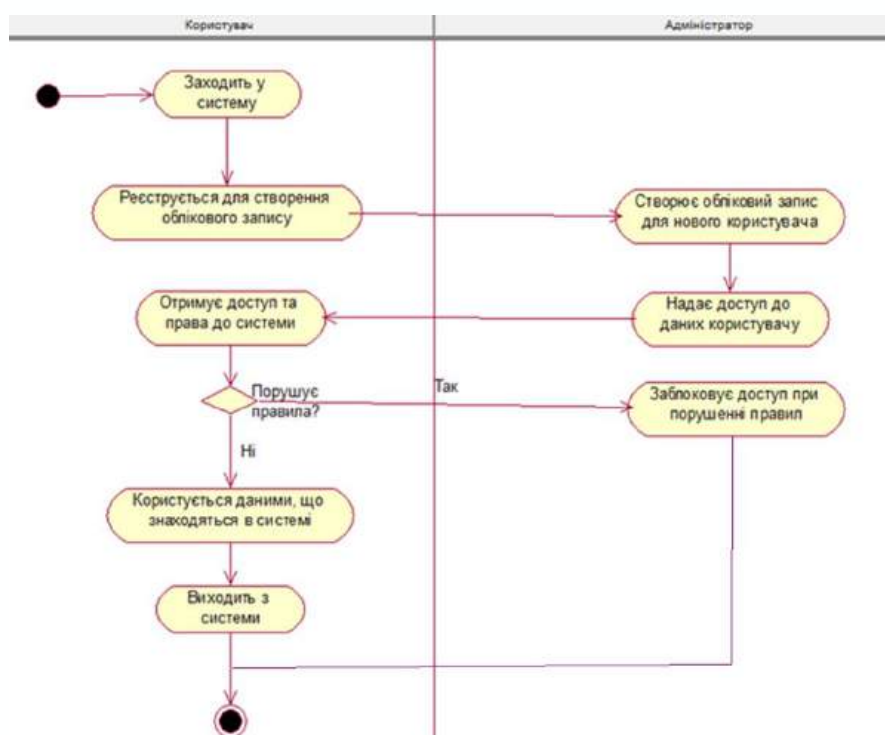
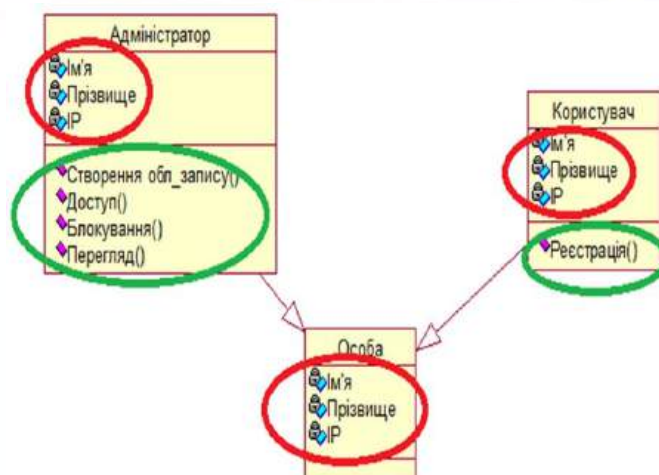


Рис. 3 Діаграма дій

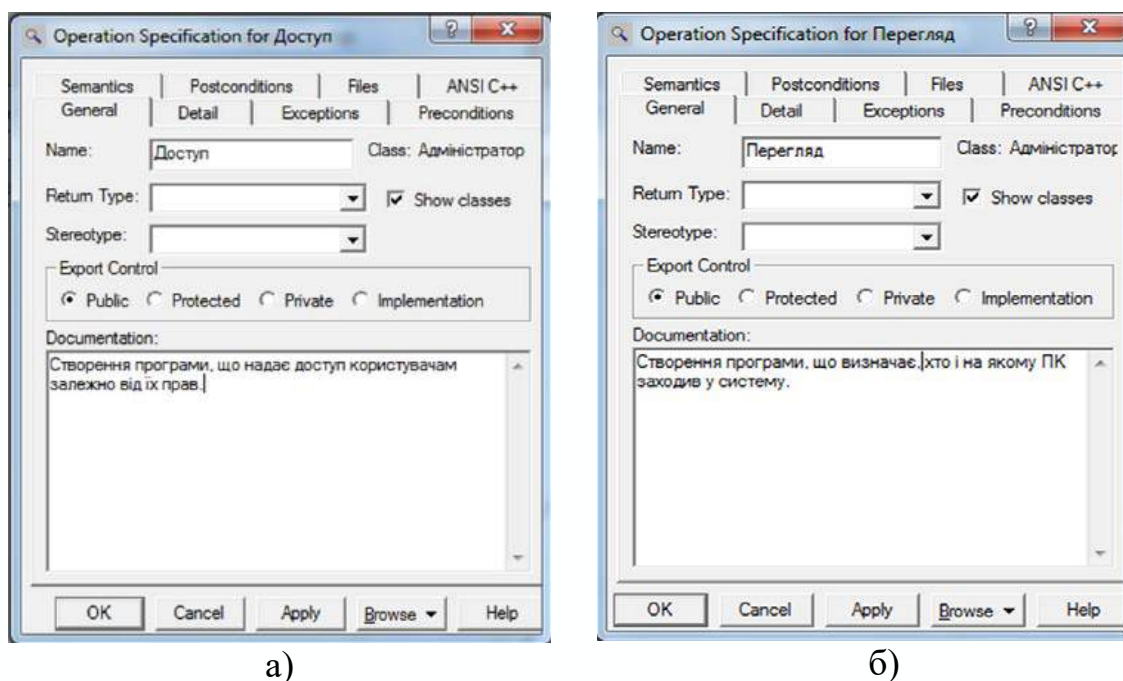
Кожен об'єкт є екземпляром конкретного класу і не може бути екземпляром декількох класів.

По міру того як нові класи додаються в систему, їх текстове представлення стає незручним. Діаграми класів допомагають графічно представити деякі або всі класи в моделі. Для розглядуваного прикладу діаграму класів наведено на рис. 4. Виділено поля та методи класів.



**Рис. 4** Діаграма класів

Методи класів визначають функціонал майбутньої системи. Опис деяких методів класів для прикладу інформаційної системи наведено на рис. 5.



а)

б)

**Рис. 5** Опис методів класу *адміністратор*: а – *доступ* ; б - *перегляд*

Наприкінці розробки інформаційної системи CASE-засоби дають змогу згенерувати програмний код. Для розглядуваного прикладу згенерований код мовою C++ наведено на рис. 6.

Отже, клас виконує ряд обов'язків, від яких залежить поведінка його об'єктів. Обов'язки виконуються за допомогою конкретних операцій. Структура об'єкту описується атрибутами класу. Кожен атрибут - це поле





*Зменшення витрат на обслуговування коду:* інструменти CASE використовуються у перевірці відстеження та узгодженості коду.

*Структурований підхід:* запровадження CASE впливає на роботу організації в цілому та робить її орієнтованою на структурований та впорядкований підхід.

*Швидші цикли розробки.* Генерація коду та тестування - це приклади операцій розробки програмного забезпечення, які можна автоматизувати, щоб скоротити цикли розробки.

*Зменшення помилок:* інструменти CASE допомагають зменшити кількість помилок, використовуючи автоматизований аналіз для виявлення можливих проблем на ранній стадії. Це дає змогу створити програмне забезпечення, яке є більш надійним і довговічним.

*Покращена співпраця:* технології CASE надають членам команди можливість покращення співпраці, що є найбільш актуальним для великих команд розробників і дає змогу покращити спілкування, зменшити непорозуміння та сприяти більш логічному процесу розробки.

Таким чином, інструменти CASE стають все більш важливими по мірі розвитку технологій, щоб задовольнити вимоги складних проектів розробки програмного забезпечення.

#### ***Література:***

1. Матеріали інформаційного сайту Comprehensive Overview of CASE Tools: Streamlining Software Development [Електронний ресурс, Posted on Oct, 2, 2023]. – Режим доступу: <https://dev.to/adityapratapbh1/comprehensive-overview-of-case-tools-streamlining-software-development-2p9p>

2. Матеріали інформаційного сайту Different Types of Reviews in the Software Development Life Cycle [Електронний ресурс, Posted on Aug 2 2024]. – Режим доступу: <https://dev.to/adityabhuyan/different-types-of-reviews-in-the-software-development-life-cycle-sdlc-3249>

3. Матеріали інформаційного сайту Computer Aided Software Engineering (CASE) [Електронний ресурс, Last Updated on Jan, 19, 2024]. – Режим доступу: <https://www.geeksforgeeks.org/computer-aided-software-engineering-case/>

4. Коваленко С., Добровська Л. Проектування інформаційних систем: загальні питання теорії проектування ІС. Київ: КПІ ім.Ігоря Сікорського, 2020. – 192 с.

5. Литвин В.В. , Пасічник В.В. , Шаховська Н.Б. Проектування інформаційних систем. Навчальний посібник, Львів: Магнолія, 2023. – 380 с. <https://openbook.in.ua/neshudozhnya-literatura/kompyuterna-literatura/proektyvannya-informacijnyh-system-lytvyn-vv/>

6. Масловський Б., Дрововозов В., Коба О. Технології проектування комп'ютерних систем. К.: Ін-т кібернетики ім. В.М. Глушкова НАН України, 2015. – 500 с. <https://profbook.com.ua/tehnologii-proektuvannya-komp-system.html?srsId=AfmBOoraOC2NhyQ9fPnaKDXu990kiQC4AhGS5wNVyiM8-l6g5Ttt3Nhg>

7. Павловський С.М., Бабков А.В. Основи автоматизованого проектування. – Одеса: Олді-Плюс, 2021. – 598 с. <https://card-file.ontu.edu.ua/handle/123456789/17523>

8. F. Miranda On the Evolution of Information Systems In book: Systems Theory: Perspectives, Applications and Developments. Nova Publishers, 2014. - p. 197-208 [https://www.researchgate.net/publication/263849621\\_On\\_the\\_Evolution\\_of\\_Information\\_Systems](https://www.researchgate.net/publication/263849621_On_the_Evolution_of_Information_Systems)

9. Третьуб В.Г. Проектування систем автоматизації. К.: Ліра-К, 2017, 344 с. <https://lira-k.com.ua/preview/12125.pdf?srsltid=AfmBOoqBePHkmb0bGNyKkdQzZpDqrDq9BbHgY1lYYEw6oQssfItpA4xm>

10. M. Voynarenko, V. Dzhuliy, L. Yemchuk Development of information systems and modeling of their implementation in the business // Problems and Perspectives in Management, N 14(3), 2016. – p. 102-107. DOI:10.21511/ppm.14(3).2016.10

11. Quatrani T. Visual modelling with Rational Rose 2000 and UML (2<sup>nd</sup> ed.) Edinburgh: Addison-Wesley Longman Ltd, 2000. – 256 p. <https://www.amazon.com/Visual-Modeling-Rational-Addison-Technology/dp/0201699613>

10. Матеріали інформаційного сайту Benefits of CASE – Software Engineering [Електронний ресурс, Last Updated on Jan, 19, 2024]. – Режим доступу: <https://www.geeksforgeeks.org/software-engineering-benefits-of-case/>

### References:

1. Materials of information site Comprehensive Overview of CASE Tools: Streamlining Software Development [Electronic resource, Posted on Oct, 2, 2023]. – Access mode: <https://dev.to/adityapratapbh1/comprehensive-overview-of-case-tools-streamlining-software-development-2p9p>

2. Materials of information site Different Types of Reviews in the Software Development Life Cycle [Electronic resource, Posted on Aug 2 2024]. – Access mode: <https://dev.to/adityabhuyan/different-types-of-reviews-in-the-software-development-life-cycle-sdlc-3249>

3. Materials of information site Computer Aided Software Engineering (CASE) [Electronic resource, Last Updated on Jan, 19, 2024]. – Access mode: <https://www.geeksforgeeks.org/computer-aided-software-engineering-case/>

4. Kovalenko S., Dobrovska L. Proektuvannia informatsiynyh system: zahalni pytannia teorii proektuvannia [Information Systems Design: General Issues in Design Theory]. Kyiv: KPI im. Ihoria Sikorskoho, 2020. – 192 p. [in Ukrainian].

5. Lytvyn V.V. , Pasichnyk V.V. , Shahovska N.B. Proektuvannia informatsiynyh system [Information systems design]. Lviv: Magnolia, 2023. – 380 p. [in Ukrainian]. <https://openbook.in.ua/ne-hudozhnya-literatura/kompyuterna-literatura/proektyvannya-informacijnyh-system-lytvyn-vv/>

6. Maslovskiy B., Drovovozov V., Koba O. Tekhnolohii proektuvannua komputernykh system [Computer systems design technologies]. Kyiv.: Instytut kibernetiky im. V.M. Hlushkova NAN Ukrainy, 2015. – 500 p. [in Ukrainian] <https://profbook.com.ua/tehnologii-proektuvannya-komp-system.html?srsltid=AfmBOoraOC2NhyQ9fPnaKDXu990KiQC4AhGS5wNVyiM8-l6g5Tt3Nxg>

7. Pavlovskiy S.M., BaBkov A.V. Osnovy avtomatyzovanoho proektuvannia [Fundamentals of computer-aided design]. – Odesa: Oldi-Plus, 2021. – 598 p. [in Ukrainian] <https://card-file.ontu.edu.ua/handle/123456789/17523>

8. F. Miranda On the Evolution of Information Systems In book: Systems Theory: Perspectives, Applications and Developments. Nova Publishers, 2014. - p. 197-208 [https://www.researchgate.net/publication/263849621\\_On\\_the\\_Evolution\\_of\\_Information\\_Systems](https://www.researchgate.net/publication/263849621_On_the_Evolution_of_Information_Systems)

9. Trehub V.G. Proektuvannia system avtomatyzatsii [Automation systems design]. Kyiv: Lira-K, 2017, 344 p. [in Ukrainian] <https://lira-k.com.ua/preview/12125.pdf?srsltid=AfmBOoqBePHkmb0bGNyKkdQzZpDqrDq9BbHgY1lYYEw6oQssfItpA4xm>

10. M. Voynarenko, V. Dzhuliy, L. Yemchuk Development of information systems and modeling of their implementation in the business // Problems and Perspectives in Management, N 14(3), 2016. – p. 102-107. DOI:10.21511/ppm.14(3).2016.10

11. Quatrani T. Visual modelling with Rational Rose 2000 and UML (2<sup>nd</sup> ed.) Edinburgh: Addison-Wesley Longman Ltd, 2000. – 256 p. <https://www.amazon.com/Visual-Modeling-Rational-Addison-Technology/dp/0201699613>

12. Materials of information site Benefits of CASE – Software Engineering [Electronic resource, Last Updated on Jan, 19, 2024]. – Access mode: <https://www.geeksforgeeks.org/software-engineering-benefits-of-case/>