

Кирийчук Д.Л.

Херсонський національний технічний університет

ВИКОРИСТАННЯ ОБ'ЄКТНО-ОРІЄНТОВАНИХ МЕТОДІВ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИ ПОБУДОВІ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ДІАГНОСТИКИ АВТОМАТИЗОВАНИХ РІЧТРАКІВ

У роботі запропоновано методологію розроблення інформаційної системи для діагностики автоматизованих річтраків. Для проектування інформаційної системи було обрано CASE-засіб *Enterprise Architect*, перевагами застосування якого є підтримка нотації UML 2.0, підтримка мов програмування C++ та Java, доступність для завантаження UML-профілів, що дозволяють створювати вузькоспеціалізовані моделі інформаційної системи, підтримка шаблонів проектування, генерація документації у форматах HTML і RTF, можливість автоматизації інтерфейсу та підтримка макросів.

Для розроблення діаграм класів, кооперації, діяльності, компонентів та розгортання було використано методологію об'єктно-орієнтованого аналізу і проектування та мову UML, що дозволило автоматизувати процес розроблення інформаційної системи діагностики автоматизованих річтраків.

Архітектура інформаційної системи представлена такими класами: Клас «Аналіз даних», Клас «Устаткування», Клас «Пристрій», Клас «Перелік устаткування» та Клас «Звіт». Клас «Аналіз даних» виконує аналіз даних, підготовку даних для побудови графіка та розрахунок оцінки стану обладнання. Виводить користувачу оцінку стану обладнання, а також у вигляді графіка відображає середньоквадратичне відхилення графіка роботи обладнання. Клас «Устаткування» є таблицею бази даних. Містить перелік ідентифікаторів обладнання та дані його місцезнаходження. Клас «Пристрій» є таблицею бази даних. Містить перелік датчиків діагностування обладнання та показників його роботи. Клас «Звіт» є таблицею бази даних. Містить дату та результати діагностування обладнання. Клас «Перелік» також є таблицею бази даних. Містить дані про кількість здійснених вимірювань під час діагностування обладнання. Розроблена інформаційна система дозволить здійснювати діагностику автоматизованих річтраків та своєчасно виявляти ряд недоліків у їх роботі.

Ключові слова: об'єктно-орієнтовані методи, програмне забезпечення, інформаційна система, CASE-засоби, автоматизовані річтраки.

Постановка проблеми. Останнім часом для розробки інформаційних систем (ІС) застосовується уніфікована мова моделювання UML, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю. UML є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення (ПЗ).

В останні роки розроблено велику кількість простих і універсальних програмних інструментальних засобів, що реалізують дружній інтерфейс мови UML, а також зручних засобів автоматизованого перетворення з моделі в програмний шаблонний код (CASE-засоби). Найбільш відомою є система IBM Rational Rose від розробника Rational Software.

Серед функціональних можливостей IBM Rational Rose є такі [1]:

- проектування систем будь-якої складності;
- підтримка мови UML;

– можливості автоматичного контролю, у тому числі перевірки відповідності двох моделей;

– надання розгорнутої інформації про проєкт (сумісно із засобами документування, зокрема SoDA);

– кодогенерування (стандартний список модулів включає C++, ADA, CORBA, Visual Basic, XML, COM, Oracle);

– зворотне проектування наявних систем.

Існують й інші виробники, які пропонують свої аналоги засобів підтримки UML, наприклад, Power Designer від компанії Sybase, Oracle Designer від компанії Oracle, ERwin від компанії CA Technologies та інші. Для побудови інформаційної системи автоматизованих річтраків в роботі буде використовуватися CASE-засіб *Enterprise Architect* від компанії Sparx System.

До переваг застосування *Enterprise Architect* можна віднести такі [2]:

- використання нотації UML 2.0 з підтримкою всіх видів діаграм;
- підтримку C++, Java, C#, VB, PHP;
- моделювання баз даних (БД), пряме проектування в DDL і зворотне проектування з ODBC;
- доступні для завантаження UML-профілі, що дозволяють створювати вузькоспеціалізовані моделі;
- підтримку шаблонів проектування;
- генерацію документації у форматах HTML і RTF;
- автоматизацію інтерфейсу, підтримку макросів.

Далі автор описує об'єктно-орієнтований підхід розробки та життєвий цикл інформаційної системи для діагностики автоматизованих річтраків із використанням UML у рамках процесу розробки ПЗ.

Аналіз останніх досліджень і публікацій. В роботі [2] розглядаються можливості застосування пакету для UML-проективання Sparx Enterprise Architect на прикладі системи обліку багажу в аеропортах та досліджено роль процесу моделювання інформаційних систем в ньому.

У роботі [3] розглядається діаграма прецедентів інформаційної системи для аналізу ринку праці IT-фахівців, яка створена за допомогою середовища Enterprise Architect з використанням уніфікованої мови моделювання UML. Обґрунтовується доцільність використання діаграми прецедентів при проектуванні інформаційної системи, описується призначення основних вузлів розробленої діаграми, подається загальна характеристика мови UML.

У роботі [4] розроблено програмний засіб для керування мережевим плануванням реінжинірингу програмного проекту. Моделювання системної архітектури програмного забезпечення в роботі виконується у рамках методології UML із використанням CASE-інструментарію Enterprise Architect.

У роботі [5] розглядається використання візуальних нотацій UML і P-схем в процесі навчання об'єктно-орієнтованих методів програмування. Пропонується новий наскрізний підхід до вивчення та використання техніки візуального моделювання програмних систем.

Формулювання цілей статті. Метою статті є побудова інформаційної системи для діагностики автоматизованих річтраків із використанням об'єктно-орієнтованих методів розроблення програмного забезпечення та уніфікованої мови моделювання UML.

Виклад основного матеріалу. Річтрак – штабелер з висувною щоглою, призначений для обслуговування висотних (від 6 м) стелажних систем [6]. Річтрак – найскладніший і високопродуктивний вид складської техніки, що поєднує в собі кращі якості штабелера і класичного вилочного навантажувача. На них встановлюється найсучасніше обладнання для обробки вантажів, включаючи преселектори висоти, камери на вилах, що дозволяють працювати на великих висотах. За технічним призначенням річтрак можна віднести до гібриду вилочного навантажувача і високопідйомного штабелера з висувним пристроєм вантажопідйомника на відстань від 600 до 1200 мм.

Основними відмінними особливостями такого симбіозу у річтраках є дуже висока продуктивність (швидкості пересування техніки та підйому вантажу практично в два рази перевищують аналогічні параметри штабелерів), економічність, відмінна маневреність і можливість оснащення колосальною кількістю додаткових опцій [6].

Основним кроком до побудови інформаційної системи для діагностики автоматизованих річтраків із використанням об'єктно-орієнтованих методів розроблення програмного забезпечення та уніфікованої мови моделювання UML є розробка діаграми класів. Діаграма класів є основним логічним поданням моделі і містить детальну інформацію про архітектуру об'єктно-орієнтованої програмної системи [1].

Архітектура об'єктно-орієнтованої програмної системи для діагностики автоматизованих річтраків представлена наступними класами: Клас «Аналіз даних», Клас «Устаткування», Клас «Пристрій», Клас «Перелік устаткування» та Клас «Звіт» (рис. 1).

Автор розглядає більш докладно класи системи. Так, Клас «Аналіз даних» виконує аналіз даних, підготовку даних для побудови графіка та розрахунок оцінки стану обладнання. Виводить користувачу оцінку стану обладнання, а також у вигляді графіка відображає середньоквадратичне відхилення графіка роботи обладнання. Клас «Устаткування» є таблицею бази даних. Містить перелік ідентифікаторів обладнання та дані його місцезнаходження. Клас «Пристрій» є таблицею бази даних. Містить перелік датчиків діагностування обладнання та показників його роботи. Клас «Звіт» є таблицею бази даних. Містить дату та результати діагностування обладнання. Клас «Перелік» також є таблицею бази даних. Містить дані про кількість здійснених вимірювань під час діагностування обладнання.

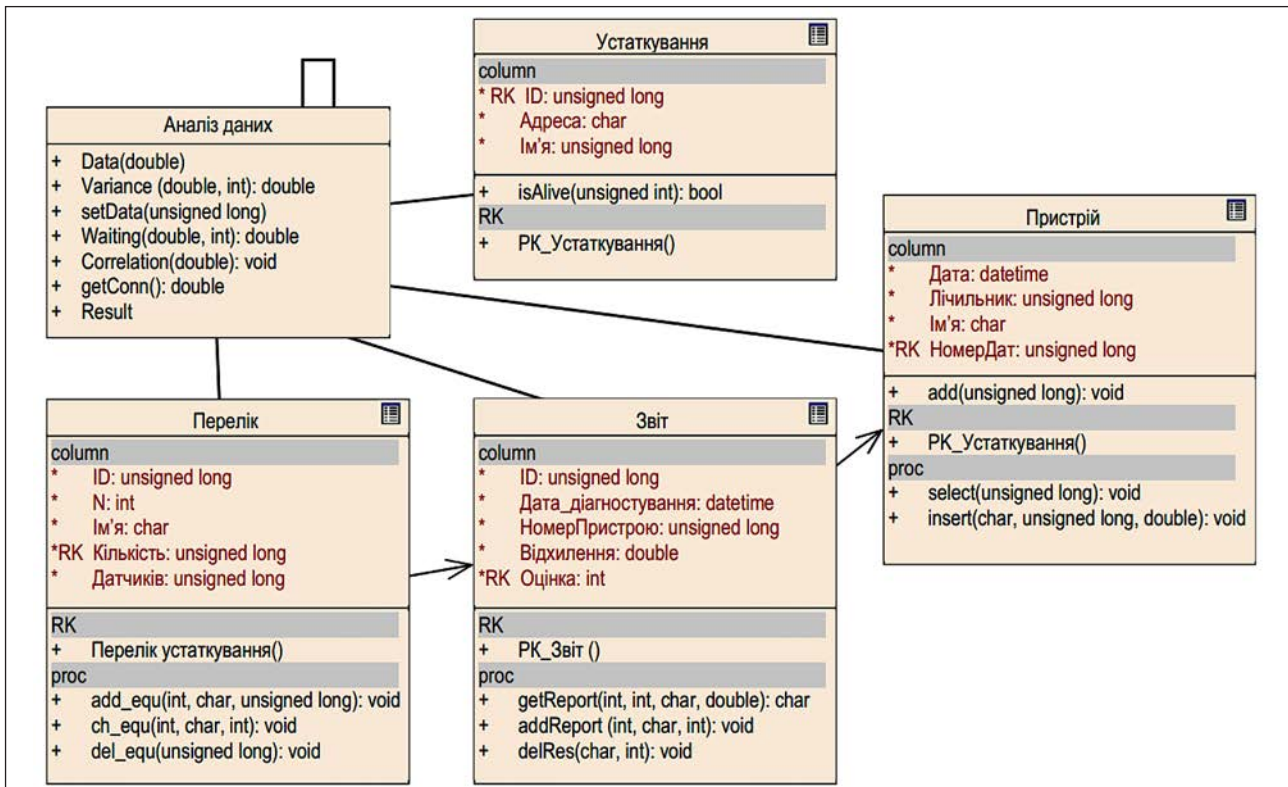


Рис. 1. Діаграма класів ІС

Діаграма послідовності дій показує послідовність дій об'єктів і класів системи в часі. За допомогою діаграми послідовності можна уявити взаємодію елементів моделі всієї сукупності об'єктів, пов'язаних між собою. На діаграмі послідовності дій зображуються тільки ті об'єкти, які безпосередньо беруть участь у взаємодії [1, 7–10].

В UML кожна взаємодія описується сукупністю повідомлень, якими ті об'єкти, що беруть участь у ньому обмінюються між собою. Повідомлення є закінченим фрагментом інформації, який відправляється одним об'єктом іншому. Прийм повідомлення ініціює виконання певних дій, спрямованих на вирішення окремого завдання тим об'єктом, якому це повідомлення відправлено.

Лінія життя об'єкта зображується пунктирною вертикальною лінією, асоційованою з єдиним об'єктом на діаграмі послідовності. Лінія життя служить для позначення періоду часу, протягом якого об'єкт існує в системі і, отже, може потенційно брати участь у всіх її взаємодіях. Якщо об'єкт існує в системі постійно, то і його лінія життя повинна продовжуватися по всій площині діаграми послідовності від самої верхньої її частини до самої нижньої [7]. Діаграма послідовності дій зображена на рис. 2.

На діаграмі зображено взаємодію файлу «Програма_діагностики» з таблицями «Устаткування», «Пристрій», «Результати_діагностики», «Перелік», а також взаємодію з користувачем: здійснення запити на перевірку зв'язку з обладнанням; передачу даних, що свідчить про наявність зв'язку і здійснює передачу даних до програми; запис отриманих даних в таблицю пристрою; завдання кількості вироблених вимірів датчиком; вибірку даних для проведення аналізу таблиці «Пристрій»; проведення математичного аналізу над даними; запис результатів діагностування обладнання в таблицю «Результати_діагностики»; відображення користувачу результатів діагностики.

Наступним логічним етапом є побудова діаграми кооперацій (рис. 3), що описує поведінку системи на рівні окремих об'єктів, які обмінюються між собою повідомленнями при реалізації деякого варіанту використання [8].

Автор розглядає повідомлення, якими обмінюються між собою об'єкти під час роботи системи: здійснення запити на перевірку зв'язку з обладнанням, додавання запису до таблиці пристрою, встановлення номера датчика та кількості вироблених вимірів датчиком, вибірка даних із таблиці «Пристрій», повернення вибраних даних

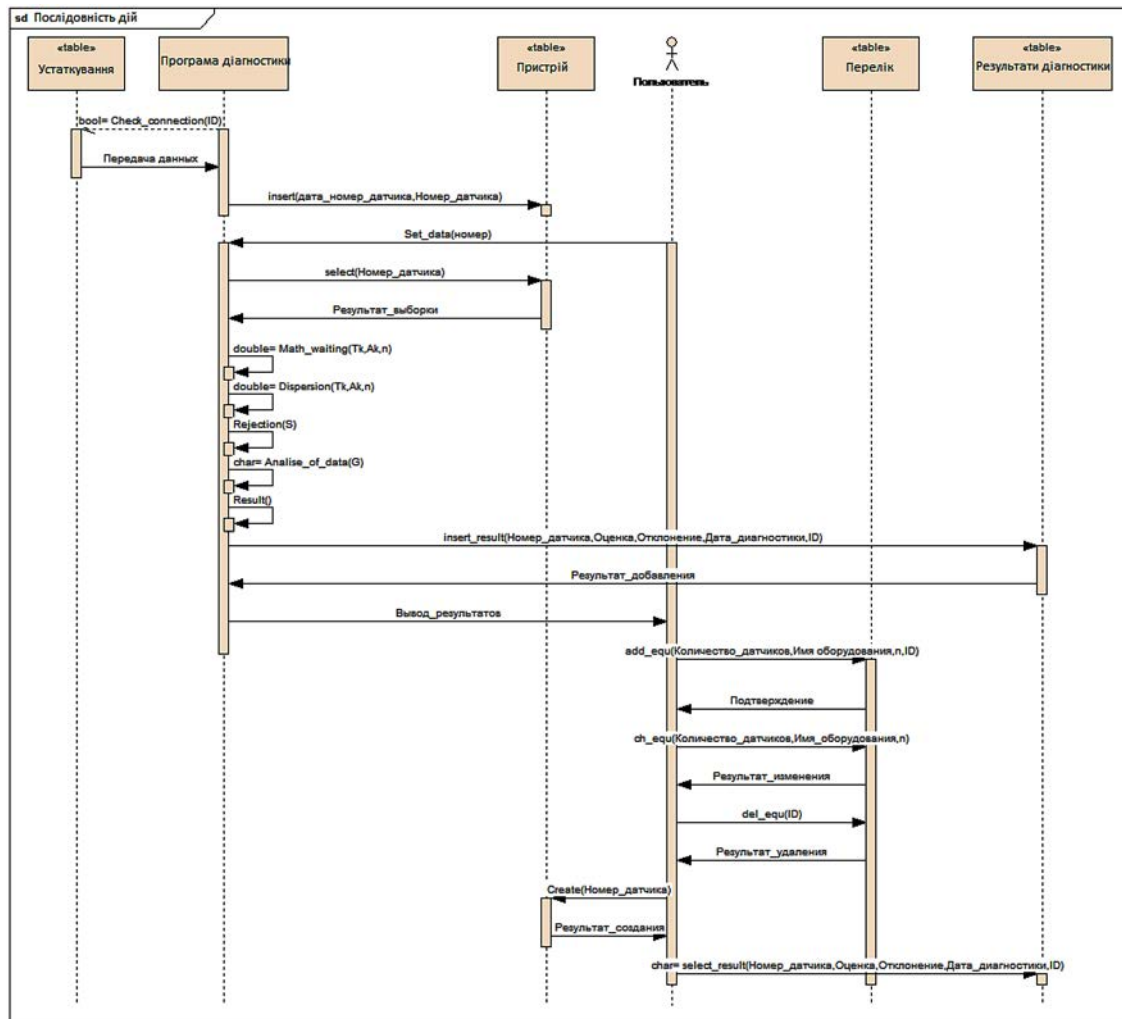


Рис. 2. Діаграма послідовності дій ІС

програмою, проведення аналізу даних, підготовка результатів виведення, запис даних в таблицю «Результати діагностики», відображення користувачеві результатів діагностики, додавання запису в таблицю «Перелік», зміна запису в таблиці «Перелік», видалення запису з таблиці «Перелік», додавання запису в таблицю «Пристрій», вибірка даних з таблиці «Результати діагностики».

Діаграма діяльності дозволяє моделювати послідовності дій, реалізованих методами класів. Діаграми діяльності є аналогом блок-схеми будь-якого алгоритму. Вони, як і діаграми станів та переходів, відображаються у вигляді орієнтованого графу, вершинами якого є дії, а ребра – переходи між діями. Кожний стан на діаграмі діяльності відповідає виконанню деякої елементарної операції, а перехід в наступний стан виконується тільки після завершення цієї операції [7–9].

Основним напрямом використання діаграми діяльності є візуалізація особливостей реалізації

операцій класів, коли необхідно надати алгоритми їх виконання. Діаграма діяльності зображена на рис. 4.

Діаграма компонентів відрізняється від раніше описаних діаграм тим, що описує особливості фізичного подання системи. Діаграма компонентів відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного забезпечення може бути представлено як компоненту. Деякі компоненти існують під час компіляції, деякі – під час компонування, інші – під час роботи програми [7]. Діаграму компонентів подано на рис. 5.

Основними елементами інформаційної системи є файл, що виконується, база даних і звіт. Компонент «Програма» має стереотип «Executable», що означає, що цей компонент є таким, який буде виконано.

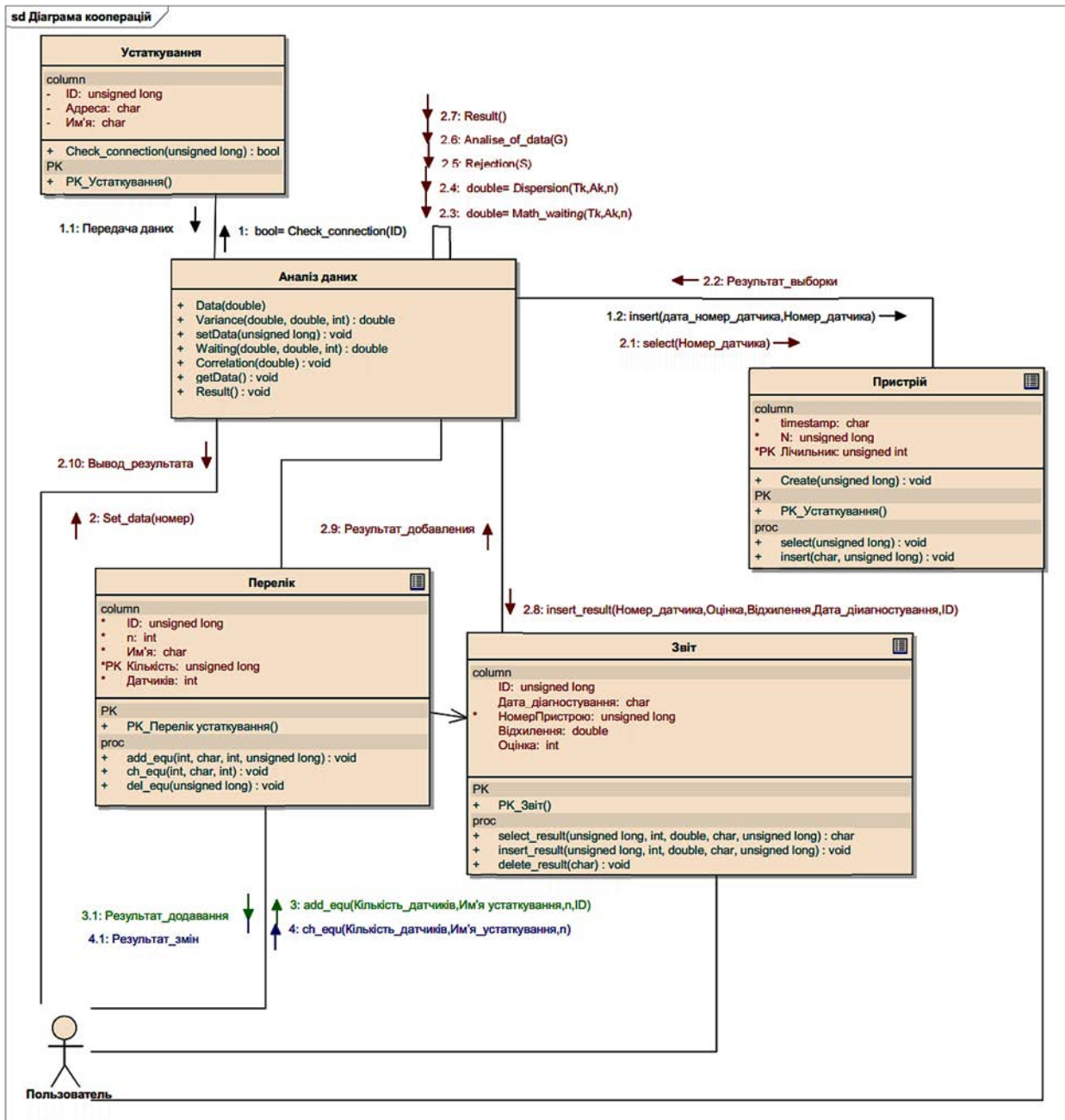


Рис. 3. Діаграма кооперації ІС

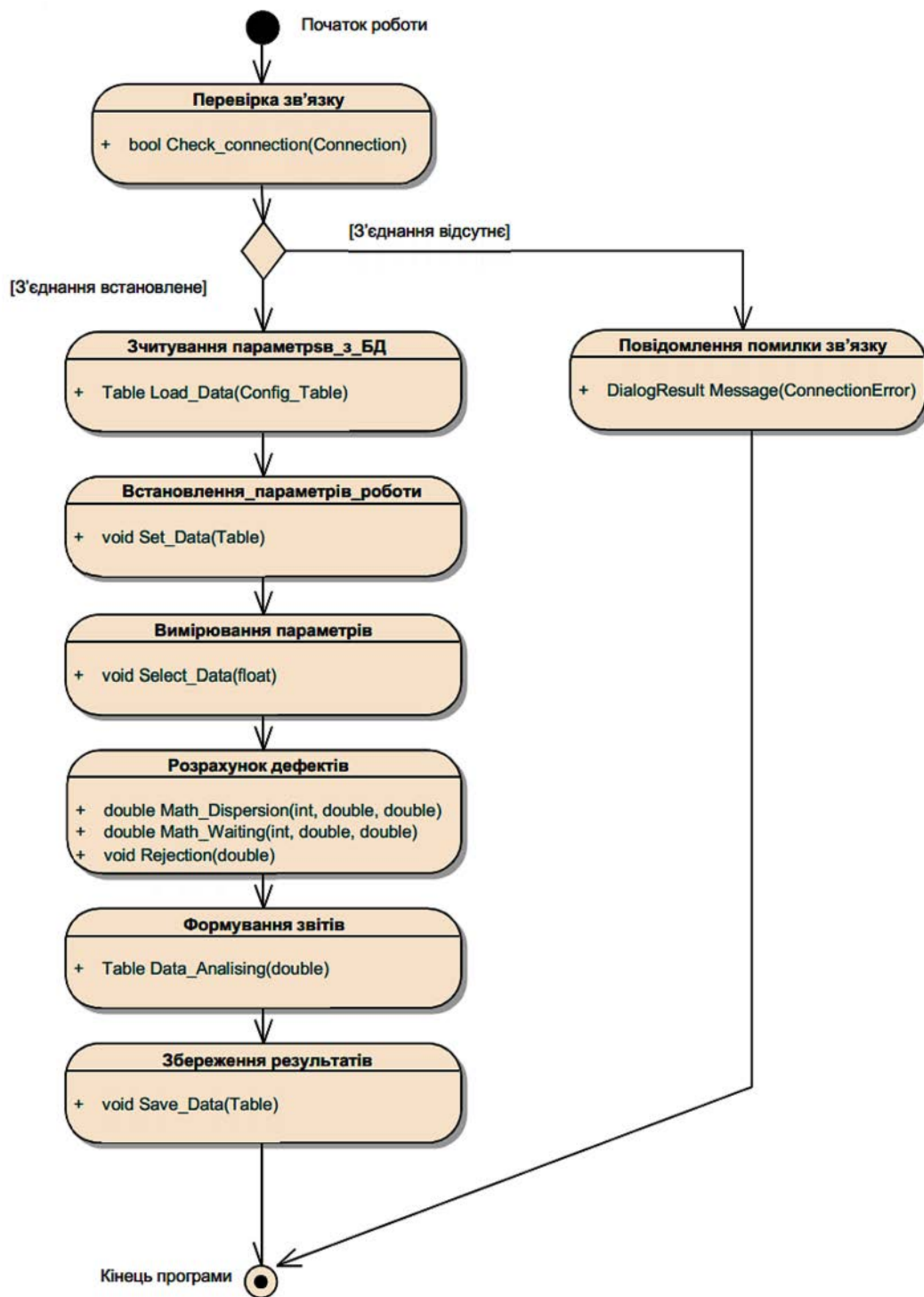


Рис. 4. Діаграма діяльності ІС

Діаграма розгортання призначена для візуалізації елементів і компонентів програми, які існують тільки на етапі її виконання (run-time). При цьому наведено тільки ті компоненти системи, які є виконуваними файлами або динамічними бібліотеками [1, 7, 9]. Діаграму розгортання ІС зображено на рис. 6.

Компоненти, які не використовуються на етапі виконання, на діаграмі розгортання не відображаються. Діаграма розгортання дозволяє проаналізувати апаратну частину системи, виділити обчислювальні ресурси, пристрої, які використовуються ними і з'єднання між ними, а також спроектувати розташування цих частин. Діаграма призначена

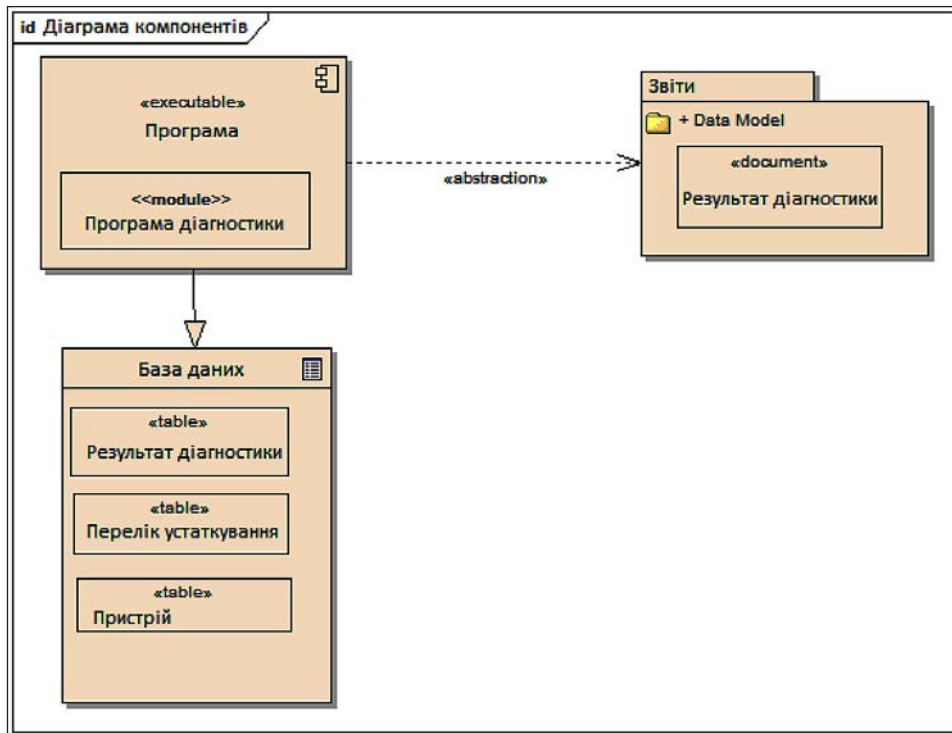


Рис. 5. Діаграма компонентів ІС

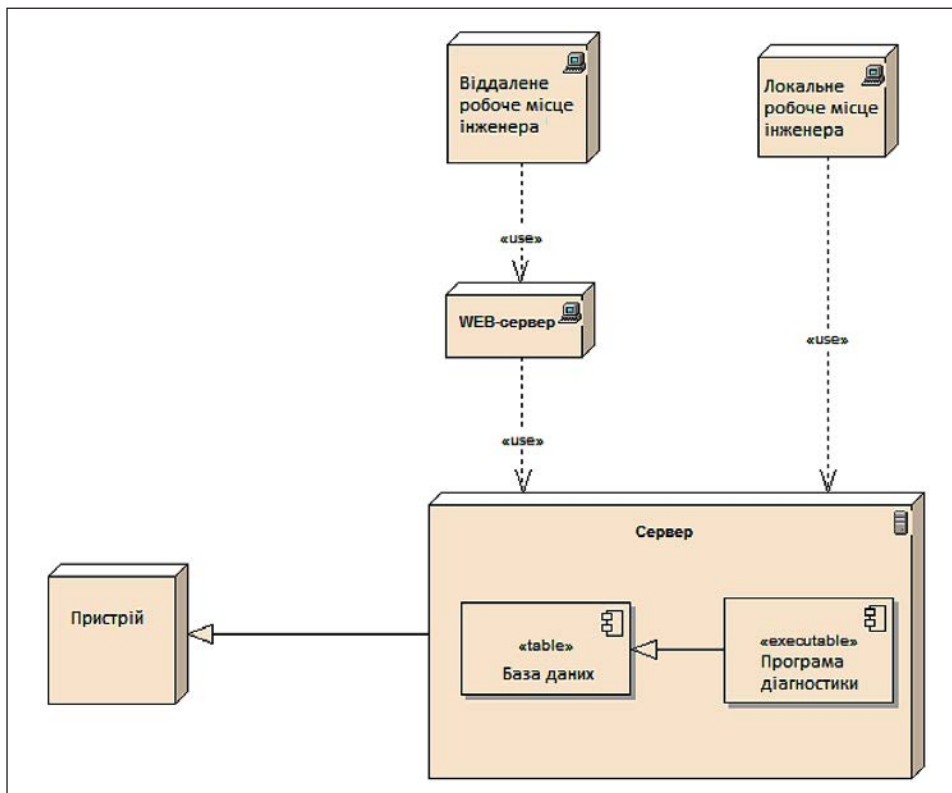


Рис. 6. Діаграма розгортання ІС

для візуалізації елементів і компонентів програми, що існують лише на етапі її виконання (runtime).

Висновки. В роботі розроблено діаграми класів, кооперації, діяльності компонентів та роз-

гортання системи діагностики автоматизованих річтраків із використанням методології об'єктно-орієнтованого аналізу і проектування та мови UML.

Список літератури:

1. Боггс У., Боггс М. UML и Rational Rose. Изд-во Лори, 2001. 299 с.
2. Боровик В.М., Андрушук О.В. Моделювання системи обліку багажу в аеропортах. *Проблеми інформатизації та управління*. 2013. Вип. 2(42). С. 14–18.
3. Шаров С.В., Шарова Т.М. (2018) Проектування діаграми прецедентів інформаційної системи для аналізу ринку праці IT-фахівців. *Сучасні проблеми моделювання : зб. наук. пр. (11)*. 2018. С. 170–175.
4. Великодний С.С., Бурлаченко Ж.В., Зайцева-Великодна С.С. Розробка архітектури програмного засобу для управління мережевим плануванням реінжинірингу програмного проекту. *Сучасний стан наукових досліджень та технологій в промисловості*. 2019. № 2 (8). С. 25–35.
5. Дробушевич Л.Ф., Совместное использование нотаций UML и P-схем в процессе обучения методам разработки программных систем, MEDIAS-2010, Кипр, с. 73–77.
6. Александров М.П., Колобов Л.Н., Лобов Н.А. и др. Грузоподъемные машины : Учебник для вузов. М. : Машиностроение, 1986, 400 с.
7. Леоненков А. Самоучитель UML. Петербург : БХВ, 2004.
8. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя 2-е изд.: Пер. с англ. Мухин Н.М. ДМК Пресс, 2006.
9. Кватрани Т. Визуальное моделирование с помощью IBM Rational Software Architect и UML. М.: Кудиц-образ. 2007.
10. Ларман К. Применение UML 2.0 и шаблонов проектирования. Практическое руководство. М. : Диалектика/Вильямс, 2013.

Kyryichuk D.L. USING OBJECT-ORIENTED SOFTWARE DEVELOPMENT METHODS TO BUILD AN INFORMATION SYSTEM FOR REACH TRUCK DIAGNOSIS

A methodology for developing an information system for the diagnosis of Reach Truck has been proposed. The Enterprise Architect CASE tool was chosen to design the information system. The benefits of Enterprise Architect include the support for UML 2.0 notation, C ++ and Java programming languages, the ability to download UML profiles to create highly specialized information system models, support for design templates, HTML and RTF documentation generation, interface automation, and macro support.

Object-oriented analysis and design methodology and the UML language were used to develop class diagram, cooperation's diagram, activities diagram, components diagram, and deployments diagram, which made it possible to automate the process of designing automated Reach Truck diagnostics.

The following classes represent the information system architecture: Data Analysis Class, Equipment Class, Device Class, Equipment List Class, and Report Class. The Data Analysis class performs data analysis, preparation of data for graphing, and calculation of equipment condition assessment. Outputs to the user an estimate of the condition of the equipment, and also in the form of a graph displays the standard deviation of the schedule of the equipment.

The Equipment class is a database table. Provides a list of equipment IDs and location data. The Device class is a database table. Provides a list of sensors to diagnose the equipment and its performance. The Report class is a database table. Provides date and results of equipment diagnostics. The List class is also a database table. Provides information on the number of measurements made during equipment diagnosis. The developed information system will allow to diagnose automated Reach Truck and to identify a number of shortcomings in their work in a timely manner.

Key words: *object-oriented methods, software, information system, CASE-tools, automated Reach Truck.*