

УДК 004.01

Гудзовата О. О.,

ORCID ID: 0000-0001-7739-9978, Researcher ID: G-5953-2019,

к.е.н., доц., доцент кафедри комп'ютерних наук, Львівський торговельно-економічний університет, м. Львів

Костирко В. І.,

к.т.н., доц., доцент кафедри комп'ютерних наук, Львівський торговельно-економічний університет, м. Львів

Артищук І. В.,

ORCID ID: 0000-0001-7287-8451, Researcher ID: G-4924-2019,

к.е.н., доц., доцент кафедри комп'ютерних наук, Львівський торговельно-економічний університет, м. Львів

ВИКОРИСТАННЯ УНІФІКОВАНОЇ МОВИ ВІЗУАЛЬНОГО МОДЕЛЮВАННЯ UML (UNIFIED MODELING LANGUAGE) ЯК ІНСТРУМЕНТУ ПІДТРИМКИ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

***Анотація.** У статті розглянуто сучасні засоби проектування інформаційних систем (ІС) і моделювання прикладних задач, зокрема уніфіковану мову візуального моделювання UML. У дослідженні викладено основні характеристики UML як інструменту розробки інформаційних систем і їх структурних елементів, а також реалізований приклад використання UML для моделювання задач. Встановлено, що більшість CASE-засобів (програм автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі, розроблені в UML, дозволяють значно спростити процес кодування і пришвидшити реалізацію самої системи. Уніфікована мова візуального моделювання UML використовується в багатьох успішних проектах інформатизації. Подальші дослідження будуть спрямовані на розроблені засобами UML моделі, що дозволяє значно спростити процес кодування і пришвидшити реалізацію самої ІС.*

Ключові слова: інформаційна система (ІС), інформаційні технології (ІТ), проектування, прикладні задачі, уніфікована мова візуального моделювання UML.

Gudzovata O.O.,

ORCID ID: 0000-0001-7739-9978, Researcher ID: G-5953-2019,

Ph.D., Associate Professor, Associate Professor of the Department of Computer Sciences, Lviv University of Trade and Economics, Lviv

Kostyrko V.I.,

Ph.D., Associate Professor, Associate Professor of the Department of Computer Sciences, Lviv University of Trade and Economics, Lviv

Artyshchuk I. V.,

ORCID ID: 0000-0001-7287-8451, Researcher ID: G-4924-2019,

Ph.D., Associate Professor, Associate Professor of the Department of Computer Sciences, Lviv University of Trade and Economics, Lviv

THE USE OF UNIFIED MODELING LANGUAGE (UML) AS A TOOL FOR INFORMATION SYSTEMS DESIGNING SUPPORT

Abstract. *The article considers modern means of information systems designing and modeling of applied problems, in particular, the unified language of visual modeling. The study reveals the main characteristics of unified modeling language as a tool for developing information systems and their structural elements, as well as shows the example of using this language for modeling tasks. It is found that most CASE-tools (automation software for analysis and design)*

have support of unified modeling language. Models developed with the help of unified modeling language allow for a significant simplification of the coding process and speed up the implementation of the system itself. The visual modeling language is used in many successful informatization projects. Further research will be aimed at developed by means of the unified modeling language models, which can significantly simplify the process of encoding and accelerate the implementation of the information system itself.

Key words: information system, information technologies, design, applied tasks, unified language of visual modeling.

JEL Classification: L86; C51; C88; O32

DOI: <https://doi.org/10.36477/2522-1256-2019-24-16>

Постановка проблеми. Зростання ролі сучасних інформаційних систем (ІС) в умовах нової економіки обумовлене конкурентними перевагами, які вони надають компаніям. Для розробки та впровадження таких систем призначені сучасні засоби їх проектування. Методологія створення інформаційних систем передбачає організацію процесу побудови інформаційної системи для підприємства та забезпечення управління цим процесом.

Починаючи з середини 60-х років і донедавна широке розповсюдження мали *структурні* методології аналізу, проектування і розробки інформаційних систем, що характеризуються штучним поділом (часто неоптимальним) системи на підсистеми, а також слабким взаємозв'язком процесів і даних, які присутні в системі. На відміну від них, *об'єктні* технології, орієнтовані на тісний взаємозв'язок процесів і даних у системах, дозволяють програмним системам бути надійнішими, легшими для реалізації та стійкішими до змін. Крім того, така філософія моделювання найкраще відповідає загальним концепціям поведінки систем реального світу. І використання уніфікованої мови візуального моделювання UML (Unified Modeling Language) є прикладом оптимізації процесу проектування та розробки як ІС, так і конкретних прикладних задач.

Аналіз останніх досліджень і публікацій. Методологія проектування ІС реалізується через конкретні технології та підтримуючі стандарти, методики та інструментальні засоби, що забезпечують підтримку життєвого циклу системи. Ця проблематика широко висвітлювалась у роботах І. Жирякова, А. Марченко, Б. Литвина, Б. Одинцова, Н. Шаховської та багатьох інших дослідників.

Постановка завдання. Метою роботи є ознайомлення з основними засадами уніфікованої мови візуального моделювання UML як інструменту розробки інформаційних систем і їх структурних елементів, а також використання UML для моделювання прикладних задач (у нашому випадку – складання розкладу).

Виклад основного матеріалу дослідження. Незважаючи на перевагу *об'єктно-орієнтованих технологій* аналізу та проектування інформаційних систем (ІС) перед *структурними*, їх поширення до останнього часу було незначним, оскільки жоден з методів не давав єдиної і цілісної об'єктної моделі системи. Кожен метод висвітлював одну або декілька сторін реальної системи, залишаючи поза увагою багато інших, не менш важливих сторін. Крім

того, відсутність єдиного стандарту не сприяла широкому поширенню об'єктно-орієнтованих методів при розробці програмного забезпечення.

Потужний поштовх до розробки цього напрямку ІТ дало поширення об'єктно-орієнтованих мов програмування в кінці 1980-х - на початку 1990-х років. Користувачам хотілось отримати єдину мову моделювання, яка об'єднала би в собі всі переваги об'єктно-орієнтованого підходу і давала би чітку модель системи, яка відображатиме всі її значимі сторони. До середини 90-х явними лідерами в цій галузі стали методи *Booch* (Grady Booch), *OMT-2* (Jim Rumbaugh), *OOSE – Object-Oriented Software Engineering* (Ivar Jacobson). Однак ці три методи мали свої сильні та слабкі сторони: OOSE був кращим на стадії аналізу проблемної області та аналізу вимог до системи, OMT-2 був найкращий на стадіях аналізу та розробки інформаційних систем, Booch найкраще підходив для стадій дизайну та розробки.

Мовою, яка об'єднала сильні сторони відомих методів і забезпечила найкращу підтримку моделювання, стала *уніфікована мова моделювання UML* (Unified Modeling Language). Вона з'явилася внаслідок розвитку методів *об'єктно-орієнтованого аналізу і проектування (OOA&D – Object-Oriented Analyses&Design)*. Мова моделювання пройшла процес стандартизації в межах консорціуму *OMG (Object Management Group)* і на сьогодні прийнята як фактичний стандарт *OMG*.

UML – це сімейство *графічних нотацій*, яке допомагає в описі та проектуванні програмних систем, особливо систем, побудованих з використанням об'єктно-орієнтованої парадигми. Графічні мови моделювання вже тривалий час широко використовуються в індустрії програмування і проектування інформаційних систем.

UML є відносно відкритим стандартом, що знаходиться під управлінням групи *OMG (Object Management Group)*, відкритого консорціуму компанії. Група *OMG* була сформована для створення стандартів, які підтримують міжсистемну взаємодію, зокрема взаємодію об'єктно-орієнтованих систем. Група *OMG* ще є відомою за стандартами *CORBA (Common Object Request Broker Architecture)*.

Існує безліч технологій та інструментальних засобів, за допомогою яких можна реалізувати в певному сенсі оптимальний проект ІС, починаючи з етапу аналізу і закінчуючи створенням програмного коду системи. У більшості випадків ці технології пред'являють досить жорсткі вимоги до процесу

розробки і використовуваних ресурсів, а спроби трансформувати їх під конкретні проекти виявляються безуспішними. Ці технології представлені CASE-засобами верхнього рівня або CASE-засобами повного життєвого циклу (upper CASE tools або full life-cycle CASE tools). Вони не дозволяють оптимізувати діяльність на рівні окремих елементів проекту, і, як наслідок, багато розробників перейшли на так звані CASE-засоби нижнього рівня (lower CASE tools). Однак вони стикнулися з новою проблемою – проблемою організації взаємодії між різними командами, які реалізують проект.

Уніфікована мова об'єктно-орієнтованого моделювання Unified Modeling Language (UML) стала засобом досягнення компромісу між цими підходами. Існує достатня кількість інструментальних засобів, що підтримують за допомогою UML життєвий цикл інформаційних систем, і одночасно UML є досить гнучким для настройки і підтримки специфіки діяльності різних команд розробників.

UML – об'єктно-орієнтована мова моделювання, що володіє наступними основними характеристиками:

- є мовою візуального моделювання, що забезпечує розробку репрезентативних моделей для організації взаємодії замовника і розробника ІС, різних груп розробників ІС;
- містить механізми розширення і спеціалізації базових концепцій мови.

Сьогодні UML є загально визнаним стандартом, який використовує більшість розробників системного та прикладного програмного забезпечення. UML підтримується багатьма *об'єктно-орієнтованими CASE-продуктами*. Знання UML є необхідним не лише для системних аналітиків і проектувальників, але й для звичайних програмістів і тестувальників програмного забезпечення. Постійно збільшується

ринок UML-орієнтованих інструментальних засобів, призначених для автоматизації процесу розробки програм. Безперечно, UML відіграватиме важливу роль у галузі розробки програмного забезпечення і в майбутньому. Розвиток UML спрямований на спрощення розв'язку однієї з найскладніших задач в галузі інформаційних технологій – *задачі проектування ПЗ*.

UML призначена для моделювання програмних систем. Самі автори UML визначають її як графічну мову моделювання загального призначення, яку використовують для специфікації, візуалізації, конструювання і документування усіх артефактів (модель, програмний код, документація тощо), які створюються під час розробки програмних систем (табл. 1).

Мова UML призначена для вирішення наступних завдань:

- надавати в розпорядження користувачів готову до використання виразну потужну мову візуального моделювання, що дозволяє розробляти осмислені моделі й обмінюватися ними;
- передбачити внутрішні механізми розширюваності й спеціалізації базових концепцій мови;
- забезпечити максимальну незалежність проекту створення ПЗ від конкретних мов програмування і процесів розробки;
- забезпечити формальну основу для однозначної інтерпретації мови;
- стимулювати розширення ринку об'єктно-орієнтованих інструментальних засобів створення програмного забезпечення;
- інтегрувати кращий практичний досвід використання мови й реалізації програмних засобів його підтримки.

UML є уніфікованою мовою моделювання, яка використовується у парадигмі ООП (об'єктно-орієнтованого моделювання).

Таблиця 1

Види робіт при моделюванні систем засобами UML

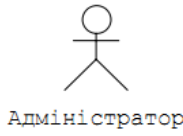
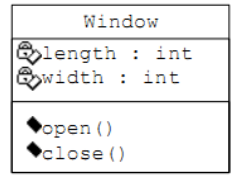

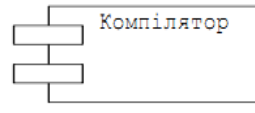

Види робіт	Характеристика робіт
Специфікація	Декларативний опис того, як усе побудоване або працює. UML надає достатньо формальні та універсальні засоби для специфікації усіх можливих артефактів, що дає змогу знизити ризик неоднозначного сприйняття специфікації.
Візуалізація	Представлення інформації у графічній формі, придатній для сприйняття людиною. Часто моделювання є єдиним засобом, який дає змогу уявити систему загалом як одне ціле. Проблема полягає в обмеженому сприйнятті людиною складних сутностей. Моделювання передбачає розділення складної системи на дещо простіші складові та окремо розглядає кожну з цих складових. Також моделювання дає змогу створювати високорівневі моделі всієї системи, відкидаючи деталі, несуттєві для цього рівня абстракції.
Конструювання	Отримання набору програмних модулів, які утворюють застосування або його компонент. Розроблені моделі системи утворюють деякий базовий каркас, на основі якого можна будувати систему. Сучасні CASE-засоби дають змогу деякою мірою автоматизувати конструювання програмного коду на підставі розроблених моделей.
Документування проектних рішень	Для підтримки та розвитку програмних продуктів потрібна вичерпна та якісна документація. Моделювання дає змогу одержати документи, які визначають високорівневу організацію системи. Такі документи необхідні для початкового ознайомлення з системою.

Для UML базовими поняттями є *сутності, відношення, діаграми*. Сутності – це певні абстракції, які є базовими елементами моделей. В UML є чотири типи сутностей: структурні (актори, класи, інтерфейси, компоненти, вузли), поведінкові (прецеденти, діяльності, стани і повідомлення), групування, анотаційні. Структурні сутності – це статичні

поняття, які відповідають концептуальним, логічним чи фізичним елементам системи. Структурні сутності зазвичай позначають іменниками. Розрізняють п'ять головних структурних сутностей: актори, класи, інтерфейси, компоненти, вузли (табл. 2). Кожна з сутностей може мати свої підвиди.

Таблиця 2

Графічне зображення структурних сутностей UML-моделей

Назва	Означення	Зображення
Актор (Actor)	Суб'єкт, який перебуває поза системою, що моделюється, і безпосередньо з нею взаємодіє.	
Клас (Class)	Сукупність однотипних сутностей предметної області (об'єктів) зі спільними атрибутами, операціями, відношеннями та семантикою.	
Інтерфейс (Interface)	Сукупність операцій, що формують деякий сервіс, який надає клас чи компонент.	
Компонент (Component)	Фізично заміщувана частина системи, яка відповідає певному набору інтерфейсів і/або забезпечує реалізацію іншого набору інтерфейсів.	
Вузол (Node)	Фізичний елемент системи, який існує під час виконання програми і є обчислювальним ресурсом.	

В UML використовуються такі види діаграм (табл. 3):

Таблиця 3

Види діаграм

<i>Структурні діаграми:</i>	<i>Structure Diagrams:</i>
Класів	Class diagram
Компонент	Component diagram
Композитної/складеної структури	Composite structure diagram
Кооперації (UML2.0)	Collaboration (UML2.0)
Розгортання	Deployment diagram
Об'єктів	Object diagram
Пакетів	Package diagram
<i>Діаграми поведінки:</i>	<i>Behavior Diagrams:</i>
Діяльності	Activity diagram
Станів	State Machine diagram
Прецедентів	Use case diagram
<i>Діаграми взаємодії:</i>	<i>Interaction Diagrams:</i>
Кооперації (UML1.x) /Комунікації (UML2.0)	Collaboration (UML1.x) /Communication diagram (UML2.0)
Огляду взаємодії (UML2.0)	Interaction overview diagram (UML2.0)
Послідовності	Sequence diagram
Синхронізації (UML2.0)	UML Timing Diagram (UML2.0)

Мова UML широко використовується для моделювання *прикладних задач*. Розглянемо використання уніфікованої мови візуального моделювання UML як інструменту підтримки проектування ІС на прикладі моделювання процесу складання розкладу.

Для розробки UML-діаграми використаємо ПЗ ModelioSoft та тип діаграми *Use-Case* або діаграму *прецедентів*.

Use-case diagram – це опис сценаріїв взаємодії учасників (зазвичай користувача та системи). Учасників може бути 2 і більше, користувачами може виступати як людина, так і інша система.

Послідовність дій:

1. *Визначимо, кому і в яких випадках потрібні сценарії:*

- *Розробникам*. Зручно, коли вимога описана за допомогою основного й альтернативного потоку подій, при якому чітким і зрозумілим є: хто, коли і що потребує, і що виявляється в результаті.

- *Замовникам*. Описано людською мовою, замовник може своєчасно підтвердити, що це саме те, чого він очікує, або внести правки.

- *Тестувальнику*. Майже готовий тест-кейс.

- *Усій проектній команді*. Якщо сценарій потрібно узгодити, а є декілька альтернативних варіантів сценарію, допоможе строго описаний потік подій.

- Іншим учасникам сценарію.

2. *В яких випадках потрібні сценарії:*

- Якщо потрібна якісна, повна специфікація вимог, що містить модель даних, опис інтерфейсу, інтеграції з іншими системами.

- Для підтримки системи. Щоб виявити помилку, зрозуміти, на якому етапі щось пішло не так.

- Якщо потрібно описати якусь частину функціональності, роботи користувача з інтерфейсом у вигляді сценарію. Тоді можна використати шаблон юзкейса за основу і далі використовувати його для опису сценарію. Наприклад, основу вимог до мобільного додатка становить опис користувацького інтерфейсу. Але виконання деяких функцій має багато нюансів, які потрібно додатково описати за допомогою таблицьки “дія-відгук системи”, або навіть поєднати таку таблицьку зі сценарієм.

3. *Опис діаграми в прикладах:*

Приклад 1. Запит на отримання розкладу на відповідну групу.

Діючі особи	Студент, система
Ціль	Отримати розклад на відповідну групу
Передумова	Група не вибрана
Успішний сценарій:	1. Студент вибирає групу і відправляє запит 2. Система приймає, обробляє запит і надсилає результат студенту
Результат	Студент отримав результат за своїм запитом

Приклад 2.

Діючі особи	Адміністратор, система
Цілі	Адміністратор: додати/змінити пару у розкладі Система: прийняти зміну і видати результат про успіх
Успішний сценарій:	1. Адміністратор переходить до адмін-панелі. Система пропонує йому додати/ змінити пару. 2. Адміністратор створює/змінює пару. 3. Система перевіряє валідність даних. 4. Система обробляє запит і при успіху додає його. 5. Система видає адміністратору повідомлення про успіх у виконанні дії.
Результат	Адміністратор вдало додав/змінив пару
Розширення:	
*a	Немає доступу до БД. Система видає повідомлення (посилання на повідомлення) Результат: адміністратор не може додати/змінити пару.
1a	В налаштуваннях було допущено помилку і БД недоступна. Результат: опція «додати/змінити» є недоступною доти, поки не буде змінено параметрів налаштування.
2a	Адміністратор робить зміни. Викликається повторний сценарій «Додати/змінити пару»
3a	Адміністратор повторює спробу – і помилка. Результат: відмова в додаванні. Система видає помилку. Переходимо на крок 2.
3б	К-ть спроб дійшла ліміту, налаштованого в налаштуваннях. Результат: система просить адміністратора переглянути налаштування. Видається повідомлення. Вхід блокується до повторного введення паролю і логіну адміністратора

Очевидно, що якщо Приклад 1 можна описати простим текстом, то Приклад 2 набагато краще сприймається у вигляді сценарію. Але якщо вся функціональність описана у вигляді юзкейсів, то краще описувати юзкейсами навіть дуже прості сценарії (кілька кроків), оскільки специфікація повинна бути в єдиному стилі.

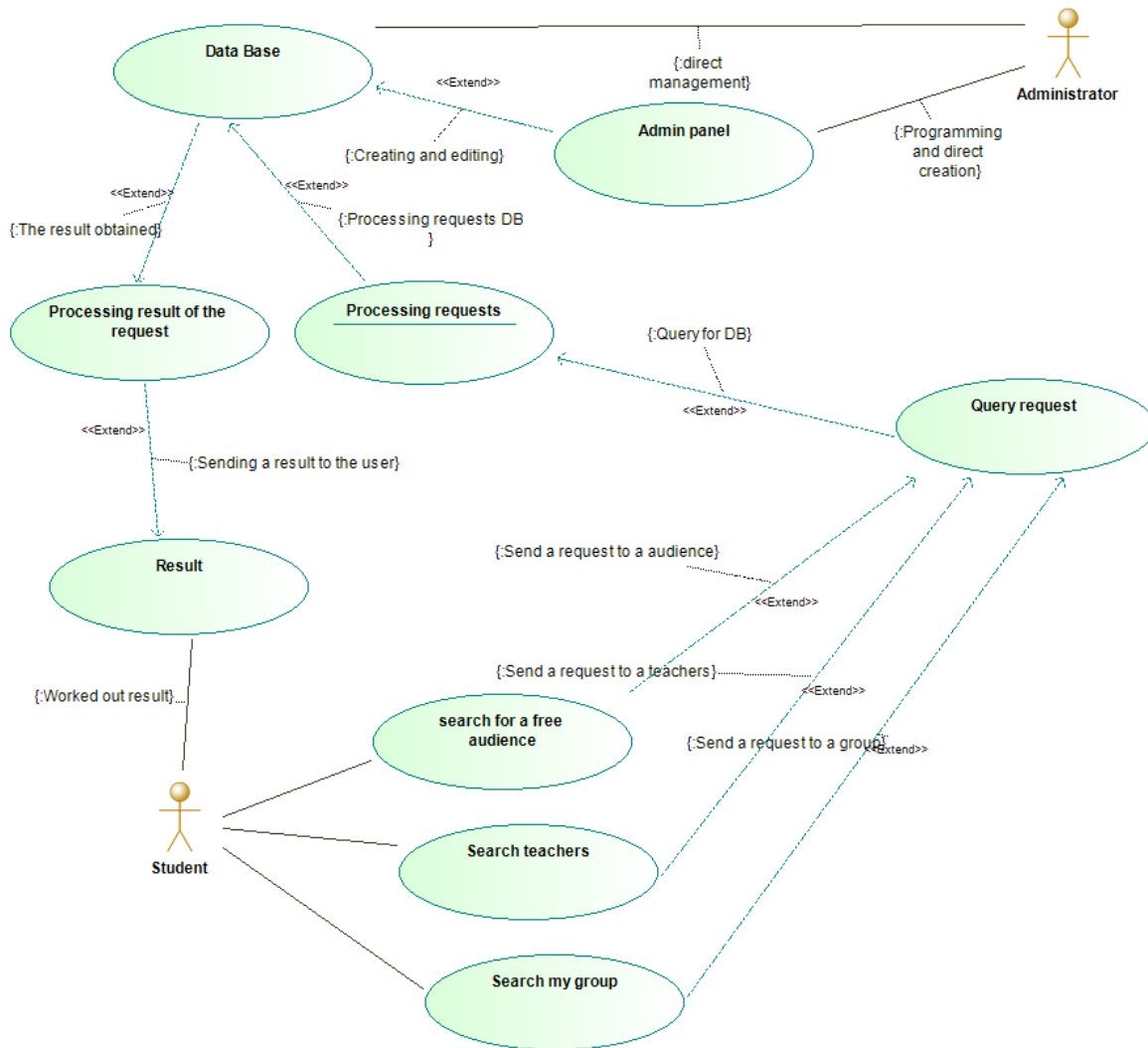


Рис. 1. Графічне представлення UML (Use-Case) Timetable

Слід використовувати мінімальну кількість слів і пунктів, необхідних для однозначного розуміння сценарію. Якщо юзкейс виявляється занадто довгий, можливо, краще буде розбити його на кілька. З дуже довгими сценаріями, з великою кількістю розширень працювати вкрай незручно.

Якщо в двох і більше сценаріях повторюється однаковий набір кроків, є сенс винести ці кроки в окремий сценарій і посилатися на них. Документ буде зручніший. А якщо щось у цих кроках зміниться, то достатньо буде внести зміни в одному місці.

Потрібно перечитати документ зі сценаріями перед тим, як віддавати його розробникам або замовникам. Завжди знаходяться моменти, які можна описати лаконічніше, або виявляються якісь невідповідності чи випадковості.

Приклад моделювання процесу складання розкладу з використанням UML ModelioSoft зображено на рис. 1.

Таким чином, уніфікована мова візуального моделювання UML, яка створена для оптимізації процесу проектування та розробки інформаційних систем, дозволяє збільшити ефективність їх реалізації,

помітно поліпшити якість кінцевого продукту шляхом вирішення великої множини різних прикладних задач.

Висновки і перспективи подальших досліджень у даному напрямі. Уніфікована мова візуального моделювання UML використовується в багатьох успішних проектах інформатизації. Більшість CASE-засобів (програм автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі, розроблені в UML, дозволяють значно спростити процес кодування і пришвидшити реалізацію самої системи.

UML необхідний:

- керівникам проектів, які керують розподілом завдань і контролем за виконання проекту;
- проектувальникам інформаційних систем, які розробляють технічні завдання для програмістів;
- бізнес-аналітикам, які досліджують реальну систему та здійснюють інжиніринг і реінжиніринг бізнес-процесів компанії;
- програмістам, які реалізують модулі інформаційної системи.

Використання UML є ефективним інструментом у проектуванні інформаційних систем підприємств,

банківських і фінансових установ, телекомунікацій, на транспорті, в торгівлі, науці тощо. UML можна використовувати, наприклад, для моделювання документообігу в юридичних фірмах чи урядових структурах, для опису структури та функціонування системи обслуговування пацієнтів у лікарнях, для проектування апаратних засобів тощо.

ЛІТЕРАТУРА

1. Горбенко А. О. Информационные системы в экономике / А. О. Горбенко. – М. : БИНОМ. Лаборатория знаний, 2010. – 292 с.
2. Гудзовата О. О. Методології структурного проектування та моделювання інформаційних систем / О. О. Гудзовата, С. Р. Майнош // Трансформаційні процеси національної економіки: проблеми та перспективи розвитку : збірник наукових праць з актуальних проблем економічних наук : у 2-х частинах. / Наукова організація “Перспектива”. – Дніпропетровськ : Видавничий дім “Тельветика”, 2014. – Ч. 1. – С. 98-110.
3. Дубейковский В. И. Практика функционального моделирования с AllFusion Process Modeler 4.1. / В. И. Дубейковский. – М. : ДИАЛОГ-МИФИ, 2004. – 464 с.
4. Жирякова І. А. Використання CASE-систем при проектуванні інформаційних систем / І. А. Жирякова, Є. Ю. Філіпов [Електронний ресурс]. – Режим доступу: http://www.rusnauka.com/13_NPN_2010/Tecnic/65951.doc.htm.
5. Информационные ресурсы и технологии в экономике / [под ред. А. Н. Романова, Б. Е. Одинцова]. – М. : ИНФРА-М, 2013. – 462 с.
6. Марченко А. В. Проектування інформаційних систем / А. В. Марченко. – К., 2016 [Електронний ресурс]. – Режим доступу: https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/content-20160217112601.pdf.
7. Описание бизнес-процессов [Электронный ресурс]. – Режим доступа: http://www.bsc-consulting.ru/services/business_consulting/business_processes_definition/.
8. Особливості проектування інформаційних систем. CASE-технології [Електронний ресурс]. – Режим доступу: <http://library.nung.edu.ua/osoblivostii-proektuvannya-i-informatsiiniikh-sistem.html>.
9. Проектирование информационных систем [Электронный ресурс]. – Режим доступа: <https://www.intuit.ru/studies/courses/2195/55/info>.

10. Шаховська Н. Б. Проектування інформаційних систем : навч. пос. / Н. Б. Шаховська, В. В. Литвин. – Львів : Магнолія-2006, 2011. – 380 с.

REFERENCES

1. Horbenko, A. O. (2010), Ynformatsyonnye systemy v ekonomyke, BYNOM. Laboratoryia znanyj, M., 292 s.
2. Hudzovata, O. O. and Majnosh, S. R. (2014), Metodolohii struktornoho proektuvannia ta modeliuвання informatsijnykh system, *Transformatsijni protsesy natsional'noi ekonomiky: problemy ta perspektyvy rozvytku* : zbirnyk naukovykh prats' z aktual'nykh problem ekonomichnykh nauk : u 2-kh chastynakh. / Naukova orhanizatsiia “Perspektyva”, Vydavnychij dim “Hel'vetyka”, Dnipropetrovs'k. – Ch. 1. – S. 98-110.
3. Dubejkovskij, V. Y. (2004), Praktyka funktsional'noho modelyrovanyia s AllFusion Process Modeler 4.1., DYALOH-MYFY, M., 464 s.
4. Zhyriakova, I. A. and Filipov, Ye. Yu. Vykorystannia CASE-system pry proektuvanni informatsijnykh system, available at: http://www.rusnauka.com/13_NPN_2010/Tecnic/65951.doc.htm.
5. Ynformatsyonnye resursy y tekhnolohyy v ekonomyke, pod red. A. N. Romanova, B. E. Odyntsova (2013), YNFRA-M, M., 462 s.
6. Marchenko, A. V. Proektuvannia informatsijnykh system. – K., 2016, available at: https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/content20160217112601.pdf.
7. Opysanye byznes-protsessov, available at: http://www.bsc-sonsulting.ru/services/business_consulting/business_processes_definition/.
8. Osoblyvosti proektuvannia informatsijnykh system. CASE-tekhnologii, available at: <http://library.nung.edu.ua/osoblivostii-proektuvannya-i-informatsiiniikh-sistem.html>.
9. Proektyrovanye ynformatsyonnykh system, available at: <https://www.intuit.ru/studies/courses/2195/55/info>.
10. Shakhovs'ka, N. B. and Lytvyn, V. V. (2011), Proektuvannia informatsijnykh system : navch. pos., Mahnoliia-2006, L'viv, 380 s.

Стаття надійшла до редакції 3 грудня 2018 р.