

Міністерство освіти і науки України
Державний вищий навчальний заклад
“Криворізький національний університет”

МЕТОДИЧНІ ВКАЗІВКИ

для самостійної роботи

з дисципліни

„Інженерія програмного забезпечення”

Для студентів денної та заочної форм навчання
Спеціальність 123 «Комп’ютерна інженерія»

Факультет інформаційних технологій

Кафедра комп’ютерні системи і мережі

Кривий Ріг
2018

Методичні вказівки до самостійної роботи студентів денної та заочної форм навчання з дисципліни „ Інженерія програмного забезпечення ”.

Укладач: доц.,к.т.н. Вдовиченко І.Н.
Комп’ютерний набір: доц., к.т.н. Вдовиченко І.Н.

Схвалено на засіданні кафедри комп’ютерні системи і мережі
(Протокол № 7 від 20.02.2018 р.)

завкафедрою _____ д.т.н. А.І. Купін
підпис

Методичні вказівки розглянуто та схвалено вченою радою факультету інформаційних технологій
(Протокол №7 від 21.02.2018 р.)

Голова вченої ради ФІТ _____ В.А. Чубаров
підпис

Начальник навчально-методичного відділу _____ Г.Х.Отверченко
підпис

Методичні вказівки рекомендовано для самостійної роботи з дисципліни „ Інженерія програмного забезпечення ”.

Наклад примірників.

Зміст

Вступ.....	4
Постановка завдання.....	5
Аналіз предметної області.....	5
Опис середовища IBM Rational Rose.....	6
Головне меню і стандартна панель інструментів.....	7
Додавання класу на діаграму класів і редагування його властивостей....	10
Додавання і редагування атрибутів класів.....	12
Додавання асоціації на діаграму класів і редагування її властивостей....	13
Формулювання проектування.....	14
Розробка діаграми прецедентів.....	16
Перелік контрольних питань.....	17
Розробка діаграми класів.....	20
Перелік контрольних питань.....	20
Розробка діаграми послідовності.....	22
Перелік контрольних питань.....	24
Створення кооперативної діаграми.....	24
Перелік контрольних питань.....	26
Діаграма діяльності.....	26
Перелік контрольних питань.....	29
Створення діаграми компонентів.....	29
Перелік контрольних питань.....	30
Створення діаграми розміщення.....	30
Перелік контрольних питань.....	31
Список використаної літератури.....	31

Вступ

В даний час розробники програмного та інформаційного забезпечення інформаційних систем все частіше використовують об'єктно-орієнтований підхід, з використанням середовища IBM Rational Rose.

Запропонована об'єктно-орієнтована модель програмного забезпечення буде розроблена з використанням мови моделювання UML в середовищі IBM Rational Rose.

Уніфікована мова моделювання UML (Unified Modeling Language) є мовою для визначення, представлення, проектування і документування програмних систем, інформаційних систем, організаційно-економічних систем, технічних систем та інших систем різної природи.

У CASE-засобі IBM Rational Rose реалізовані загальноприйняті стандарти на робочий інтерфейс програми, аналогічно відомим в середовищах візуального програмування.

Rational Rose має досить простий і інтуїтивно зрозумілий інтерфейс користувача, що дозволяє створювати моделі складних програмних продуктів за допомогою уніфікованої мови моделювання UML.

Моделювання в середовищі IBM Rational Rose

Тема: Використання CASE-засобу IBM Rational Rose для побудови моделі інформаційної системи.

Мета: аналіз та розробка об'єктно-орієнтованої моделі програмного забезпечення автоматизованого робочого місця працівника, а також здобуття навичок роботи з пакетом програмних продуктів Rational Rose.

Постановка завдання

В якості предметної області використовується склад товарів.

Склад - приміщення (також їх комплекс), призначене для зберігання матеріальних цінностей і надання складських послуг.

Сукупність складських технологічних операцій:

- Організація розвантаження та переміщення товарів (вантажів) до місця приймання.
- Організація приймання.
- Організація розміщення та укладання.
- Організація відпуску товарів.
- Інвентаризація товаро - матеріальних цінностей.

Документи:

- Журнал обліку товарів на складі.
- Товарно-транспортні накладні.
- Товарний ярлик.
- Картка маршруту.

Актуальність розробки об'єктно-орієнтованої моделі інформаційної підсистеми для обліку руху товарів на складі обумовлюється автоматизацією основних функцій, які виконують завідувач складом, в обов'язки якого входить проведення ревізій і складання оборотів за період, і комірник, який здійснює інвентаризацію і прийом/видачу товару.

В результаті проектування має бути отримана об'єктно-орієнтована модель інформаційної підсистеми для обліку руху товарів на складі.

Аналіз предметної області

На складах здійснюється цілий комплекс різноманітних послідовно виконуваних операцій по вступу, зберігання і відпустці товарів. Ці операції складають складський технологічний процес.

Види технологічних операцій і їх зміст :

- *організація розвантаження та переміщення товарів (вантажів) до місця приймання.*

– **організація приймання.** Разом з товарами / вантажами склад отримує супровідні документи: товарно-транспортні накладні, рахунки-фактури, інвойси тощо.

– **організація розміщення та укладання.** Для раціонального розміщення товарів складають економічно і технологічно обґрунтовані схеми розміщення.

– **Організація відпуску товарів.** Відпустка товарів зі складу - заключна стадія складського технологічного процесу. Вона включає в себе наступні операції:

- Відбірки товарів з місць зберігання (комплектація);
- Підготовка до відпуски (упаковка, окантовка, маркування тощо);
- Оформлення відпуску;
- Відправлення за призначенням (доставка), або відвантаження.

Всіма операціями по руху товарів на складі керує завідувач складом.

Опис середовища IBM Rational Rose

Робочий інтерфейс програми IBM Rational Rose складається з різних елементів, основними з яких є:

- Головне меню;
- стандартна панель інструментів;
- спеціальна панель інструментів;
- вікно браузера проекту;
- робоча область зображення діаграми або вікно діаграми;
- вікно документації;
- вікно журналу.

Розглянемо призначення і основні функції кожного з цих елементів.

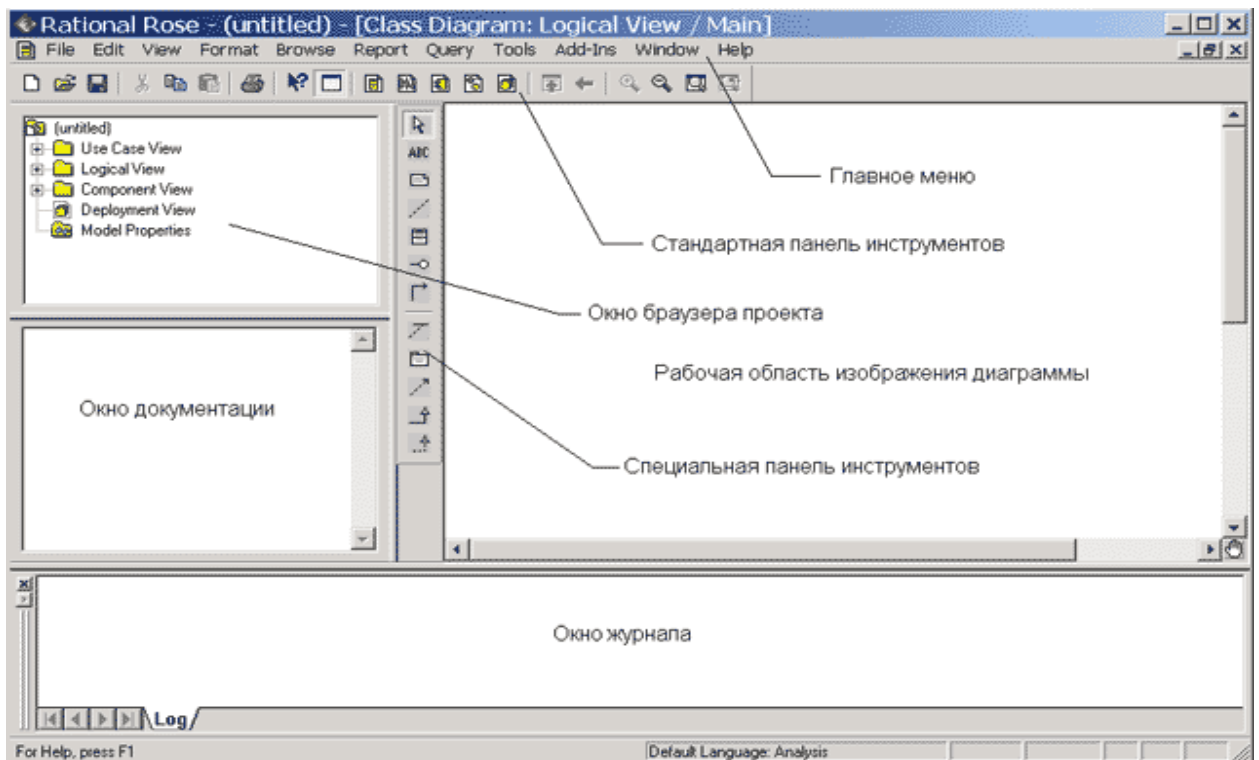


Рис. 1 Загальний вигляд робочого інтерфейсу CASE-засобу IBM Rational Rose

Головне меню і стандартна панель інструментів

Головне меню програми IBM Rational Rose виконано в загальноприйнятому стандарті і має наступний вигляд (рис. 2).

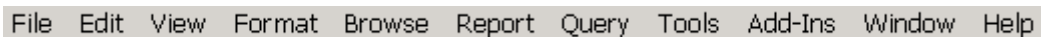


Рис.2. Зовнішній вигляд головного меню програми

Окремі пункти меню об'єднують подібні операції, відносяться до всього проекту в цілому. Деякі з пунктів меню містять добре знайомі операції, такі як відкриття проекту, висновок на друк діаграм, копіювання в буфер і вставка з буфера різних елементів діаграм. Інші операції настільки специфічні, що можуть знадобитися додаткові зусилля для їх вивчення (властивості операцій генерації програмного коду або перевірки узгодженості моделей).

Стандартна панель інструментів розташовується нижче рядка головного меню і має такий вигляд (рис. 3). Деякі з інструментів недоступні для нового проекту, який не має ніяких елементів. Стандартна панель інструментів забезпечує швидкий доступ до тих команд меню, які виконуються розробниками найбільш часто.

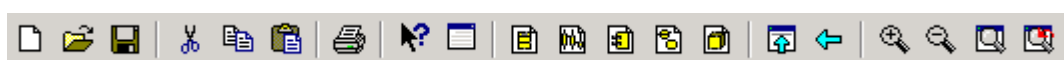


Рис.3. Зовнішній вигляд стандартної панелі інструментів

Користувач може налаштувати зовнішній вигляд цієї панелі на свій розсуд. Для цього необхідно виконати операцію головного меню:

Tools> Options (Інструменти>; Параметри), відкрити вкладку Toolbars (Панелі інструментів) діалогового вікна і натиснути кнопку Standard (Стандартна). У додатково відкритому вікні можна переносити необхідні кнопки з лівого списку в правий список, а непотрібні кнопки - з правого списку в лівий. Даним способом можна показати або приховати різні кнопки інструментів, а також змінити їх розмір.




Призначення операцій головного меню File і Edit

Робочий інтерфейс IBM Rational Rose використовує головне меню, яке дозволяє користувачеві завантажувати і зберігати інформацію в зовнішніх файлах, змінювати зовнішній вигляд елементів графічного інтерфейсу, викликати довідкову інформацію, викликати інші діалогові вікна для роботи з програмою IBM Rational Rose і т.д. Розглянемо призначення окремих пунктів головного меню засобу IBM Rational Rose.


Операції головного меню File (Файл) дозволяють створювати нові моделі в нотації мови UML, завантажувати і зберігати розробляється модель в зовнішньому файлі, роздруковувати на принтері розроблені діаграми. Призначення операцій цього пункту головного меню представлено в таблиці 1.

Операції головного меню Edit (Редагування) дозволяють виконувати дії по редагуванню елементів моделі і їх властивостей, а також виконувати пошук елементів в рамках проекту, що розробляється.

Таблиця 1. Операції пункту головного меню File (Файл)

Назва операції меню	Наявність кнопки на стандартній панелі	Призначення операції головного меню
New		Створює нову модель IBM Rational Rose. При цьому нова модель за замовчуванням має ім'я untitled
Open		Викликає стандартне діалогове вікно відкриття зовнішнього файлу з диска. Відкрити можна або файл моделі (файл з розширенням ".mdl"), або файл підмоделі (файл з розширенням ".ptl")
Save		Дозволяє зберегти розробляється модель в зовнішньому файлі на диску
Save As		Дозволяє зберегти модель, що розробляється під іншим ім'ям в зовнішньому файлі на диску. При цьому викликається стандартне діалогове вікно збереження файлу на диску з пропозицією поставити ім'я відповідного файлу моделі або

		подмоделі.
Save Log As		Дозволяє зберегти зміст журналу в зовнішньому файлі на диску з ім'ям error.log. При цьому викликається стандартне діалогове вікно збереження файлу на диску з пропозицією змінити пропонуване за замовчуванням ім'я відповідного файлу.
AutoSave Log		Дозволяє автоматично зберігати зміст журналу в зовнішньому файлі на диску з ім'ям error.log. При першому виконанні цього пункту меню також викликається стандартне діалогове вікно збереження файлу на диску з пропозицією змінити пропонуване за замовчуванням ім'я відповідного файлу.
Clear Log		Очищає зміст журналу
Load Model Workspace		Дозволяє завантажити робочу область з зовнішнього файлу на диску. Викликає стандартне діалогове вікно відкриття зовнішнього файлу з диска, при цьому відкрити можна файл з розширенням "wsp"
Save Model Workspace		Дозволяє зберегти робочу область моделі в зовнішньому файлі на диску. При виконанні цього пункту меню викликається стандартне діалогове вікно збереження файлу з розширенням "wsp"
Save Model Workspace As		Дозволяє зберегти робочу область моделі в зовнішньому файлі на диску. Викликається стандартне діалогове вікно збереження файлу з пропозицією змінити пропонуване за замовчуванням ім'я відповідного файлу
Units		Дозволяє завантажити категорію елементів моделі з зовнішнього файлу на диску. Викликає стандартне діалогове вікно відкриття зовнішнього файлу з диска, при цьому відкрити можна файл з розширенням "cat"
Import		Зберігає інформацію з зовнішніх файлів різних форматів, включаючи файли моделей, підмоделей, категорій і підсистем
Export Model		Дозволяє експортувати інформацію про модель в зовнішній файл. Вид цього пункту меню залежить від виділеного елемента моделі

Update		Дозволяє вставити інформацію зворотного проектування з зовнішнього файлу з розширенням "ged" в розроблювану модель
Print		Дозволяє роздрукувати на принтері окремі діаграми і специфікації різних елементів моделі, що розробляється. В цьому випадку викликається діалогове вікно вибору діаграм і специфікацій для друку на підключеному до комп'ютера принтера
Print Setup		Викликається стандартне діалогове вікно макета сторінки для настройки властивостей друку
Edit Path Map		Викликає вікно завдання шляхів доступу до файлів системи IBM Rational Rose. Як правило, значення шляхів, встановлені за замовчуванням, слід змінювати тільки в разі крайньої необхідності
<Имена файлов>		Секція з іменами останніх файлів, з якими здійснювалася робота в IBM Rational Rose
Exit		Припиняє роботу і закриває IBM Rational Rose

При створенні нового проекту зазначена ієрархічна структура формується програмою автоматично.

Спеціальна панель інструментів і вікно діаграми

Спеціальна панель інструментів розташовується між вікном браузера і вікном діаграми в середній частині робочого інтерфейсу. За замовчуванням пропонується панель інструментів для побудови діаграми класів моделі (рис. 4).



Рис. 4. Зовнішній вигляд спеціальної панелі інструментів для діаграми класів













Діаграма класів є основним логічним поданням моделі і містить детальну інформацію про внутрішній устрій об'єктно-орієнтованої програмної системи або, висловлюючись сучасною термінологією, про архітектуру програмної системи.

Активізувати робоче вікно діаграми класів можна декількома способами:

- ✓ вікно діаграми класів з'являється за замовчуванням в робочому вікні діаграми після створення нового проекту;
- ✓ клацнути на кнопці із зображенням діаграми класів на стандартній панелі інструментів;
- ✓ розкрити логічне уявлення (Logical View) в браузері проекту і двічі клацнути на піктограмі Main (Головна);
- ✓ виконати операцію головного меню: Browse> Class Diagram (Огляд> Діаграма класів).

При цьому з'являється нове вікно з чистим робочим листом діаграми класів і спеціальна панель інструментів, що містить кнопки із зображенням графічних примітивів, необхідних для розробки діаграми класів (табл. 2). Призначення окремих кнопок панелі можна дізнатися також з спливаючих підказок.

Таблиця 2. Призначення кнопок спеціальної панелі інструментів для діаграми класів

Графічне зображення	Спливаюча підказка	Призначення
	Selection Tool	Перетворює зображення курсору в форму стрілки для подальшого виділення елементів на діаграмі
	Text Box	Додає на діаграму текстову область
	Note	Додає на діаграму примітки
	Anchor Note to Item	Додає на діаграму зв'язок примітки з відповідним графічним елементом діаграми
	Class	Додає на діаграму клас
	Interface	Додає на діаграму інтерфейс
	Unidirectional Association	Додає на діаграму спрямовану асоціацію
	Association Class	Додає на діаграму асоціацію клас
	Package	Додає на діаграму пакет
	Dependency or Instantiates	Додає на діаграму відношення залежності
	Generalization	Додає на діаграму відношення узагальнення
	Realize	Додає на діаграму відношення реалізації

На спеціальній панелі інструментів за замовчуванням відображена тільки частина піктограм елементів, які можуть бути використані для побудови діаграми класів. Додати кнопки з піктограмами інших графічних елементів таких як, наприклад, відносини агрегації і композиції, шаблон, клас бізнес-сутність, керуючий клас, або видалити непотрібні кнопки можна за допомогою настройки спеціальної панелі інструментів. Відповідне діалогове вікно настройки спеціальної панелі інструментів для діаграми класів можна викликати аналогічно іншим панелям за допомогою операції контекстного меню Customize (Налаштування) при позиціонуванні курсора на спеціальній панелі інструментів.

Додавання класу на діаграму класів і редагування його властивостей

Для додавання класу на діаграму класів потрібно за допомогою лівої кнопки миші натиснути кнопку із зображенням піктограми класу на спеціальній панелі інструментів, відпустити ліву кнопку миші і клацнути лівою кнопкою миші на вільному місці робочого листа діаграми.

На діаграмі з'явиться зображення класу з маркерами зміни його геометричних розмірів і запропонованим середовищем ім'ям за умовчанням NewClass.

Для класу можна уточнити його призначення в моделі за допомогою вказівки стереотипу і пояснювального тексту у вигляді документації. З цією метою подвійним клацанням лівою кнопкою миші на зображенні цього класу на діаграмі або в браузері проекту слід відкрити діалогове вікно специфікації властивостей цього класу і на вкладці General (Загальні) вибрати з вкладеного списку Stereotype стереотип entity (сутність).

Для окремого класу можна уточнити також і інші його властивості, доступні для редагування на вкладці Detail (Детально) вікна специфікації властивостей цього класу.

Наприклад, на цій вкладці за допомогою вкладеного списку Multiplicity (Кратність) можна задати кількість об'єктів або екземплярів даного класу, для чого слід вибрати рядок з буквою n. Дане значення означає, що у класі може бути будь-яке кінцеве число примірників. Поле введення з ім'ям Space (Простір) служить для вказівки обсягу абсолютної або відносної пам'яті, яка потрібна, за оцінкою розробника, для реалізації кожного об'єкта даного класу. Що стосується аналізованої моделі це поле можна залишити порожнім.

Додавання і редагування атрибутів класів

З усіх графічних елементів середовища IBM Rational Rose клас володіє максимальним набором властивостей, головними з яких є його атрибути і операції. Оскільки саме діаграма класів використовується в середовищі IBM

Rational Rose, для генерації програмного коду, будуть детально розглянуті відповідні властивості атрибутів і операцій.

За допомогою операції контекстного меню: New> Attribute (Новий> Атрибут) для класу, виділеного в браузері проекту. В цьому випадку активізується курсор введення тексту в області ієрархічного уявлення класу в браузері проекту під ім'ям відповідного класу.

За допомогою операції контекстного меню Insert (Вставити), викликаного при позиціонуванні курсору в області відкритої вкладки атрибутів в діалоговому вікні властивостей Class Specification відповідного класу виконуємо дію.





Після додавання атрибута до класу за замовчуванням йому присвоюється ім'я name і деякий квантор видимості.

Додати атрибут до створеного раніше класу можна одним з таких способів:

За допомогою операції контекстного меню New Attribute (Новий атрибут) для класу, виділеного на діаграмі класів. В цьому випадку активізується курсор введення тексту в області графічного зображення класу на діаграмі.

Імена атрибутів і операцій класів повинні починатися з малої літери. Видимість атрибутів на діаграмі класів зображується у формі спеціальних піктограм або прикрас. Використовувані піктограми видимості зображуються перед ім'ям відповідного атрибута і мають наступний сенс (табл. 3).

Таблиця 3. Піктограми видимості атрибутів класів

Графічне зображення	Текстовий аналог	Призначення піктограми
	Public	Загальнодоступний або відкритий. В нотатції мови UML такому атрибуту відповідає знак "+"
	Protected	Захищений. В нотатції мови UML такому атрибуту відповідає знак "#"
	Private	Закритий. В нотатції мови UML такому атрибуту відповідає знак "-"
	Implementation	Реалізація. В нотатції мови UML такому атрибуту відповідає знак " <i>sim</i> "

Для окремого атрибута можна також визначити додаткові властивості, доступні для редагування на вкладці Detail (Детально) діалогового вікна специфікації властивостей обраного атрибута.

На вкладці Detail в групі вибору Containment (Локалізація) можна специфікувати умови зберігання атрибута у об'єктів обраного класу.

Для вибору можуть бути використані наступні властивості:

- ✓ By value (За значенням) - властивість за замовчуванням, яке означає, що значення атрибута зберігаються в межах адресного простору, виділеного для об'єкту даного класу. Наприклад, якщо є атрибут типу String, то значення цього рядка міститься в межах визначення класу.
- ✓ By reference (За посиланням) - означає, що значення атрибута зберігається поза адресного простору, виділеного для об'єкту даного класу, але у об'єктів класу є покажчик на цей атрибут.
- ✓ Unspecified (Не визначено) - означає, що метод локалізації даного атрибута не визначено. В цьому випадку при генерації програмного коду для даного атрибута за замовчуванням вибирається значення By value.

Далі можна визначити атрибут як статичний, виставивши позначку в рядку вибору Static. Статичний атрибут по визначенню має одне і теж значення для всіх об'єктів даного класу. Нарешті, на вкладці Detail можна визначити атрибут як похідний, виставивши позначку в рядку вибору Derived. Значення похідного атрибута за визначенням може бути обчислено на підставі значень інших атрибутів цього або іншого класу.

Діаграма класів є логічним поданням структури моделі, тому вона повинна містити стільки класів, скільки необхідно для реалізації всього проекту. При цьому для повного уявлення структури моделі необхідно встановити і уточнити відносини між класами.

Додавання асоціації на діаграму класів і редагування її властивостей

Додавання на діаграму асоціацію між двома класами виконується наступним чином.

На спеціальній панелі інструментів необхідно натиснути кнопку із зображенням піктограми спрямованої асоціації і відпустити ліву кнопку миші. Якщо асоціація - спрямована, то на діаграмі класів треба виділити перший елемент асоціації або джерело, від якого виходить стрілка, і, не відпускаючи натиснуту ліву кнопку миші, перемістити її покажчик до другого елементу відносини або приймача, до якого спрямована стрілка. Після переміщення до другого елементу кнопку миші слід відпустити, в результаті чого на діаграму класів буде додана спрямована асоціація з ім'ям Untitled між двома вибраними класами.

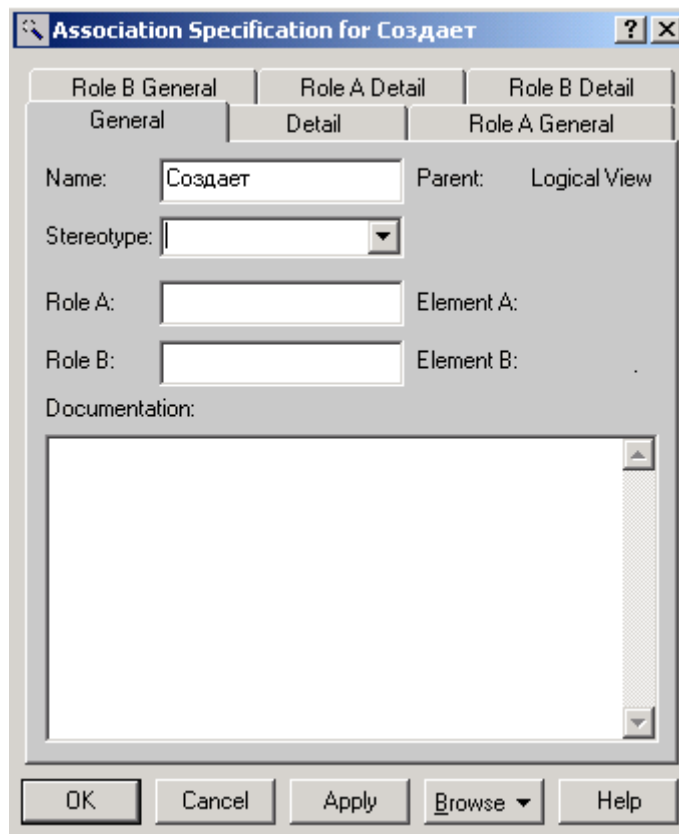


Рис. 5. Діалогове вікно специфікації властивостей асоціації

Для завдання імені асоціації слід на вкладці General (Загальні) в поле введення Name (Ім'я) ввести текст її імені: Створює і натиснути кнопку Apply або OK, щоб зберегти результати редагування імені асоціації. Для асоціації можна задати також кратність кожного з кінців асоціації, стереотип, використовувати обмеження і ролі, а також деякі інші властивості.

Для доданої на діаграму класів асоціації задамо кратність кінця асоціації у класу, рівну 1. Для цього слід у вікні специфікації властивостей асоціації перейти на вкладку Role B Detail і вибрати значення 1 з вкладеного списку Multiplicity. Аналогічним чином слід задати кратність кінця асоціації у другому класі рівну 1..n, для чого на вкладці Role A Detail і слід вибрати значення 1..n з вкладеного списку Multiplicity. Змістовно це буде означати, що кожен об'єкт класу може бути пов'язаний з одним або декількома об'єктами другого класу.

Слід зауважити, що при зображенні діаграми класів всі класи представлені в формі графічних стереотипів, при цьому обраний спосіб відображення сигнатури операцій класів. Для більш компактного представлення діаграми можна прибрати відображення атрибутів, операцій або сигнатури операцій окремих класів за допомогою відповідних операцій контекстного меню Options (Налаштування).

Формулювання проектування

Завданням є розробка об'єктно-орієнтованої моделі інформаційної підсистеми для обліку руху товарів на складі фірми з використанням мови UML.

В результаті проектування має бути отримана об'єктно-орієнтована модель інформаційної підсистеми для обліку руху товарів на складі. Модель включатиме:

- діаграму варіантів використання;
- діаграму послідовності;
- діаграму співпраці;
- діаграму класів;
- діаграму активності;
- діаграму компонентів;
- діаграму розміщення.

Розробка діаграми прецедентів

Поведінка системи (тобто функціональність, що забезпечується системою), що розробляється, описується за допомогою функціональної моделі, яка відображає системні прецеденти (use case), системне оточення (дійових осіб або акторів - actors) і зв'язки між прецедентами і акторами (діаграми прецедентів - use case diagrams).

Діаграми прецедентів при описі і аналізі предметної області дають можливість виділити бізнес-процеси, не вдаючись до деталей їх реалізації. Крім того, робиться попереднє виділення об'єктів предметної області і їх класифікація. На підставі побудованої моделі складається план розробки системи.

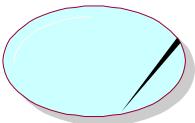
Елементи діаграми :

- варіант використання — цей логічний опис певної частини діяльності системи. Він не є чіткою конструкцією, яку можна безпосередньо реалізувати в програмному коді. Кожен варіант використання визначає послідовність дій, які мають бути виконані проєктованою системою при взаємодії її з відповідним актором.

- актор є будь-якою зовнішньою по відношенню до модельованої системи суттю, яка взаємодіє з системою і використовує її функціональні можливості. Актори використовуються для моделювання зовнішніх по відношенню до проєктованої системи сутностей, які взаємодіють з системою і використовують її як окремих користувачів.

- стосунки асоціації встановлюють, яку конкретну роль грає актор при взаємодії з екземпляром варіанту використання.

Таблиця 5. Опис елементів

Графічна нотація	Назва	Визначення
1	2	3
	"стереотип" Business Use Case	Business Use Case (бізнес-прецедент) визначає безліч екземплярів бізнес-прецедентів, в якому кожен екземпляр є послідовністю виконуваних бізнес-дій, які призводять до спостережуваного конкретним бізнес-актором результату. Клас бізнес-прецеденту містить усі основні, альтернативні потоки робіт, що відносяться до виробництва "спостережуваного результату". Бізнес-прецедент описує бізнес-процес із зовнішньої, додаткової точки зору. Бізнес-прецеденти є бізнес-процесами, які переходять межі організації, включаючи, можливо, партнерів і

		постачальників, для надання значення акціонерів.
	"стереотип" Business Actor	Визначає набір екземплярів бізнес-актора (хтось або щось поза бізнесом, що взаємодіє з бізнесом), в якому кожен екземпляр бізнес-актора грає однакову роль по відношенню до бізнесу. Важливо, що бізнес-актор представляє деякого учасника поза зоною дії бізнесу і, отже, має представлення тільки зовнішньої видимої поведінки бізнесу.
	Бізнес-актор (business actor)	індивідуум, група, організація, компанія або система, які взаємодіють з модельованою бізнес-системою, але не входять в неї, тобто не є частиною модельованої системи.
	асоціації (association relationship)	служить для позначення специфічної ролі актора при його взаємодії з окремим варіантом використання. Іншими словами, асоціація специфікує семантичні особливості взаємодії акторів і варіантів використання в графічній моделі системи.
	розширення (extend)	визначає взаємозв'язок базового варіанту використання з іншим варіантом використання, функціональна поведінка якого задіється базовою не завжди, а тільки при виконанні додаткових умов.
	включення include	визначає взаємозв'язок базового варіанту використання з іншим варіантом використання, функціональна поведінка якого задіється базовою не завжди, а тільки при виконанні додаткових умов.

Основне завдання діаграми прецедентів - бути єдиним засобом, що дає можливість замовникові, кінцевому користувачеві і розробникові спільно обговорювати функціональність і поведінку системи.

Діаграма прецедентів (use case diagram) - це графічне представлення усіх або частини акторів, прецедентів і їх взаємодій в системі.

Суб'єктом в даному випадку виступає: комірник.

Прецедентами є:
відпускати товар,
приймати товар,

складувати товар,
оформити витратну накладну,
зробити запис в журналі руху товарів,
оформити товарний ярлик,
оформити маршрутний лист.

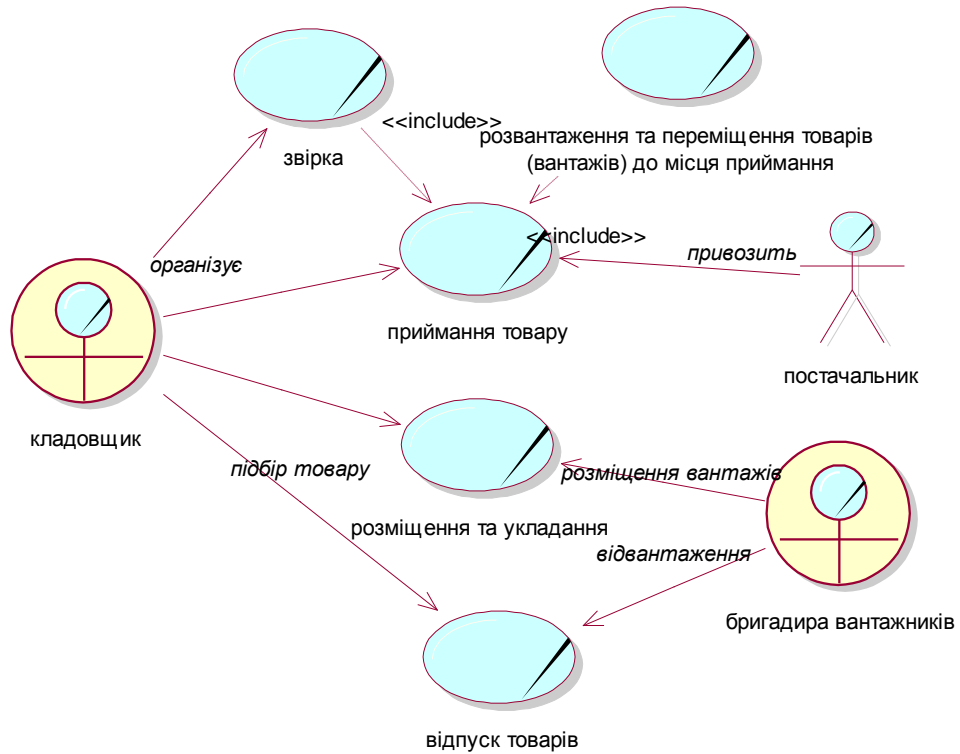


Рис. 5 Діаграма прецедентів «Бізнес процесу предметної області»

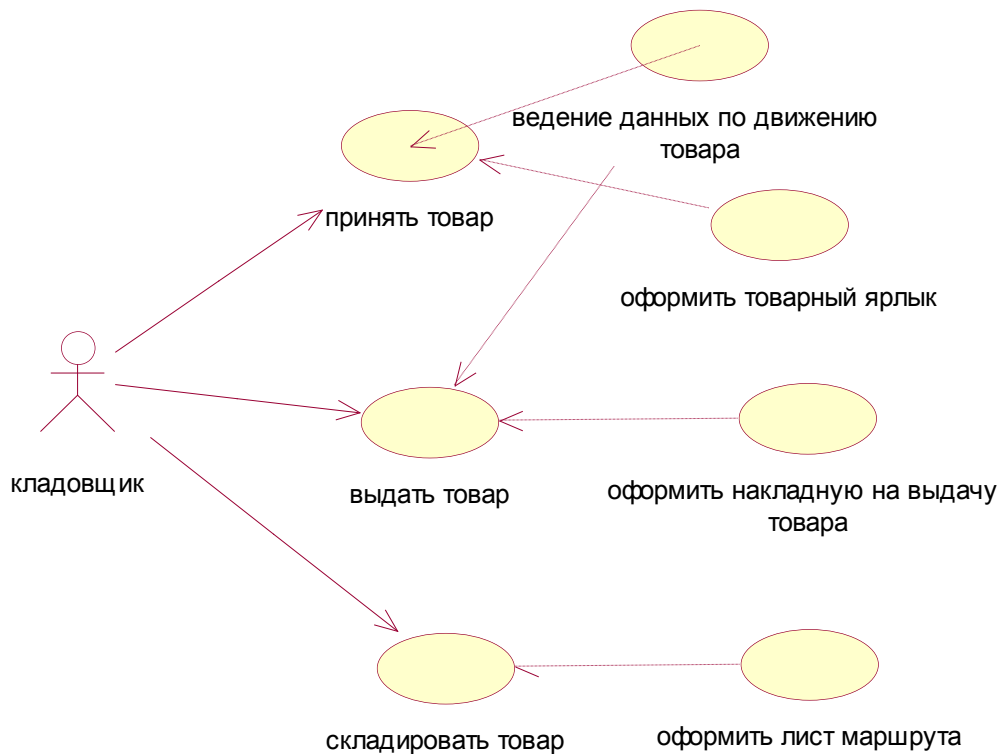


Рис. 6 Діаграма прецедентів програмного засобу

Перелік контрольних питань

1. Що таке нефункціональні вимоги? Як вони відображаються на діаграмах прецедентів?
2. Які способи зображення акторів ви знаєте?
3. В які відносини можуть вступати актори між собою?
4. У чому полягає сенс відносин включення та розширення?
5. Що таке точка розширення?
6. Перерахуйте відомі вам причини використання прецедентів.
7. Як прецеденти застосовують в прямому і зворотному проектуванні?

Розробка діаграми класів

Діаграма класів (class diagram) дозволяє створювати логічне представлення системи, на основі якого створюється початковий код описаних класів. Значки діаграми дозволяють відображати складну ієрархію систем, взаємозв'язки класів (classes) і інтерфейсів (interfaces).

На діаграмах класів зображуються також атрибути класів, операції класів і обмеження, які накладаються на зв'язки між класами. Цей тип

діаграм протилежний за змістом діаграмі співпраці, на якій відображаються об'єкти системи.

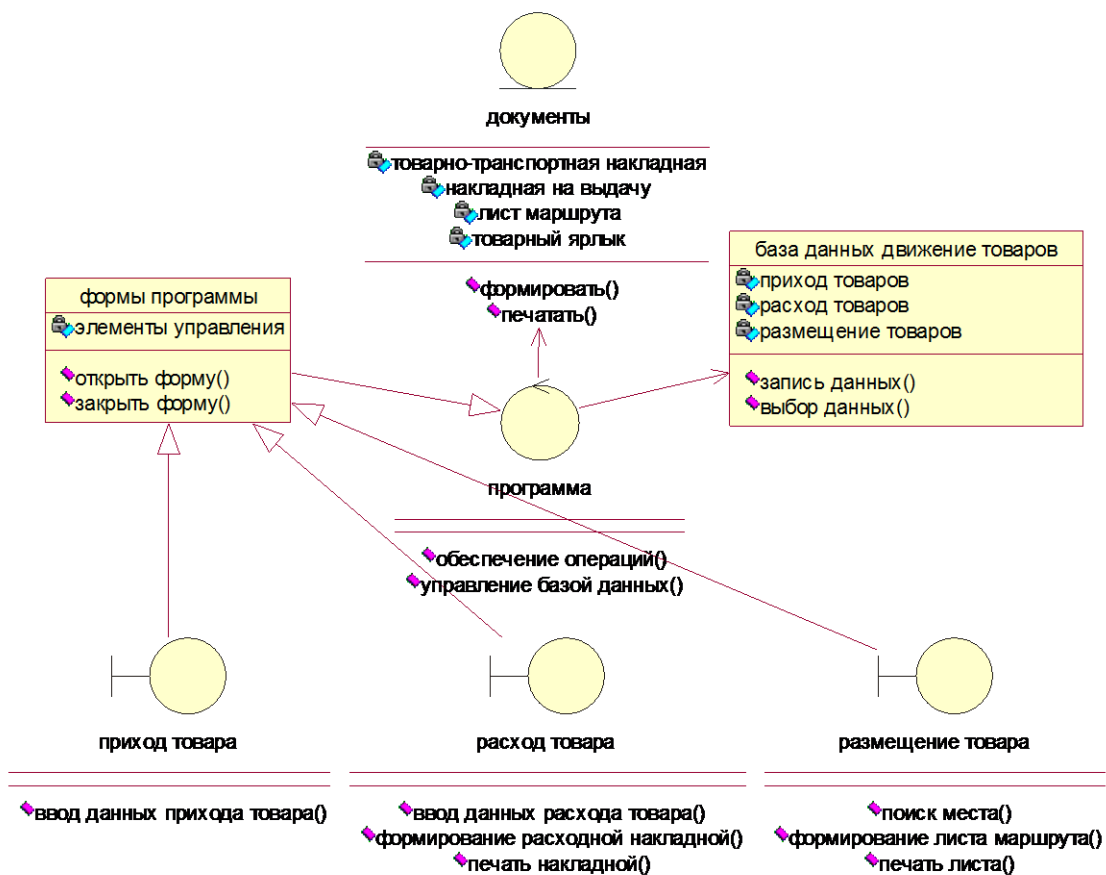
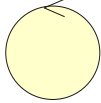


Рис. 7. Діаграма класів програмного засобу

Таблиця 6. Опис компонент діаграми класів

Графічна нотація	Назва	Визначення
1	2	3
	клас-суть	Класом (Class) називається іменований опис сукупності об'єктів із загальними атрибутами, операціями, зв'язками і семантикою.
	Boundary	клас, який розташовується на межі системи із зовнішнім середовищем і безпосередньо взаємодіє з акторами, але є складовою частиною системи.

Графічна нотація	Назва	Визначення
1	2	3
	Control	<p>клас, що відповідає за координацію дій інших класів. На кожній діаграмі класів має бути хоч би один клас, що управляє, причому кількість відправлених об'єктам класу повідомлень, що управляє, мало, в порівнянні з числом тих, що розсилаються ними. Клас, що управляє, відповідає за координацію дій інших класів. У кожній діаграмі класів має бути хоч би один клас, що управляє, контролюючий послідовність виконання дій цього варіанту використання. Як правило, цей клас являється акт</p>

Перелік контрольних питань

1. Значення діаграми класів
2. Графічні символи діаграми класів
3. Атрибути та методи
4. Типи зв'язків між класами

Розробка діаграми послідовності

Для моделювання взаємодії об'єктів в мові UML використовуються діаграми послідовності. Діаграма послідовності відбиває потік подій, що відбуваються у рамках варіанту використання.

На діаграмі послідовності зображуються тільки ті об'єкти, які безпосередньо беруть участь у взаємодії. Ключовим моментом для діаграм послідовності є динаміка взаємодії об'єктів в часі.

На діаграмі послідовності об'єкт зображується у вигляді прямокутника, від якого вниз проведена пунктирна вертикальна лінія. Ця лінія називається лінією життя об'єкту. Вона є фрагментом життєвого циклу об'єкту в

процесі взаємодії. Кожне повідомлення представляється у вигляді стрілки між лініями життя двох об'єктів і позначається ім'ям.

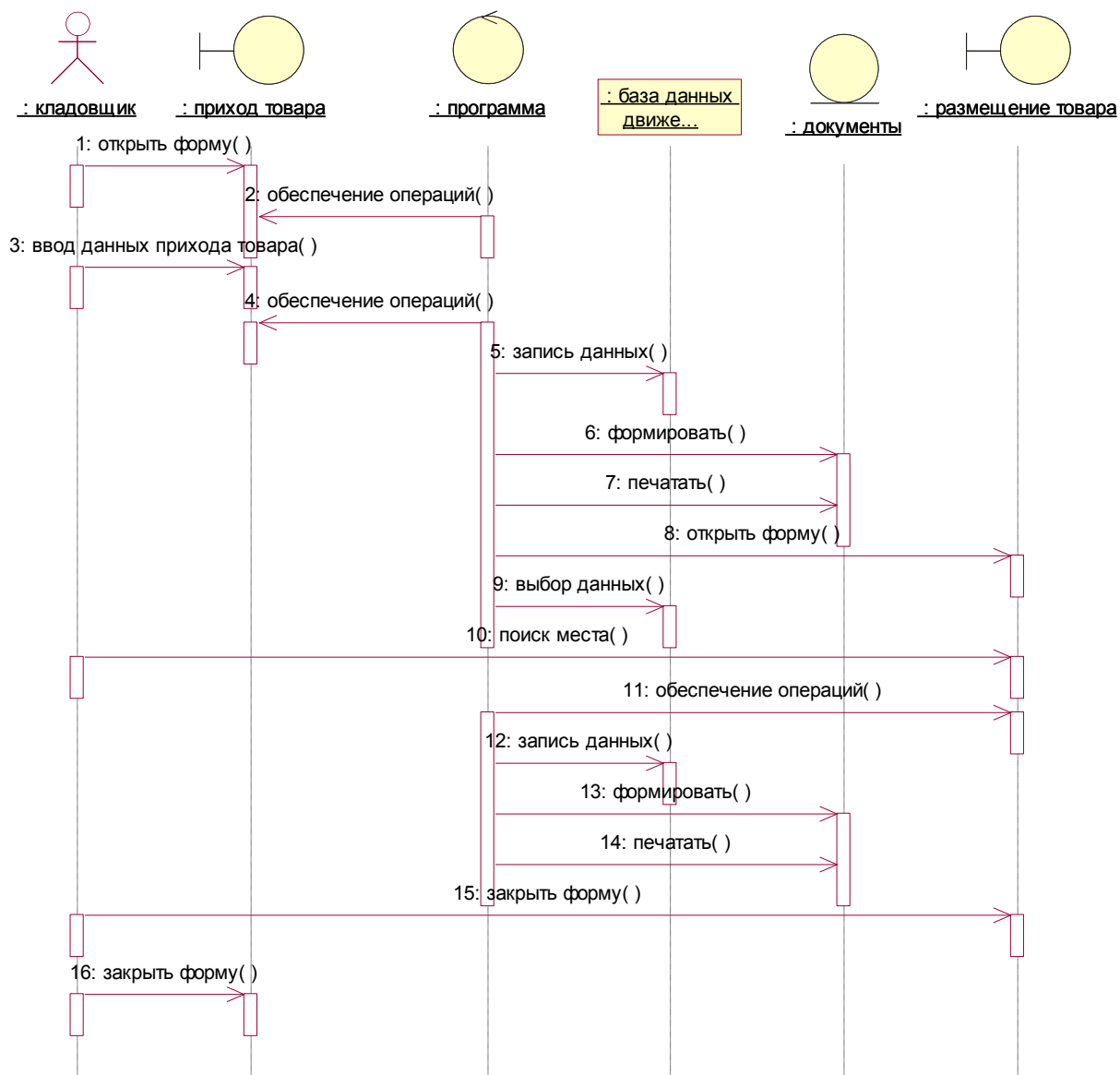


Рис. 8 Діаграма послідовності прецеденту «Прибуття товару»

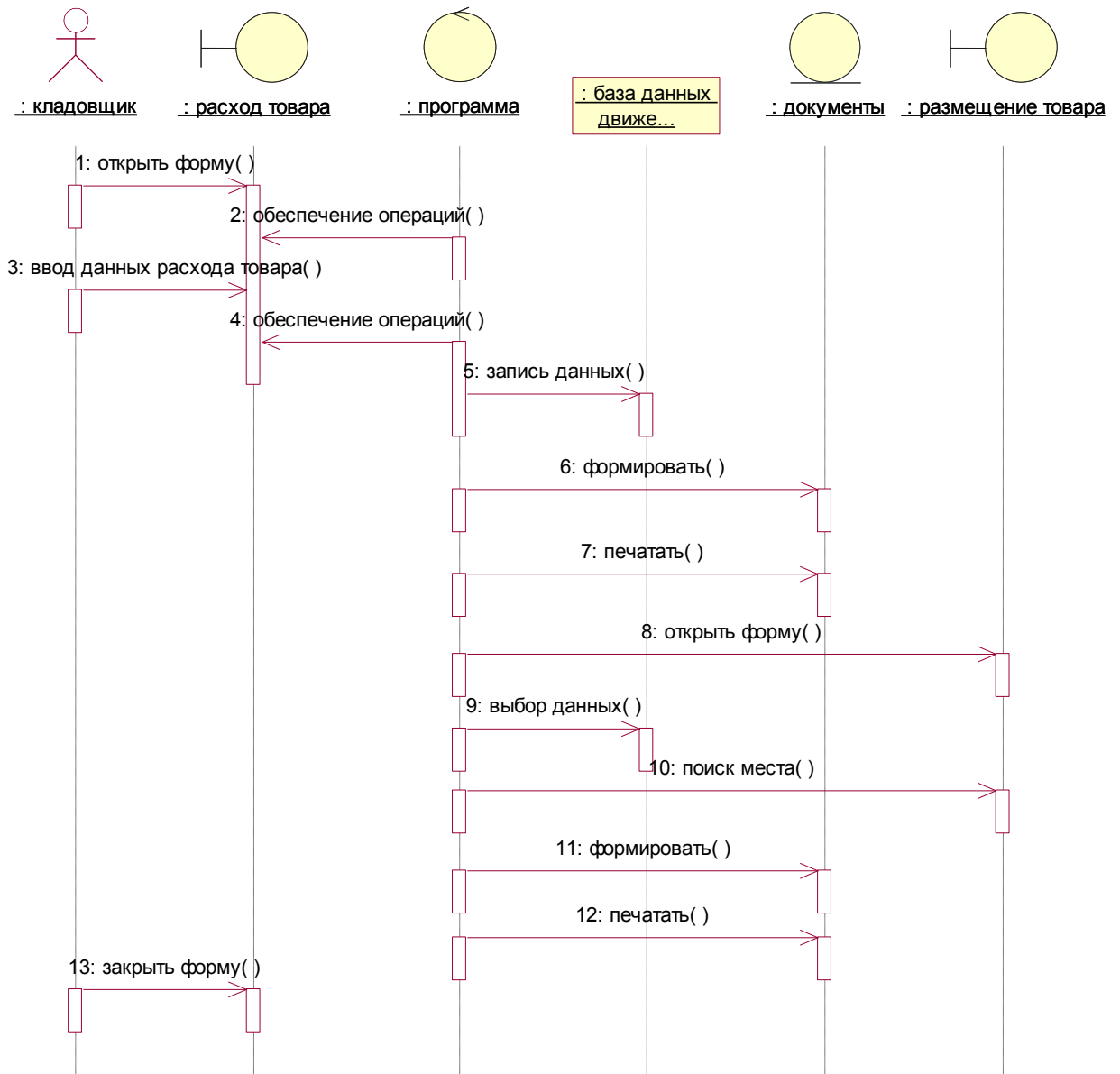


Рис. 9 Діаграма послідовності прецеденту «Вибуття товару»

Перелік контрольних питань

1. Значення діаграма послідовності.
2. Графічні символи діаграми послідовності.
3. Що показує лінія життя.

Створення кооперативної діаграми

Головна особливість діаграми кооперації полягає в можливості графічно представити не лише послідовність взаємодії, але і усі структурні стосунки між об'єктами, що беруть участь в цій взаємодії.

Передусім, на діаграмі кооперації у вигляді прямокутників зображуються об'єкти, що беруть участь у взаємодії, містять ім'я об'єкту,

його клас і, можливо, значення атрибутів. Далі, як і на діаграмі класів, вказуються асоціації між об'єктами у вигляді різних сполучних ліній. При цьому можна явно вказати імена асоціації і ролей, які грають об'єкти в цій асоціації. Додатково можуть бути зображені динамічні зв'язки - потоки повідомлень. Вони представляються також у вигляді сполучних ліній між об'єктами, над якими розташовується стрілка з вказівкою напрямку, імені повідомлення і порядкового номера в загальній послідовності ініціалізації повідомлень.

На відміну від діаграми послідовності, на діаграмі кооперації зображуються тільки стосунки між об'єктами, що грають певні ролі у взаємодії. На цій діаграмі не вказується час у вигляді окремого виміру. Тому послідовність взаємодій і паралельних потоків може бути визначена за допомогою порядкових номерів.

Отже, якщо необхідно явно специфікувати взаємозв'язки між об'єктами в реальному часі, краще це робити на діаграмі послідовності. Кооперативні діаграми створюються практично одночасно з діаграмами послідовності.

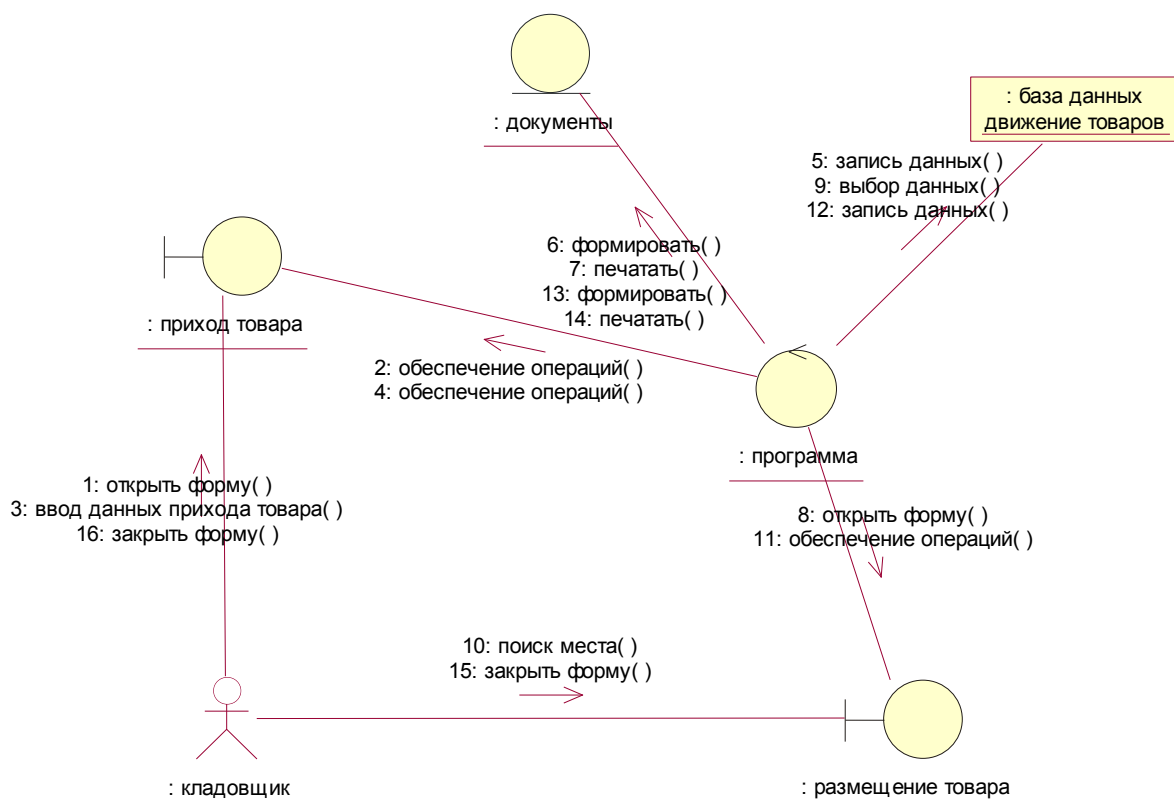


Рис. 10. Діаграма співпраці прецеденту «Прибуття товару»

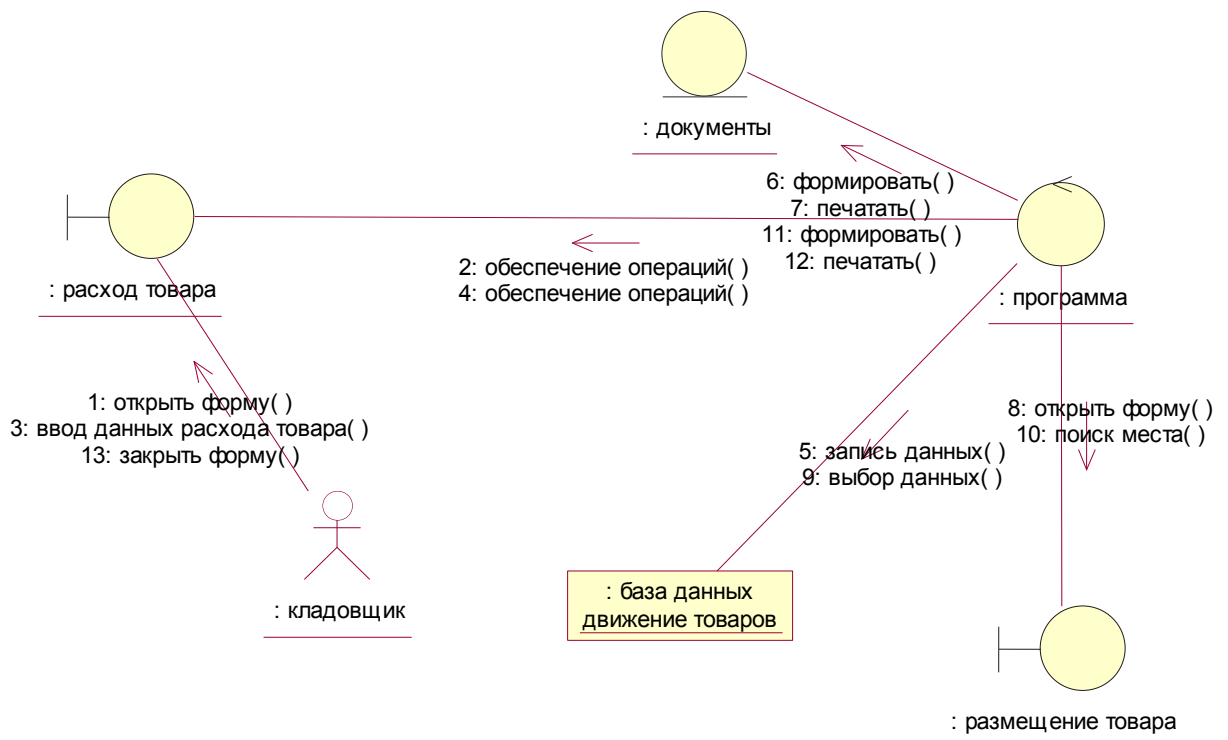


Рис. 11. Діаграма співпраці прецеденту «Вибуття товару»

Перелік контрольних питань

1. Значення діаграма кооперації.
2. Графічні символи діаграми кооперації.
3. Що відображає діаграма кооперації.
4. Що показують порядкові номери в діаграмі кооперації.

Діаграма діяльності

Діаграма діяльності (activity diagram) — діаграма, на якій показано розкладання деякої діяльності на її складові частини. Під діяльністю (activity) розуміється специфікація виконуваної поведінки у вигляді координованого послідовного і паралельного виконання підпорядкованих елементів — вкладених видів діяльності і окремих дій (action), сполучених між собою потоками, які йдуть від виходів одного вузла до входів іншого.

Загальна діаграма діяльності, яка охоплює програмне забезпечення ведення складського обліку, представлено на рис. 12.

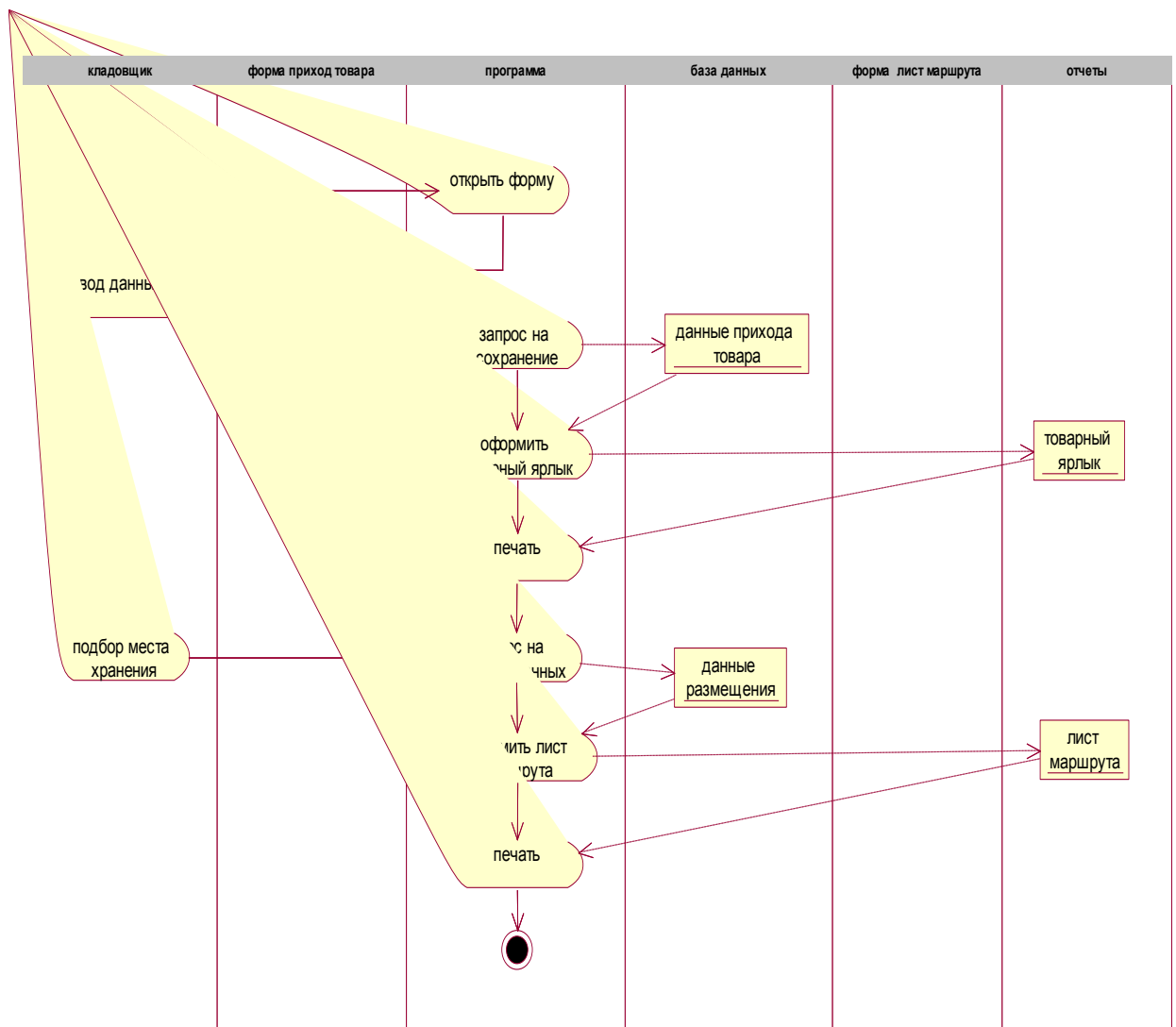


Рис. 12 Діаграма діяльності прецеденту «Прибуття товару»

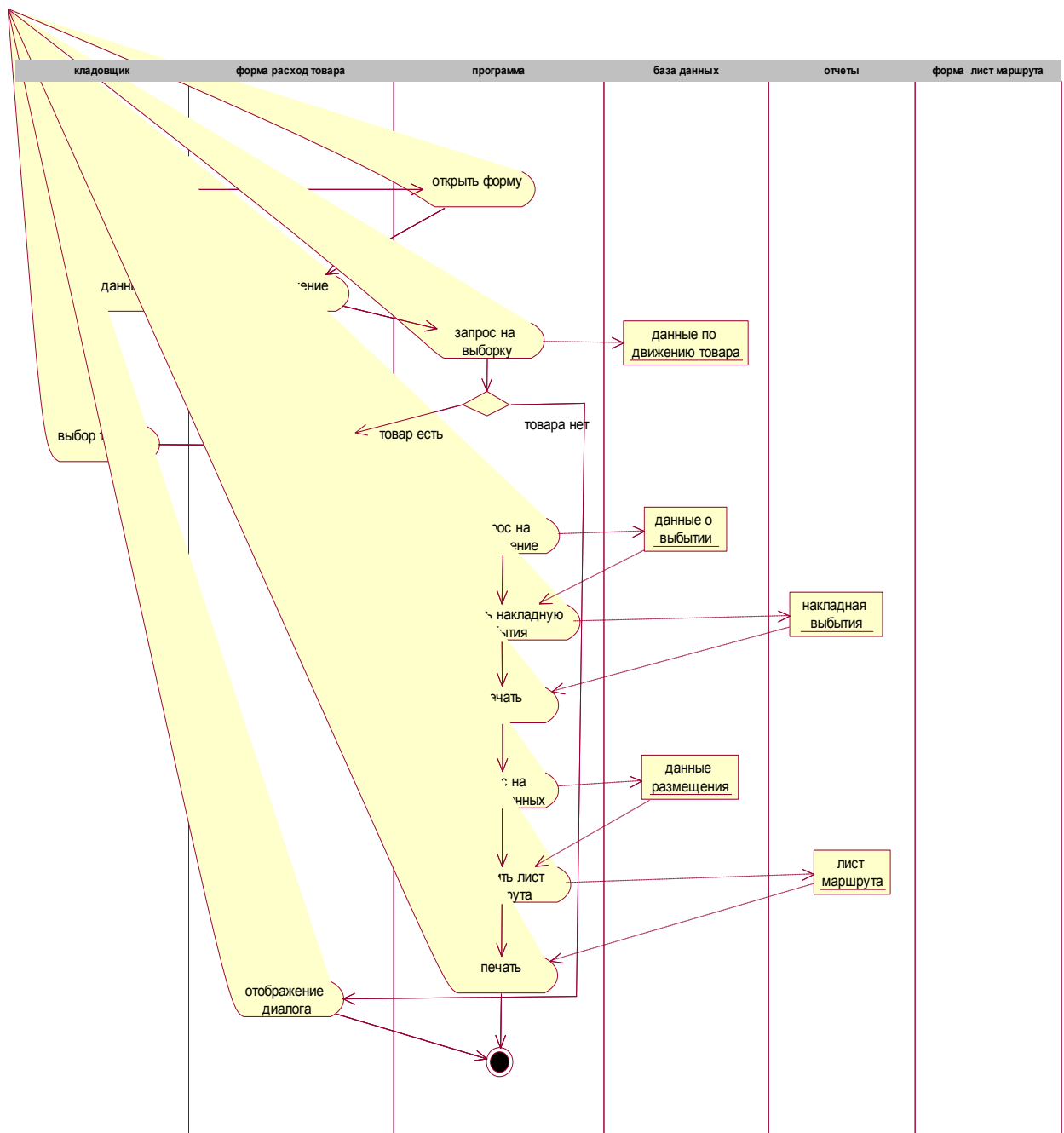






Рис. 13. Диаграмма діяльності прецеденту «Вибуття товару»

Таблица 7. Опис компонент діаграми

Графічне зображення	Назва	Призначення
	початковий стан	не містить ніяких внутрішніх дій, в цьому стані знаходиться об'єкт за умовчанням в початковий момент часу
	кінцевий стан	який також не містить ніяких внутрішніх дій. У цьому стані знаходиться за умовчанням після завершення роботи в кінцевий момент

Графічне зображення	Назва	Призначення
		часу. Воно служить для вказівки графічної області, в якій завершується процес зміни стану
	стан дій	Усередині цієї фігурки записується вираження дій, яке має бути унікальним в межах однієї діаграми діяльності
	Галуження	вживане у випадках, коли послідовно виконувана діяльність повинна розділитися на альтернативні гілки залежно від значення деякого проміжного результату. У цей ромб може входити тільки одна стрілка від того стану, після виконання якого, потік управління має бути продовжений по одній з гілок, що взаємно виключають. Стрілок, що виходять, може бути дві або більше, але для кожної з них явно вказується відповідна умова
	Перехід	відношення між двома послідовними станами, яке вказує на факт зміни одного стану іншим
	Розпаралелювання обчислень.	Може мати один перехід, що входить, і декілька що виходять, або декілька що входять і що один виходить

Перелік контрольних питань

1. Значення діаграма діяльності.
2. Графічні символи діаграми діяльності.
3. Що відображає діаграма діяльності.
4. Як показують паралельні процеси в діаграмі діяльності.

Створення діаграми компонентів

Діаграма компонентів показує організацію набору компонентів і зв'язок між ними. Діаграма компонентів застосовується для моделювання статичного виду системи з точки зору її реалізації.

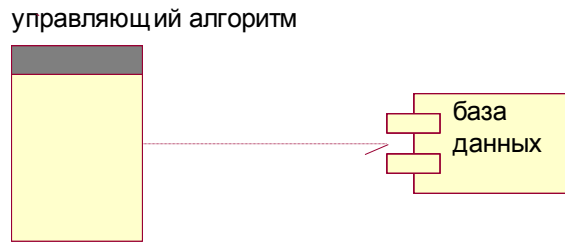


Рис. 14. Діаграма розгортання

Перелік контрольних питань

1. Значення діаграма компонентів.
2. Графічні символи діаграми компонентів.
3. Що відображає діаграма компонентів.

Створення діаграми розміщення

Фізичне представлення програмної системи не може бути повним, якщо відсутня інформація про те, на якій платформі і на яких обчислювальних засобах вона реалізована. Для представлення загальної конфігурації і топології розподіленої програмної системи в UML призначені діаграми розміщення.

Діаграма розміщення призначена для візуалізації елементів і компонентів програми, існуючих лише на етапі її виконання.

При цьому видаються тільки компоненти-екземпляри програми, що є виконуваними файлами або динамічними бібліотеками. Ті компоненти, які не використовуються на етапі виконання, на діаграмі розгортання не показуються. Так, компоненти з початковими текстами програм можуть бути присутніми тільки на діаграмі компонентів. На діаграмі розміщення вони не вказуються.

Діаграма розміщення відбиває фізичні взаємозв'язки між програмними і апаратними компонентами системи. Вона є хорошим засобом для того, щоб показати маршрути переміщення об'єктів і компонентів в розподіленій системі. Кожним вузлом на діаграмі розміщення є деякий тип обчислювального пристрою - у більшості випадків, частина апаратури. Ця апаратура може бути простим пристроєм або датчиком, а може бути і мейнфреймом.

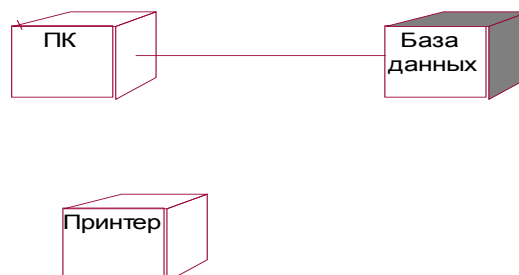


Рис. 15. Діаграма розгортання

Зафарбовані частини діаграми є процесорами, тобто комп'ютери, які можуть обробляти дані. Тут це сервер бази даних, каса і клієнт - сервер комп'ютерів. Не зафарбовані - пристрої, нездатні обробляти дані. В даному випадку, це принтер і сукупність екранів, використовуваних для відображення і введення інформації.

Діаграма розміщення містить графічні зображення процесорів, пристроїв, процесів і зв'язків між ними. На відміну від діаграм логічного представлення, діаграма розміщення є єдиною для системи в цілому, оскільки повинна цілком відбивати особливості її реалізації. Розробка діаграми розміщення, як правило, є останнім етапом специфікації моделі програмної системи.

Перелік контрольних питань

1. Значення діаграма розміщення.
2. Графічні символи діаграми розміщення.
3. Що відображає діаграма розміщення.

Список використаної літератури

1. Кватрани Т. Rational Rose 2000 до UML. Візуальне моделювання: Пер. з англ. - М.: ДМК Прес, 2001. - 176 с.
2. Лешек А. Мацяшек. Аналіз вимог і проектування систем. Розробка інформаційних систем з використанням UML. «Вільямс», Москва-Санкт-Петербург-Київ, 2012