

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт з дисципліни
**“ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП’ЮТЕРНИХ
СИСТЕМ”**
для студентів напряму 123 ”Комп'ютерна інженерія”
спеціалізацій (освітніх програм) ”Комп'ютерні системи та мережі”,
”Спеціалізовані комп'ютерні системи”
усіх форм навчання

2017 р.

Методичні вказівки до виконання лабораторних робіт з дисципліни "програмне забезпечення комп'ютерних систем" для студентів напряму 123 "Комп'ютерна інженерія" спеціалізацій (освітніх програм) "Комп'ютерні системи та мережі", "Спеціалізовані комп'ютерні системи" усіх форм навчання/ Укл.: К.М. Касьян, М.М. Касьян.- Запоріжжя: ЗНТУ, 2017. – 70 с.

Укладачі: К.М. Касьян, доцент, к.т.н.
М.М. Касьян, доцент, к.т.н.

Рецензент М.Ю. Тягунова, доцент, к.т.н.

Відповідальний за випуск: К.М. Касьян, доцент, к.т.н.

Затверджено
на засіданні кафедри КСМ
Протокол № 4 від 14.11.2017

Затверджено
на засіданні НМК КНТ
Протокол № 4 від 28.11.2017

ЗМІСТ

| | |
|--|----|
| ЗАГАЛЬНІ ВІДОМОСТІ..... | 5 |
| 1 ЛАБОРАТОРНА РОБОТА №1 Створення діаграми Варіантів | |
| Використання | 7 |
| 1.1 Мета роботи..... | 7 |
| 1.2 Постановка задачі | 7 |
| 1.3 Завдання на виконання..... | 7 |
| 1.4 Контрольні питання..... | 10 |
| 2 ЛАБОРАТОРНА РОБОТА №2 Створення діаграм взаємодії. | 11 |
| 2.1 Мета роботи..... | 11 |
| 2.2 Постанова задачі | 11 |
| 2.3 Завдання на виконання..... | 11 |
| 2.4 Контрольні питання..... | 16 |
| 3 ЛАБОРАТОРНА РОБОТА №3 Створення діаграм класів | 17 |
| 3.1 Мета роботи..... | 17 |
| 3.2 Постанова задачі | 17 |
| 3.3 Завдання на виконання..... | 17 |
| 3.4 Контрольні питання..... | 20 |
| 4 ЛАБОРАТОРНА РОБОТА №4 Створення діаграм | |
| Послідовностей (з урахуванням нових вимог) | 21 |
| 4.1 Мета роботи..... | 21 |
| 4.2 Постановка задачі | 21 |
| 4.3 Завдання на виконання..... | 22 |
| 4.4 Контрольні питання..... | 25 |
| 5 ЛАБОРАТОРНА РОБОТА №5 Доповнення діаграми класів | |
| (додавання зв'язків між класами) | 26 |
| 5.1 Мета роботи..... | 26 |
| 5.2 Постанова задачі | 26 |
| 5.3 Завдання на виконання..... | 26 |
| 5.4 Контрольні питання..... | 27 |
| 6 ЛАБОРАТОРНА РОБОТА №6 Створення діаграми станів | 28 |
| 6.1 Мета роботи..... | 28 |

| | |
|---|----|
| 6.2 Постановва задачі | 28 |
| 6.3 Завдання на виконання..... | 28 |
| 6.4 Контрольні питання..... | 31 |
| 7 ЛАБОРАТОРНА РОБОТА №7 Створення діаграми компонентів | 32 |
| 7.1 Мета роботи..... | 32 |
| 7.2 Постановка задачі | 32 |
| 7.3 Завдання на виконання..... | 32 |
| 7.4 Контрольні питання..... | 38 |
| 8 ЛАБОРАТОРНА РОБОТА №8 Створення діаграми розміщення | 39 |
| 8.1 Мета роботи..... | 39 |
| 8.2 Постановка задачі | 39 |
| 8.3 Завдання на виконання..... | 39 |
| 8.4 Контрольні питання..... | 41 |
| 9 ЛАБОРАТОРНА РОБОТА №9 Генерація коду C++..... | 42 |
| 9.1 Мета роботи..... | 42 |
| 9.2 Постановка задачі | 42 |
| 9.3 Завдання на виконання..... | 42 |
| 9.4 Контрольні питання..... | 43 |
| 10 ЛАБОРАТОРНА РОБОТА №10 Проектування власної системи | 44 |
| 10.1 Мета роботи..... | 44 |
| 10.2 Постановка задачі | 44 |
| 10.3 Контрольні питання..... | 68 |
| ЛІТЕРАТУРА | 69 |

ЗАГАЛЬНІ ВІДОМОСТІ

Об'єктне моделювання є важливим інструментом при розробці складних систем. Воно дозволяє увявити систему на абстрактному рівні, крім надмірних подробиць, і подає інформацію про її структуру. Це дуже корисно на початковому етапі проектування оскільки, грамотно побудована, ця структура може лягти в основу програмної реалізації системи.

Так чи інакше, розроблювачі завжди використовували моделювання, наприклад, у вигляді малювання схем на папері електронній дошці або документування. Але, з розвитком UML (Unified Modeling Language), у розроблювачів з'явилася можливість представляти свої, інколи зрозумілі тільки їм моделі, у загальних обрисах. Це дало величезний поштовх розвитку об'єктно-орієнтованого моделювання.

Отже, UML - це графічна мова для візуалізації, проектування й документування програмних систем, покликана полегшити роботу розроблювача на всіх її етапах.

Rational Rose - це популярний засіб візуального моделювання об'єктно-орієнтованих інформаційних систем компанії Rational Software Corp. Робота продукту заснована на універсальній мові моделювання UML. Завдяки унікальній мові моделювання Rational Rose здатен вирішувати практично будь-які завдання в проектуванні інформаційних систем: від аналізу бізнес-процесів до генерації коду певною мовою програмування. Rational Rose дозволяє розробляти як високорівневі, так і низькорівневі моделі, здійснюючи тим самим або абстрактне проектування, або логічне.

Rational Rose має весь необхідний набір візуальних засобів проектування. Він вирішує проблеми з генерацією коду певною мовою програмування й здійснює такі підходи, як пряме й зворотне проектування, а також Round Trip Engineering. Такий арсенал дозволяє не тільки спроектувати нову систему, але й доробити стару, використовуючи процес зворотного проектування.

Rational Rose Professional не створює 100% коду, що виконується. На виході розроблювач отримує шаблон інформаційної системи певною мовою програмування, яка згодом потрібно запрограмувати.

Курс лабораторних робіт присвячений одному з CASE-засобів структурного і об'єктно – орієнтованого аналізу, а саме Rational Rose Professional, і направлений на навчання студентів технології складання діаграм і отриманню з них, зрештою, програмного коду на мові високого рівня, наприклад C++.

Моделювання проводиться як порівневий спуск від концептуальної моделі до логічної, а потім до фізичної моделі програмної системи.

Концептуальна модель виражається у вигляді діаграми варіантів використання (use-case diagram). Цей тип діаграм служить для ітераційного циклу загальної постановки задачі разом із замовником. Діаграми варіантів використання саме і служать основою для досягнення взаєморозуміння між програмістами - професіоналами, які розробляють проект, і замовниками проекту. Усередині кожного прецеденту можуть бути визначені:

- вкладена діаграма використання;
- діаграма взаємодії об'єктів (collaboration diagram);
- діаграма послідовності взаємодії (sequence diagram);
- діаграма класів (class diagram);
- діаграма переходу Станів (state diagram).

Звіт до кожної з лабораторних робіт повинен містити:

- тему і мету роботи;
- основні отримані результати роботи (діаграми, програмний код, тощо);
- відповіді на контрольні запитання.

1 ЛАБОРАТОРНА РОБОТА №1

СТВОРЕННЯ ДІАГРАМИ ВАРІАНТІВ ВИКОРИСТАННЯ

1.1 Мета роботи

Вивчити основні етапи проведення проектування в Rational Rose Professional, інтерфейс Rational Rose Professional і основні принципи роботи з ним. Створити діаграму Варіантів Використання

1.2 Постановка задачі

Необхідно розробити Систему роботи бібліотеки.

Система складається з наступних компонентів:

- Читач
- Бібліотекар
- Завідувач

Користувач здійснює пошук книг за каталогом, наголошує на книгах, які він хотів би узяти. Далі він відзначає із списку літератури, що є у нього, ті книги, які він хотів би здати і відправляє обидва списки Бібліотекарю.

Відповідно до одержаних даних Бібліотекар здійснює видачу і прийом літератури. Також він подає запит в систему на наявність у Читача заборгованостей по незданій літературі.

Завідувач періодично подає в систему запит на наявність в бібліотеці і у читачів книг, і порівнює одержаний результат з книгами, що реально є в бібліотеці.

1.3 Завдання на виконання

Створіть діаграму Варіантів Використання для системи роботи бібліотеки. Необхідні для цього дії детально описані далі. Готова діаграма Варіантів Використання повинна виглядати як на рисунку 1.2.

Створити діаграму Варіантів Використання

1. Двічі клацніть на Головній діаграмі Варіантів Використання (Main) в браузері, щоб відкрити її. Назвіть цю діаграму «Поиск книги по каталогу».

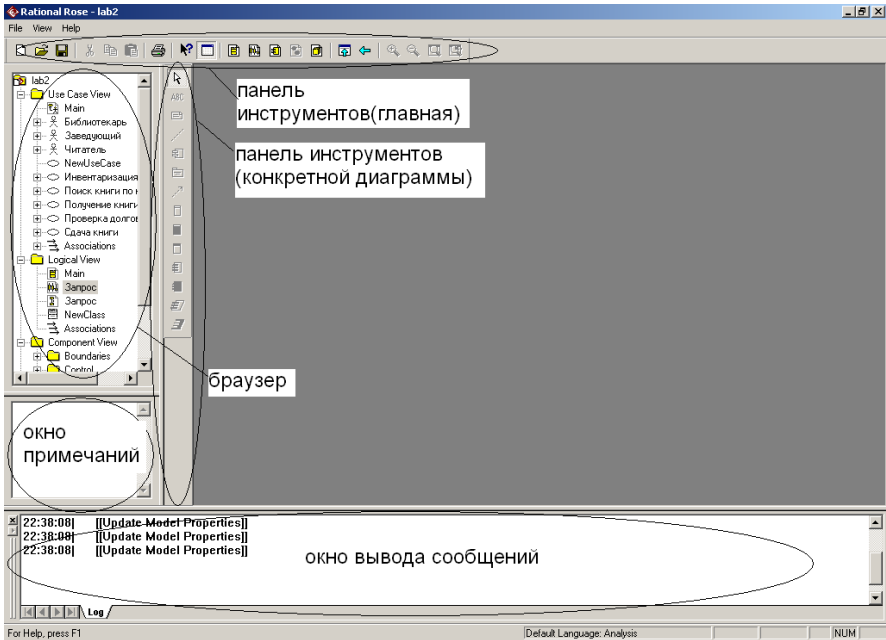


Рисунок 1.1 – Вікно програми

2. За допомогою кнопки Use Case (Варіант Використання) панелі інструментів (діаграми) помістите на діаграму новий варіант використання.

3. Назвіть цей новий варіант використання "Поиск книги по каталогу".

4. Повторіть етапи 2 і 3, щоб помістити на діаграму решту варіантів використання: Получение книги, Сдача книги, Проверка долгов читателя , Инвентаризация.

5. За допомогою кнопки Actor (Дійова особа) панелі інструментів помістите на діаграму нову дійову особу.

6. Назвіть його "Читатель"

7. Повторіть кроки 5 і 6, помістивши на діаграму решти дійових осіб: Библиотекарь, Заведующий.

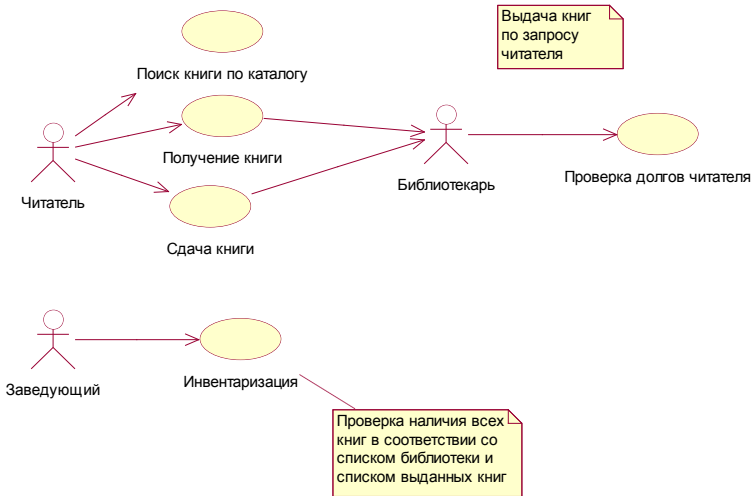


Рисунок 1.2 – Діаграма Варіантів Використання для системи складання розкладу

Додати асоціації

1. За допомогою кнопки Unidirectional Association (Однонаправлена асоціація) панелі інструментів намалюйте асоціацію між дійовою особою Читатель і варіантом використання "Поиск книги по каталогу".

2. Повторіть цей етап, щоб помістити на діаграму решту асоціацій.

Додати коментарі до варіантів використання

1. За допомогою кнопки Note (Коментар) панелі інструментів помістите на діаграму коментар до сутності «Библиотекарь»: «Выдача книг по запросу читателя».

2. Аналогічно додаванню асоціацій, створіть зв'язок коментаря з елементом за допомогою кнопки «Anchor Note to Item» (Зв'язок коментаря з елементом).

3. Повторіть етапи 1 і 2, щоб помістити на діаграму коментар до функції «Инвентаризация»: «Проверка наявності всіх книг відповідно до списку бібліотеки і списку виданих книг».

Додати описи до варіантів використання і дійових осіб

1. Виділіть у браузері варіант використання "Поиск книги по каталогу".
2. У вікні документації введіть наступний опис до цього варіанту використання: «Цей варіант використання дає читачу можливість перевірки наявності потрібної йому книги».
3. За допомогою вікна документації введіть свої описи до всієї решти варіантів використання і дійовим особам.

Прикріплення файлу до варіанту використання

1. Для опису головного потоку подій варіанту використання "Получение книги" створіть файл BookOrder.doc, що містить наступний текст.

1. *Читач здійснює пошук книги за запитом.*
2. *Система виводить результати пошуку.*
3. *Читач наголошує на книгах, які він хоче узяти.*
4. *Також читач наголошує із списку книг, що є у нього, на тих, які він збирається взяти*
5. *Результат відправляється бібліотекарю.*

2. Клацніть правою кнопкою миші на варіанті використання "Получение книги".

3. У меню, що відкрилося, виберіть пункт Open Specification (Відкрити специфікацію)

4. Перейдіть на вкладку файлів.

5. Клацніть правою кнопкою миші на білому полі і з меню, що відкрилося, виберіть пункт Insert File (Ввести файл).

6. Вкажіть файл BookOrder.doc і натискуйте на кнопку Open (Відкрити), щоб прикріпити файл до варіанту використання.

1.4 Контрольні питання

1. Що таке CASE- засоби? Навіщо вони потрібні? Наведіть приклад одного з таких засобів.

2. Які види діаграм ви знаєте? Для чого потрібна діаграма використання?

3. Які елементи використовуються для побудови діаграми використання?

4. Яким чином можна додавати коментарі до елементів діаграми? У чому різниця між такими коментарями?

2 ЛАБОРАТОРНА РОБОТА №2 СТВОРЕННЯ ДІАГРАМ ВЗАЄМОДІЇ

2.1 Мета роботи

Розробити діаграму Послідовності і Кооперативну діаграму, що описують введення нового замовлення в нашу систему обробки замовлень.

2.2 Постанова задачі

У цій лабораторній роботі необхідно провести аналіз складових частин діаграми і скласти два види діаграм взаємодії: діаграму послідовностей і кооперативну діаграму. Розглянемо варіант використання - Получение книг. Опис сценарію виглядає таким чином:

1. Читач здійснює пошук книг за каталогом.
2. Читач здійснює пошук книг за каталогом, але книги немає в наявності.
3. Читач здійснює пошук книг за каталогом, але при передачі запиту на отримання книги бібліотекарю відбувається помилка.

2.3 Завдання на виконання

Створіть діаграму Послідовності і Кооперативну діаграму, що відображає введення запиту в систему обробки запитів. Готова діаграма Послідовності повинна виглядати як на рисунку 2.3.

Ця діаграма відповідає успішному варіанту ходу подій. Для опису того, що трапиться, якщо виникне помилка, або якщо користувач вибере інші дії із запропонованих, доведеться розробити інші діаграми. Кожний альтернативний потік варіанту використання може бути промодельований за допомогою своїх власних діаграм Взаємодії.

Налаштування

1. У меню моделі виберіть пункт Tools > Options (Інструменти > Параметри).
2. Перейдіть на вкладку діаграм.
3. Контрольні перемикачі Sequence Numbering, Collaboration Numbering і Focus Control повинні бути помічені.
4. Натискуйте ОК, щоб вийти з вікна параметрів.

Створення діаграми Послідовності

1. Клацніть правою кнопкою миші на Логічному представленні браузера.
2. У меню, що відкрилося, виберіть пункт New > Sequence Diagram.
3. Назвіть нову діаграму "Введення запиту".
4. Двічі клацніть на ній, щоб відкрити її.

Додавання на діаграму дійової особи і об'єктів

1. Перетягніть дійову особу Читатель з браузера на діаграму.
2. На панелі інструментів натискуйте кнопку Object (Об'єкт).
3. Клацніть мишею у верхній частині діаграми, щоб помістити туди новий об'єкт.
4. Назвіть об'єкт "Форма выбора действия".
5. Повторіть етапи 3 і 4, щоб помістити на діаграму всю решту об'єктів: Форма запроса і Запрос Библиотекаряу.

Додавання повідомлень на діаграму

1. На панелі інструментів натискуйте кнопку Object Message (Повідомлення об'єкту).
2. Проведіть мишею від лінії життя актора Читатель до лінії життя об'єкту Форма выбора действия.
3. Виділивши повідомлення, введіть його ім'я „Обращение к системе”.
4. Повторіть етапи 2 і 3, щоб помістити на діаграму додаткові повідомлення відповідно до рисунку:
 - Відкрити форму
 - Ввести подробиці замовлення
 - Відіслати запит бібліотекарю
 - Створити порожній запит
 - Заповнити запит
 - Передача запиту бібліотекарю

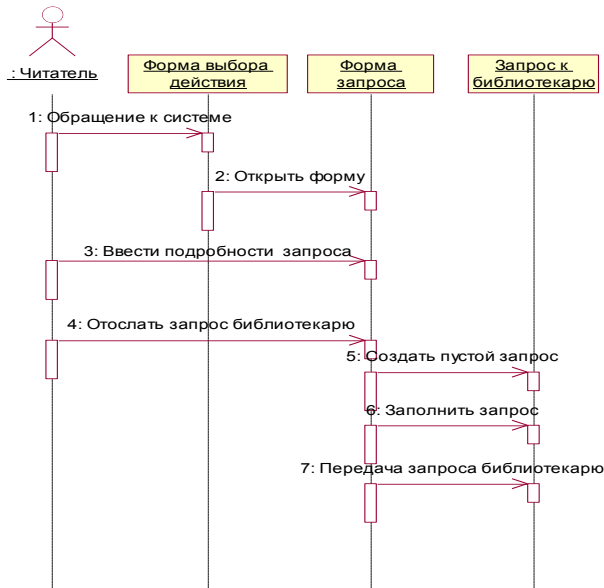


Рисунок 2.1- Діаграма Послідовності введення запиту

Співвідношення об'єктів з класами

1. Клацніть правою кнопкою миші на об'єкті Форма выбора действия.
2. У меню, що відкрилося, виберіть пункт Open Specification (Відкрити специфікацію).
3. У списку класів, що розкривається, виберіть пункт <New> (Створити). З'явиться вікно специфікації класів.
4. У полі імені введіть ім'я ВыборЗапроса.
5. Клацніть на кнопці ОК. Ви повернетеся до вікна специфікації об'єкту.
6. У списку класів виберіть тепер клас ВыборЗапроса.
7. Клацніть на кнопці ОК, щоб повернутися до діаграми. Тепер об'єкт називається Форма выбора действия: ВыборЗапроса.
8. Для співвідношення решти об'єктів з класами повторите етапи з 1 по 7:
9. Форма запроса : Запрос.
10. Запрос Библиотекарю : ЗапросБиблиотекарю.

Після завершення цих дій ваша діаграма повинна виглядати як на рисунку 2.2.

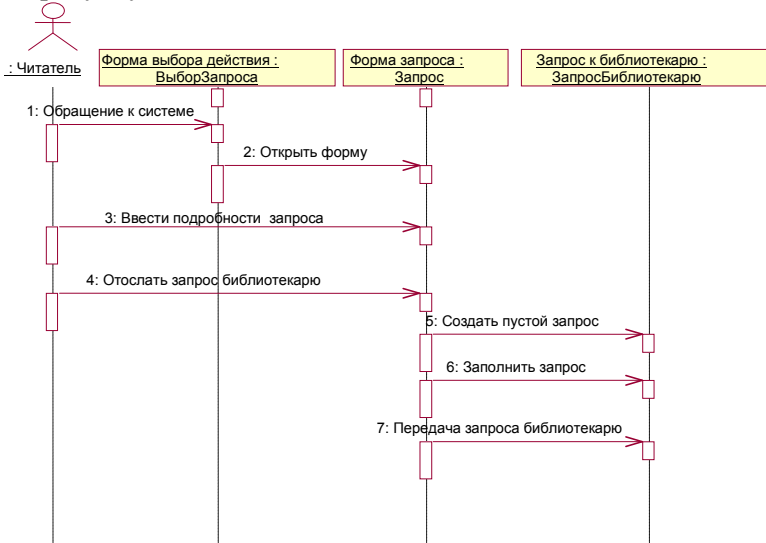


Рисунок 2.2 – Діаграма Послідовності з іменами класів

Співвідношення повідомлень з операціями

1. Клацніть правою кнопкою на повідомленні 1: Звернення до системи
 2. У меню, що відкрилося, виберіть пункт <new operation> (створити операцію). З'явиться вікно специфікації операції.
 3. У полі імені введіть ім'я операції - **login**
 4. Натискуйте на кнопку ОК, щоб закрити вікно специфікації операції і повернутися на діаграму.
 5. Ще раз клацніть правою кнопкою миші на повідомленні 1.
 6. У меню, що відкрилося, виберіть нову операцію **login**
 7. Повторіть повідомлення з 1 по 6, поки не співвіднесете з операціями всю решту повідомлень:
- 2: Відкрити форму - **open**
 - 3: Ввести подробиці запиту - **info**
 - 4: Відіслати запит бібліотекарю - **send**
 - 5: Створити порожній запит - **create**
 - 6: Заповнити запит - **submit**

7: Передача запиту бібліотекарю – **show**

Ваша діаграма повинна виглядати як на рисунку 2.3.

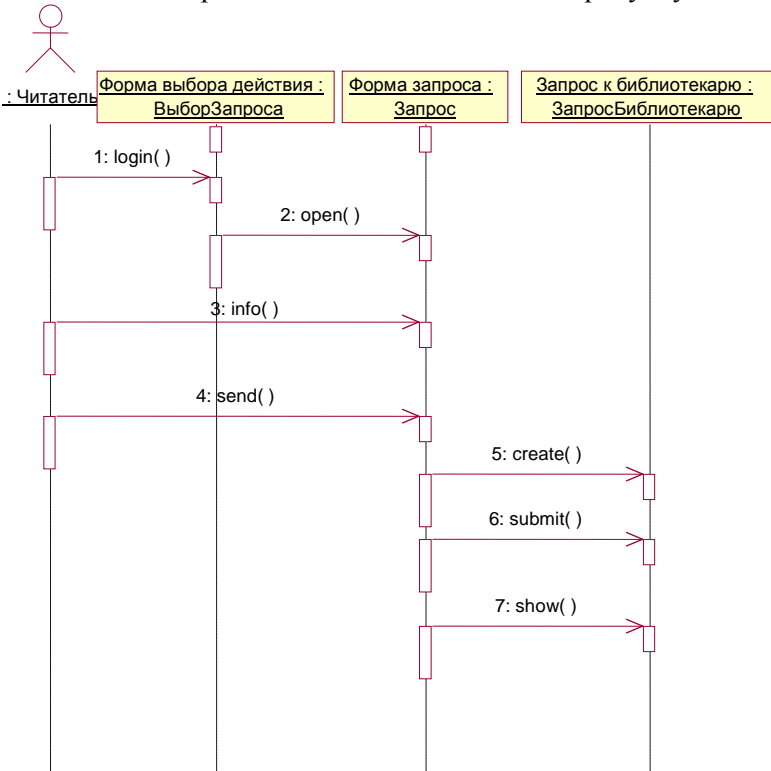


Рисунок 2.3 – Діаграма Послідовності з показаними на ній операціями

Створення Кооперативної діаграми

Для створення Кооперативної діаграми достатньо просто натискувати клавішу F5.

Ваша діаграма повинна виглядати як на рисунку 2.4.

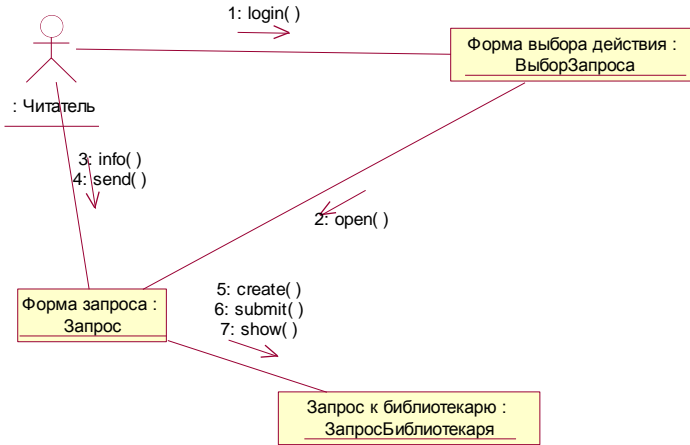


Рисунок 2.4 - Кооперативна діаграма з показаними на ній операціями

2.4 Контрольні питання

1. Які діаграми взаємодії використані у даній лабораторній роботі? Яка з них є більш наочною? Чому?
2. Які елементи застосовуються для побудови діаграми послідовностей?
3. Для чого потрібно робити співвідношення повідомлень з операціями?
4. Яким чином найшвидше пересуватися між діаграмами взаємодії?

3 ЛАБОРАТОРНА РОБОТА №3 СТВОРЕННЯ ДІАГРАМ КЛАСІВ

3.1 Мета роботи

У ході лабораторної роботи треба буде згрупувати в пакети класи, створені під час виконання попередньої роботи. Потім вам належить створити декілька діаграм Класів, на яких ви також покажете класи і пакети системи.

3.2 Постанова задачі

Необхідно об'єднати їх в пакети за стереотипом. Для цього створимо пакети Entities (Сутності), Boundaries (Границі) і Control (Управління), помістивши в них відповідні класи. Потім для кожного пакету створимо діаграми Класів; крім того, на Головній діаграмі будуть показані пакети і на діаграмі Поиск книги по каталогу - всі класи цього варіанту використання.

3.3 Завдання на виконання

Створення діаграми Класів

Об'єднайте знайдені класи в пакети. Створіть діаграму Класів для відображення пакетів, діаграми Класів для представлення класів в кожному пакеті і діаграму Класів для представлення всіх класів варіанту використання "Поиск книги по каталогу".

Етапи виконання.

Налаштування

1. У меню моделі виберіть пункт Tools > Options (Інструменти > Параметри).
2. Перейдіть на вкладку діаграм.
3. Переконайтеся, що помічений контрольний перемикач Show Stereotypes (Показати стереотипи).
4. Переконайтеся, що помічені контрольні перемикачі Show All Attributes (Показати всі атрибути) і Show All Operations (Показати всі операції).
5. Переконайтеся, що не помічені перемикачі Suppress Attributes (Подавити виведення атрибутів) і Suppress Operations (Подавити виведення операцій).

Створення пакетів

1. Клацніть правою кнопкою миші на Логічному представленні браузера.
2. У меню, що відкрилося, виберіть пункт New > Package (Створити > пакет).
3. Назвіть новий пакет Entities (Сутності).
4. Повторіть етапи з першого по третій, створивши пакети Boundaries (Границі) і Control (Управління).

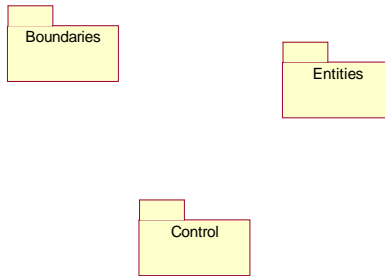


Рисунок 3.1 – Головна діаграма Класів системи обробки замовлень

Створення Головної діаграми Класів

1. Двічі клацніть на Головній діаграмі Класів прямо під Логічним представленням браузера, щоб відкрити її.
2. Перетягніть пакети Entities, Boundaries і Control з браузера на діаграму.

Головна діаграма Класів повинна виглядати як на рисунку 3.1.

Створення діаграми Класів для сценарію "Поиск книги по каталогу" зі всіма класами.

1. Клацніть правою кнопкою миші на Логічному представленні браузера.
2. У меню, що відкрилося, виберіть пункт New > Class Diagram (Створити > Діаграму Класів).
3. Назвіть нову діаграму Класів Search a book (Пошук книги).
4. Клацніть в браузері на цій діаграмі двічі, щоб відкрити її.
5. Перетягніть з браузера всі класи (ВыборЗапроса, Запрос, ЗапросБиблиотекаря).

Діаграма Класів повинна виглядати як на рисунку 3.2.

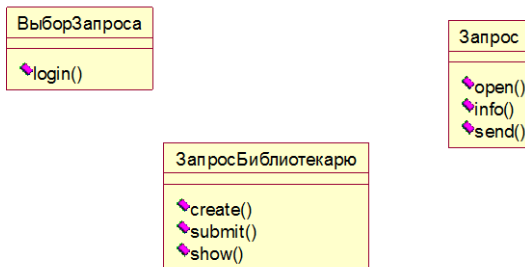


Рисунок 3.2 – Діаграма Класів Search a book

Додавання стереотипів до класів

1. Клацніть правою кнопкою миші на класі ВыборЗапроса діаграми.
2. У меню, що відкрилося, виберіть пункт Open Specification (Відкрити специфікацію).
3. У полі стереотипу введіть слово Boundary.
4. Натискуйте на кнопку ОК.
5. Клацніть правою кнопкою миші на класі Запрос діаграми.
6. У меню, що відкрилося, виберіть пункт Open Specification (Відкрити специфікацію).
7. У списку, що розкривається, в полі введіть слово Control. Вкажіть його.
8. Натискуйте на кнопку ОК.
9. Повторіть етапи 1 - 4, зв'язавши клас ЗапросБиблиотекарю із стереотипом Entity.

Тепер діаграма Класів повинна виглядати як на рисунку 3.3.

Об'єднання класів в пакети

1. Перетягніть в браузері клас ВыборЗапроса на пакет Boundaries.
2. Перетягніть класи Запрос на пакет Control.
3. Перетягніть клас ЗапросБиблиотекарю на пакет Entities.

Додавання діаграм Класів до кожного пакету

1. Клацніть правою кнопкою на пакеті Boundaries браузера.
2. У меню, що відкрилося, виберіть пункт New > Class Diagram (Створити > Діаграму Класів).
3. Введіть ім'я нової діаграми - Main (Головна).
4. Двічі клацніть мишею на цій діаграмі, щоб відкрити її.

5. Перетягніть на неї з браузера клас ВыборЗапроса.
6. Закрийте діаграму.
7. Клацніть правою кнопкою на пакеті Entities браузера.
8. У меню, що відкрилося, виберіть пункт New > Class Diagram (Створити > Діаграму Класів).
9. Введіть ім'я нової діаграми - Main (Головна).
10. Двічі клацніть мишею на цій діаграмі, щоб відкрити її.
11. Перетягніть на неї з браузера клас ЗапросБиблиотекарю.
12. Закрийте діаграму.
13. Клацніть правою кнопкою на пакеті Control браузера.
14. У меню, що відкрилося, виберіть пункт New > Class Diagram (Створити > Діаграму Класів).
15. Введіть ім'я нової діаграми - Main (Головна).
16. Двічі клацніть мишею на цій діаграмі, щоб відкрити її.
17. Перетягніть на неї з браузера клас Запрос.
18. Закрийте діаграму.

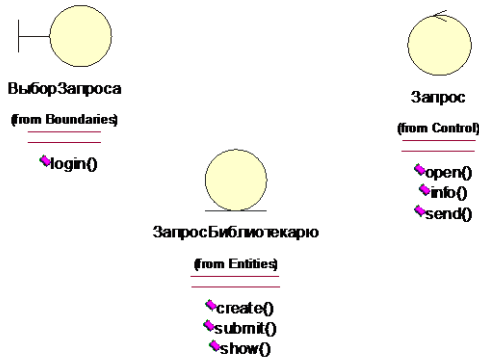


Рисунок 3.3 – Стереотипи класів для варіанту використання «Поиск книги по каталогу»

3.4 Контрольні питання

1. Що собою представляє діаграма класів?
2. Що собою представляє елемент «пакет»? Які його функції?
3. Чим відрізняються один від одного класи на рисунку 3.3?
4. Які стереотипи можуть бути у класів?

4 ЛАБОРАТОРНА РОБОТА №4 СТВОРЕННЯ ДІАГРАМ ПОСЛІДОВНОСТЕЙ (З УРАХУВАННЯМ НОВИХ ВИМОГ)

4.1 Мета роботи

У лабораторній роботі №2 було створено декілька операцій для класів нашої задачі. В попередній лабораторній роботі ми нанесли класи на діаграму. В цій роботі до описів операцій будуть додані деталі, включаючи параметри і типи значень, що повертаються. Крім того, біля класів будуть визначені атрибути.

4.2 Постановка задачі

Мовою програмування виберемо C++, що дозволить додати до класів параметри операцій, типи даних і типи значень, що повертаються.

Для визначення атрибутів звернемося до потоку подій. В результаті, до класу Запрос діаграми Класів додамо атрибути OrderNumber (номер запиту) і ReaderName (Ім'я читача). Оскільки в одному запиті можна вказати велику кількість книг, і біля кожного з них є свої власні дані, моделюватимемо їх як самостійні класи, а не як атрибути класу Запрос.

Щоб привести модель у відповідність з новими ідеями, перебудуємо діаграму Послідовностей, як показано на рисунку 4.1.

Введемо також додаткові дані – Дата узяття книги.

Введення нових вимог, пов'язаних з датами, привели до необхідності ввести нові атрибути до класу Запрос. Після цього модель знову стала відповідати останнім пред'явленим до системи вимогам.

Додавання атрибутів і операцій

Додамо атрибути і операції до класів діаграми Класів "Search a book". Для атрибутів і операцій використовуємо специфічні для мови особливості. Встановимо параметри так, щоб показувати всі атрибути, всі операції і їх сигнатури. Видимість покажемо за допомогою нотації UML.

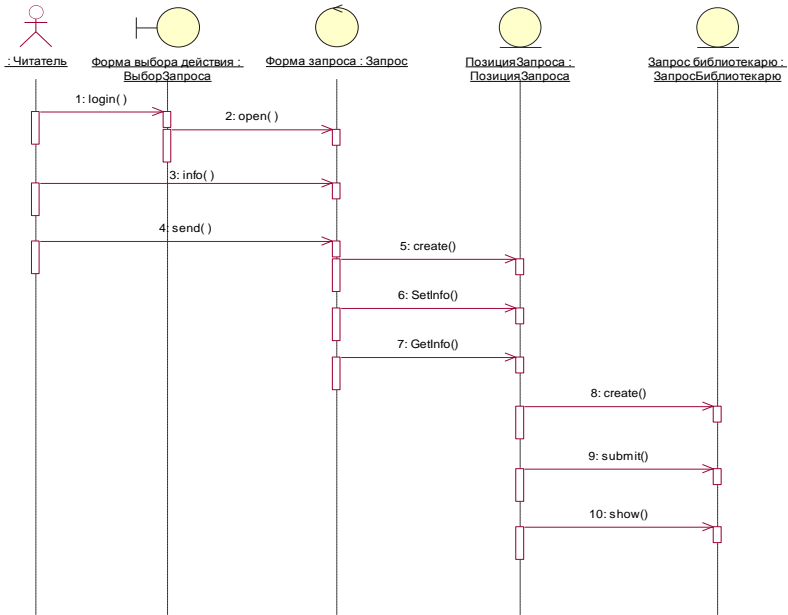


Рисунок 4.1 – Перебудована діаграма Послідовностей

4.3 Завдання на виконання

Налаштування

1. У меню моделі виберіть пункт Tools > Options.
2. Перейдіть на вкладку Diagram.
3. Переконайтеся, що перемикач Show Visibility помічений.
4. Переконайтеся, що перемикач Show Stereotypes помічений.
5. Переконайтеся, що перемикач Show Operation Signatures помічений.
6. Переконайтеся, що перемикачі Show All Attributes і Show All Operations помічені.
7. Переконайтеся, що перемикачі Suppress Attributes і Suppress Operations не помічені.
8. Перейдіть на вкладку Notation.
9. Переконайтеся, що перемикач Visibility as Icons помічений.

Додавання нового класу

1. Знайдіть в браузері діаграму Класів "Search a book".
2. Клацніть на ній двічі, щоб її відкрити.
3. Натискуйте кнопку Class панелі інструментів.
4. Клацніть мишею усередині діаграми, щоб помістити там новий клас.
5. Виберіть в меню, що з'явилося, назву нового класу – ПозиціяЗапроса.
6. Перевірте, щоб цьому класу відповідав стереотип Entity і він знаходився в пакеті Entities.

Додавання атрибутів

1. Клацніть правою кнопкою миші на класі Запрос.
2. У меню, що відкрилося, виберіть пункт New Attribute (Створити атрибут).
3. Введіть новий атрибут OrderNumber : Integer (НомерЗапроса)
4. Натискуйте клавішу Enter.
5. Введіть наступний атрибут ReaderName : String (ИмяЧитателя).
6. Повторіть етапи 4 і 5, додавши атрибут OrderDate : Date (ДатаЗапроса).
7. Клацніть правою кнопкою миші на класі ПозицияЗапроса.
8. У меню, що відкрилося, виберіть пункт New Attribute (Створити атрибут).
9. Введіть новий атрибут ItemID : Integer (ИдентификаторКниги).
10. Натискуйте клавішу Enter.
11. Введіть наступний атрибут Author : String (АвторКниги).
12. Введіть наступний атрибут BookName : String (НазваКниги).
13. Натискуйте клавішу Enter.

Докладний опис операцій за допомогою діаграми Класів

1. Клацніть мишею на класі ПозицияЗапроса, виділивши його у такий спосіб.
2. Клацніть на цьому класі ще один раз, щоб перемістити курсор всередину.

3. Відредагуйте операцію Create(), щоб вона виглядала таким чином: Create() : Boolean

4. Відредагуйте операцію SetInfo(), щоб вона виглядала таким чином: SetInfo(OrderNumber : Integer, ReaderName : String, OrderDate : Date): Boolean

5. Відредагуйте операцію GetInfo(), щоб вона виглядала таким чином: GetInfo() : String

Докладний опис операцій за допомогою браузера

1. Знайдіть в браузері клас ПозицияЗапроса
2. Щоб розкрити цей клас, клацніть на значку "+" поряд з ним. В браузері з'являться його атрибути і операції.
3. Двічі клацніть на операції GetInfo(), щоб відкрити вікно її специфікації.
4. У списку Return Type (клас, що повертається), що розкривається, вкажіть String.
5. Клацніть на кнопці ОК, закривши вікно специфікації операції.
6. Двічі клацніть в браузері на операції SetInfo класу ПозицияЗапроса, щоб відкрити вікно її специфікації.
7. У списку Return Type, що розкривається, вкажіть Boolean.
8. Перейдіть на вкладку Detail (Детально).
9. Клацніть правою кнопкою миші на білому полі в області аргументів, щоб додати туди новий параметр.
10. У меню, що відкрилося, виберіть пункт Insert. Rose додасть туди аргумент під назвою argname.
11. Змініть ім'я аргументу argname на ID.
12. Клацніть на імені 2 рази, у вікні, що відкрилося, в полі Type виберіть тип Integer.
13. Клацніть на полі Default, щоб додати значення аргументу за умовчанням. Введіть туди число 0.
14. Натискуйте на кнопку ОК, закривши вікно специфікації операції.
15. Двічі клацніть на операції Create() класу ПозицияЗапроса, щоб відкрити вікно її специфікації.
16. У списку Return Type, що розкривається, вкажіть Boolean.
17. Натискуйте на кнопку ОК, закривши вікно специфікації

операції.

Докладний опис операцій за допомогою будь-якого з описаних методів

1. Використовуючи браузер або діаграму Класів, введіть наступну сигнатуру операцій класу Запрос:

open() : Boolean

info() : Boolean

send() : Boolean

2. Використовуючи браузер або діаграму Класів, введіть наступну сигнатуру операцій класу ЗапросБиблиотекарю: Create() : Boolean.

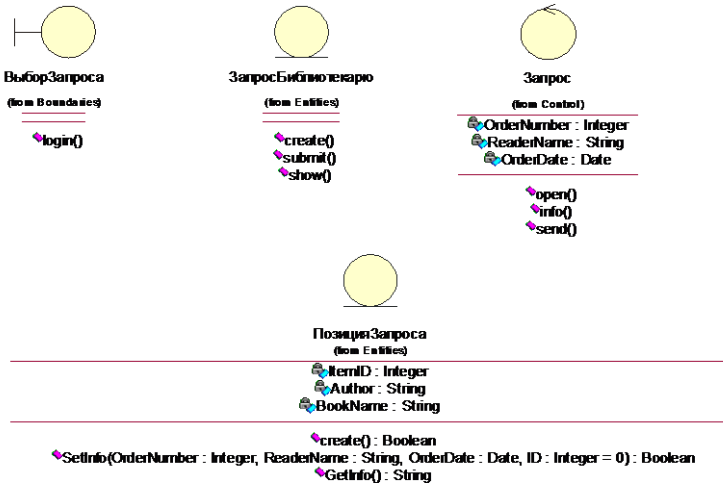


Рисунок 4.2 – Сценарій "Search a book"

4.4 Контрольні питання

1. Які додаткові властивості можна додати до класу?
2. Що означає тип, що повертає значення? Наведіть приклади таких типів.
3. Що таке атрибут? Які атрибути застосовуються для класу Запрос?
4. Як саме була перебудована діаграма класів (які класи, операції, тощо були додані)?

5 ЛАБОРАТОРНА РОБОТА №5 ДОПОВНЕННЯ ДІАГРАМИ КЛАСІВ (ДОДАВАННЯ ЗВ'ЯЗКІВ МІЖ КЛАСАМИ)

5.1 Мета роботи

Визначити зв'язки між класами, що беруть участь у варіанті використання "Поиск книги по каталогу".

5.2 Постанова задачі

Щоб знайти зв'язки, необхідно вивчити діаграми Послідовності. Всі взаємодіючі там класи потребують визначення відповідних зв'язків на діаграмах Класів. Після виявлення зв'язків необхідно додати їх в модель.

Додавання зв'язків

Додамо зв'язки до класів, що беруть участь при пошуку книги (діаграма "Search a book").

5.3 Завдання на виконання

Налаштування

1. Знайдіть в браузері діаграму Класів "Search a book"
2. Двічі клацніть на ній, щоб відкрити її.
3. Перевірте, чи є на панелі інструментів діаграми кнопка Unidirectional Association. Якщо її немає, продовжуйте налаштування, виконавши етапи 4 і 5. Якщо є, приступайте до виконання самої вправи.
4. Клацніть на панелі управління Tools, виберіть пункт Create.
5. А далі виберіть із списку Unidirectional Association.

Додавання асоціацій

1. Виберіть опцію Unidirectional Association.
2. Намалюйте асоціацію від класу ВыборЗапроса до класу Запрос.
3. Повторіть етапи 1 і 2, створивши ще асоціації:
 - Від класу Запрос до класу ПозицияЗапросов;
 - Від класу ПозицияЗапросов до класу ЗапросБиблиотекарю.

4. Клацніть правою кнопкою миші на однонаправленій асоціації між класами ВыборЗапроса і Запрос, з боку класу ВыборЗапроса.
5. У меню, що відкрилося, виберіть пункт Multiplicity > Zero or One.
6. Клацніть правою кнопкою миші на іншому кінці однонаправленої асоціації.
7. У меню, що відкрилося, виберіть пункт Multiplicity > Zero or One.
8. Повторіть етапи 4 - 7, додавши на діаграму значення множинності для решти асоціацій, як показано на рисунку 5.1.

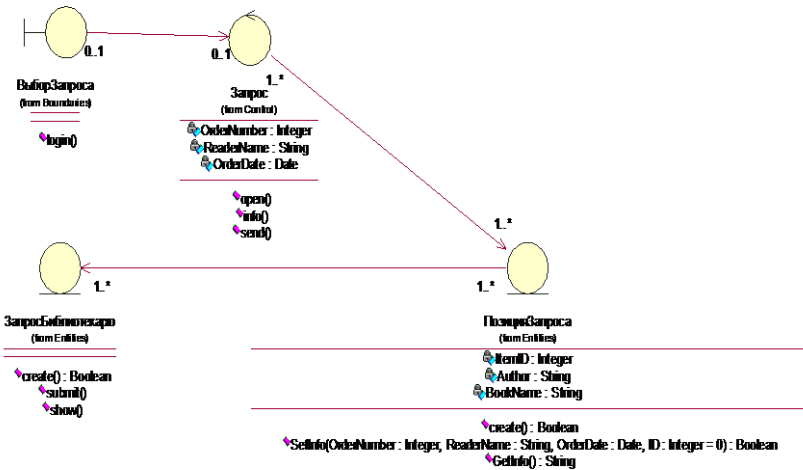


Рисунок 5.1 – Асоціації сценарію "Search a book"

5.4 Контрольні питання

1. Яким чином додати зв'язок до діаграми класів?
2. Що він (зв'язок) позначає?
3. Що таке «множинність»?
4. Що означає «множинність» у розрізі зв'язків між класами (саме для представленої діаграми)?

6 ЛАБОРАТОРНА РОБОТА №6 СТВОРЕННЯ ДІАГРАМИ СТАНІВ

6.1 Мета роботи

Створити діаграму Станів для класу ЗапросБиблиотекарю.

6.2 Постановка задачі

Щоб переконатися, що проект задовольняє всім вимогам необхідно створити діаграму станів для класу ЗапросБиблиотекарю для остаточного розуміння, як треба писати код для цього класу.

6.3 Завдання на виконання

Створення діаграми станів

Розробіть діаграму Станів для класу ЗапросБиблиотекарю, показану на рисунку 6.1.

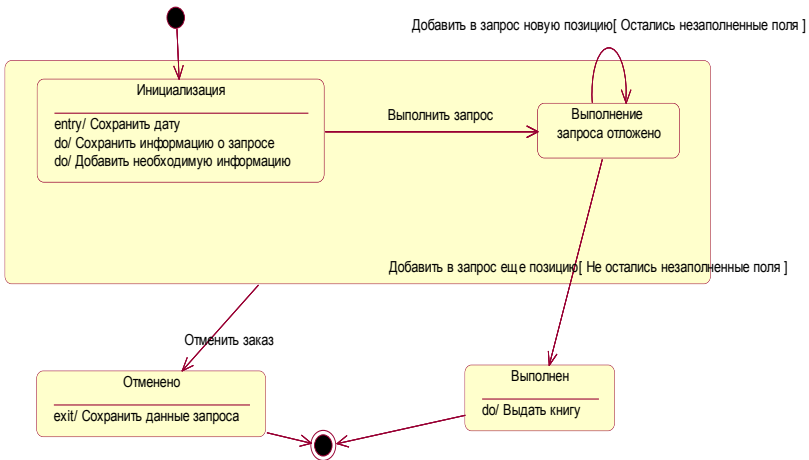


Рисунок 6.1 – Діаграма Станів для класу Order

Створення діаграми

1. Знайдіть в браузері клас ЗапросБиблиотекарю.
2. Клацніть на панелі управління на кнопці State Diagram.

3. У меню, що відкрилося, вкажіть ім'я «ЗапросБиблиотекарю».

Додавання початкового і кінцевого полягань

1. На панелі інструментів натискуйте кнопку Start State (Початковий стан).
2. Помістіть цей стан на діаграму.
3. На панелі інструментів натискуйте кнопку End State (Кінцевий стан).
4. Помістіть цей стан на діаграму.

Додавання суперстану

1. На панелі інструментів натискуйте кнопку State (Стан).
2. Помістіть цей стан на діаграму, поле з ім'ям залиште порожнім.

Додавання інших станів

1. На панелі інструментів натискуйте кнопку State.
2. Помістіть цей стан на діаграму.
3. Назвіть стан Отменено.
4. На панелі інструментів натискуйте кнопку State.
5. Помістіть цей стан на діаграму.
6. Назвіть стан Выполнено.
7. На панелі інструментів натискуйте кнопку State.
8. Помістіть цей стан на діаграму всередину суперстану.
9. Назвіть стан Инициализация.
10. На панелі інструментів натискуйте кнопку State.
11. Помістіть цей стан на діаграму всередину суперстану.
12. Назвіть стан Выполнение запроса отложено.

Докладний опис полягань

1. Двічі клацніть на стані Инициализация.
2. Перейдіть на вкладку Actions (Дії).
3. У меню, що відкрилося, виберіть пункт Insert (Вставити).
4. Двічі клацніть мишею на новій дії.
5. Назвіть його Сохранить дату.
6. Переконайтеся, що у вікні When (Коли) вказаний пункт On Entry (На вході).
7. Повторіть етапи 3 - 7, додавши наступні дії:

Зберегти інформацію про запит (Collect Customer Info), у вікні When вказати пункт Do

Додати необхідну інформацію (Add Order Items), у вікні When вказати Do

8. Натискуйте на кнопки ОК двічі, щоб закрити специфікацію.

9. Двічі клацніть на стані Отменено.

10. Повторіть етапи 2 - 7, додавши дію Сохранить данные запроса, вказати пункт On Exit (на виході).

11. Натискуйте на кнопки ОК двічі, щоб закрити специфікацію.

12. Двічі клацніть на стані Выполнено.

13. Повторіть етапи 2 - 7, додавши дію Выдать книгу, вказати пункт Do.

14. Натискуйте на кнопки ОК двічі, щоб закрити специфікацію.

Додавання переходів

1. На панелі інструментів натискуйте кнопку State Transition (Перехід).

2. Клацніть мишею на початковому стані.

3. Проведіть лінію переходу до стану Инициализация.

4. Повторіть етапи з першого по третій, створивши наступні переходи:

- від стану Инициализация до стану Выполнение запроса обложено;

- від стану Выполнение запроса обложено до стану Выполнено;

- від суперстану до стану Отложено;

- від стану Отложено до кінцевого стану;

- від стану Выполнено до кінцевого стану.

5. На панелі інструментів натискуйте кнопку Transition to Self (Перехід до себе).

Клацніть на стані Выполнение запроса отложено.

Докладний опис переходів

1. Двічі клацніть на переході від стану Инициализация до стану Выполнение запроса отложено, відкривши вікно його специфікації.

2. У полі Event (Подія) введіть фразу Виконати запит.

3. Клацніть на кнопці ОК, закривши вікно специфікації.

4. Повторіть етапи з першого по третій, додавши подію Отме-

нить заказ до переходу між суперстаном і станом Отменено.

5. Двічі клацніть на переході від стану Выполнение запроса отложено до стану Выполнено, відкривши вікно його специфікації.

6. У полі Event (Подія) введіть фразу Добавить в запрос еще позицию.

7. Перейдіть на вкладку Detail (Детально).

8. У полі Guard Condition (Умова) введіть “Не остались незаполненные поля”.

9. Клацніть на кнопці ОК, закривши вікно специфікації.

10. Двічі клацніть мишею на переході (Transition to Self) рефлексії стану Выполнение запроса отложено.

11. У полі Event (Подія) введіть фразу Добавить в запрос новую позицию.

12. Перейдіть на вкладку Detail (Детально).

13. У полі Guard Condition (Умова) введіть “Остались незаполненные поля”.

14. Клацніть на кнопці ОК, закривши вікно специфікації.

6.4 Контрольні питання

1. Які види переходів ви знаєте?
2. Що таке «суперстан»? Чим відрізняється «суперстан» від «стану»?
3. За допомогою якого інструменту та яким чином отримати на діаграмі «суперстан»?
4. Які функції виконує діаграма станів?

7 ЛАБОРАТОРНА РОБОТА №7 СТВОРЕННЯ ДІАГРАМИ КОМПОНЕНТІВ

7.1 Мета роботи

Створити діаграму Компонентів системи роботи бібліотеки. На даний момент вже визначені всі класи, що вимагаються для варіанту використання "Поиск книги". У міру реалізації інших варіантів використання на діаграму слід додавати нові компоненти.

7.2 Постановка задачі

Щоб переконатися, що проект задовольняє всім вимогам необхідно створити діаграму Станів для класу ЗапросБиблиотекарю для остаточного розуміння, як треба писати код для цього класу.

Завершивши аналіз і проектування системи необхідно розробити діаграми Компонентів. Вибравши мовою програмування - C++, для кожного класу необхідно створити відповідні цій мові компоненти.

7.3 Завдання на виконання

Створення пакетів компонентів

1. Клацніть правою кнопкою миші на представленні компонентах в браузері (Component View).
2. У меню, що відкрилося, виберіть пункт New > Package (Створити > пакет).
3. Назвіть цей пакет Entities (Сутності).
4. Повторіть етапи з першого по третій, створивши пакети Boundaries (Межі) і Control (Управління).

На рисунку 7.1 показана головна діаграма Компонентів всієї системи. Увагу на ній надається пакетам створюваних компонентів.

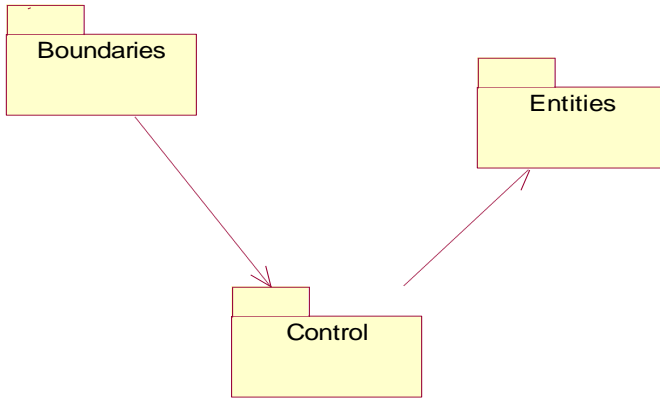


Рисунок 7.1 – Головна діаграма Компонентів системи

Малювання залежностей між пакетами

1. На панелі інструментів натискуйте кнопку Dependency (Залежність).
2. Клацніть мишею на пакеті Boundaries Головної діаграми Компонентів.
3. Проведіть лінію залежності до пакету Control.
4. Повторіть етапи 1 - 3, провівши ще залежність від пакету Control до пакету Entities.

Додавання компонентів до пакетів і малювання залежностей

1. Двічі клацніть мишею на пакеті Entities Головної діаграми Компонентів, відкривши Головну діаграму Компонентів цього пакету.
2. На панелі інструментів натискуйте кнопку Package Specification (Специфікація пакету).
3. Помістіть специфікацію пакету на діаграму.
4. Введіть ім'я специфікації пакету `ПозицияЗапроса`.
5. Повторіть етапи 2 - 4, додавши специфікацію пакету `ЗапросБиблиотекаря`.
6. На панелі інструментів натискуйте кнопку Package Body (Тіло пакету).
7. Помістіть його на діаграму.

8. Введіть ім'я тіла пакету `ПозицияЗапроса`.

9. Повторіть етапи 6 - 8, додавши тіло пакету `ЗапросБиблиотекарю`.

10. На панелі інструментів натискуйте кнопку `Dependency` (Залежність).

11. Клацніть мишею на тілі пакету `ПозицияЗапроса`.

12. Проведіть лінію залежності від нього до специфікації пакету `ПозицияЗапроса`.

13. Повторіть етапи 10 - 12, додавши лінію залежності між тілом пакету `ЗапросБиблиотекарю` і специфікацією пакету `ЗапросБиблиотекарю`.

14. Повторіть етапи 10 - 12, додавши лінію залежності від специфікації пакету `ЗапросБиблиотекарю` до специфікації пакету `ПозицияЗапроса`.

На рисунку 7.2 показані всі компоненти пакету `Entities`. Ці компоненти містять класи пакету `Entities` Логічного представлення системи.

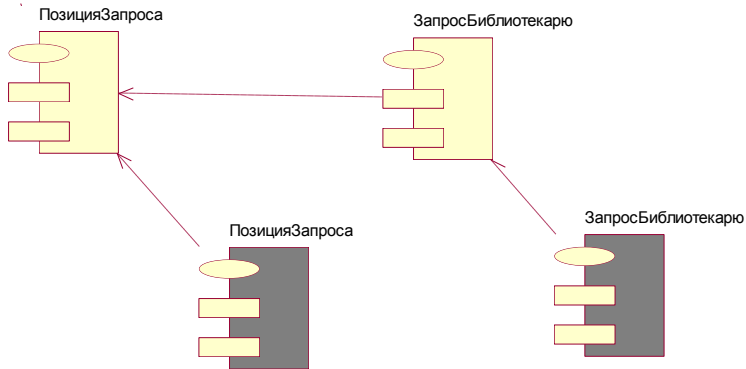


Рисунок 7.2 – Діаграма Компонентів пакету `Entities`

15. За допомогою описаного методу створіть наступні компоненти і залежності:

Для пакету `Boundaries`:

- специфікацію пакету `ВыборЗапроса`;
- тіло пакету `ВыборЗапроса`;

Залежності в пакеті Boundaries:

- від тіла пакету ВыборЗапроса до специфікації пакету ВыборЗапроса.

На рисунку 7.3 показані компоненти пакету Boundaries. Вони також відповідають класам однойменного пакету Логічного представлення системи.

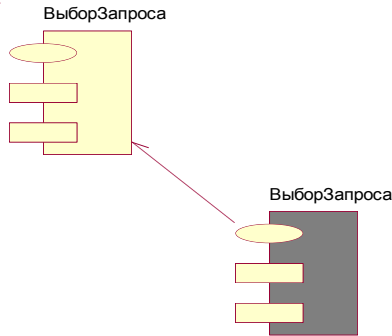


Рисунок 7.3 – Діаграма Компонентів пакету Boundaries

Для пакету Control:

- специфікацію пакету Запрос;
- тіло пакету Запрос.

Залежності в пакеті Control:

- від тіла пакету Запрос до специфікації пакету Запрос.

На рисунку 7.4 показані компоненти пакету Control. Вони містять класи пакету Control Логічного представлення системи.

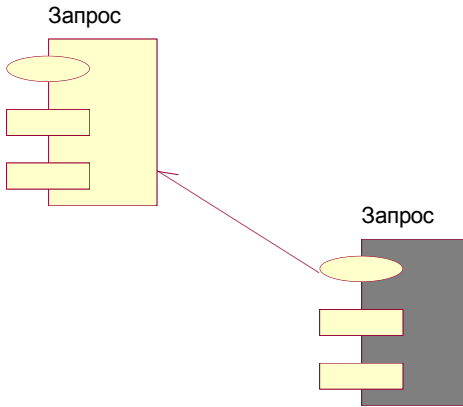


Рисунок 7.4 – Діаграма Компонентів пакету Control.

Створення діаграми Компонентів системи

1. Клацніть правою кнопкою миші на представленні Компонентів в браузері.
2. У меню, що відкрилося, виберіть пункт New > Component Diagram

3. Назвіть нову діаграму System.

4. Двічі клацніть на цій діаграмі.

Розміщення компонентів на діаграмі Компонентів системи

1. Якщо це було ще не зроблено, розверніть в браузері пакет компонентів Entities, щоб відкрити його.

2. Клацніть мишею на специфікації пакету ЗапросБиблиотекарю в пакеті компонентів Entities.

3. Перетягніть цю специфікацію на діаграму.

4. Повторіть етапи 2 і 3, помістивши на діаграму специфікацію пакету ПозицияЗапроса.

5. За допомогою цього методу помістите на діаграму наступні компоненти:

З пакету компонентів Boundaries:

- специфікацію пакету ВыборЗапроса.

З пакету компонентів Control:

- специфікацію пакету Запрос.

6. На панелі інструментів натискуйте кнопку Task Specification

(Специфікація задачі).

7. Помістіть специфікацію задачі на діаграму і назвіть її QueryReaderExe.

8. Повторіть етапи 6 і 7 для специфікації задачі QueryLibExe.

Додавання залежностей, що залишилися, на діаграму Компонентів системи

Вже існуючі залежності будуть автоматично показані на діаграмі Компонентів системи після додавання туди відповідних компонентів. Тепер треба додати решту залежностей.

1. На панелі інструментів натискуйте кнопку Dependency (Залежність).

2. Клацніть на специфікації пакету ВыборЗапроса.

3. Проведіть лінію залежності до специфікації пакету Запрос.

4. Повторіть етапи 1 - 3, створивши наступні залежності:

- від специфікації пакету Запрос до специфікації пакету ЗапросБиблиотекарю;

- від специфікації пакету Запрос до специфікації пакету ПозицияЗапроса;

- від специфікації задачі QueryReaderExe до специфікації пакету ВыборЗапроса;

- від специфікації задачі QueryLibExe до специфікації пакету Запрос.

Співвідношення класів з компонентами

1. У Логічному представленні браузера знайдіть клас ЗапросБиблиотекарю пакету Entities.

2. Перетягніть цей клас на специфікацію пакету компоненту ЗапросБиблиотекарю в представленні Компонентів браузера. В результаті клас ЗапросБиблиотекарю буде співвіднесений із специфікацією пакету компоненту ЗапросБиблиотекарю.

3. Перетягніть клас ЗапросБиблиотекарю на тіло пакету компоненту ЗапросБиблиотекарю в представленні Компонентів браузера. В результаті клас ЗапросБиблиотекарю буде співвіднесений з тілом пакету компоненту ЗапросБиблиотекарю.

4. Повторіть етапи 1 - 3, співвідношуючи з класами наступні компоненти:

- клас ПозицияЗапроса із специфікацією пакету ПозицияЗа-

проса;

- клас `ПозицияЗапроса` з тілом пакету `ПозицияЗапроса`;
 - клас `ВыборЗапроса` із специфікацією пакету `ВыборЗапроса`;
- са;
- клас `ВыборЗапроса` з тілом пакету `ВыборЗапроса`;
 - клас `Запрос` із специфікацією пакету `Запрос`;
 - клас `Запрос` з тілом пакету `Запрос`;

На рисунку 7.5 показані всі компоненти системи. Ми назвали цю діаграму діаграмою Компонентів системи. На ній ви можете бачити всі залежності між всіма компонентами проектованої системи.

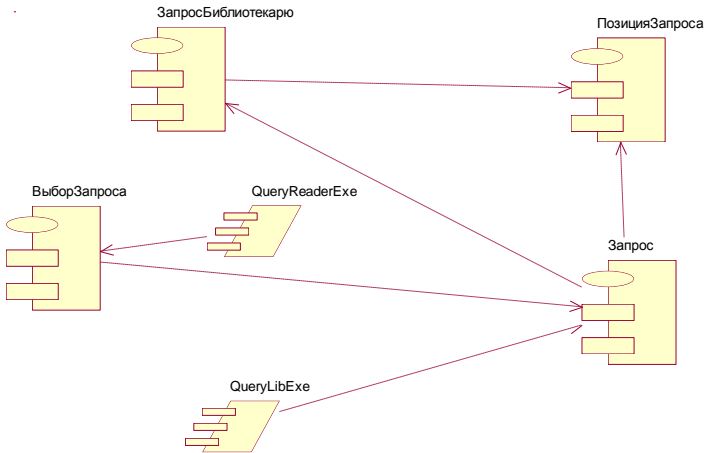


Рисунок 7.5 – Діаграма Компонентів системи

7.4 Контрольні питання

1. Що входить до складу діаграми «system»? До якого типу діаграм вона належить?
2. Чи можна провести зв'язок від компонента `Запрос` до компонента `ВыборЗапроса`? Чому?
3. З якою метою робиться співвідношення класів з компонентами системи?

8 ЛАБОРАТОРНА РОБОТА №8 СТВОРЕННЯ ДІАГРАМИ РОЗМІЩЕННЯ

8.1 Мета роботи

У цій лабораторній роботі буде створена діаграма Розміщення для системи роботи бібліотеки.

8.2 Постановка задачі

Варіанти використання, взаємодії між об'єктами і компоненти були вже чітко описані вище. Проте, для адміністрування мережі необхідно знати, на яких комп'ютерах розміщуватимуться різні компоненти системи. У зв'язку з цим необхідно розробити діаграму Розміщення для системи роботи бібліотеки.

Створення діаграми Розміщення

Розробіть діаграму Розміщення для системи обробки замовлень. Готова діаграма повинна виглядати як на рисунку 8.1.

8.3 Завдання на виконання

Додавання вузлів до діаграми Розміщення

1. Двічі клацніть мишею на представленні Розміщення в браузері, щоб відкрити діаграму Розміщення.
2. На панелі інструментів натискуйте кнопку Processor (Процесор).
3. Клацніть на діаграмі, помістивши туди процесор.
4. Введіть ім'я процесора «Сервер бази даних».
5. Повторіть етапи 2 - 4, додавши наступні процесори:
 - клиентская рабочая станция №1;
 - клиентская рабочая станция №2;
6. На панелі інструментів натискуйте кнопку Device (Пристрій).
7. Клацніть на діаграмі, помістивши на неї пристрій.
8. Назвіть його "Принтер".

Додавання зв'язків

1. На панелі інструментів натискуйте кнопку Connection (Зв'язок).
2. Клацніть на процесорі «Сервер бази даних».

3. Проведіть наступні зв'язки:

- від процесора «Сервер бази даних» до процесора " Клиентская рабочая станция №1";
- від процесора " Сервер бази даних " до процесора " Клиентская рабочая станция ";
- від процесора " Сервер бази даних " до пристрою "Принтер".

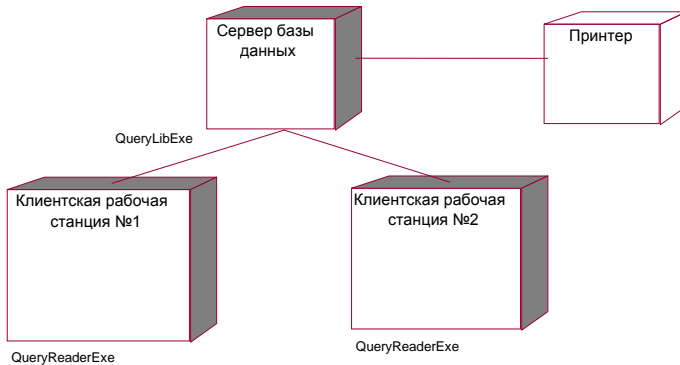


Рисунок 8.1 – Діаграма Розміщення для системи обробки замовлень

Додавання процесів

1. Клацніть правою кнопкою миші на процесорі " Сервер бази даних " в браузері.
2. У меню, що відкрилося, виберіть пункт New > Process (Створити > Процес).
3. Введіть ім'я процесу QueryLibExe.
4. Повторіть етапи 1 - 3, додавши ще процеси:
 - на процесорі " Клиентская рабочая станция №1 " -- процес QueryReaderExe;
 - на процесорі " Клиентская рабочая станция №2 " -- процес QueryReaderExe.

Відображення процесів на діаграмі

1. Клацніть правою кнопкою миші на процесорі « Сервер бази даних».

2. У меню, що відкрилося, виберіть пункт Show Processes (Показати процеси).

3. Повторіть етапи 1 і 2, показавши процеси на наступних процесорах:

- клиентская рабочая станция №1;
- клиентская рабочая станция №2.

8.4 Контрольні питання

1. Для чого потрібна діаграма розміщення?
2. Як створити процес?
3. Чи є дана діаграма основоположною для побудови програмного коду?
4. Які елементи можуть бути використані для побудови діаграми розміщення?

9 ЛАБОРАТОРНА РОБОТА №9 ГЕНЕРАЦІЯ КОДУ C++

9.1 Мета роботи

Згенерувати програмний код C++ для системи роботи бібліотеки.

9.2 Постановка задачі

У попередніх лабораторних роботах була створена модель для системи роботи бібліотеки. Тепер згенеруємо програмний код C++ для цієї системи. При цьому скористаємося діаграмою Компонентів системи, представленої на рисунку 9.1. Для генерації програмного коду необхідно виконати описані нижче кроки.

9.3 Завдання на виконання

Введення тіл пакетів на діаграму системи

1. Відкрийте діаграму Компонентів системи System.
2. Виберіть в браузері Entities: тіло пакету ЗапросБиблиотекарю.
3. "Перетягніть" тіло пакету ЗапросБиблиотекарю на діаграму Компонентів системи.
4. Повторіть пп. 2 і 3 для наступних компонентів:
 - Entities: тіло пакету ПозycjaЗапроса.
 - Boundaries: тіло пакету ВыборЗапроса.
 - Control: тіло пакету Запрос.

Встановлення мови C++

1. Відкрийте специфікацію компоненту ЗапросБиблиотекарю (специфікацію пакету) в пакеті компонентів Entities.
2. Виберіть мову ANSI C++.
3. Повторіть пп. 1 і 2 для наступних компонентів:
 - Entities: тіло пакету ЗапросБиблиотекарю;
 - Entities: специфікація пакету ПозycjaЗапроса;
 - Entities: тіло пакету ПозycjaЗапроса;
 - Boundaries: специфікація пакету ВыборЗапроса;
 - Boundaries: тіло пакету ВыборЗапроса;
 - Control: специфікація пакету Запрос;

- Control: тіло пакету Запрос;
- Специфікація задачі QueryReaderExe;
- Специфікація задачі QueryLibExe.

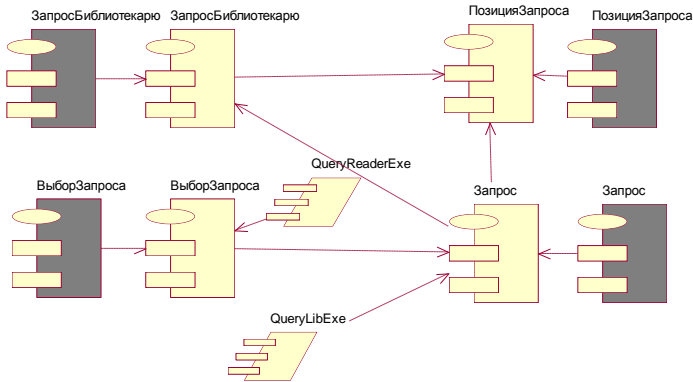


Рисунок 9.1 – Діаграма компонентів системи роботи бібліотеки

Генерація програмного коду C++

1. Відкрийте діаграму Компонентів системи.
2. Виберіть всі об'єкти на діаграмі Компонентів системи.
3. Виберіть Tools > C++ > Code Generation в меню для генерації програмного коду.
4. Поясніть отриманий програмний код.

9.4 Контрольні питання

1. Які файли отримані в результаті генерації коду?
2. Яка структура цих файлів (що вони вміщують)?
3. Чи є серед отриманих файлів такі, що мають однаковий вміст? Поясніть, чому?
4. Для яких систем доцільно застосовувати запропонований метод створення програмного коду?

10 ЛАБОРАТОРНА РОБОТА №10 ПРОЕКТУВАННЯ ВЛАСНОЇ СИСТЕМИ

10.1 Мета роботи

Побудувати діаграми та згенерувати програмний код C++ для системи, заданої відповідно до варіанту.

10.2 Постановка задачі

У попередніх лабораторних роботах була створена модель для системи роботи бібліотеки, та був згенерований програмний код C++ для цієї системи. У даній лабораторній роботі необхідно побудувати діаграми та згенерувати програмний код C++ для власної системи.

Відповідно до варіанту, вказаному у таблиці 10.1 побудувати для заданої системи діаграму використання, кооперативну діаграму, діаграму класів, діаграму переходу станів та діаграму компонентів системи; отримати програмний код для даної системи.

Варіант взяти рівним номеру комп'ютера, за яким закріплений студент.

Таблиця 10.1 – Вибір системи для проектування

| Номер варіанту | Назва системи для проектування |
|----------------|---|
| 1 | Магазин відеопрокату |
| 2 | Торговий автомат |
| 3 | Табло на станції метро |
| 4 | Система автоматизації для пункту прокату відеокасет |
| 5 | Міні-АТС |
| 6 | Система складського обліку |
| 7 | Система торгової компанії |
| 8 | Керування контактами з клієнтами |
| 9 | Банкомат |
| 10 | Система кредитування банку |
| 11 | Система табельного обліку |

| | |
|----|---|
| 12 | Web-сайт авіакомпанії |
| 13 | Система обліку товарів |
| 14 | Інтернет-магазин |
| 15 | WWW-конференція |
| 16 | Каталог ресурсів Інтернет |
| 17 | Система для введення інформації при прийнятті співробітника на роботу |
| 18 | Служба зайнятості в рамках вузу |
| 19 | Система підтримки складання розкладу занять |
| 20 | Система нарахування зарплатні |

Варіант 1. Магазин відеопрокату.

Відеомагазин має потребу в комп'ютерній системі. Асортименти магазину становить біля тисячі відеокaset і п'ятсот відеодисків. Запас уже замовлений в одного постачальника, однак, для майбутніх замовлень директор має намір користуватися послугами більшого числа постачальників. Всі відеокasetи й диски позначені штрих-кодом, таким чином, сканер, інтегрований у систему, може підтримувати операції видачі напрокат і повернення відеофільмів. Членські картки клієнтів також позначені штрих-кодом.

Клієнти мають можливість резервувати відео таким чином, щоб комплект відеофільмів був зібраний до певної дати. Система повинна мати пошуковий механізм для відповідей на запити клієнтів, включаючи питання, що стосуються фільмів, яких немає в асортименті магазину (але які він може замовити на прохання клієнта).

Для кожного фільму встановлений конкретний період прокату (обчислюваний у днях) з відповідною платою за прокат за цей період.

Відеомагазин повинен швидко дати відповідь на будь-який запит по наявності фільмів у запасі, а також кількості касет або дисків (поточні умови по кожній стрічці й диску повинні бути відомі й зафіксовані).

Плата за прокат відрізняється залежно від відеоносія: касета або диск.

Хоча магазин тримає в запасі відеодиски тільки одного формату - DVD, користувачі бажали б розширити в майбутньому систему про-

кату й на інші формати дисків.

Працівники відеомагазину прагнуть запам'ятати коди найбільш популярних стрічок. Найчастіше при ідентифікації фільму вони використовують саме код фільму, а не його назву (оскільки фільм з однією назвою міг випускатися різними режисерами).

Додаткові вимоги:

За касети й диски, повернуті пізніше строку, стягується додаткова плата за період, що перевищує строк прокату. Кожен відеоносій має унікальний ідентифікаційний номер.

Фільми замовляються в постачальника, що, у загальному випадку, може поставити касети й диски протягом одного тижня. Звичайно одне замовлення робиться на кілька фільмів.

Забронювати можна ті фільми, які замовлені в постачальника й/або всі копії яких перебувають у прокаті. Можна також бронювати ті фільми, яких немає в запасі і які не замовлені в постачальника; при цьому із клієнта потрібен задаток за один період прокату.

Клієнт може також зробити кілька попередніх замовлень, однак для кожного заброньованого фільму готується окремий запит на бронювання. Бронювання може бути скасовано. Через відсутність реакції з боку клієнта, більш точно, протягом одного тижня з моменту, коли клієнтові було повідомлено про можливість взяти фільм напрокат. Якщо за фільм був сплачений задаток, він записується на рахунок клієнта.

База даних зберігає звичайну інформацію про постачальників і клієнтів, тобто адреси, телефонні номери й т.д. У кожному замовленні постачальникові вказуються фільми, що замовляються, їхня кількість, формати касети/диска, а також дата очікуваної доставки, відпускна ціна, можливі знижки й т.д..

Коли касета повертається клієнтом або надходить від постачальника, спочатку виконуються попередні замовлення. Працівники магазину встановлюють контакт із клієнтами, що зробили попереднє замовлення. Для правильної обробки бронювання фільмів інформація, пов'язана із бронюванням, обновляється двічі: після встановлення контакту із клієнтом, коли йому повідомляється, що "заброньований фільм прийшов", і після здачі фільму клієнтові напрокат. Ці кроки га-

рантують правильне проведення операції бронювання.

Клієнт може взяти кілька касет або дисків, однак кожному взятому відеоносію ставиться у відповідність окремий запис. Для кожного видаваного напрокат фільму фіксуються дата й час видачі, установлений і фактичний строк повернення. Пізніше запис про прокат обновлюється, щоб відбити факт повернення відеофільму й факт остаточного платежу (або повернення грошей). Крім того, запис зберігає інформацію про продавця, відповідального за прокат фільму. Детальна інформація про клієнта й по прокату зберігається протягом року, щоб можна було легко визначити рівень довіри до клієнта. Стара інформація із прокату зберігається протягом року з метою проведення аудита.

Всі операції виконуються з використанням готівки, електронно-го переказу грошей або кредитних карток. Від клієнтів потрібно внести плату за прокат при видачі касет/дисків. Якщо касета/диск повернуті пізніше встановленого строку (або не можуть бути повернуті за якимись причинами), плата знімається або з рахунку клієнта, або приймається безпосередньо від клієнта.

Якщо касета/диск затримані більш ніж на два дні, клієнтові відправляється повідомлення про затримку. Після відправлення двох повідомлень про затримку однієї й тієї ж касети/диска, клієнт попереджається про те, що він є "порушником" і при наступному його зверненні до магазину керівництво розглядає питання про зняття з його статусу "порушника".

Варіант 2. Торговий автомат.

Потрібно розробити засобами Rational Rose модель програмного забезпечення убудованого процесора універсального торговельного автомата.

В автоматі є п'ять лотків для зберігання й видачі товарів. Завантаження товарів на лотки здійснюються обслуговуючим персоналом. Автомат стежить за наявністю товару. Якщо який-небудь товар розпроданий, автомат відправляє повідомлення про це на станцію обслуговування й інформує покупців (засвітиться червона лампочка поруч із лотком даного товару).

Автомат приймає до оплати паперові купюри й монети. Спеціа-

льний індикатор висвічує поточну суму грошей, прийнятих автоматом до оплати. Після уведення грошей клієнт натискає на кнопку видачі товару. Видача товару виробляється тільки в тому випадку, якщо уведена сума грошей відповідає ціні товару. Товар видається поштучно. При натисканні на кнопку "Повернення" клієнтові повертаються всі прийняті від нього до оплати гроші. Повернення грошей не відбувається після видачі товару. Автомат повинен коректно працювати при одночасному натисканні на кнопки видачі товару й повернення грошей.

У спеціальному відділенні автомата, що закривається на замок, є "секретна кнопка", що використовується обслуговуючим персоналом для виїмки виторгу. При натисканні на цю кнопку відкривається доступ до ящика із грішми.

Автомат одержує зі станції обслуговування дані про товари й зберігає їх у своїй пам'яті. Дані містять у собі ціну, найменування товару, номер лотка, на якому перебуває товар і кількість товару на лотку. Варіант завдання містить у собі розробку схеми бази даних про товари.

Варіант 3. Табло на станції метро.

Потрібно розробити засобами Rational Rose модель програмного забезпечення табло для інформаційної служби метрополітену.

Табло розташовані на кожній станції метро. Вони працюють під керуванням єдиного пункту керування (ПК) інформаційної служби метро. Табло відображає поточний час (години, хвилини, секунди) і час, що пройшов з моменту відправлення останнього поїзда (хвилини, секунди). Момент прибуття й відправлення поїзда визначається за допомогою датчиків, установлених на шляхах. Всі табло метро синхронізовані, поточний час веде відлік й установлюється із центральної служби часу, що перебуває на ПК.

На табло висвічується кінцева станція призначення поїзда, що прибуває. Ці дані містяться у розкладі руху поїздів, що зберігається в пам'яті табло й періодично оновлюється з ПК.

В рядку, що біжить, на табло відображається рекламна інформація. Пам'ять табло зберігає до 10 рекламних повідомлень. Повідом-

лення відображаються один за одним з невеликими паузами, циклічно. Зміст рекламних повідомлень надходить із ПК.

Додаткова функція табло - по запиту з ПК воно пересилає дані про порушення розкладу (передчасних відправленнях поїздів або запізненнях).

У ході виконання завдання повинна бути створена схема бази даних для зберігання рекламних повідомлень, розкладу й відомостей про порушення розкладів.

Пояснення: у завданні потрібно розробити модель ПЗ тільки для табло, але не для пункту керування інформаційної служби.

Варіант 4. Система автоматизації для пункту прокату відеокасет.

Потрібно розробити засобами Rational Rose модель програмної системи автоматизації роботи пункту прокату відеокасет (далі в тексті - системи).

Пункт прокату містить каталог касет, що є в наявності в цей момент часу. Система підтримує роботу каталогу, дозволяючи працівникам прокату додавати нові найменування касет, видаляти старі й редагувати дані про касети.

Клієнт, що звернувся в пункт, вибирає касету по каталогу, вносить заставу й забирає її на певний строк. Строк прокату, вимірюваний у днях, обирається при видачі касети. Вартість прокату обчислюється системою, виходячи з тарифу за добу й строку прокату. Клієнт повертає касету й оплачує прокат. Якщо касета не ушкоджена, клієнтові повертається застава. Працівник пункту прокату реєструє здачу касети клієнтом і її повернення в системі. Якщо клієнт пошкодив касету, то касета видаляється з каталогу, а застава залишається в касі прокату.

При необхідності працівник може запросити у системи наступні дані:

- чи є в наявності касета з даною назвою;
- коли буде повернута будь-яка касета з тих, що здано в прокат;
- чи є даний клієнт постійним клієнтом пункту прокату (чи

користувався він прокатом 5 або більше разів).

Постійним клієнтам надаються знижки, а також від них приймаються заявки на поповнення асортименту касет. Заявки реєструються в системі. По них готується підсумковий звіт, керуючись яким, працівники пункту прокату обновляють асортименти касет.

Необхідно розробити схему бази даних для зберігання каталогу, облікових записів про прокат касет і заявок на поповнення асортименту.

Варіант 5. Міні-АТС.

Потрібно розробити засобами Rational Rose модель програмного забезпечення убудованого мікропроцесора міні-АТС (автоматичної телефонної станції) установи.

Міні-АТС здійснює зв'язок між службовцями установи. Кожен абонент підключений до неї лінією зв'язку. Міні-АТС сполучає лінії абонентів (здійснює комутацію ліній). Абоненти мають номери, що складаються із трьох цифр. Спеціальний номер "9" зарезервований для зовнішнього зв'язку.

Телефонне з'єднання абонентів відбувається в такий спосіб. Абонент піднімає трубку телефону, і міні-АТС отримує сигнал "Трубка". У відповідь міні-АТС посилає сигнал "Тон". Приймавши цей сигнал, абонент набирає телефонний номер (посилає три сигнали "Цифра"). Міні-АТС перевіряє готовність абонента, якого викликають. Якщо абонент не готовий (його лінія зайнята), міні-АТС посилає абонентові, що викликає, сигнал "Зайнято". Якщо абонент готовий, міні-АТС посилає обом абонентам сигнал "Виклик". При цьому телефон абонента, якого викликають, починає дзвонити, а абонент, що викликає, чує в трубці довгі гудки. Абонент, до якого телефонують, знімає трубку, і міні-АТС одержує від нього сигнал "Трубка", після чого здійснює комутацію лінії. Абоненти обмінюються сигналами "Дані", які міні-АТС повинна передавати від одного абонента до іншого. Коли один з абонентів опускає трубку, міні-АТС одержує сигнал "Кінець" і посилає іншому абонентові сигнал "Тон".

У будь-який момент абонент може покласти трубку, при цьому міні-АТС одержує сигнал "Кінець". Після одержання цього сигналу

сеанс обслуговування абонента завершується.

Якщо абонент бажає з'єднатися з абонентом за межами установи, то він набирає номер "9". Міні-АТС посилає по лінії, що з'єднує із зовнішньою (міською) АТС, сигнал "Трубка" і надалі служить посередником між телефоном абонента й зовнішньою АТС. Вона приймає й передає сигнали й дані між ними, не вносячи ніяких змін. Єдине виключення стосується завершення сеансу. Одержавши від міської АТС сигнал "Кінець", міні-АТС посилає абонентові сигнал "Тон" і чекає сигналу "Кінець" для завершення обслуговування абонента. Якщо абонент, що викликав, першим вішає трубку, то міні-АТС одержує сигнал "Кінець" і передає його міській АТС і завершує сеанс.

Міні-АТС може одержати сигнал "Виклик" від міської АТС. Це відбувається, коли немає з'єднань із зовнішніми абонентами. Сигнал "Виклик" від міської АТС передається абонентові з кодом "000". Тільки цей абонент може відповідати на зовнішні дзвінки.

Варіант 6. Система складського обліку.

Система складського обліку - програмна система, що включає всі аспекти, пов'язані з рухом товару на склад і зі складу. За результатами аналізу можна виділити сім основних функцій системи:

1. Облік замовлень. Прийом замовлень від клієнтів і відповіді на запити клієнтів про стан замовлень.

2. Ведення рахунків. Напрямок рахунків клієнтам і відстеження платежів. Прийом рахунків від постачальників і відстеження платежів постачальникам.

3. Відвантаження зі складу. Складання специфікацій на комплектацію товарів, що відправляються зі складу клієнтам.

4. Складський облік. Постановка товарів, що прибувають, на облік і зняття товарів з обліку при відправленні замовлень.

5. Закупівлі. Замовлення товарів постачальникам і відстеження поставок.

6. Прийом товарів. Прийняття на склад товарів від постачальників.

7. Планування. Випуск звітів, у тому числі тенденції, що відбивають попит на окремі види товарів й активність постачальників.

Стратегія по проникненню компанії, що займається торгівлею по каталогах, на нові ділянки ринку, вимагає створення ряду відносно автономних регіональних складів продукції. Кожен такий склад відповідає за облік товарів і виконання замовлень. З метою підвищення ефективності своєї роботи склад зобов'язаний сам підтримувати ту номенклатуру товарів, що у найкращим чином відповідає потребам місцевого ринку. Номенклатура може бути різною для кожного регіону. Крім того, номенклатура повинна оперативнo змінюватися відповідно до мінливих потреб клієнтів. Головна компанія хотіла б мати на всіх складах однакові системи обліку.

Основними функціями системи є:

1. Облік товарів, що приходять від різних постачальників, при їхньому прийомі на склад.
2. Облік замовлень у міру їхнього надходження із центральної віддаленої організації; замовлення також можуть прийматися поштою. Їхня обробка ведеться на місцях.
3. Генерація вказівок персоналу, зокрема, про упакування товарів.
4. Генерація рахунків і відстеження оплат.
5. Генерація запитів про поставку й відстеження платежів постачальникам.

Крім автоматизації стандартних складських операцій, система також повинна надавати широкі можливості по генерації різних форм звітності, у тому числі тенденції, що відбивають розвиток ринку, списки найбільш надійних і ненадійних постачальників і клієнтів, матеріали для рекламних компаній.

Варіант 7. Система торговельної компанії.

Компанія - торговельний посередник - продає товари різних виробників. Для забезпечення своєї діяльності вона має потребу в програмній системі обробки замовлень.

Двічі на рік компанія публікує каталог продуктів, який розсилається клієнтам й іншим зацікавленим особам.

Клієнти отримують товари, направляючи компанії перелік продуктів з інформацією про оплату. Компанія виконує замовлення й від-

правляє товари за адресами клієнтів.

Система повинна відслідковувати замовлення від моменту його одержання до відправлення товару.

Клієнти можуть повертати товари, можливо, оплачуючи при цьому деякі витрати.

Деякі клієнти користуються замовленням товарів через Інтернет.

Компанія користується послугами різних транспортних і страхових компаній.

Варіант 8. Керування контактами з клієнтами.

Компанія, що займається дослідженням ринку, має стабільну клієнтську базу організацій, які купують звіти з аналізу ринку. Деякі крупні клієнти купують у компанії також спеціалізоване ПЗ, призначене для створення звітів. Цих клієнтів компанія також забезпечує неопрацьованою й агрегованою інформацією для генерації їхніх власних звітів.

Нова система керування контактами повинна перебувати в розпорядженні всіх працівників компанії, але з наданням різного рівня доступу. Система повинна забезпечити гнучке планування й перепланування видів діяльності, пов'язаних з контактами.

Система підтримує функції "постійного контакту" з наявною й потенційною клієнтською базою так, щоб реагувати на її потреби й одержувати нові контракти на придбання товарів.

Система зберігає імена, номери телефонів, звичайні поштові й кур'єрські адреси й т.д. організацій і контактних осіб у цих організаціях.

Система дозволяє співробітникам планувати завдання й заходи, які необхідно провести відносно контактних осіб. Співробітники планують завдання й заходи для інших співробітників або для себе.

Завдання - це група заходів, які здійснюються для досягнення певного результату. Результатом може бути перетворення потенційного клієнта в клієнта, організація доставки товару або вирішення проблеми клієнта. До звичайних типів заходів ставляться телефонний дзвінок, візит, відправлення факсу, пристрій навчання й т.д..

Клієнт розглядається як наявний, якщо існує контракт із цим клієнтом на поставку товарів або послуг. Однак, функції керування контрактами виходять за рамки системи.

Система дозволяє виробляти різні звіти по контактах на основі поштової й кур'єрської адрес (наприклад, знаходити всіх клієнтів по поштовому індексу). Дата й час створення завдання фіксуються. Можна також зберегти значення "доходу", очікуваного від здійснення завдання.

Заходи для співробітника відображаються на екрані його комп'ютера у вигляді сторінки календаря (один день на сторінку). Пріоритет кожного заходу (низький, середній, високий) візуально виділяється на екрані.

Не з усіма заходами зв'язане поняття "строк виконання" - деякі з них є "безстроковими" (вони можуть виконуватися в будь-який час протягом дня, на який вони заплановані).

Час створення заходу не може змінюватися, а строк виконання - може. По завершенні заходу дата й час його завершення фіксуються. Система також зберігає відмітні ознаки для співробітників, які створюють завдання, й заходи, які заплановано здійснити ("доручено співробітникам"), а також завершені заходи.

Варіант 9. Банкомат.

Потрібно розробити засобами Rational Rose модель програмного забезпечення банкомата. Банкомат - це автомат для видачі готівки по кредитних пластикових картках. У його склад входять наступні пристрої: дисплей, панель керування із кнопками, приймач кредитних карток, сховище грошей і лоток для їхньої видачі, сховище конфіскованих кредитних карток, принтер для друку довідок.

Банкомат підключений до лінії зв'язку для обміну даних з банківським комп'ютером, що зберігає відомості про рахунки клієнтів.

Обслуговування клієнта починається з моменту поміщення пластикової картки в банкомат. Після розпізнавання типу пластикової картки, банкомат видає на дисплей запрошення ввести персональний код. Персональний код являє собою чотиризначне число. Потім банкомат перевіряє правильність уведеного коду. Якщо код введено не-

правильно, користувачеві надаються ще дві спроби для уведення правильного коду. У випадку повторних невдач картка переміщається в сховище карток, і сеанс обслуговування закінчується. Після уведення правильного коду банкомат пропонує користувачеві вибрати операцію. Клієнт може або зняти наявні з рахунку, або довідатися про залишок на його рахунку.

При знятті наявних з рахунку банкомат пропонує вказати суму (5, 10, 20, 50, 100, 200, 500 гривень). Після вибору клієнтом суми банкомат запитує, потрібно чи друкувати довідку по операції. Потім банкомат надсилає запит на зняття обраної суми центральному комп'ютеру банку. У випадку одержання дозволу на операцію, банкомат перевіряє, чи є необхідна сума в його сховищі грошей. Якщо він може видати гроші, то на дисплей виводиться повідомлення "Вийміть карту". Після видалення картки із приймача, банкомат видає зазначену суму в лоток видачі. Банкомат друкує довідку по зробленій операції, якщо вона була замовлена клієнтом.

Якщо клієнт хоче довідатися про залишок на рахунку, то банкомат надсилає запит центральному комп'ютеру банку й виводить суму на дисплей. На вимогу клієнта друкується й видається відповідна довідка.

У спеціальному відділенні банкомата, що закривається на замок, є "секретна кнопка", що використовується обслуговуючим персоналом для завантаження грошей. При натисканні на цю кнопку відкривається доступ до сховища грошей і конфіскованих кредитних карток.

Варіант 10. Система кредитування банку.

Невеликий банк автоматизує діяльність, пов'язану із кредитуванням фізичних й юридичних осіб (індивідуальних клієнтів й організацій).

У цей час кандидат на одержання кредиту заповнює паперову форму, додає необхідні документи (фінансовий звіт, перспективну оцінку фінансового стану й ін.) і відправляє до банку. Референт по кредитуванню аналізує запит на предмет можливих помилок і підтверджує його достовірність.

Потім референт замовляє звіт про кредитні операції клієнта у

відділі кредитування. Копія звіту переглядається службовцем банку, а референт перевіряє фінансове становище й доходи клієнта. Службовець також звертається до існуючої системи керування рахунками клієнтів, щоб одержати необхідну інформацію про стан рахунку й попередніх кредитів клієнта.

Вся інформація комплектується в кредитний запит і направляється для оцінки інспекторові по кредитах. Якщо запит затверджується, інспектор визначає найкращі умови кредитування й повідомляє про це клієнта. Якщо клієнт приймає умови, то кредит оформляється.

На обробку запиту звичайно витрачається мінімум два тижні (як для індивідуальних клієнтів, так і для організацій). Мета автоматизації - скоротити час обробки запиту до 48 годин для індивідуальних клієнтів й до 72 годин для організацій, скоротити кількість співробітників, зайнятих у процесі обробки й збільшити кількість запитів, оброблюваних у заданий період часу.

Варіант 11. Система табельного обліку.

Завдання контролю робочого часу, виконуване відділом кадрів, включає збір видрукованих табелів, перевірку повноти й правильності заповнення табелів, одержання при необхідності підтвердження від менеджера й уведення даних табельних карток у діючу систему розрахунку заробітної плати, для підготовки друку платіжної відомості. Крім того, штат відділу кадрів повинен підтримувати коди зарплати (звичайний час, понаднормовий час, відпустка через хворобу й т.д.) і поширювати список кодів серед працівників компанії при кожній його зміні.

Ця робота у свою чергу впливає на здатність відділу кадрів виконувати інші обов'язки, у тому числі готувати звіти по зарплаті, займатися посібниками й виконувати функції найму, переміщення й звільнення. Успішне вирішення проблеми повинно скоротити час, який відділ кадрів витрачає на завдання, пов'язані з табельним обліком, на 60% і більше.

Зацікавленими особами в цьому проекті є працівники компанії, що мають наступні посадові обов'язки (див. табл.10.2).

Таблиця 10.2 - Працівники та їх обов'язки

| Зацікавлена особа | Обов'язки |
|------------------------------|---|
| Адміністратор відділу кадрів | Вести облік службовців. Розглядати й заносити дані табелів службовців. Підтримувати й розподіляти коди зарплатні. Готувати звіти про зарплатню. Звіряти дані про зарплатню. Управляти посібниками. Консультувати службовців з питань посібників. Спостерігати за процесом найму й відповідним обліком. Спостерігати за переміщенням і відповідним обліком. Спостерігати за процесом звільнення й відповідним обліком. Реагувати на скарги службовців. |
| Менеджер відділу кадрів | Управляти відділом кадрів. |
| Службовці | Надавати відділу кадрів інформацію, пов'язану з працівниками і їхнім робочим часом. |

Відомості про користувачів системи і їхні обов'язки наведено у таблиці 10.3.

Нова система повинна надавати наступні функціональні можливості:

- для службовця - представляти й переглядати інформацію свого табеля;
- для менеджера - затверджувати табелі його службовців;
- для адміністратора відділу кадрів - контролювати правиль-

ність табелів і вносити необхідні уточнення;

- для адміністратора відділу кадрів - підтримувати коди зарплатні.

Таблиця 10.3 – Користувачі системи та їх обов'язки

| Користувач | Обов'язки |
|------------------------------|---|
| Адміністратор відділу кадрів | Розглядати й корегувати табельні картки. Підтримувати коди зарплати. |
| Менеджер відділу кадрів | Затверджувати особливі типи корегування табельних карток. |
| Службовці | Вводити дані табельних карток. |
| Менеджери | Підтверджувати табельні картки службовців. |

Варіант 12. Web-сайт авіакомпанії.

Комерційний відділ авіакомпанії запропонував розширити свій Web-сайт, щоб дозволити користувачам:

- довідатися про виконання рейсів на поточний день;
- запросити інформацію про розклад рейсів, вартість квитків і наявність місць;
- купити квитки.

Постійні клієнти, які часто літають літаками авіакомпанії, зможуть використати також наступні функції:

- одержати поточну інформацію про стан свого особистого рахунку (кількість кілометрів, проведених у повітрі з початку року на дане число, кількість налітаних кілометрів для одержання заохочувальної винагороди (безкоштовного перельоту) і т.д. ;
- купити квитки, використовуючи або інформацію про налітані кілометри (для постійних клієнтів), або кредитну картку.

Щоб гарантувати таємницю приватної інформації й запобігти несанкціонованому використанню даних про постійних клієнтів, при доступі до особистих рахунків необхідно видати запит про реєстрацію користувача, для чого треба ввести номер рахунку й особистий ідентифікаційний номер власника картки (PIN). Після реєстрації користу-

вач повинен побачити початкову сторінку з урахуванням його вподобань і звичок, взятих із бази даних, що зберігає інформацію про перельоти постійних клієнтів. Постійні клієнти можуть оперативно обновляти відомості про себе.

Щоб заощадити гроші, керівництво компанії прийняло рішення використати ряд існуючих систем:

- систему керування рахунками, що зберігає інформацію про постійних клієнтів і баланс "преміальних кілометрів";
- маркетингову базу даних, що відслідковує дані про виконані рейси, клас оплати й ін. (Ці дані використовуються для формування спеціальних повідомлень, які включаються в щомісячні виписки з особового рахунку постійних клієнтів.);
- базу даних тарифів;
- базу даних наявності квитків.

Варіант 13. Система обліку товарів.

Потрібно розробити засобами Rational Rose модель системи підтримки замовлення й обліку товарів у бакалійній крамниці.

У бакалійній крамниці для кожного товару фіксується місце зберігання (певна полиця), кількість товару і його постачальник. Система підтримки замовлення й обліку товарів повинна забезпечувати додавання інформації про новий товар, зміну або видалення інформації про наявний товар, зберігання (додавання, зміну й видалення) інформації про постачальників, що включає в себе назву фірми, її адресу й телефон. За допомогою системи робляться замовлення постачальникам. Кожне замовлення може містити кілька позицій, у кожній позиції вказується найменування товару і його кількість у замовленні. Система обліку на вимогу користувача формує й готує до друку наступну довідкову інформацію:

- список всіх товарів;
- список товарів, що є в наявності;
- список товарів, кількість яких необхідно поповнити;
- список товарів, що постачаються даним постачальником.

У ході виконання цього варіанта завдання повинна бути розроблена схема бази даних, що зберігає інформацію про товари, замовлен-

ня й постачальників.

Варіант 14. Інтернет-магазин.

Потрібно розробити засобами Rational Rose модель програмного забезпечення Інтернет-магазину.

Виробник комп'ютерів пропонує можливість придбання своєї продукції через Internet. Клієнт може вибрати комп'ютер на Web-сторінці виробника. Комп'ютери підрозділяються на сервери, настільні й портативні. Замовник може вибрати стандартну конфігурацію або побудувати необхідну конфігурацію в діалоговому режимі. Компоненти конфігурації (такі, як оперативна пам'ять) представляються як список для вибору з доступних альтернатив. Для кожної нової конфігурації система може підрахувати ціну.

Щоб оформити замовлення, клієнт повинен заповнити інформацію для доставки й оплати. Як платіжні засоби допускається використання кредитних карток або чеків. Після уведення замовлення система відправляє клієнтові по електронній пошті повідомлення з підтвердженням одержання замовлення. Поки клієнт очікує прибуття комп'ютера, він може перевірити стан замовлення в будь-який час у діалоговому режимі. Серверна частина обробки замовлення складається із завдань, необхідних для перевірки кредитоспроможності й способу розрахунку клієнта за покупку, отримання замовленої конфігурації зі складу, друку рахунку й подачі заявки на склад про доставку комп'ютера клієнтові.

Додаткові вимоги:

Для знайомства зі стандартною конфігурацією обираного сервера, настільного або портативного комп'ютера клієнт використовує Web-сторінку Internet-магазину. При цьому також вказується ціна конфігурації. Клієнт вибирає деталі конфігурації, з якими він хоче познайомитися, можливо, з наміром купити готову або скласти більш підходящу конфігурацію. Ціна для кожної конфігурації може бути підрахована на вимогу користувача.

Клієнт може вибрати варіант замовлення комп'ютера по Internet або попросити, щоб продавець зв'язався з ним для пояснення деталей замовлення, домовився про ціну й т.п. перш, ніж замовлення буде фа-

ктично розміщене.

Для розміщення замовлення клієнт повинен заповнити електронну форму з адресами для доставки товару й відправлення рахунку-фактури, а також деталями, що стосуються оплати (кредитна картка або чек).

Після уведення замовлення клієнта в систему продавець відправляє на склад електронна вимогу, що містить деталі замовленої конфігурації.

Деталі угоди, включаючи номер замовлення, номер рахунку клієнта, відправляються електронною поштою клієнтові, так що замовник може перевірити стан замовлення через Internet.

Склад одержує рахунок-фактуру від продавця й відвантажує комп'ютер клієнтові.

При виконанні цього варіанта завдання рекомендуємо ознайомитися з роботою Коналлен-2001. Варто визначитися, по якому архітектурному шаблону буде будуватися Web-додаток ("тонкий клієнт" або "товстий клієнт"). Відповідно до обраного шаблону варто побудувати моделі клієнтської частини магазину й серверної частини, промодельовати зв'язки між частинами додатка. Для Web-додатків типовими є наступні класи:

- клієнтська Web-сторінка;
- серверна Web-сторінка (наприклад, CGI-скрипт);
- HTML-форма;
- об'єкт JavaScript.
- Додаткові зв'язки між класами Web-додатків:
- link - посилання з однієї сторінки на іншу;
- build - зв'язок між CGI-скриптом і клієнтською сторінкою, що генерується при його виконанні;
- submit - зв'язок між формою й серверною Web-сторінкою, що приймає дані з форми.
- Типові компоненти:
- Web-сторінка (HTML-файл),
- Active Server Page (ASP),
- Java Server Page (JSP),
- сервлет,

- бібліотека скриптів (наприклад, що підключає файл із Javascript-функціями).

Варіант 15. WWW-конференція.

Потрібно розробити засобами Rational Rose модель програмного забезпечення WWW-конференції.

WWW-конференція являє собою сховище повідомлень у мережі Інтернет, доступ до якого здійснюється за допомогою браузера. Для кожного повідомлення конференції зберігаються значення наступних полів: номер повідомлення, автор, тема, текст повідомлення, дата додавання повідомлення, посилання на батьківське повідомлення. Початковою сторінкою конференції є ієрархічний список повідомлень. Верхній рівень ієрархії становлять повідомлення, що відкривають нові теми, а підрівні становлять повідомлення, отримані у відповідь на повідомлення верхнього рівня. Повідомлення-відповідь завжди має посилання на вихідне повідомлення. У списку відображаються тільки теми повідомлень, їхні автори й дати додавання. Переглядаючи список, користувач вибирає повідомлення й по гіперпосиланню відкриває сторінку з текстом повідомлення. Крім тексту на цій сторінці відображається список (ієрархічний) повідомлень, що є відповідями, відповідями на відповіді й т.д. Для зручності користувачів необхідно передбачити пошук повідомлень по автору або по ключових словах у темі або тексті повідомлення.

Повідомлення додаються в конференцію зареєстрованими користувачами, які при відправленні повідомлення повинні вказати своє ім'я й пароль. Реєструє нових користувачів модератор конференції - її ведучий. При реєстрації користувач заповнює спеціальну форму, уміст якої потім пересилається модератору й запам'ятовується в базі користувачів. Модератор вирішує, реєструвати чи користувача ні, і відправляє свою відповідь.

При додаванні повідомлень користувач має можливість почати нову тему або відповісти на раніше додані повідомлення. Після додавання повідомлення воно доступно для читання всім користувачам (навіть незареєстрованим), і список повідомлень обновляється.

Модератор має право з тих чи інших причин видаляти повідом-

лення будь-яких авторів. Він також може карати користувачів, що порушують правила поведінки в конференції, лишаючи на певний час користувача можливості додавати й редагувати повідомлення.

Варіант завдання містить у собі розробку схеми бази даних для зберігання повідомлень конференції й інформації про її учасників.

Виконуючи це завдання корисно ознайомитися із заключним зауваженням до варіанта 14 "Інтернет-магазин". Найбільш підходящою архітектурою для WWW-конференції є "тонкий клієнт", оскільки клієнтська частина практично не містить "бізнес-логіки". Єдиним її елементом, що може виконуватися на стороні клієнта, є перевірка правильного заповнення полів форми, перед відправленням її вмісту на сервер.

Варіант 16. Каталог ресурсів Інтернет.

Потрібно розробити засобами Rational Rose модель програмного забезпечення каталогу ресурсів мережі Інтернет.

У каталозі зберігається наступна інформація про ресурси: назва ресурсу, унікальний локатор ресурсу (URL), розділ каталогу, у якому міститься ресурс, список ключових слів, короткий опис, дата останнього відновлення, контактна інформація.

Доступ користувачів до каталогу здійснюється за допомогою браузера. Користувачі каталогу можуть додавати нові ресурси, інформація про які не була внесена раніше. Ресурси в каталозі класифікуються за розділами. Повний список ресурсів кожного розділу повинен бути доступний користувачам. Користувачам каталогу повинні бути надані можливості з пошуку ресурсів. Пошук здійснюється по ключових словах. Якщо користувач не задоволений результатами пошуку, він може уточнити запит (здійснити пошук серед результатів попереднього пошуку). Повинна бути можливість видавати результати пошуку в різній формі (висновок всієї інформації про ресурси або часткової). Користувач може відсортувати список ресурсів за релевантністю (відповідністю ключовим словам із запиту) або за датою оновлення.

Оскільки зміст ресурсів Інтернет з часом змінюється необхідно стежити за датою останнього оновлення, періодично опитуючи Web-

сайти, URL яких зберігаються в каталозі.

Варіант завдання містить у собі розробку схеми бази даних для зберігання повідомлень конференції й інформації про її учасників.

Виконуючи це завдання, корисно ознайомитися із заключним зауваженням до варіанта 14 "Інтернет-магазин". Як й у варіанті 15 "WWW-конференція" найбільш підходящою архітектурою для каталогу є "тонкий клієнт", оскільки клієнтська частина практично не містить у собі функцій "бізнес-логіки" крім перевірки вмісту форм перед пересиланням на сервер.

Варіант 17. Система для введення інформації при прийнятті співробітника на роботу.

Підрозділ/кафедра вибирається шляхом позиціонування на певному (потрібному) рядку. Підрозділи змінюються порівняно часто й тому їхні назви й коди зберігаються у відповідних довідниках.

Посади також вибираються у відповідності зі штатним розкладом. Якщо відповідні посади вже зайняті, то уведення не може бути здійснене.

Кожному співробітникові відповідно до наявних обмежень повинен бути привласнений тарифний розряд.

Поряд з іншими відомостями, вводиться інформація про знання іноземних мов. Мови вибираються зі списку. Вибір може бути множинним (тобто співробітник може володіти більш ніж однією мовою). Всі обрані позиції повинні бути видні; користувач може коректувати свій вибір перед тим, як остаточно занести дані до БД. Крім назви мови, фіксується ще й ступінь володіння мовою.

Забороняється приймати на роботу осіб пенсійного віку.

При прийомі на викладацьку посаду освіта повинна бути вищою.

При прийомі на посаду вище асистента науково-педагогічний стаж повинен бути більше 3 років. Науково-педагогічний стаж не може бути більше загального стажу.

Після занесення всієї інформації про співробітника до БД кожному співробітникові автоматично привласнюється табельний номер.

Варіант 18. Служба зайнятості в межах вузу.

Система призначена для того, щоб допомогти студентові влаштуватися на роботу вже в процесі його навчання у вузі. Подавши заявку в систему, студент стає її клієнтом і починає обслуговуватися протягом усього навчання у вузі. Заява являє собою анкету. Система пропонує професійні (засновані на досліджуваних предметах), психологічні тестування, проведені регулярно (раз у семестр (півроку)). Особлива увага приділяється навчанню студента, за підсумками успішності складаються експертні оцінки. На основі зібраної інформації складається резюме, що представляє собою повну характеристику людини. Це резюме відсилається всім організаціям, що мають необхідні вакансії.

Основним призначенням системи є автоматизація уведення й зберігання звітних даних по студентах, складання характеристик і резюме, пошуку вакансій у фірмах. Система дозволяє змінювати, доповнювати, вести пошук і перегляд інформації про студентів, накладати обмеження доступу до системи, зберігати списки студентів, що закінчили навчання, у вигляді архіву, контролювати видачу студентові завдань на курсові роботи й проекти, зв'язувати інститут з фірмами, зацікавленими в пошуку співробітників.

Дана система також може бути використана для складання окремих списків груп, друку залікових відомостей, друку повної бази даних і для статистики.

Система складається із чотирьох підсистем:

- контролю успішності студентів;
- професійних і психологічних тестів;
- обробки запитів, визначення категорій повноважень користувачів;
- експертних оцінок.

Підсистема контролю успішності студентів відповідає за статистичну звітність по успішності окремого студента, групи або цілого факультету, а також за зберігання й правильність її уведення.

Вхідними даними підсистеми є: оцінки, дати здачі іспитів, імена студентів, номери груп, факультет. На виході підсистема видає оброб-

лені дані: середній бал по студенту, групі або факультету, процентне співвідношення оцінок у студента в групі або на факультеті, імена й кількість стипендіатів у групі або на факультеті. Підсистема "Контроль успішності студентів" може функціонувати окремо від всієї системи, що дає можливість установити й використовувати її незалежно, якщо це необхідно.

Варіант 19. Система підтримки складання розкладу занять.

Потрібно розробити засобами Rational Rose модель системи підтримки складання розкладу занять.

Система забезпечує складання розкладу деякого навчального закладу, внесення в розклад змін, видачу повного розкладу й додаткової інформації (наприклад, за підсумковим розкладом складається розклад зазначеної групи на заданий день або тиждень).

У розкладі фіксуються час і місце проведення заняття, предмет і викладач, що проводить заняття, а також номер групи, для якої це заняття проводиться. Розклад не повинен містити колізій (наприклад, різні заняття не повинні перетинатися одне з одним за місцем і часом їхнього проведення, один викладач не може вести одночасно два різних заняття, у той же самий час в однієї й тієї ж групи не може бути два різних заняття й т.д.).

При роботі над цим варіантом завдання необхідно розробити схему бази даних для зберігання розкладу.

Варіант 20. Система нарахування зарплатні.

Перед інформаційною службою компанії поставлене завдання створення нової системи нарахування зарплатні замість морально застарілої існуючої системи. Нова система повинна надавати службовцям можливість записувати електронним способом інформацію з картки обліку робочого часу й автоматично формувати чеки на оплату, що враховують кількість відпрацьованих годин і загальний обсяг продажів (для службовців, що одержують комісійну винагороду).

Нова система повинна надавати службовцям можливість вводити інформацію з картки обліку робочого часу, вводити замовлення на поставку, змінювати свої параметри (такі, як спосіб оплати за роботу),

і формувати різні звіти. Система повинна працювати на персональних комп'ютерах із всієї компанії. З метою забезпечення безпеки й аудита службовці повинні мати можливість доступу й редагування тільки своїх власних карток обліку робочого часу й замовлень на поставку.

У системі повинна зберігатися інформація про всіх співробітників компанії у різних країнах. Система повинна забезпечувати правильну й своєчасну оплату роботи кожного службовця відповідно до зазначеного ним способу. Компанія з міркувань економії витрат бажає зберегти без змін одну з існуючих баз даних (БД керування проектами), що містить всю інформацію щодо проектів і тарифів. БД керування проектами функціонує в середовищі DB2 на мейнфреймі IBM. Нова система може читати дані із БД керування проектами, але не може оновлювати їх.

Деякі службовці одержують погодинну зарплатню. Вона нараховується на основі карток обліку робочого часу, кожна з яких містить дату й кількість годин, відпрацьованих відповідно до конкретного тарифу. Якщо який-небудь службовець відробив у день більше 8 годин, понаднормовий час оплачується з коефіцієнтом 1.5. Службовці-погодинники одержують зарплату щоп'ятниці .

Деякі службовці одержують фіксований оклад, однак вони теж представляють свої картки обліку робочого часу. Завдяки цьому система може вести облік кількості годин, відпрацьованих відповідно до конкретних тарифів. Такі службовці одержують зарплату в останній робочий день місяця.

Деякі із службовців з фіксованим окладом також одержують комісійну винагороду, що враховує обсяг продажів. Вони представляють замовлення на поставку, що відбивають дату й обсяг продажів. Відсоток комісійної винагороди визначається індивідуально для кожного службовця й може становити 10%, 15%, 25% або 35%.

Однією з найбільш часто використовуваних можливостей нової системи є формування різних звітів - запрошення кількості відпрацьованих годин, сумарної зарплатні, що залишилася, часу відпустки й т.д.

Службовці можуть вибрати спосіб оплати за працю. Вони можуть одержувати свої чеки на оплату поштою, на рахунок у банку або на руки в офісі.

Адміністратор системи веде інформацію про службовців. У його обов'язки входить уведення даних про нових службовців, видалення даних і зміну будь-якої інформації про що служить, такий, як ім'я, адреса й спосіб оплати, а також формування різних звітів для керівництва.

Додаток нарахування зарплатні запускається автоматично щоп'ятниці й в останній робочий день місяця, розраховуючи в ці дні зарплатню відповідних службовців. Нарахування зарплатні повинне виконуватися автоматично, без втручання користувача.

10.3 Контрольні питання

1. Поясніть розроблені діаграми.
2. Поясніть отриманий програмний код.

ЛІТЕРАТУРА

1. Федотова, Д.Э. CASE-технологии: Практикум. / Д.Э. Федотова, Ю.Д.Семенов, К.Н.Чижик.– М.: Горячая линия – Телеком, 2003. – 160 с.
2. Боггс, У. UML и Rational Rose. Упражнения. / У.Боггс, М. Боггс. – М.:Лори, 2008. – 286 с.
3. Трофимов, С.А. CASE-технологии: практическая работа в Rational Rose. / С.А. Трофимов. - М.:Бином-пресс, 2002. – 288 с.
4. Кватрани, Т. Rational Rose 2000 и UML. Визуальное моделирование. / Т. Кватрани – М.:ДМК Пресс, 2002. – 176 с.
5. Фаулер, М. UML. Основы. 3-е издание. / М.Фаулер, К.Скотт. – СПб.: Символ-Плюс, 2005. – 192 с.
6. Ларман, К. Применение UML и шаблонов проектирования. / К. Ларман. – М.:Вильямс, 2007. – 624 с.
7. Леоненков, А.В. Самоучитель UML. Эффективный инструмент моделирования информационных систем. / А.В. Леоненков. – ВHV - Санкт – Петербург, 2007. – 519 с.
8. Крачтен, Ф. Введение в Rational Unified Process. / Ф.Крачтен. – М.:Вильямс, 2002. – 240 с.
9. Буч, Г. UML. Руководство пользователя. / Г.Буч, Д.Рамбо, А.Джекобсон. – М.:ДМК Пресс, 2007. – 496 с.
10. Грекул, В.И. Проектирование информационных систем. / ., В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. – М.:Бином-пресс, 2008. – 304 с.
11. Леоненков, А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose. / А.В.Леоненков. –М.: Бином-пресс, 2006. – 320 с.
12. Амблер, С. Гибкие технологии: экстремальное программирование и унифицированный процесс разработки. / С.Амблер. – СПб.: Питер, 2005. – 412 с.
13. Рамбо, Д. UML 2.0. Объектно-ориентированное моделирование и разработка. / Д.Рамбо, М. Блаха. – СПб.: Питер, 2007. – 544 с.
14. Rational Rose. [Электронный ресурс] // Режим доступа: <http://www.interface.ru/rational/rosemain.htm>.
15. Арлоу, Д. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. / Д. Арлоу,

А. Нейштадт. – СПб.: Символ-плюс, 2007. – 624 с.

16. Новиков, Ф.А. Моделирование на UML. Теория, практика, видеокурс. / Ф.А. Новиков, Д.Ю.Иванов. – СПб.: Наука и техника, 2010. – 640 с.

17. Коберн, А. Современные методы описания функциональных требований к системам. / А. Коберн. – М.: Лори, 2011. – 288 с.

18. Орлов, С.А. Технологии разработки программного обеспечения. Учебник для вузов. / С.А.Орлов. – СПб.: Питер, 2002. – 464 с.

19. Орлов, С. А. Программная инженерия: технологии разработки программного обеспечения. Учебник для вузов / С. А. Орлов - 5-е изд., обновл. и доп. – СПб.: Питер, 2016. - 640 с.

20. Лаврищева, Е.М. Software Engineering компьютерных систем. Парадигмы, технологии и CASE- средства программирования/ Е. М. Лаврищева. – К.: Наук. думка, 2013. – 283 с.