# Практична робота № 9 ПОБУДОВА ДІАГРАМ ВПРОВАДЖЕННЯ, КОДОГЕНЕРАЦІЯ ТА ЗВОРОТНЕ ПРОЕКТУВАННЯ

**Теоретичні відомості** *Діаграми впровадження* *(****deployment diagrams****)* представляють фізичну модель предметної області й відображають апаратну частину реалізації програмного комплексу. Головними елементами діаграми впровадження є процесори *(processors),* пристрої *(devices*) та зв’язки між ними *(connections).*

В якості процесорів на діаграмі впровадження відображаються апаратні комплекси, сервери, робочі станції, контролери, тобто пристрої, що мають CPU. В якості пристроїв найчастіше виступає допоміжне обладнання, таке як пристрої вводу виводу, мережеве обладнання, таймери, датчики тощо. Графічне зображення процесорів та пристроїв наведено на рис. 26.



**Рис. 26.** Графічне зображення процесорів та пристроїв на діаграмі впровадження

Процесори та пристрої на діаграмі впровадження поєднуються зв’язками в залежності від їх фізичного поєднання в реальному світі. Розглянемо діаграму впровадження для абстрактного програмного комплексу з практичної роботи № 8. Програмний копмлекс має тришарову архітектуру. У якості першого шару виступає робоча станція з пристроями вводу-виводу; другий шар є веб-сервером, де виконується бізнес логіка програми; третім шаром виступає сервер баз даних, доступ до якого відбувається з веб-сервера (рис. 27).

***Кодогенерація*** ***(code generation)*** – це процес автоматизованого перетворення елементів діаграм UML у програмний код.



**Рис. 27.** Приклад діаграми впровадження для програмного комплексу з тришаровою архітектурою

Необхідними умовами проведення генерації програмного коду є наявність діаграми класів з переліком класів, які в сукупності формують програмний комплекс, а також діаграми компонентів, між якими дані класи розподілені. Кожен компонент повинен мати визначену мову реалізації. Послідовність кроків для проведення кодогенерації залежить від обраної мови програмування.

Перед проведенням кодогенерації необхідно впевнитись у тому, що для всіх атрибутів та операцій визначені типи даних, що властиві для мови програмування, визначеної у компоненті, до якого належить даний клас. Усі логічні типи даних (наприклад, такі як Integer, Boolean, String) мають бути замінені на типи даних відповідної мови програмування (у випадку С++ на int, bool, та char\* відповідно).

***Послідовність кроків для проведення кодогенерації з використанням***

***ANSI C++***

1. Визначити типи даних властиві для мови програмування С++ для всіх атрибутів та операцій всіх класів.
2. На жорсткому диску створити директорію для проведення кодогенерації.
3. На діаграмі компонентів для одного з компонентів у контекстному меню обрати пункт ANSI C++ / Open ANSI C++ Specification.
4. Задати параметри для генерації програмного коду (директорія для кодогенерації, директорії для h та cpp файлів, стиль для блоків коду, наявність коментарів, генерацію backup файлів тощо).
5. Для даного компоненту в контекстному меню обрати пункт ANSI C++ / Generate Code.
6. Вибрати зі списку класів, наведених у ListControl ті, для яких необхідно генерувати програмний код та натиснути Ок.
7. Програмний код буде згенеровано, а список помилок та попереджень виведено у лог.

Програмний код для конкретного класу можна згенерувати безпосередньо з діаграми класів, обравши в контекстному меню даного класу пункт ANSI C++ / Generate Code. За допомогою контекстного меню ANSI C++ / Class customization можна додати до класу специфічні операції (конструктори, деструктори, різноманітні оператори, селектори і модифікатори). Переглянути згенерований програмний код можна за допомогою команди ANSI C++ / Browse header та ANSI C++ / Browse body з використанням вбудованого редактора Rational Rose, а також із використанням будь-якого іншого текстового редактора безпосередньо в заданій для кодогенерації директорії.

***Послідовність кроків для проведення кодогенерації з використанням***

***Visual C++***

1. Визначити типи даних, властиві для мови програмування С++ для всіх атрибутів та операцій усіх класів.
2. На діаграмі компонентів для одного з компонентів у контекстному меню обрати пункт Update Code.
3. У вікні Code Update Tool обрати ті компоненти (чи окремі класи), для яких необхідно згенерувати код.
4. Якщо компонент не було призначено до жодного проекту VC++ відкриється нове вікно, в якому необхідно задати шлях до файла workspace проекта. Натиснувши ..., можна призначити компонент до існуючого проекта (Existing), чи створити новий (New). Після цього необхідно обрати класи чи компоненти для кодогенерації ще раз.
5. Натиснути Next, і на екрані з’явиться перелік класів, які будуть згенеровані.
6. Натиснути Finish, у вікні Code Update Tool буде відображено лог повідомлень, серед яких при наявності будуть вказані помилки та попередження, що виникли у процесі кодогенерації.

Кодогенерацію можна виконати також для окремого класу на діаграмі класів, якщо для даного класу обрати в контекстному меню пункт Update Code. Для редагування класу можна скористатися інструментом Model Assistant (обравши пункт Model Assistant контекстного меню), який дозволяє додавати до класу атрибути, операції; створювати конструктори, деструктори, різноманітні оператори, селектори і модифікатори.

***Зворотнє проектування*** (***reverse engineering***) – процес оновлення моделі проекту Rational Rose зі створеного, існуючого чи модифікованого програмного коду.

Генерація програмного коду є стартовою точкою розробки програмного забезпечення. В залежності від якості виконаного проектування у процесі розробки доводиться декілька разів змінювати дизайн із метою врахування додаткових вимог чи виправлення певних помилок. У процесі розробки кількість програмного коду, класів, інтейфесів, операцій та атрибутів критично зростає. Для відповідності моделі до програмного коду, який постійно змінюється виконується процес зворотнього проектування.

***Послідовність кроків для проведення зворотнього проектування з використанням ANSI C++***

1. Модифікувати програмний код, що було згенеровано попередньо (додати нові чи видалити існуючі атрибути, операції, параметри) для певного переліку класів.
2. На діаграмі компонентів (чи діаграмі класів) обрати компонент (клас), для якого необхідно провести процес зворотнього проектування.
3. У контекстному меню визначеного об’єкта (об’єктів) обрати пункт ANSI C++ / Reverse Engineer.
4. Модель Rational Rose буде приведено у відповідність до існуючого програмного коду.

***Послідовність кроків для проведення зворотнього проектування з використанням Visual C++***

1. Модифікувати програмний код, що було згенеровано попередньо (додати нові чи видалити існуючі атрибути, операції, параметри) для певного переліку класів.
2. На діаграмі компонентів (чи діаграмі класів) обрати будь-який компонент (клас).
3. У контекстному меню визначеного об’єкта (об’єктів) обрати пункт Update Model.
4. Натиснути Next, у вікні Model Update Tool обрати компоненти (класи), для яких необхідно провести процес зворотнього проектування. Натиснути Next.
5. У вікні Model Update Tool буде відображатися список класів, для яких буде виконано процес зворотнього проектування. Натиснути Finish.
6. Модель Rational Rose буде приведено у відповідність до існуючого програмного коду. Відомості про попередження чи помилки у процесі виконання зворотнього проектування будуть занесені в лог.

**Завдання** 1. Створити в середовищі Rational Rose діаграму впровадження для обраного програмного продукту.

1. Задати для всіх атрибутів та операцій типи даних відповідно до обраної мови реалізації програмного продукту.
2. Провести кодогенерацію відповідно до обраної мови реалізації.
3. У разі необхідності (генерація з використанням ANSI C++) створити новий проект у середовищі Visual С++ та додати до нього файли згенерованого коду.
4. Провести компіляцію програмного коду, виправити помилки процесу кодогенерації.
5. Додати до довільно обраного класу перелік атрибутів та операцій.
6. Перевірити вплив процесу зворотнього проектування на модель проекту.

## Вимоги

1. Діаграма впровадження повинна мати не менше 5 елементів.
2. Згенерований код має містити всі класи, присутні на діаграмі класів. 3. Модель Rational Rose і програмний код мають повністю відповідати один одному.

4. При компіляції програмного коду не повинно виникати помилок та попереджень.

**Контрольні запитання** 1. Для чого призначена діаграма впровадження (deployment diagram)? Наведіть її основні елементи.

1. У чому відмінність між процесорами та пристроями? Наведіть приклади для кожних з них.
2. Яким чином поєднуються елементи на діаграмі впровадження.
3. Дайте визначення процесу кодогенерації.
4. Що є необхідною умовою проведення кодогенерації?
5. Які UML діаграми впливають на генерацію програмного коду?
6. Як провести кодогенерацію, використовуючи стандарт ANSI C++?
7. Як провести кодогенерацію, використовуючи середовище MS Visual

C++ 6.0?

1. Дайте визначення процесу зворотного проектування.
2. Як провести зворотне проектування, використовуючи стандарт

ANSI C++?

1. Як провести зворотне проектування, використовуючи середовище MS Visual C++ 6.0?