



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №3
**Технологія розроблення програмного
забезпечення**

*«Діаграма розгортання. Діаграма компонентів.
Діаграма взаємодій та послідовностей.»*

Варіант 25

Виконала
студентка групи ІА-24
Мелешко Юлія Сергіївна

Перевірив:
Мягкий Михайло
Юрійович

Київ 2024р.

Мета: Навчитися розробляти діаграму розгортання. діаграма компонентів. діаграма взаємодій та послідовностей.

Хід роботи

25. Installer generator (iterator, builder, factory method, bridge, interpreter, client-server)

Генератор інсталяційних пакетів повинен мати якийсь спосіб налаштування файлів, що входять в установку, установки вікон з інтерактивними можливостями (галочка - створити ярлик на робочому столі; ввести в текстове поле деякі дані, наприклад, ліцензійний ключ і т.д.). Генератор повинен вивести один файл .exe або .msi.

Завдання.

1. Ознайомитися з короткими теоретичними відомостями.
2. Розробити діаграму розгортання для проектованої системи.
3. Розробити діаграму компонентів для проектованої системи.
4. Розробити діаграму послідовностей для проектованої системи.
5. Скласти звіт про виконану роботу

Діаграма розгортання (Deployment Diagram) - це один з типів діаграм UML (Unified Modeling Language), який використовується для моделювання фізичної структури та розміщення компонентів програмної системи на апаратному обладнанні, такому як сервери, комп'ютери, сенсори, мережеві пристрої тощо. Діаграма розгортання допомагає візуалізувати, як компоненти програмної системи розташовані на різних об'єктах обладнання та як вони взаємодіють між собою через мережу.

Вузли (Nodes): Вузли представляють фізичні або віртуальні об'єкти обладнання, такі як сервери, робочі станції, маршрутизатори, бази даних, сенсори тощо. Кожен вузол може мати атрибути, такі як IP-адреси або характеристики обладнання.

Артефакти (Artifacts): Артефакти представляють програмні компоненти або файли, які розгортаються на вузлах. Це можуть бути виконувані файли, конфігураційні файли, бази даних, додатки тощо.

Зв'язки (Connections): Зв'язки показують, як компоненти взаємодіють між собою на різних вузлах. Зв'язки можуть позначати мережеві з'єднання, передачу даних, доступ до служб тощо.

Діаграми розгортання корисні для інженерів програмного забезпечення та системних архітекторів для візуалізації архітектурної концепції системи, розміщення компонентів на конкретних серверах чи обладнанні

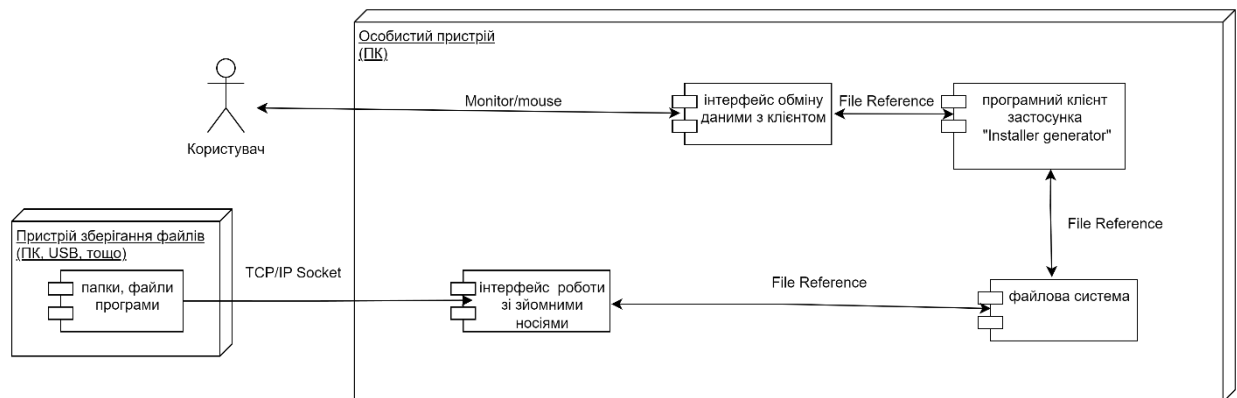


Рис 1 діаграма розгортання генератора інсталяційних пакетів

На діаграмі розгортання показані пристрої та інтерфейси що використовуються для генератора інсталяційних пакетів. Користувач взаємодіє з програмним клієнтом застосунка за допомогою інтерфейсу обміну даними з клієнтом. Програмний клієнт застосунка в свою чергу використовує файлову систему. Файлова система взаємодіє з інтерфейсом роботи зі зйомними носіями. Все це знаходиться на пристрої «Персональний ПК». На пристрої «USB» або також «Персональний ПК» надаються папки та файли, які необхідні для створення файла.exe, інтерфейсу роботи зі зйомними носіями.

Діаграма компонентів

Діаграма компонентів (іноді також називається діаграмою компонентної структури) - це тип діаграми, яка використовується в інженерії програмного забезпечення для візуального представлення архітектури системи або програмного продукту та її компонентів (або модулів). Діаграми компонентів допомагають розуміти, як система складається з окремих частин (компонентів), як вони взаємодіють між собою та як вони спільно працюють для досягнення цілей системи.

Компоненти: Це основні модулі або блоки, які складають систему або програмний продукт. Кожен компонент зазвичай виконує певну функцію і може бути реалізований як окремий програмний модуль або об'єкт.

Зв'язки: Діаграми компонентів включають зв'язки між компонентами, що показують, як вони взаємодіють між собою. Зв'язки можуть представляти залежності між компонентами, наприклад, використання одного компонента іншим, або вони можуть вказувати на комунікацію між компонентами через інтерфейси.

Інтерфейси: Компоненти можуть мати інтерфейси, які описують спосіб взаємодії з іншими компонентами. Інтерфейси вказують, які методи або функції можуть бути викликані із зовнішніх компонентів.

Залежності: Діаграми компонентів можуть також показувати залежності між компонентами, що вказують на те, як один компонент може впливати на інший.

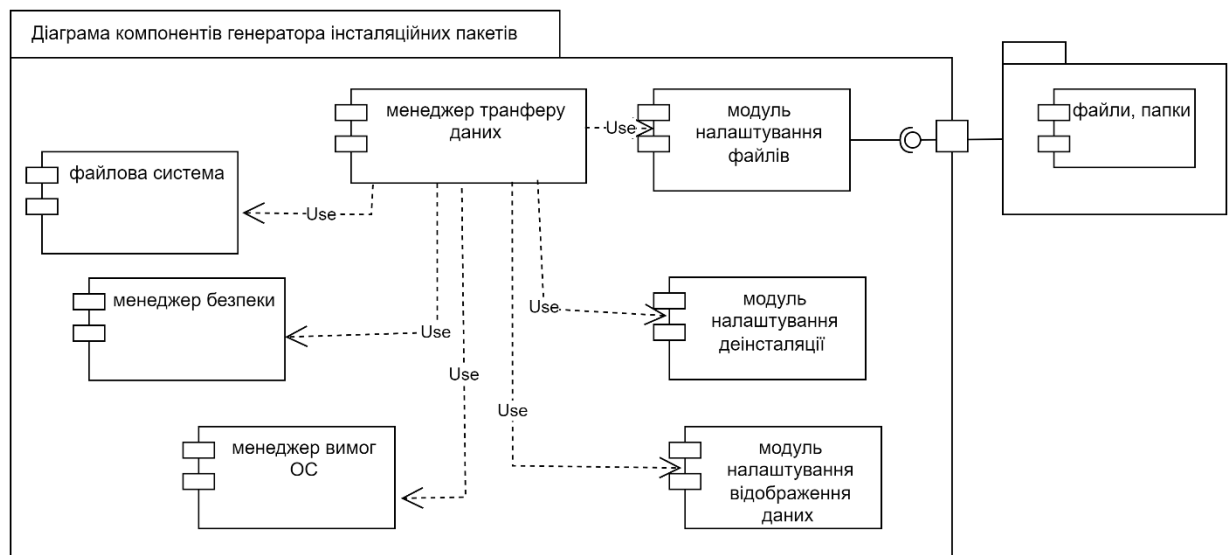


Рис 2 діаграма компонентів генератора інсталяційних пакетів

На діаграмі компонентів ми можемо бачити як модулі взаємодіють між собою для створення інсталяційного пакета. Модуль «Менеджер трансферу даних» використовує такі модулі як: модуль «Налаштування файлів», модуль «Налаштування деінсталяції», модуль «Налаштування відображення даних», модуль «Менеджер вимог ОС», модуль «Менеджер безпеки», модуль «файлова система». Модулю «Налаштування файлів» надаються файли зі зйомного носія, або ПК.

Діаграми послідовностей

Ці діаграми використовуються для візуалізації взаємодії між об'єктами або компонентами в системі впродовж певного проміжку часу. Вони дозволяють показати, як об'єкти взаємодіють між собою, обмінюючи повідомленнями або виконуючи методи, і як ця взаємодія впливає на стан системи.

Основні елементи діаграм послідовностей включають:

Об'єкти (Objects): Об'єкти або сутності, які беруть участь у взаємодії, представлені у верхній частині діаграми. Кожен об'єкт позначається ім'ям і може мати життєвий цикл, який відображається на діаграмі зліва направо.

Лінії життєвого циклу (Lifelines): Лінії життєвого циклу об'єкта відображають, як триває його існування під час взаємодії. Вони позначаються вертикальними лініями, на яких розташовані повідомлення, інтервали активації та інші моменти життєвого циклу об'єкта.

Повідомлення (Messages): Повідомлення показують передачу інформації або виконання дій між об'єктами. Існують різні типи повідомлень, такі як синхронні (з блокуванням), асинхронні (без блокування), відповіді на повідомлення тощо.

Діаграми послідовностей корисні для аналізу та проектування взаємодії в системі. Вони допомагають визначити порядок виконання операцій та повідомлень між об'єктами та можуть бути використані для детального опису архітектурних взаємодій в програмному забезпеченні перед його реалізацією.



Рис.3 Діаграма послідовностей Генератора інсталяційних пакетів

Ця діаграма описує часові особливості роботи Генератора інсталяційних пакетів. В нас є Користувач, який крок за кроком створює файл.exe

Користувач заходить до програми Генератор інсталяційних пакетів і створює новий проект. Додає назву майбутньому файлу.exe, потім додає назву компанії, обирає місце, де буде розташований файл.exe. Далі додає файли, які він хоче щоб були у файлу.exe, обирає ОС на яких програма зможе працювати, вибирає, чи хоче щоб програма при встановленні запрошувала права адміністратора. Користувач додає ліцензію з якою буде порівнюватися ліцензії при встановленні програми, а також файл з додатковою інформацією, яку можна буде почитати при встановленні. Додає помітку, що хоче створити ярлик та надає шлях, де буде розташовуватися ярлик, додає назву ярлику. Обирає мови, логотип для програми та фон при встановленні. Також можна вибрати файли, які будуть видалені при деінсталяції, якщо не обирати, то буде за дефолтом всі видалятися. Інсталятор створює файл.exe, зберігає його в обраній папці. Користувач завершує роботу програми Генератора інсталяційних пакетів.

Висновок: Під час виконання лабораторної роботи я навчилася створювати діаграми розгортання, компонентів та послідовностей для проектованої системи