Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут”

Кафедра АСОІУ

**ЗВІТ**

про виконання комп’ютерного практикуму №6

з дисципліни

«Архітектура програмного забезпечення»

Тема: ««Створення діаграми компонентів»

|  |  |
| --- | --- |
| Прийняв: | Виконав: |
| Кузнєцов  Олександр  Вікторович | студент 3-го курсу  гр. ІП-51 ФІОТ  Зарічковий Олексадр Анатолійович |

Київ – 2017

**ЗМІСТ:**

1. Мета РОБОТИ 3
2. Постановка задачі 4
3. Результати та пояснення 5
4. ЗАПИТАННЯ ТА ВІДПОВІДІ 14
5. Висновок 16

# Мета РОБОТИ

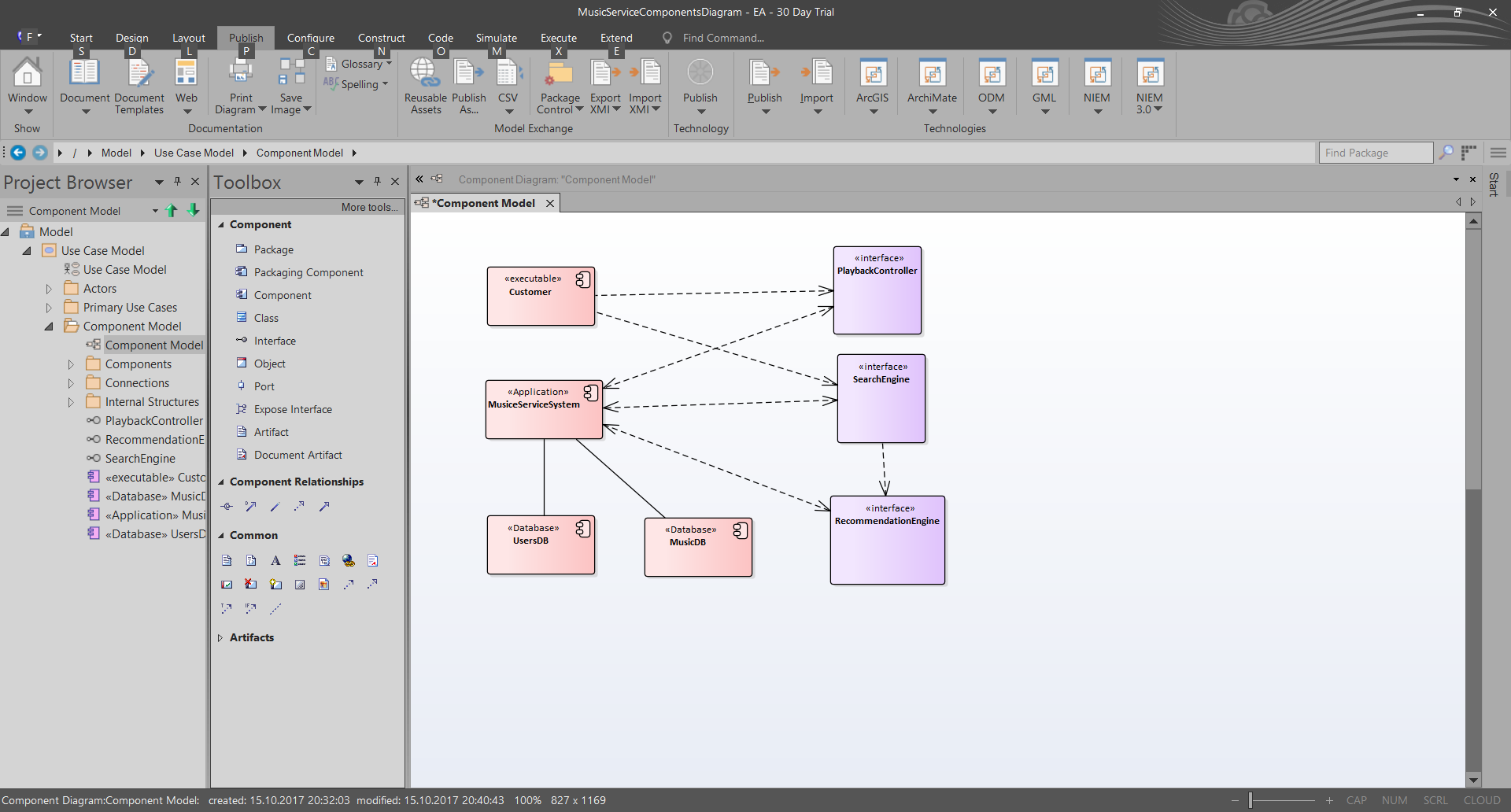
Набути навичок побудови діаграми компонентів.

# Постановка задачі

* Створити діаграму компонентів та додати до неї компоненти
* Додати інтерфейси
* Додати відношення
* Створити короткий опис усіх компонентів.

# Результати та пояснення

**Діаграма компонентів**:



**Кількісний аналіз діаграми**:

Маємо 3 інтерфейси, 4 компоненти, 6 відношень типу «Залежність», 2

відношення типу «Асоціація».

Sobj = 5\*4 = 20

Slnk = 1\*6 = 6

Tobj = 1

Tlnk = 1

Oobj = 5

S = (20 + 6)/(1 + 5 + (1+1)^0.5) = 26/7.4 = 3.42

**Опис компонентів та інтерфейсів**:

1. *База даних користувачів* – набір файлів, що зберігають всю інформації про користувачів системи.
2. *База музикальних файлів* – набір файлів, що зберігають мультимедійну інформацію.
3. *Система* – додаток, що координує роботу інтерфейсів між собою та їх взаємодію з базами даних.
4. *Користувач* – файл, що реалізує бізнес-логіку користувача.
5. *Інтерфейс програвача* – інтерфейс, що надає можливість контролювати процес програвання музикальних треків на сайті.
6. *Інтерфейс пошукової двигуна* – інтерфейс, що надає можливість знаходити музикальні файли на сайті.
7. *Інтерфейс рекомендаційної системи* – інтерфейс, що надає можливість генерувати набори рекомендованої музики для заданого користувача.

# ЗАПИТАННЯ ТА ВІДПОВІДІ

1. **Що таке пряме та зворотнє проектування?**

Пряме проектування (forward engineering) - це процес трансформації моделі в код з відображенням на мову реалізації. В результаті прямого проектування відбувається втрата інформація, оскільки моделі, описані на UML, семантично багатшими, ніж будь-який сучасний об'ектно-орієнтованний підхід в програмуванні.

Зворотне проектування (reverse engineering) - це процес трансформації коду в модель. Зворотне проектування породжує надлишок інформації, частина якої представлена на більш низькому рівні деталізації, ніж потрібно для побудови зручною моделі.

1. **З чого складається діаграма компонентів? Дайте визначення основного будівного блоку.**

Діаграма компонентів складається з так званих компонент, які з’єднуються між собою певними відношення. Компонент (component) - фізично існуюча частина системи, яка забезпечує реалізацію класів і відносин, а також функціонального поведінки модельованої програмної системи. Компонент призначений для подання фізичною організації асоційованих з ним елементів моделі. Компонентом може бути виконуваний код окремого модуля, командні файли або файли, що містять інтерпретовані скрипти. Компонент служить для загального позначення елементів фізичного представлення моделі і може реалізовувати певний набір інтерфейсів**.**.

1. **З яких етапів складається створення діаграми компонентів?**

Створення діаграми станів має наступні етапи:

• вибір предметної області

• вибір компонентів

• виділення інтерфейсів

• встановлення відношень

• опис компонентів та відношень

1. **Для чого потрібні інтерфейси на діаграмі компонентів?**

Різні компоненти можуть не мати можливості взаємодіяти один з одним напряму. Наприклад, розробник не може просто сказати адміністратору, що його додаток треба завантажити у систему, інакше б адміністратору було дуже складно усім власноруч відповідати. Для цього існує спеціальний інтерфейс Запиту, що робить такі звертання зручнішими для обох сторін. Відповідним чином, він зображується на діаграмі компонентів.

1. **Перелічіть стереотипи, які визначені в стандарті UML для діаграми компонентів?**

* << file >> (файл) - визначає найбільш загальну різновид компонента, який представляється у вигляді довільного фізичного файлу.
* << executable >> (здійсненний) - визначає різновид компонента-файлу, який є виконуваним файлом і може виконуватися на комп'ютерній платформі.
* << document >> (документ) - визначає різновид компонента-файлу, який представляється в формі документа довільного змісту, який не є виконуваним файлом або файлом з вихідним текстом програми.
* << library >> (бібліотека) - визначає різновид компонента-файлу, який представляється в формі динамічної або статичної бібліотеки.
* << source >> (джерело) - визначає різновид компонента-файлу, що представляє собою файл з вихідним текстом програми, який після компіляції може бути перетворений в здійсненний файл.
* << table >> (таблиця) - визначає різновид компонента, який представляється в формі таблиці бази даних.

1. **Які відношення існують на діаграмі компонентів?**

Стрілка Асоціація виконує роль відношення, що визначає взаємозв’язок між інтерфейсом, та компонентом, що використовує інтерфейс. Стрілка Залежність виконує роль відношення залежності між інтерфейсом та компонентом, коли внесення змін в бібліотеки або початковий код призводить до змін самого компоненту.

# Висновок

Отже, я отримав навички зображення діаграм компонентів засобами UML-діаграм, навички документації цих діаграм за допомогою опису їх компонентів. Також я ознайомився з принципами виокремлення інтерфейсів системи, компонентів та їх відношень.