

Учреждение образования
«Белорусский государственный технологический университет»
Кафедра программной инженерии

Лабораторная работа №6
**«Объектно-ориентированное моделирование.
Физические диаграммы UML»**

Выполнила:
студентка 4 курса 5 группы ФИТ
Каспер Н.В.
Проверила:
Ющик Д.С.

Минск 2020

Цель работы: изучить методологию объектно-ориентированного моделирования средствами UML; ознакомиться с основными принципами объектно-ориентированного проектирования программного обеспечения, получить навыки проектирования архитектуры информационной системы с применением методологии UML.

Теоретические вопросы:

1. Укажите назначение физических диаграмм: компонентов и развертывания.

Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код.

Диаграмма развертывания содержит графические изображения процессоров, устройств, процессов и связей между ними.

2. Дайте определение понятиям: узел, артефакт, интерфейс.

Узел (node) представляет собой некоторый физически существующий элемент системы, обладающий некоторым вычислительным ресурсом.

Артефакт – некая физическая сущность, программный компонент, который используется или создаётся во время работы программного обеспечения.

Интерфейс – это внешне видимый, именованный набор операций, который класс, компонент или подсистема может предоставить другому классу, компоненту или подсистеме, для выполнения им своих функций.

3. Опишите нотации, которые используются для представления компонентов (их вариации).

Компоненты развертывания, которые обеспечивают непосредственное выполнение системой своих функций: динамически подключаемые библиотеки с расширением dll, Web-страницы на языке разметки гипертекста с расширением html и файлы справки с расширением hlp.

Компоненты-рабочие продукты: файлы с исходными текстами программ, например, с расширениями h или srr для языка C++.

Компоненты исполнения, представляющие исполнимые модули – файлы с расширением exe.

Интерфейс служит для описания атрибутов и операций, которые должен реализовать компонент.

4. Опишите основные нотации, которые используются для представления архитектуры системы в виде диаграммы развертывания.

Узел (node) представляет собой некоторый физически существующий элемент системы, обладающий некоторым вычислительным ресурсом.

Соединения указывают отношения между узлами и являются разновидностью ассоциации.

5. Укажите основные виды связей между компонентами и между узлами.

В качестве отношений выступают физические соединения между узлами и зависимости между узлами и компонентами. Соединения являются разновидностью ассоциации и изображаются отрезками линий без стрелок.

Ход работы:

На основе технического задания были определены компоненты, узлы и взаимоотношения между ними. Результат построения представлен на рисунке 1.

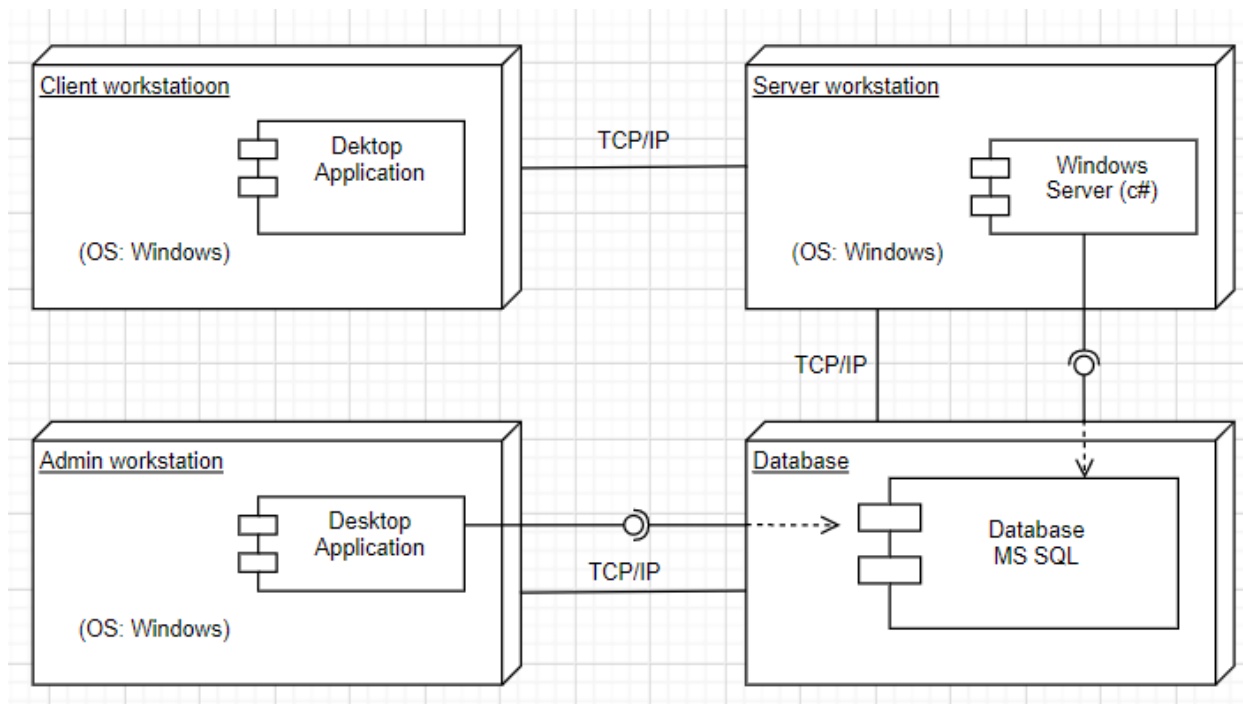


Рисунок 1. Диаграмма компонентов

Из рисунка видно, что у нас есть рабочие станции: сервер, СУБД, ПК администратора и ПК пользователя. А также такие компоненты как десктопное приложение, база данных, сервер. Определены протоколы, которыми связаны узлы.

Постановка задачи:

Система управления автовокзалом. Проектируемая ИС предоставляет интерфейс для просмотра доступных поездов.

Пользователь может авторизоваться; просматривать список доступных поездов; осуществлять фильтрацию поездов по дате выезда либо приезда; бронировать билет.

Диспетчер может просматривать все возможные страницы; добавлять, изменять и удалять информацию об автостанциях и автобусах, просматривать список доступных поездов; осуществлять фильтрацию поездов по дате отправления или прибытия.

Менеджер компаний-перевозчиков может просматривать все возможные страницы; добавлять, изменять и удалять информацию об автобусах и компаниях-перевозчиках.

Топ-менеджер может просматривать все возможные страницы; добавлять, изменять и удалять любые виды информации.

Описание программно-аппаратных средств, используемые при выполнении работы:

Построение моделей выполнялось с помощью бесплатного онлайн-инструмента Draw.io.

Draw.io — инструмент, который позволяет создавать блок-схемы, сетевые диаграммы, интеллект-карты, отношения сущностей, программные блоки, UML, макеты и т. д. Богатая функциональность Draw.io позволяет пользователям отслеживать и восстанавливать изменения, импортировать и экспортировать в PDF, PNG, XML, VSDX, HTML, а также автоматически публиковать и делиться работами. С помощью сервиса можно создавать: графики, блок-схемы, диаграммы, ментальные карты, макеты.