**Государственное бюджетное профессиональное образовательное учреждение города Москвы "Московский автомобильно-дорожный колледж им. А.А. Николаева"**

**ДОКЛАД**

**по инструментальным средствам разработки программного обеспечения**

**на тему:**

**service**-**oriented** **architecture**

**Выполнил студент группы 2ИП1**

**Копачински Ольга**

**SOA (англ. Service Oriented Architecture) — это прикладная архитектура, в которой функции определены как независимые сервисы с вызываемыми интерфейсами. Обращение к этим сервисам в определенной последовательности позволяет реализовать тот или иной бизнес-процессов**

**Функции SOA:**

* Разделение интерфейсов: Функции SOA разделены на отдельные слои, каждый из которых предоставляет определенный сервис, который может быть использован другими слоями или клиентским приложением.
* Использование стандартизованных протоколов: Функции SOA используют стандартизованные протоколы, такие как SOAP (Simple Object Access Protocol) и HTTP (Hypertext Transfer Protocol), для взаимодействия между слоями и клиентскими приложениями.
* Управление сервисами: Функции SOA включают в себя механизмы управления сервисами, такие как регистрация сервисов, мониторинг и масштабирование, которые позволяют эффективно управлять и поддерживать функции SOA.
* Разделение данных: Функции SOA предоставляют возможность разделения данных между различными слоями, что позволяет каждому слою работать со своей собственной копией данных и обеспечивает более гибкую архитектуру.
* Гибкость: функции SOA позволяют независимую разработку и изменение сервисов без влияния на всю систему. Это также позволяет добавлять или удалять сервисы по мере необходимости без изменения всей архитектуры.

**Основные принципы использования Soa архитектуры в разработке ПО:**

**Модульность:**

*Сервисы разбиваются на отдельные модули, что позволяет разработчикам работать над ними независимо друг от друга. Это упрощает процесс разработки, сопровождения и масштабирования приложения.*

**Возможность повторного использования:**

*Сервисы могут быть использованы повторно в различных приложениях и системах. Это позволяет сэкономить время и усилия на разработку нового функционала.*

**Легкая интеграция:**

*Сервисы разрабатываются с учетом возможности их интеграции с другими компонентами системы. Это позволяет легко обмениваться данными и функциональностью между сервисами.*

**Гибкость:**

*Soa архитектура позволяет быстро реагировать на изменения в бизнес-требованиях и добавлять новый функционал или изменять существующий без необходимости переработки всей системы.*

**Легкая масштабируемость:**

*Поскольку каждый сервис предоставляет определенную функциональность, его можно масштабировать независимо от других компонентов системы. Это позволяет более эффективно использовать ресурсы и увеличить производительность системы при увеличении нагрузки.*

**Недостатки SOA:**

**Сложность развертывания:**

*SOA требует дополнительных инфраструктурных изменений и инвестиций, чтобы реализовать полностью функционирующую архитектуру, что может быть сложным и затратным процессом.*

**Избыточность данных:**

*По мере роста количества сервисов, данные в разных сервисах могут быть дублированы, что может привести к проблемам согласованности данных.*

*.* **Сложности взаимодействия:**

*Клиентам требуется научиться взаимодействовать с новой архитектурой SOA, а это может потребовать.*

**Зависимость от сторонних сервисов:**

*SOA позволяет использовать сервисы, предоставляемые другими организациями, но это также означает, что ваша система становится зависимой от этих сторонних сервисов. Если один из них перестает работать или изменяет свои условия, это может иметь негативное влияние на вашу систему.*

**Безопасность:**

*Распределенная природа SOA может создавать дополнительные проблемы безопасности, например, в области аутентификации и авторизации, поскольку требуется управлять доступом к множеству сервисов.*

**SOA требует хорошей архитектурной проработки и дизайна, чтобы гарантировать эффективное взаимодействие между сервисами. Реализация SOA также может потребовать внесения изменений внутри компании, включая обучение сотрудников и повышение уровня согласованности между отделами.**

**В заключение, service-oriented architecture - это подход к проектированию систем, который позволяет разделять функциональность на отдельные сервисы и облегчает интеграцию и модернизацию системы. Он предлагает множество преимуществ, но требует тщательного планирования и грамотного внедрения.**