Disynergy
Страница материала ↗
EXQT
Готовые ответы — на EXQT
Доступ к ответам приобретается отдельно.
Найти ответы ↗

УНИВЕРСИТЕТ СИНЕРГИЯ

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ

ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»

Факультет Информационных технологий
(наименование факультета/ института)

Направление подготовки /специальность: 09.03.03 Прикладная информатика
(код и наименование направления подготовки /специальности)

Профиль/специализация: Инженер искусственного интеллекта (AI инженер)
(наименование профиля/специализации)

УТВЕРЖДАЮ
Декан факультета
Информационных технологий
Университета «Синергия»

Захаров
(подпись)

А.В. Захаров
(И.О. Фамилия)

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

(вид практики)

Технологическая (проектно-технологическая) практика
(тип практики)

обучающегося группы _____ _____.
(Шифр и № группы) (Ф.И.О. обучающегося)

Место прохождения практики:
Университет Синергия/
(наименование Профильной организации/ профильного структурного подразделения Образовательной организации)

Срок прохождения практики: с «___» _____ 20__ г. по «__» _____20__ г.

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

Содержание индивидуального задания на практику:

№ п/п	Виды работ
1.	Инструктаж по соблюдению правил противопожарной безопасности, правил охраны труда, техники безопасности, санитарно-эпидемиологических правил и гигиенических нормативов.
2.	Выполнение определенных практических кейсов-задач, необходимых для оценки умений, навыков и (или) опыта деятельности по итогам учебной практики (технологической (проектно-технологической) практики) (вид практики, тип практики)
2.1.	Кейс-задача № 1 Проанализируйте и опишите состав компонентов трехуровневой архитектуры, по следующим критериям: • Уровни архитектуры: Трехуровневая архитектура обычно состоит из трех уровней: презентационного, бизнес-логики и данных. Презентационный уровень отвечает за отображение информации пользователю, уровень бизнес-логики обрабатывает данные и выполняет бизнес-правила, а уровень данных управляет доступом к базе данных. • Компоненты презентационного уровня: Презентационный уровень может включать в себя веб-серверы, серверы приложений, клиентские приложения (например, веб-браузеры, мобильные приложения), пользовательские интерфейсы и т.д. • Компоненты уровня бизнес-логики: Уровень бизнес-логики может включать в себя серверы приложений, бизнес-объекты, сервисы бизнес-правил, компоненты обработки транзакций и т.д. • Компоненты уровня данных: Уровень данных может включать в себя базы данных, системы управления базами данных, хранилища данных, серверы баз данных и т.д. • Связь между уровнями: Важно проанализировать, как компоненты различных уровней взаимодействуют друг с другом. Например, как презентационный уровень взаимодействует с уровнем бизнес-логики и уровнем данных, какие протоколы и технологии используются для обмена данными между уровнями. • Безопасность и аутентификация: Необходимо учесть компоненты, отвечающие за безопасность и аутентификацию пользователей, такие как системы управления доступом, механизмы аутентификации и шифрования. • Масштабируемость и отказоустойчивость: Важно проанализировать, как архитектура обеспечивает масштабируемость и отказоустойчивость. Это может включать в себя использование кластеризации, балансировки нагрузки, резервирования и т.д. • Производительность и оптимизация: Необходимо учесть компоненты, отвечающие за оптимизацию производительности, такие как кэширование, оптимизация запросов к базе данных, использование асинхронной обработки и т.д. • Разработка и поддержка: Важно учесть компоненты, связанные с разработкой и поддержкой системы, такие как системы контроля версий, инструменты сборки, системы непрерывной интеграции и доставки (CI/CD). • Совместимость и интеграция: Необходимо учесть компоненты, отвечающие за интеграцию с другими системами и сервисами, а также за совместимость с различными платформами и технологиями. • Управление и мониторинг: Важно учесть компоненты, отвечающие за управление и мониторинг системы, такие как системы мониторинга, логирования, управления конфигурацией и т.д. • Ресурсы и требования к инфраструктуре: Важно учесть требования к

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

№ п/п	Виды работ
	ресурсам и инфраструктуре для каждого компонента, такие как вычислительные мощности, хранение данных, сетевые ресурсы и т.д. • Стандарты и лучшие практики: Необходимо учесть соответствие архитектуры стандартам и лучшим практикам в области разработки программного обеспечения и архитектуры информационных систем. Это лишь примерный список параметров, которые могут быть использованы для описания задания. Конкретные параметры могут варьироваться в зависимости от задачи.
2.2.	Кейс-задача № 2 Проведите анализ оптимизации обработки исключений в C++, по следующим критериям: • Производительность: Обработка исключений должна быть оптимизирована для минимизации влияния на производительность программы. Это может включать в себя использование исключений только в исключительных ситуациях и избегание их для нормального управления потоком выполнения программы. • Масштабируемость: Система обработки исключений должна быть масштабируемой, чтобы корректно работать в условиях высокой нагрузки и больших объемов данных. • Надежность: Система обработки исключений должна быть надежной и устойчивой к сбоям. Это может включать в себя механизмы восстановления после сбоев и корректную обработку исключений. • Безопасность: Система обработки исключений должна обеспечивать безопасность данных, предотвращая несанкционированный доступ и утечки данных. • Управление ресурсами: Система обработки исключений должна эффективно управлять ресурсами, такими как память и процессорное время. Это может включать в себя оптимизацию использования памяти и управление ресурсами в случае исключений. • Иерархия исключений: Система обработки исключений должна поддерживать иерархию исключений, чтобы можно было обрабатывать их на разных уровнях абстракции. • Логирование и отладка: Система обработки исключений должна предоставлять возможности для логирования и отладки, чтобы можно было анализировать возникшие исключения и устранять проблемы. • Интеграция с другими системами: Система обработки исключений должна быть способна интегрироваться с другими системами и сервисами, такими как базы данных и облачные хранилища. • Поддержка стандартов: Система обработки исключений должна поддерживать стандарты и протоколы, такие как C++ Standard Library и Boost, для обеспечения совместимости с различными приложениями и библиотеками. Это лишь примерный список параметров, которые могут быть использованы для описания задания. Конкретные параметры могут варьироваться в зависимости от задачи.
2.3.	Кейс-задача № 3 Создайте проект в RAD среде Delphi выводящий с помощью компонента Label текст “Прикладная программа”. Ответом на задачу будет ссылка на репозиторий GitHub, где хранится Ваша программа. Или иным удобным для Вас способом.
2.4.	Кейс-задача № 4 Проведите анализ выполненной кейс-задачи №3. Критерии анализа выполненной задачи могут включать следующие аспекты: • Функциональность: Компонент Label должен корректно отображать текст "Прикладная программа" на форме приложения.

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

№ п/п	Виды работ
	• Дизайн интерфейса: Внешний вид приложения должен быть аккуратным и профессиональным, с хорошо организованными элементами управления. • Совместимость: Приложение должно корректно работать на целевой операционной системе (например, Windows) и не должно вызывать конфликтов с другими приложениями или системой. • Безопасность: Приложение не должно содержать уязвимостей, которые могут поставить под угрозу безопасность данных пользователя. • Производительность: Приложение должно работать стабильно и быстро, не занимая чрезмерного количества системных ресурсов. • Документация: Должна быть предоставлена документация, описывающая функциональность приложения, его настройки и требования к системе. • Тестирование: Приложение должно пройти все необходимые тесты на соответствие заданным требованиям. • Соответствие стандартам: Приложение должно соответствовать стандартам программирования и безопасности, принятым в индустрии. • Поддержка: Должна быть предусмотрена возможность поддержки пользователей, включая ответы на вопросы и исправление ошибок.
2.5.	Кейс-задача № 5 Спроектируйте базу данных «Вуз». При проектировании базы данных необходимо создать 4-5 таблиц предметной области: 3-4 таблицы- справочника и 1 таблицу переменной информации. Для всех таблиц создать первичные ключи. Построить связи между таблицами при помощи внешних ключей: атрибуты таблицы переменной информации должны ссылаться на ключевые атрибуты таблиц справочников.
3.	Систематизация собранного нормативного и фактического материала.
4.	Оформление отчета о прохождении практики.
5.	Защита отчета по практике.

Обучающий(ая)ся индивидуальное задание получил(а):

_____ _____

(Ф.И.О.) (Подпись)

«___»_____ 20__г.[1]

[1] В соответствии с календарным учебным графиком указывается дата первого дня практики.

Помогу с учебным заданием без предоплаты. Оплата после выполнения

Нажмите на мессенджер, чтобы написать

Telegram ↗ WhatsApp ↗ MAX ↗

+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru