

Дополнительная профессиональная программа повышения квалификации «Системный анализ для начинающих»

Программа представляет собой комплексный подход к изучению процесса разработки программного обеспечения, включая составление требований к программному продукту и изучение его жизненного цикла, управление рисками, взаимодействие участников команды и их роли. В процессе обучения слушатели развивают навыки сбора, анализа и моделирования требований к программному обеспечению. В программу включен модуль по языку SQL с выполнением практических заданий на SQL-тренажере.

№	Наименование	Характеристика
1	Наименование и цели программы	
1.1	Наименование	«Системный анализ для начинающих»
1.2	Вид программы	Дополнительная профессиональная программа повышения квалификации
1.3	Цель	Формирование у слушателей знаний об основах теории системного анализа и навыков их применения
1.4	Планируемые результаты (формируемые или совершенствующиеся компетенции в соответствии с указанным профстандартом)	Изучение и описание деятельности, подлежащей автоматизации, и работы пользователей в системах-аналогах; Извлечение и описание образцов данных и агрегированных значений из систем-аналогов Профессиональный стандарт 06.022 «Системный аналитик» (утв. приказом Министерства труда и социальной защиты РФ от 27.04.2023 № 367н). Подготовка к выполнению задания на тестирование ПО Профессиональный стандарт 06.004 «Специалист по тестированию в области информационных технологий» (утв. приказом Министерства труда и социальной защиты РФ от 02.08.2021 № 531н).
2	Формат реализации	
2.1	Язык обучения	Русский
2.2	Форма обучения	Очно-заочная с применением ДОТ
2.3	Объем программы	58 академических часа
2.4	Режим занятий	В течение 7 недель в удобное для слушателя время. Вебинары – согласно расписанию занятий.
2.5	Выдаваемый документ	Удостоверение о повышении квалификации Академии ПСБ

2.6	Целевая аудитория	Аналитики данных; руководители проектов и менеджеры в области информационных технологий и аналитики; студенты, которые интересуются системным анализом и хотят приобрести знания и навыки в этой области
2.7	Численность обучаемых по программе	Численность слушателей определяется договором об образовании, заключаемым при приеме на обучение
3	Организационно-педагогические условия	
3.1	Учебный план	см. Приложение № 1
3.2	Календарный учебный график	см. Приложение № 2
3.3	Рабочая программа (учебные предметы, курсы, дисциплины (модули))	см. Приложение № 3
3.4	Профессорско-преподавательский состав	Реализация программы обеспечивается квалифицированными научно-педагогическими кадрами, имеющими базовое образование, соответствующее профилю преподаваемой дисциплины, и систематически занимающимися научной и научно-методической деятельностью
3.5	Форма итоговой аттестации	Экзамен в форме выполнения задания
3.6	Оценочные материалы	Пример задания см. Приложение № 4 . Результаты выполнения оцениваются на основании 4 критериев, с присвоением баллов по каждому критерию оценки. Максимальный балл, который может набрать слушатель за выполнение итогового задания – 100 баллов. Условием успешного завершения обучения и получения удостоверения о повышении квалификации является выполнение итогового задания с результатом не менее 40 баллов
3.7	Методические материалы	Учебная литература (учебники, пособия, книги) – предоставляется доступ к электронной библиотечной системе «Знаниум» (www.znanium.ru).
4	Контактная информация	
4.1	Руководитель программы	Мубаракшина Алсу Ирековна, директор Центра развития обучения по управлению данными АНО ДПО «Академия ПСБ» a.mubarakshina@psb-academy.ru

УЧЕБНЫЙ ПЛАН
дополнительной профессиональной программы повышения квалификации
«Системный анализ для начинающих»

Количество часов – 58 академических часа.

Форма обучения – очно-заочная с применением дистанционных образовательных технологий.

№ п/п	Наименование раздела/ модуля, темы	Общая трудоемкость, час.	Аудиторная работа, час.					С применением дистанционных образовательных технологий, электронного обучения, час.					Самостоятельная работа, час.	Текущий контроль успеваемости	Промежуточная аттестация (форма/час.)	Итоговая аттестация (вид/час.)	Код компетенции
			Всего	В форме практической подготовки	В том числе			Всего	В форме практической подготовки	В том числе							
					Лекции/в интерактивной форме	Практические (семинарские) занятия/в интерактивной форме	Контактная самостоятельная работа			Лекции/в интерактивной форме	Практические (семинарские) занятия/в интерактивной форме	Контактная самостоятельная работа					
1	Введение	2						2		2							A/02.4
2	Процесс разработки программного обеспечения	6						5	2	3	1	1	1				A/02.4
3	Сбор бизнес-требований	6						5	2	3	1	1	1				A/02.4
4	Разработка и описание требований к программному обеспечению	9						8	2	6	1	1	1				A/01.4
5	SQL	16						15	5	12	1	2	1				A/04.4
6	Моделирование процессов	8						7	1	5	1	1	1				A/04.4
7	Тестирование программного обеспечения	9						8	2	6	1	1	1				A/02.4 A/01.4
	Итоговая аттестация	2														Э¹/2	A/01.4 A/02.4 A/04.4
	ИТОГО:	58						50	14	37	6	7	6				

¹ Вид итоговой аттестации: Э – экзамен.

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Период обучения:		
35 дней	7 недель	2 месяца

РАБОЧИЕ ПРОГРАММЫ РАЗДЕЛОВ / МОДУЛЕЙ / ДИСЦИПЛИН

1. ВВЕДЕНИЕ	
1.1. Входное тестирование.	
1.2. Введение в системный анализ.	• Определение системного аналитика. • Задачи системного аналитика. • Отличие системного аналитика от бизнес-аналитика.
1.3. Основное противоречие.	• Противоречие в изменениях требований в системном анализе.
2. ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	
2.1. Процесс разработки ПО.	• Существующие риски проекта. • Управление процессами проекта. • Фазы жизненного цикла. • Каскадная модель. • Инкрементная модель. • Семейство методологий Agile. • Методология Scrum и Kanban.
2.2. Жизненный цикл программного продукта.	• Характеристики жизненного цикла программного продукта. • Важность бизнес-требований. • Пользовательские требования. • Функциональные требования.
2.3. Участники и роли в процессе разработки.	• Понимание процессов взаимодействия участников и их ролей в разработке.
3. СБОР БИЗНЕС-ТРЕБОВАНИЙ	
3.1. Понимание и задачи бизнес-требований.	• Виды требований. • ПО как инструмент оптимизации бизнес-процессов.
3.2. Сбор бизнес-требований.	• Владелец требований. • Атрибуты качества требований. • Назначение приоритетов требований. • Заинтересованные лица разработки системы. • Матрица ролей и прав доступа.
3.3. Структурирование бизнес-требований.	• Цели моделирования. • Прототипы моделей ПО. • Риски при работе с прототипами.
4. РАЗРАБОТКА И ОПИСАНИЕ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ	
4.1. Понимание требований к ПО.	• Область проблем пользователя. • Пользовательские требования.
4.2. Описание процессов.	• Контекстная диаграмма.

• Матрица CRUD.
4.3. Моделирование требований. • Требование пользователей User Story. • Разработка в Agile. • Оценка User Story.
4.4. Инструменты моделирования требований. • Требование пользователей Use Cases. • Диаграмма Use Cases. • Моделирование Use Cases. • Ассоциации.
4.5. Постановка задачи разработчику. • Приоритеты при создании документа. • Нефункциональные требования. • Чек-лист по качеству описания требований.
5. SQL
5.1. SQL. • Понимание реляционных баз данных.
5.2. Клиент-серверная архитектура. • Описание файл-серверной и клиент-серверной архитектуры.
5.3. Столбцы и строки. • Столбцы как структура данных. • 4 основные инструкции для пользователя при работе со строками.
5.4. Понимание таблиц. • Таблицы связей.
5.5. Инструкция SELECT. • Описание работы секций SELECT и FROM.
5.6. Фильтрация и выборка. • Фильтрация по столбцам. • Фильтрация по строкам.
5.7. Использование функций. • Применение функций в запросе.
5.8. Неизвестные значения. • Недостаток реляционных баз данных. • Способы устранения недостатков реляционных баз данных.
5.9. Сортировка. • Способы управления порядком передачи результата.
5.10. Отсечка. • Понятие «отсечка» и примеры применения.
5.11. Устранение дубликатов. • Как работать с дубликатами.
5.12. Порядок операций в запросе. • Важность порядка операций в запросе.
5.13. Трансформирующие операции. • Знакомство с трансформирующими операциями. • Понимание чем является каждая строка.
5.14. Агрегация. • Устройство агрегатных функций. • Различие агрегатных и скалярных функций.
5.15. Группировка.

• Важность использования группировки. • Практическое применение группировки.
5.16. Фильтрация групп. • Операция HAVING. • Примеры решения задач.
6. МОДЕЛИРОВАНИЕ ПРОЦЕССОВ
6.1. Понимание процессов и потоков данных. • Модели предметной области. • ER-модель. • Основные понятия: объект, сущность, связь, атрибут. • Типы связей.
6.2. Моделирование процессов и потоков данных. • Диаграмма потоков данных. • Нотация Йордона-Де Марко. • Понятия «процесс», «накопитель данных», «поток данных». • Нотация Гейна-Сарсона. • Уровни детализации.
6.3. Описание архитектуры решения. • Структурные UML-диаграммы.
6.4. Виды архитектур и их компоненты. • Использование файл-серверной и клиент-серверной архитектуры. • Выбор сертификата для шифрования. • Конвейер бизнес-аналитики. • Репликация данных.
7. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
7.1. Понимание тестирования ПО. • Цель тестирования. • Модель качества продукта. • Верификация и валидация. • Среда разработки.
7.2. Виды тестирования. • Разновидности видов тестирования. • Особенности различных видов тестирования.
7.3. Описание тест-кейсов. • Тест-дизайн. • «Минорные» тесты. • Чек-лист для тестирования продуктов.
7.4. Проведение тестирования. • Идентификатор. • Приоритеты проведения тестирования. • Описание шаблона тест-кейса.
7.5. Документирование ПМИ и ПСИ. • Описание тестовой документации. • Этапы тестирования. • Само тестирование. • Спецификация тестирования.

Пример задания итоговой аттестации

Задание №1. Создайте ER-модель, показывающую сущности и их взаимосвязи.

Описание задачи:

Вам поставили задачу разработать ПО для авиакомпании. Начав строить модель предметной области, вы выделили сущности:

- пассажир;
- билет;
- рейс (маршрут с присвоенным ему обозначением);
- самолёт (воздушное судно, имеющееся в распоряжении у авиакомпании).

1. Используйте любой инструмент для построения ER-модели и представьте три варианта решения:

- только с одной связью «многие-ко-многим»;
- с двумя связями «многие-ко-многим»;
- с тремя связями «многие-ко-многим».

2. По каждому варианту решения перечислите его недостатки и ограничения.

3. Из этих трёх вариантов выберите лучший и доработайте его (при помощи дополнительных сущностей), заменив связи «многие-ко-многим» на пары связей «1:M» + «M:1».

4. В качестве результата необходимо представить:

- три схемы со связями "M:M".
- комментарии к каждой схеме (недостатки и ограничения).
- итоговую схему с дополнительными сущностями.

Используйте любую нотацию («Воронья лапка», Чена, Баркера и т.д.).

Задание № 2.

1. Выберите правильные ответы из предложенных вариантов.

2. Обоснуйте, почему Вы считаете правильным именно этот вариант.

3. Обоснуйте, почему Вы считаете остальные варианты неверными.

Описание задачи:

Отметьте три наиболее важные цели тестирования:

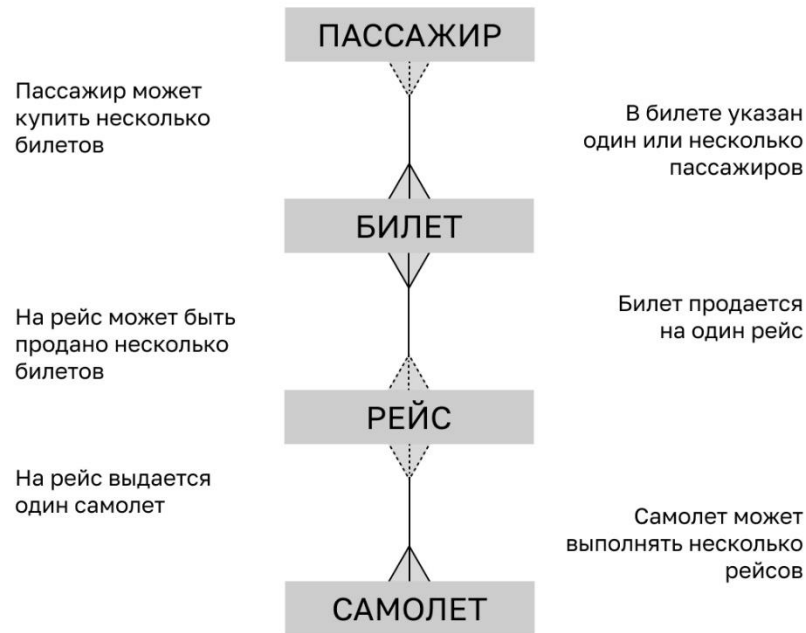
1. Исправить ошибки в ПО.
2. Минимизировать риски при внедрении ПО.
3. Обеспечить разработчиков информацией, которую они смогут использовать для исправления ошибок.
4. Обеспечить менеджеров информацией, которую они смогут использовать для оценки рисков использования ПО.
5. Оценить соответствие ПО сформулированным требованиям.
6. Оценить соответствие ПО не сформулированным, но подразумеваемым требованиям.

Пример решения итоговых заданий:

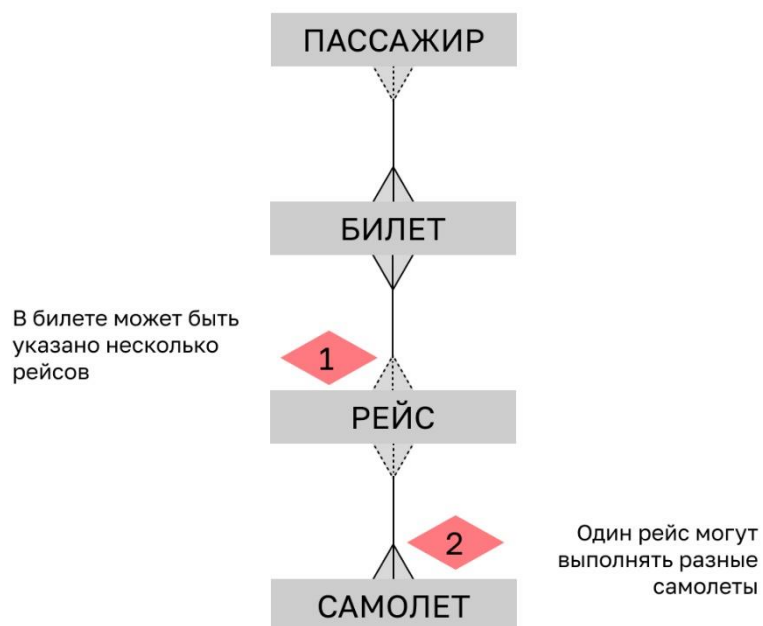
Задание № 1.

1. ER-модели:

- ER-модель с одной связью «многие-ко-многим»:



- ER-модель с двумя и тремя связями «многие-ко-многим»:



2. Недостатки и ограничения по каждому решению:

Недостатком решения с одной связью М:М будет необходимость оформлять несколько отдельных билетов для составного маршрута. Второй недостаток — невозможна работа одного самолёта на разных маршрутах.

Вторую связь М:М имеет смысл поставить между билетом и рейсом, чтобы можно было оформить билет на сложный маршрут, состоящий из нескольких перелётов.

Третья связь М:М позволит авиакомпании ставить один самолёт на разные маршруты. Дополнив решение с тремя связями М:М промежуточными сущностями, получим наиболее удобную модель.

3. Итоговая схема с дополнительными сущностями:



Задание № 2.

1. Правильные ответы из предложенных вариантов: 2, 3, 4.

2. Вышеуказанные варианты являются правильными, потому что:

- вариант 2: это главная цель тестирования – выявить и оценить риски серьёзных проблем при эксплуатации ПО. Тестирование помогает выявлять такие ситуации и обеспечивать разработчиков информацией, которую они смогут использовать для исправления ошибок, а также обеспечивать менеджеров информацией, которую они смогут использовать для оценки рисков использования ПО;
- варианты 3, 4: для выработки решения разработчикам и менеджерам нужна подробная информация (в связке с вариантом 2 по выявленным рискам).

3. Остальные варианты не правильные, потому что:

- вариант 1: исправление ошибок – не цель, а один из эффектов, к которому приводит тестирование. Программа может поставляться и эксплуатироваться с имеющимися ошибками, даже если и разработчик, и заказчик в курсе наличия ошибок. Некоторые ошибки исправить невозможно, другие ошибки обойдутся в исправлении слишком дорого, третьи ошибки настолько незначительны, что до их исправления никогда не дойдут. Ценность ПО определяется не наличием ошибок.
- варианты 5, 6: несоответствие требованиям – это лишь один из возможных рисков.