

**ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«ТРАНСИНФО»**



**«УТВЕРЖДАЮ»
ООО «ТРАНСИНФО»**

Салафаниди Г.С.

«10» февраля 2023 г.

**Дополнительная профессиональная программа повышения
квалификации «Введение в нагрузочное тестирование»**

категория обучающихся: лица, имеющие или получающие высшее образование

трудоемкость: 41 академический час

срок реализации программы: 12 дней

Автор-составитель образовательной программы:
Салафаниди Гоар Самвеловна

г. Краснодар, 2023 г.

I. РАЗДЕЛ «КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ОБРАЗОВАНИЯ»

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная профессиональная программа повышения квалификации «Введение в нагрузочное тестирование» (далее по тексту - программа) направлена на совершенствование компетенции, необходимой для профессиональной деятельности, и (или) повышение профессионального уровня в рамках имеющейся квалификации инженеров-тестировщиков программных обеспечений.

Настоящая программа разработана в соответствии с:

- Федеральным законом от 29.12.2012 N 273-ФЗ "Об образовании в Российской Федерации";
- Приказом Министерства образования и науки Российской Федерации (Минобрнауки России) от 1 июля 2013 г. N 499 г. Москва "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам";
- Приказом Министерства образования и науки РФ от 19 сентября 2017 г. N 929 "Об утверждении федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника";
- Приказ Минтруда России от 02.08.2021 N 531н "Об утверждении профессионального стандарта "Специалист по тестированию в области информационных технологий";
- Приказом Министерства образования и науки РФ от 23 августа 2017 г. N 816 "Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ";
- иными нормативно-правовыми актами Российской Федерации в сфере образования.

Категория слушателей

К освоению дополнительной профессиональной программы допускаются лица, имеющие высшее образование, а также лица, получающие высшее образование по направлению подготовки «Информатика и вычислительная техника».

Документ, выдаваемый по окончании обучения по программе повышения квалификации и прохождении слушателями итоговой аттестации:

- слушателям, успешно прошедшим итоговую аттестацию, выдается **удостоверение о повышении квалификации установленного организацией образца;**
- слушателям, не прошедшим итоговую аттестацию, выдается справка об обучении установленного организацией образца.

2. ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

Цель: сформировать у слушателей целостные, систематизированные, комплексные знания и профессиональные компетенции, необходимые для выполнения трудовой функции по проведению тестирования ПО и исследованию результатов. Получение знаний и основ практики в области тестирования ПО по разработанным тестовым случаям.

Задачи: формирование совокупности комплексных представлений о выполнении комплекса работ по тестированию ПО.

3. УЧЕБНЫЙ ПЛАН

№ пп	Наименование раздела (урока)	Общее количество во ак. часов:	Виды учебной работы		Форма контроля (аттестации)
			Теоретические занятия	Практические занятия	
1.	Вводный урок.	1	1	-	-

2.	Процессы разработки ПО, место тестирования в нем, виды тестирования.	5	3	2	-
3.	Цели нагрузочного тестирования и виды тестов.	3	2	1	-
4.	План НТ. Постановка задачи на НТ. МНТ.	5	2	3	-
5.	Архитектура LoadRunner. Запись первого скрипта. Первая параметризация.	3	2	-	1 Дифзачет (промежуточная аттестация)
6.	Параметризация, корреляция, верификация.	3	2	-	1 Дифзачет (промежуточная аттестация)
7.	Завершение скрипта.	3	2	-	1 Дифзачет (промежуточная аттестация)
8.	Структура скриптов. Профиль и пейсинг. Создание теста в Controller.	3	2	-	1 Дифзачет (промежуточная аттестация)
9.	Разбор проблем со скриптом. Анализ результатов в Analysis.	3	2	-	1 Дифзачет (промежуточная аттестация)
10.	Структура результатов теста. Мониторинг и метрики.	3	2	-	1 Дифзачет (промежуточная аттестация)
11.	Основы БД. Отчеты. Основы загрузок.	5	2	3	-
12.	Итоговая аттестация.	4	-	-	Экзамен
Итого:		41 ак.ч.			

4. СОДЕРЖАНИЕ ПРОГРАММЫ

Программа состоит из следующих разделов (уроков): 1) Вводный урок. 2) Процессы разработки ПО, место тестирования в нем, виды тестирования. 3) Цели нагрузочного тестирования и виды тестов. 4) План НТ. Постановка задачи на НТ. МНТ. 5) Архитектура LoadRunner. Запись первого скрипта. Первая параметризация. 6) Параметризация, корреляция, верификация. 7) Завершение скрипта. 8) Структура скриптов. Профиль и пейсинг. Создание теста в Controller. 9) Разбор проблем со скриптом. Анализ результатов в Analysis. 10) Структура результатов теста. Мониторинг и метрики. 11) Основы БД. Отчеты. Основы загрузок, трудоемкость и последовательность освоения которых приводится в «Учебном плане» настоящей программы.

Трудоемкость программы 41 академический час (1 час = 45 минут), из которых на теоретическую часть отводится 22 часа, на практическую часть – 9 часов, на промежуточную аттестацию - 6 часов, на итоговую аттестацию – 4 часа.

5. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

В результате освоения программы повышения квалификации слушатель должен в соответствии с Приказом Министерства образования и науки РФ от 19 сентября 2017 г. N 929 "Об утверждении федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника" **обладать общепрофессиональными компетенциями:**

ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;

ОПК-3. Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;

ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем;

ОПК-7. Способен участвовать в настройке и наладке программно-аппаратных комплексов;

ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

ОПК-9. Способен осваивать методики использования программных средств для решения практических задач.

В соответствии с пунктом 3.4. Приказа Министерства образования и науки РФ от 19 сентября 2017 г. N 929 **профессиональные компетенции** определяются организацией самостоятельно на основе профессионального стандарта "Специалист по тестированию в области информационных технологий"

ПК-1 Проведение тестирования ПО по разработанным тестовым случаям

ПК-2 Определение и описание тестовых случаев для выполнения процесса тестирования ПО, включая разработку автотестов

ПК-3 Разработка тестовых случаев, проведение тестирования ПО и исследование результатов

В результате освоения программы формируются **знания и умения** в соответствии с Приказом Министерства труда и социальной защиты РФ от 5 октября 2015 г. N 686н "Об утверждении профессионального стандарта "Специалист по администрированию сетевых устройств информационно-коммуникационных систем":

умения:

- Составлять сценарии поведения пользователей ПО;
- Выполнять интеграционное и модульное тестирование ПО;
- Выполнять статическое тестирование ПО;
- Использовать специальное ПО для автоматизированного тестирования ПО при необходимости;
- Составлять отчет о проведении тестирования ПО по разработанным тестовым случаям;
- Взаимодействовать с членами команды разработчиков ПО;
- Использовать системы автоматизированного тестирования ПО

знания:

- Техники тестирования ПО, базирующиеся на интуиции и опыте инженера;
- Техники тестирования ПО, базирующиеся на спецификации;
- Техники тестирования ПО, ориентированные на код;
- Тестирование ПО, ориентированное на дефекты;
- Техники тестирования ПО, базирующиеся на условиях использования;

- Тестирование ПО, базирующееся на надежности инженерного процесса;
- Техники тестирования ПО, базирующиеся на природе приложения;
- Стандарты оформления кода для используемых языков программирования;
- Основные термины и сокращения, используемые в технической документации и принятые в организации;
- Основы алгоритмизации и программирования;
- Жизненный цикл программного продукта

трудовые действия:

- Выполнение начальных настроек для проведения тестирования ПО;
- Выполнение необходимых видов тестирования ПО в соответствии с планом тестирования;
- Проведение автоматизированного тестирования ПО при необходимости;
- Составление статистики выполнения тестов;
- Проведение анализа полученных результатов тестирования ПО по разработанным тестовым случаям на соответствие ожидаемым результатам;
- Оптимизация тестовых наборов;
- Составление новых тестовых случаев и повторение тестирования при необходимости;
- Формирование и представление отчетности о проведенном тестировании ПО в соответствии с установленными регламентами.

**II. РАЗДЕЛ «КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ, ВКЛЮЧАЯ ФОРМЫ АТТЕСТАЦИЙ»
1. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК НА 2023 Г.:**

Трудоемкость программы: 41 академический час.

Режим занятий: 1-5 часов в день.

Продолжительность обучения: 12 дней

№ п/п	Наименование разделов (уроков)	Всего ак.ч.	Количество дней на освоение программы														
			1	2	3	4	5	6	7	8	9	10	11	12			
1.	Вводный урок.	1	1														
2.	Процессы разработки ПО, место тестирования в нем, виды тестирования.	5		5													
3.	Цели нагрузочного тестирования и виды тестов.	3			3												
4.	План НТ. Постановка задачи на НТ. МНТ.	5				5											
5.	Архитектура LoadRunner. Запись первого скрипта. Первая параметризация.	3					3										
6.	Параметризация, корреляция, верификация.	3						3									
7.	Завершение скрипта.	3							3								
8.	Структура скриптов. Профиль и пейсинг. Создание теста в Controller.	3								3							
9.	Разбор проблем со скриптом. Анализ результатов в Analysis.	3									3						
10.	Структура результатов теста. Мониторинг и метрики.	3											3				
11.	Основы БД. Отчеты. Основы заглашек.	5												5			
12.	Итоговая аттестация (экзамен)	4															4
Итого			41 ак.ч., 12 дней по 1-5 ак. часов в день														

2. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Требования к обучающимся: Программа предназначена для лиц, имеющих или получающих высшее образование по направлению подготовки Информатика и вычислительная техника.

Форма обучения: заочная с применением дистанционных образовательных технологий и электронного обучения

Формы организации занятий:

- ✓ Онлайн вебинары через платформу для проведения вебинаров GetCourse - <https://diuniversity.getcourse.ru/>
- ✓ Самостоятельное освоение предзаписей видеолекций и учебно-методических материалов во внеучебное время на адаптированной образовательной платформе GetCourse - <https://diuniversity.getcourse.ru/>.
- ✓ Выполнение заданий по промежуточной и итоговой аттестации, представленных на адаптированной образовательной платформе GetCourse - <https://diuniversity.getcourse.ru/>.

Формы контроля и диагностики результатов:

- Задания по промежуточной аттестации
- Устный экзамен по итоговой аттестации
- Учет посещаемости адаптированной образовательной платформы и освоения модулей образовательной программы

Для реализации программы задействован следующий кадровый потенциал:

- Преподаватели – обеспечивается необходимый уровень компетенции педагогического состава в соответствии с Приказом Министерства здравоохранения и социального развития РФ от 26 августа 2010 г. N 761н "Об утверждении Единого квалификационного справочника должностей руководителей, специалистов и служащих, раздел "Квалификационные характеристики должностей работников образования";
- Административный персонал – обеспечивает условия для эффективной работы педагогического коллектива, осуществляет контроль и текущую организационную работу
- Информационно-технологический персонал - обеспечивает функционирование информационной структуры (включая ремонт техники, оборудования, иного технического обеспечения образовательного процесса, поддержание сайта и т.п.)

Материально-технические условия реализации программы

Обучение с применением дистанционных образовательных технологий и/или электронного обучения по программе основывается на дистанционных занятиях и самостоятельной работе обучающихся. Реализация программы обеспечивается доступом каждого обучающегося к учебным материалам - текстовой, графической, аудио-, видео- информации по программе через сеть «Интернет» в режиме 24 часа в сутки 7 дней в неделю без учета объемов потребляемого трафика за исключением перерывов для проведения необходимых ремонтных и профилактических работ при обеспечении совокупной доступности услуг посредством регистрации и предоставления индивидуальных логина и пароля к адаптированной образовательной платформе GetCourse по адресу ссылки <https://diuniversity.getcourse.ru/>.

Вебинары проводятся через адаптированную образовательную платформу GetCourse - https://diuniversity.getcourse.ru в соответствии с расписанием занятий.

Условия освоения программы обучающимися:

При освоении учебного материала посредством электронной информационно-образовательной среды организация доводит до поступающих информацию об обязанностях обучающихся при освоении программы использовать свой персональный компьютер/ноутбук с доступом к сети интернет в соответствии с рекомендованными техническими параметрами:

1. Наличие персонального компьютера, вебкамеры, наушников, динамика.
2. Наличие Интернета в момент запуска видеурока/вебинара и далее.
3. Наличие на ПК программы для просмотра видеолекций и организации видеосвязи.

3. ФОРМЫ АТТЕСТАЦИИ

Формы контроля знаний и требования к его проведению:

Текущий контроль знаний проводится посредством учета и контроля посещаемости системы электронного обучения, фиксации на образовательной платформе освоения уроков программы (преподаватель через свой аккаунт на образовательной платформе GetCourse просматривает фиксацию посещаемости и какие уроки модуля освоил обучающийся).

Промежуточная аттестация: проводится в форме выполнения заданий.

Итоговая аттестация - экзамен будет проводится преподавателем курса в форме устного собеседования с каждым слушателем индивидуально.

Критерии оценивания промежуточной и итоговой аттестации – дифференцированного зачета и экзамена:

Оценка зачета	Требования к знаниям
Зачтено / 5 «отлично»	Глубокие теоретические знания программы Способность применять теоретические знания к практическим ситуациям
Зачтено / 4 «хорошо»	Твердые теоретические знания программы Способность применять теоретические знания к практическим ситуациям
Зачтено / 3 «удовлетворительно»	Основные теоретические знания программы Испытывает затруднения при применении теоретических знаний к практическим ситуациям
Не зачтено / 2 «неудовлетворительно»	Значительные пробелы в теоретических знаниях программы Нет способности применять теоретические знания к практическим ситуациям

4. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ:

Задания по промежуточной аттестации также указаны в разделе «Рабочая программа модулей» образовательной программы.

Задание №1

1. Продолжаем учебный проект: на основе постановки задачи пишем методику НТ, используя примеры и шаблон. Все перечисленные доки представлены в предыдущем ДЗ.

2. Записываем первый скрипт. Сценарий следующий:

- Открытие стартовой страницы
- Логин
- Открытие страницы поиска тура
- Выполнение поиск тура с билетом туда и обратно
- Выбор из результатов поиска случайных рейсов туда и обратно
- Завершение бронирования с оплатой
- Переход на домашнюю страницу
- Открытие списка бронирований
- Удаление созданной брони
- Выполнение логаута

3. Требования к записи скрипта:

- Настраиваем фильтр для избавления от лишних запросов к сторонним сайтам
- Транзакции расставляются во время записи: перед началом каждого шага сценария открываем, после отрисовки всех элементов сайта - закрываем транзакцию. Каждая транзакция соответствует шагу сценария (см. выше)

- Разбиваем скрипт на 3 экшна: Login, Action, Logout
- Выполняем простую параметризацию (HOST, PORT)
- Расставляем think_times: 5сек для простых переходов по формам, 10сек - перед операциями, требующими небольшого ввода данных (1-2 поля ввода, выбор checkbox) и 15сек - там, где заполняется много полей ввода

Задание №2

Задача на создание генератора юзеров в системе

1 Создаем в excel или программой пул данных в формате CSV (.dat файл)

2 Делаем скрипт для генерации юзеров в системе на основе этого файла

a. Обычный способ - через форму создания юзера в приложении с помощью LoadRunner скрипта

b. Способ повышенной сложности - разобраться в способе хранения данных в системе и генерить пользователей программно напрямую в хранилище

Задача на доработку скрипта, записанного ранее в предыдущем ДЗ:

1 Выполняем параметризацию на основе единого созданного ранее файла с данными юзеров (login, password, firstname, ..., cardnumber, expdate)

2 Выполняем параметризацию дат отлета/прилета

3 Выполняем параметризацию выбора класса и места в салоне (рандомно)

4 Параметризация названия городов отлета и прилета будет делаться на следующем уроке

5 Выполняем корреляцию идентификаторов рейсов (внимание: пока значения outboundflight, returnflight берем первыми из списка!)

6 Делаем корреляцию параметра flightID (пока при условии, что у юзера только одна бронь в списке)

7 Делаем верификацию на логин, бронирование и удаление брони

8 Проверяем верификацию, подавая неверные данные, чтоб убедиться в ее работоспособности

9 Проверяем прогоном скрипта в несколько итераций с различными учетками

Не забываем про написание методики!!!

Задание №3

1 Нужно подготовить первую версию МНТ, включая профиль. Подумайте, как вам покрыть его скриптами.

2 Отладьте скрипты для генерации учеток юзеров в системе. Для тех, кто делает корреляцию только одного номера брони в запросе ее удаления, нужно придумать механизм подготовки всех учеток к тесту: чтоб у всех был пустой список броней. Варианты: автоматизировать скриптом LR процесс удаления всех учеток (синтаксис C позволяет работать с командами консоли, можно удалять нужные файлы) и последующее создание свежих учеток. Второй вариант более интеллектуальный: любым языком программирования проверять хранилище учеток и в файлах удалять существующие брони.

3 Подготовьте рабочую версию скрипта по сценарию. Напоминаю требования к скрипту:

a. Расставляем think_times

b. Расставляем верификацию на все транзакции

c. Разделяем скрипт на три экшена: Action, Login, Logout, все они в блоке RUN

d. Выполняем параметризацию по двум сущностям: стенд (URL, Port) и пользователь (login, password, firstname, ..., cardnum, expdate) из 2 DAT-файлов

e. Выполняем корреляцию (используем параметр-массив, где это нужно, выбираем из него рандомные значения в новый параметр)

f. Делаем рандомный выбор названия городов отлета и прилета. Делаем выбор массива-параметра из запроса открытия страницы поиска, вытаскивая оттуда все города (возможно, придется использовать регулярку). Далее в цикле Do-While выбираем в 2 параметра рандомные значения при помощи lr_save_string(), сравнивать их, если совпадают, то повторяем выборку.

g. Делаем рандомный выбор класса салона и места, тут все просто, ничего сравнивать не надо, просто получаем параметр-массив и из него рандомно берем

значение в новый параметр.

h. Сделайте корреляцию параметра flightID. По умолчанию достаточно пока сделать для случая удаления единственной брони. Кто чувствует в себе силы и время есть, попробуйте сделать запись этой операции с галочкой “Use only web_custom_request”, тем самым посмотрите, как меняется вид запроса с увеличением кол-ва броней в списке. На основе этой информации нужно будет придумать механизм генерации тела запроса (Body=...).

i. Добавьте ручной вывод в лог. Пример вида лога, к которому нужно стремиться: <https://disk.yandex.ru/d/S658x4r6ifJsqw>

j. Проверяем прогоном скрипта на 10 итераций, с разными учетками

k. Скрипт выкладываем в свои папки в облако

l. Формат названия скрипта: <YourFamily>_<HomeWorkNumber>_<Date>, например: “Minaev_6_20210419”.

Задание № 4

- 1 Доделывайте скрипт согласно прежним заданиям. Выкладывайте, сообщайте о готовности.
- 2 Рассчитайте профиль НТ на основе статистики из постановки задачи. Создайте необходимые сценарии, которыми будет покрываться профиль НТ. Вычислите raising, исходя из профиля и кол-ва юзеров, которые будут эмулировать нагрузку.
- 3 Занесите результаты расчетов в соответствующие разделы МНТ. Отправьте МНТ преподавателю.
- 4 На основе расчетов создайте тест поиска максимума в Controller. Длины ступеней и их высоту рассчитайте, исходя из числа юзеров и настроенных raising.
- 5 Выполните первые тесты, требуемые по МНТ. В приоритете поиск максимума (maxperf), потом стабильности.
- 6 Анализ результатов будем учиться делать на следующем уроке.

Задание №5

- 1 Заканчивайте МНТ, направьте МНТ преподавателю. На следующем уроке разберем проблемы с МНТ.
- 2 На основе МНТ проведите тесты (макс.перф по всем правилам и стабильность). Тесты проводятся последовательно: провели один - проанализировали, поняли, какие характеристики нужны в следующем тесте и т.д.
- 3 Каждый результативный тест проанализируйте и создайте в нем графики, на основе которых вы делаете выводы по тесту. Сразу готовьте их для отчета (русифицируйте, назовите осознанно, разбейте, чтоб было не более 3-5 кривых, настройте granularity).
- 4 Каждый такой анализ сохраните: это будет Session, назовите его Family_YYYYMMDD_ННММ, он представляет собой отдельную папку с исполняемым файлом *.lra, которую можно положить в папку с основным результатом теста. Именно эту папку с сохраненным и нужно выложить архивом в облако, чтоб я мог посмотреть.

Задание №6

1. Завершите МНТ и проведите тесты по вашей методике. Преподавателю отправляйте проанализированные результаты с сохраненными графиками, которые можно вставлять в отчет.
2. Изучите шаблон и примеры отчетов (см. ниже). По результатам ваших НТ начните готовить отчет по полному шаблону (не экспресс). На следующем уроке преподаватель прокомментирует форматы отчетов:

Шаблон полного отчета: https://disk.yandex.ru/i/nEc7-OBvI_tINg

Шаблон экспресс-отчета (краткий): <https://disk.yandex.ru/i/qKHmpdEu3wo8dw>

Пример полного отчета: <https://disk.yandex.ru/i/dX6vPlsctEGtEg>

Пример экспресс-отчета: https://disk.yandex.ru/i/_zKRLCKBn12qYw

Примерные темы к собеседованию – итоговой аттестации (экзамену)

1. Введение в процесс разработки ПО

Чем промышленный процесс разработки отличается от программирования дома

Роли в промышленной разработке
Виды процессов разработки
Контроль качества в процессах разработки
Тестирование в контроле качества

2. Место НТ в процессе разработки

3. Введение в методологию НТ

Типовой план работ по НТ (основные этапы)
Сбор целей, виды тестов и соответствие между ними
Поиск максимальной производительности
Проверка стабильности
Проверка стрессоустойчивости
Проверка отказоустойчивости
Проверка масштабируемости
Объемное тестирование

4. Анализ архитектуры систем

Промышленный и тестовый стенды
Источники нагрузки
Смежные системы
Протоколы (сетевые: какие бывают, какая классификация и стеки, подробнее про HTTP)

5. Расчет профиля НТ

На основе аналитики
На основе статистики (какие источники, за какие интервалы собирают статистику и как она превращается в профиль)
Дневные и ночные профили
Час пиковой нагрузки (что это)

6. Постановка задачи на НТ (что это, основные разделы). МНТ (что это, для чего, основные разделы)

7. LoadRunner

Архитектура LR (компоненты, связь, как устроены скрипты, тесты, результаты)
7.1 Разработка скриптов в VUGEN
Критерии готовности скриптов к НТ (параметризация, корреляция, верификация)
Параметры в LR (одинарные, массивы, как создать свои вручную)
Think-times (для чего, как выставлять)
Автоматические средства корреляции LR (плюсы и минусы)
Ручная корреляция (основные функции и принцип их работы)
Параметризация различных типов (встроенные типы параметров)
Верификация через web_reg_save_param и через web_reg_find
7.2 Работа в Controller
Составление сценария по профилю из МНТ
Pasing (что это, как рассчитывать)
Проверка корректности профиля тестовыми запусками
7.3 Основные возможности Analysis

8. Метрики, собираемые во время НТ

9. Базы данных

Oracle, AWR, планы запросов

10. Введение в заглушки

Назначение
Типы и виды

11. Создание отчетов HT

Виды отчетов
Основные разделы
Связь с МНТ

12. Дополнительно: вспомните SQL (порешайте на sql-ex.ru)

Операции, какие есть (Select, Insert, Update, Delete...)
Join (какие бывают) и Union - что это отличия
Вложенные запросы
Агрегатные функции и группировки (Group by)
Where и Having

5. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Методическое обеспечение программы включает:

- Предзаписи видеолекций, методические пособия, презентация к программе, которые представлены на образовательной платформе GetCourse.
- Оценочные материалы по промежуточной и итоговой аттестациям.
- Материалы для практических занятий, представленные на образовательной платформе GetCourse.
- Перечень литературы для самостоятельной проработки тем программы.
- указанные методические материалы размещены на адаптированной образовательной платформе GetCourse - <https://diuniversity.getcourse.ru>

6. РАБОЧИЕ ПРОГРАММЫ РАЗДЕЛОВ РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 1 «ВВОДНЫЙ УРОК»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия (вебинар в режиме онлайн на образовательной платформе GetCourse)	Практические занятия	
1	Вводный урок.	1 ак.ч.	1 ак.ч.	-	-

Краткое содержание:

Урок является вводным по программе и состоится в форме вебинара. На вебинаре слушатели познакомятся с преподавателем программы и остальными слушателями группы, узнают о правилах и порядке, которые действуют в ООО «Трансинфо» на период освоения программы.

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 2 «ПРОЦЕССЫ РАЗРАБОТКИ ПО, МЕСТО ТЕСТИРОВАНИЯ В НЕМ, ВИДЫ ТЕСТИРОВАНИЯ»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Виды учебной работы
------	-----------------------------	---------------------

		Общее количество часов:	Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной платформе GetCourse	Форма контроля
1.	Процессы разработки ПО, место тестирования в нем, виды тестирования.	5 ак.ч.	3 ак.ч.	2 ак.ч.	-

Краткое содержание:

Тестирование производительности в AGILE.
 Задачи, которые решает тестирование производительности.
 Гибкая методология.
 Тестирование производительности при гибкой разработке.
 Оптимизация на уровне модулей.
 Начальные условия.
 Последовательности действий.
 Оптимизация на уровне компонент.
 Оптимизация на уровне приложения.
 Три модели: уровни успеха.
 Распределение компетенций.
 Полное погружение.
 Гибкое тестирование производительности и снижение рисков.

Практическое занятие: изучение и анализ предлагаемых к уроку материалов по нагрузочному тестированию в гибких методологиях.

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 3 «ЦЕЛИ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ И ВИДЫ ТЕСТОВ»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной платформе	

				ме GetCourse	
1.	Цели нагрузочного тестирования и виды тестов.	3 ак.ч.	2 ак.ч.	1 ак.ч.	-

Краткое содержание:

Терминология в нагрузочном тестировании
 Нагрузочное тестирование или Тестирование производительности
 Интенсивность выполнения операции (Operation Intensity)
 Итерация (Iteration)
 Виртуальный пользователь (Virtual User)

Практическое занятие: изучение и анализ предлагаемых к уроку материалов.

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 4 «ПЛАН НТ. ПОСТАНОВКА ЗАДАЧИ НА НТ. МНТ»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной платформе GetCourse	
1.	План НТ. Постановка задачи на НТ. МНТ.	5 ак.ч.	2 ак.ч.	3 ак.ч.	-

Краткое содержание:

Процесс инвентаризации проекта нагрузочного тестирования.
 Методика нагрузочного тестирования.
 Постановка задачи по нагрузочному тестированию АС «Наименование системы»
 Постановка задачи на НТ WebTours

Практическое занятие:

1. Изучите шаблон постановки задачи на НТ:

<https://disk.yandex.ru/i/j9pBbpMukcVfMw>

2. Изучите шаблон МНТ:

<https://disk.yandex.ru/i/5Cwg1zui0Brg7A>

В помощь к МНТ - глоссарий: <https://disk.yandex.ru/i/8DaouArlOpjPng>

3. Посмотрите пример МНТ:

<https://disk.yandex.ru/d/NTsROiDoKdlK7A>

Еще один пример МНТ: <https://disk.yandex.ru/i/ItJ-artKDYXb3Q>

4. Начинаем учебный проект НТ. По ссылке постановка задачи на НТ системы WebTours. Играем в игру: я заказчик, каждый из вас - специалист по НТ, пришедший на проект. Вы получили постановку задачи. Если есть вопросы - пишите их в общий чат, чтоб «заказчик» не отвечал на однотипные вопросы

<https://disk.yandex.ru/i/qRg-RhvgxZbCoA>

5. Поставьте себе на машины LoadRunner. Дистриб тут:

https://disk.yandex.ru/d/U_nH9yq03PnftQ

6. Поставьте себе учебное приложение WebTours (будем его нагружать). Дистриб тут:

<https://disk.yandex.ru/d/AIQNksCzjMJW9w>

7. У кого есть время, можете попробовать пройти туториал по LR:

<https://disk.yandex.ru/i/YrJL7F7Y9t1yuw>

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 5 «АРХИТЕКТУРА LOADRUNNER. ЗАПИСЬ ПЕРВОГО СКРИПТА. ПЕРВАЯ ПАРАМЕТРИЗАЦИЯ»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной платформе GetCourse	
1.	Архитектура LoadRunner. Запись первого скрипта. Первая параметризация.	3 ак.ч.	2 ак.ч.	-	1 ак.ч. Дифзачет (промежуточная аттестация)

Краткое содержание:

Как устроена архитектура LoadRunner. Принцип его работы.

Инструменты нагрузочного тестирования.

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

Промежуточная аттестация (дифференцированный зачет) – задание:

1. Продолжаем учебный проект: на основе постановки задачи пишем методику НТ, используя примеры и шаблон. Все перечисленные доки представлены в предыдущем ДЗ.

2. Записываем первый скрипт. Сценарий следующий:

- Открытие стартовой страницы
- Логин
- Открытие страницы поиска тура
- Выполнение поиск тура с билетом туда и обратно
- Выбор из результатов поиска случайных рейсов туда и обратно
- Завершение бронирования с оплатой
- Переход на домашнюю страницу
- Открытие списка бронирований
- Удаление созданной брони
- Выполнение логаута

3. Требования к записи скрипта:

- Настраиваем фильтр для избавления от лишних запросов к сторонним сайтам
- Транзакции расставляются во время записи: перед началом каждого шага сценария открываем, после отрисовки всех элементов сайта - закрываем транзакцию. Каждая транзакция соответствует шагу сценария (см. выше)
- Разбиваем скрипт на 3 экшна: Login, Action, Logout
- Выполняем простую параметризацию (HOST, PORT)
- Расставляем think_times: 5сек для простых переходов по формам, 10сек - перед операциями, требующими небольшого ввода данных (1-2 поля ввода, выбор checkbox) и 15сек - там, где заполняется много полей ввода

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 6 «ПАРАМЕТРИЗАЦИЯ, КОРРЕЛЯЦИЯ, ВЕРИФИКАЦИЯ»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной платформе GetCourse	
1.	Параметризация, корреляция, верификация	3 ак.ч.	2 ак.ч.	-	1 ак.ч. Дифзачет (промежуточная аттестация)

Краткое содержание:

Параметризация, корреляция, верификация
Постановка задач.

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

Промежуточная аттестация (дифференцированный зачет) – задание:

Задача на создание генератора юзеров в системе

1 Создаем в excel или программой пул данных в формате CSV (.dat файл)

2 Делаем скрипт для генерации юзеров в системе на основе этого файла

a. Обычный способ - через форму создания юзера в приложении с помощью LoadRunner скрипта

b. Способ повышенной сложности - разобраться в способе хранения данных в системе и генерить пользователей программно напрямую в хранилище

Задача на доработку скрипта, записанного ранее в предыдущем ДЗ:

1 Выполняем параметризацию на основе единого созданного ранее файла с данными юзеров (login, password, firstname, ..., cardnumber, expdate)

2 Выполняем параметризацию дат отлета/прилета

3 Выполняем параметризацию выбора класса и места в салоне (рандомно)

4 Параметризация названия городов отлета и прилета будет делаться на следующем уроке

5 Выполняем корреляцию идентификаторов рейсов (внимание: пока значения outboundflight, returnflight берем первыми из списка!)

6 Делаем корреляцию параметра flightID (пока при условии, что у юзера только одна бронь в списке)

7 Делаем верификацию на логин, бронирование и удаление брони

8 Проверяем верификацию, подавая неверные данные, чтоб убедиться в ее работоспособности

9 Проверяем прогоном скрипта в несколько итераций с различными учетками

Не забываем про написание методики!!!

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 7 «ЗАВЕРШЕНИЕ СКРИПТА»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной платформе GetCourse	
1.	Завершение скрипта	3 ак.ч.	2 ак.ч.	-	1 ак.ч. Дифзачет (промежуточная аттестация)

Краткое содержание:

Завершение сеанса авторизации пользователя.

Сценарий тестирования

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

Промежуточная аттестация (дифференцированный зачет) – задание:

1 К следующему уроку нужно подготовить первую версию МНТ, включая профиль. Подумайте, как вам покрыть его скриптами.

2 Отладьте скрипты для генерации учеток юзеров в системе. Для тех, кто делает корреляцию только одного номера брони в запросе ее удаления, нужно придумать механизм подготовки всех учеток к тесту: чтоб у всех был пустой список броней. Варианты: автоматизировать скриптом LR процесс удаления всех учеток (синтаксис C позволяет работать с командами консоли, можно удалять нужные файлы) и последующее создание свежих учеток. Второй вариант более интеллектуальный: любым языком программирования проверять хранилище учеток и в файлах удалять существующие брони.

3 Подготовьте рабочую версию скрипта по сценарию. Напоминаю требования к скрипту:

- a. Расставляем think_times
- b. Расставляем верификацию на все транзакции
- c. Разделяем скрипт на три экшена: Action, Login, Logout, все они в блоке RUN
- d. Выполняем параметризацию по двум сущностям: стенд (URL, Port) и пользователь (login, password, firstname, ..., cardnum, expdate) из 2 DAT-файлов
- e. Выполняем корреляцию (используем параметр-массив, где это нужно, выбираем из него рандомные значения в новый параметр)
- f. Делаем рандомный выбор названия городов отлета и прилета. Делаем выбор массива-параметра из запроса открытия страницы поиска, вытаскивая оттуда все города (возможно, придется использовать регулярку). Далее в цикле Do-While выбираем в 2 параметра рандомные значения при помощи lr_save_string(), сравнивать их, если совпадают, то повторяем выборку.
- g. Делаем рандомный выбор класса салона и места, тут все просто, ничего сравнивать не надо, просто получаем параметр-массив и из него рандомно берем значение в новый параметр.
- h. Сделайте корреляцию параметра flightID. По умолчанию достаточно пока сделать для случая удаления единственной брони. Кто чувствует в себе силы и время есть, попробуйте сделать запись этой операции с галочкой “Use only web_custom_request”, тем самым посмотрите, как меняется вид запроса с увеличением кол-ва броней в списке. На основе этой информации нужно будет придумать механизм генерации тела запроса (Body=...).
- i. Добавьте ручной вывод в лог. Пример вида лога, к которому нужно стремиться: <https://disk.yandex.ru/d/S658x4r6ifJsqw>
- j. Проверяем прогоном скрипта на 10 итераций, с разными учетками
- k. Скрипт выкладываем в свои папки в облако
- l. Формат названия скрипта: <YourFamily>_<HomeWorkNumber>_<Date>, например: “Minaev_6_20210419”.

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 8 «СТРУКТУРА СКРИПТОВ. ПРОФИЛЬ И ПЕЙСИНГ. СОЗДАНИЕ ТЕСТА В CONTROLLER»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной	

				платформе GetCourse	
1.	Структура скриптов. Профиль и пейсинг. Создание теста в Controller	3 ак.ч.	2 ак.ч.	-	1 ак.ч. Дифзачет (промежуточная аттестация)

Краткое содержание:

Управление логикой скриптов – Controllers
Скрипты и сценарии НТ

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

Промежуточная аттестация (дифференцированный зачет) – задание:

- 1 Доделывайте скрипт согласно прежним заданиям. Выкладывайте, сообщайте о готовности.
- 2 Рассчитайте профиль НТ на основе статистики из постановки задачи. Создайте необходимые сценарии, которыми будет покрываться профиль НТ. Вычислите pasing, исходя из профиля и кол-ва юзеров, которые будут эмулировать нагрузку.
- 3 Занесите результаты расчетов в соответствующие разделы МНТ. Отправьте МНТ преподавателю.
- 4 На основе расчетов создайте тест поиска максимума в Controller. Длины ступеней и их высоту рассчитайте, исходя из числа юзеров и настроенных pasing.
- 5 Выполните первые тесты, требуемые по МНТ. В приоритете поиск максимума (maxperf), потом стабильности.
- 6 Анализ результатов будем учиться делать на следующем уроке.

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 9 «РАЗБОР ПРОБЛЕМ СО СКРИПТОМ. АНАЛИЗ РЕЗУЛЬТАТОВ В ANALYSIS»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной платформе GetCourse	
1.	Разбор проблем со скриптом. Анализ результатов в Analysis	3 ак.ч.	2 ак.ч.	-	1 ак.ч. Дифзачет (промежуточная аттестация)

Краткое содержание:

Пример для анализа модели использования

Пример для распределения задач

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

Промежуточная аттестация (дифференцированный зачет) – задание:

1 Закончивайте МНТ, направьте МНТ преподавателю. На следующем уроке разберем проблемы с МНТ.

2 На основе МНТ проведите тесты (макс.перф по всем правилам и стабильность). Тесты проводятся последовательно: провели один - проанализировали, поняли, какие характеристики нужны в следующем тесте и т.д.

3 Каждый результативный тест проанализируйте и создайте в нем графики, на основе которых вы делаете выводы по тесту. Сразу готовьте их для отчета (русифицируйте, назовите осознанно, разбейте, чтоб было не более 3-5 кривых, настройте granularity).

4 Каждый такой анализ сохраните: это будет Session, назовите его Family_YYYYMMDD_NHMM, он представляет собой отдельную папку с исполняемым файлом *.lra, которую можно положить в папку с основным результатом теста. Именно эту папку с сохраненным и нужно выложить архивом в облако, чтоб я мог посмотреть.

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 9 «РАЗБОР ПРОБЛЕМ СО СКРИПТОМ. АНАЛИЗ РЕЗУЛЬТАТОВ В ANALYSIS»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной платформе GetCourse	
1.	Разбор проблем со скриптом. Анализ результатов в Analysis	3 ак.ч.	2 ак.ч.	-	1 ак.ч. Дифзачет (промежуточная аттестация)

Краткое содержание:

Пример для анализа модели использования

Пример для распределения задач

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

Промежуточная аттестация (дифференцированный зачет) – задание:

1. Заканчивайте МНТ, направьте МНТ преподавателю. На следующем уроке разберем проблемы с МНТ.
2. На основе МНТ проведите тесты (макс.перф по всем правилам и стабильность). Тесты проводятся последовательно: провели один - проанализировали, поняли, какие характеристики нужны в следующем тесте и т.д.
3. Каждый результативный тест проанализируйте и создайте в нем графики, на основе которых вы делаете выводы по тесту. Сразу готовьте их для отчета (русифицируйте, назовите осознанно, разбейте, чтоб было не более 3-5 кривых, настройте granularity).
4. Каждый такой анализ сохраните: это будет Session, назовите его Family_YYYYMMDD_NHMM, он представляет собой отдельную папку с исполняемым файлом *.Ira, которую можно положить в папку с основным результатом теста. Именно эту папку с сохраненным и нужно выложить архивом в облако, чтоб я мог посмотреть.

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 10 «СТРУКТУРА РЕЗУЛЬТАТОВ ТЕСТА. МОНИТОРИНГ И МЕТРИКИ»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной платформе GetCourse	
1.	Структура результатов теста. Мониторинг и метрики	3 ак.ч.	2 ак.ч.	-	1 ак.ч. Дифзачет (промежуточная аттестация)

Краткое содержание:

Анализ использования кода при выполнении тестов
 Таблицы входных значений
 Матрицы входных значений
 Требуется ручной труд для оценки покрытия тестами
 Пример: функция от двух параметров

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

Промежуточная аттестация (дифференцированный зачет) – задание:

1. Завершите МНТ и проведите тесты по вашей методике. Преподавателю отправляйте проанализированные результаты с сохраненными графиками, которые можно вставлять в отчет.
2. Изучите шаблон и примеры отчетов (см. ниже). По результатам ваших НТ начните готовить отчет по полному шаблону (не экспресс). На следующем уроке преподаватель прокомментирует форматы отчетов:

Шаблон полного отчета: https://disk.yandex.ru/i/nEc7-OBvI_tiNg

Шаблон экспресс-отчета (краткий): <https://disk.yandex.ru/i/qKHmpdEu3wo8dw>

Пример полного отчета: <https://disk.yandex.ru/i/dX6vPlsctEGtEg>

Пример экспресс-отчета: https://disk.yandex.ru/i/_zKRLCKBn12qYw

РАБОЧАЯ ПРОГРАММА РАЗДЕЛА 11 «ОСНОВЫ БД. ОТЧЕТЫ. ОСНОВЫ ЗАГЛУШЕК»

Учебно-тематическое планирование:

№ пп	Наименование модуля, уроков	Общее количество часов:	Виды учебной работы		Форма контроля
			Теоретические занятия - вебинар в режиме онлайн на образовательной платформе GetCourse	Практические занятия - самостоятельная работа на образовательной платформе GetCourse	
1.	Основы БД. Отчеты. Основы заглушек	5 ак.ч.	2 ак.ч.	3 ак.ч.	-

Краткое содержание:

Разработка заглушек.

Когда возникает необходимость в заглушках.

Недостатки использования заглушек.

Практические занятия: для закрепления информации по теоретическим разделам “Заглушки” изучите представленные к уроку материалы

Текущий контроль знаний: проводится в форме учета посещаемости уроков на адаптированной образовательной платформе GetCourse.

12. ИТОГОВАЯ АТТЕСТАЦИЯ (УСТНЫЙ ЭКЗАМЕН)

Форма проведения устного экзамена – онлайн вебинар в соответствии с расписанием занятий на образовательной платформе GetCourse

Примерные темы к собеседованию – итоговой аттестации (экзамену)

1. Введение в процесс разработки ПО

Чем промышленный процесс разработки отличается от программирования дома

Роли в промышленной разработке

Виды процессов разработки

Контроль качества в процессах разработки

Тестирование в контроле качества

2. Место НТ в процессе разработки

3. Введение в методологию НТ

Типовой план работ по НТ (основные этапы)

Сбор целей, виды тестов и соответствие между ними

Поиск максимальной производительности

Проверка стабильности

- Проверка стрессоустойчивости
- Проверка отказоустойчивости
- Проверка масштабируемости
- Объемное тестирование

4. Анализ архитектуры систем

- Промышленный и тестовый стенды
- Источники нагрузки
- Смежные системы

- Протоколы (сетевые: какие бывают, какая классификация и стеки, подробнее про HTTP)

5. Расчет профиля НТ

- На основе аналитики

- На основе статистики (какие источники, за какие интервалы собирают статистику и как она превращается в профиль)

- Дневные и ночные профили

- Час пиковой нагрузки (что это)

6. Постановка задачи на НТ (что это, основные разделы). МНТ (что это, для чего, основные разделы)

7. LoadRunner

- Архитектура LR (компоненты, связь, как устроены скрипты, тесты, результаты)

7.1 Разработка скриптов в VUGEN

- Критерии готовности скриптов к НТ (параметризация, корреляция, верификация)

- Параметры в LR (одинарные, массивы, как создать свои вручную)

- Think-times (для чего, как выставлять)

- Автоматические средства корреляции LR (плюсы и минусы)

- Ручная корреляция (основные функции и принцип их работы)

- Параметризация различных типов (встроенные типы параметров)

- Верификация через web_reg_save_param и через web_reg_find

7.2 Работа в Controller

- Составление сценария по профилю из МНТ

- Pacing (что это, как рассчитывать)

- Проверка корректности профиля тестовыми запусками

7.3 Основные возможности Analysis

8. Метрики, собираемые во время НТ

9. Базы данных

- Oracle, AWR, планы запросов

10. Введение в загрузки

- Назначение

- Типы и виды

11. Создание отчетов НТ

- Виды отчетов

- Основные разделы

- Связь с МНТ

12. Дополнительно: вспомните SQL (порешайте на sql-ex.ru)

- Операции, какие есть (Select, Insert, Update, Delete...)

- Join (какие бывают) и Union - что это отличия

Вложенные запросы
Агрегатные функции и группировки (Group by)
Where и Having

7. СПИСОК ЛИТЕРАТУРЫ К ПРОГРАММЕ

Перечень основной литературы для обучающихся

1. Методическое пособие «Постановка задачи по нагрузочному тестированию АС «Наименование системы». Автор Салафаниди Г.С., электронный документ, размещенное на адаптированной образовательной платформе GetCourse - <https://diuniversity.getcourse.ru/>.
2. Методическое пособие «Основы нагрузочного тестирования с использованием LoadRunner». Автор Салафаниди Г.С., электронный документ, размещенное на адаптированной образовательной платформе GetCourse - <https://diuniversity.getcourse.ru/>.
3. Методическое пособие «Введение в тестирование производительности». Автор Салафаниди Г.С., электронный документ, размещенное на адаптированной образовательной платформе GetCourse - <https://diuniversity.getcourse.ru/>.
4. Методическое пособие «LoadRunner Tutorial». Автор Салафаниди Г.С., электронный документ, размещенное на адаптированной образовательной платформе GetCourse - <https://diuniversity.getcourse.ru/>.
5. Глоссарий нагрузочного тестирования. Электронный документ, размещенное на адаптированной образовательной платформе GetCourse - <https://diuniversity.getcourse.ru/>.
6. Постановка задачи по нагрузочному тестированию АС «Web Tours». Электронный документ, размещенное на адаптированной образовательной платформе GetCourse - <https://diuniversity.getcourse.ru/>.