Информационные технологии

Методические указания
к выполнению практических работ для студентов направления
38.02.01«Экономика и бухгалтерскийучет (по отраслям)»

Методические указания обсуждены на заседании методического совета СПК «19» 03 2021 года.
Протокол № 7,

Председатель методического совета СПК
Сергеева С.И._____ _____.
(подпись)

Методические указания одобрены на заседании педагогического совета СПК

«26» 03 2021 года. Протокол № 7.

Председатель педагогического совета СПК
Облиенко А.В._____.
(подпись)

Воронеж 2021

Министерство науки и образовании Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Воронежский государственный технический университет»

Информационные технологии

Методические указания
к выполнению практических х работ
для студентов, обучающихся по направлению 38.02.01
«Экономика и бухгалтерский учет (по отраслям)»

Воронеж 2021

УДК 721:53(073)
ББК 38.113я7-5

Составитель Д.Г. Сергиенко

Информационные технологии: метод. указания к выполнению практических работ по дисциплине «Информационные технологии» для студентов направления 38.02.01 «Экономика и бухгалтерский учет (по отраслям)»/ ФГБОУ ВО «Воронежский государственный технический университет» ; сост.: Д.Г. Сергиенко. – Воронеж: Изд-во ВГТУ, 2021.– с.

Приводится последовательность выполнения практических работ по всем разделам курса «Информационные технологии»: цель работы, соответствующие теоретические положения, порядок проведения опытов, способы обработки результатов и описание применяемых приборов, а также справочные данные из строительных нормативов.

Предназначены для студентов направления 38.02.01 «Экономика и бухгалтерский учет (по отраслям)».

УДК 721:53(073)
ББК 38.113я7-5

Содержание

Тема 1. Отладка программных модулей.
Практическая работа №1. Разработка и отладка модуля вывода и суммирования элементов массива

Цель работы

Знакомство с технологией "разработка через тестирование". Изучение инструментов позволяющих применять данную технологию.

Общие сведения

Разработка через тестирование (англ. *test-driven development, TDD*) — техника разработки программного обеспечения, которая основывается на повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, затем пишется код, который позволит пройти тест, и под конец проводится рефакторинг нового кода к соответствующим стандартам.

TDD требует от разработчика создания автоматизированных модульных тестов, определяющих требования к коду непосредственно перед написанием самого кода. Тест содержит проверки условий, которые могут либо выполняться, либо нет. Когда они выполняются, говорят, что тест пройден. Прохождение теста подтверждает поведение, предполагаемое программистом. Разработчики часто пользуются библиотеками для тестирования (англ. *testing frameworks*) длясоздания и автоматизации запуска наборов тестов. На практике модульные тесты покрывают критические и нетривиальные участки кода. Это может быть код, который подвержен частым изменениям, код, от работы которого зависит работоспособность большого количества другого кода, или код с большим количеством зависимостей.

Среда разработки должна быстро реагировать на небольшие модификации кода. Архитектура программы должна базироваться на использовании множества сильно связанных компонентов, которые слабо сцеплены друг с другом,благодаря чему тестирование кода упрощается.

TDD не только предполагает проверку корректности, но и влияет на дизайн программы. Опираясь на тесты, разработчики могут быстрее представить, какая функциональность необходима пользователю. Таким образом, детали интерфейса

появляются задолго до окончательной реализации решения

Задание

Изучить библиотеки для тестирования Рассмотреть применение NUnit, ReSharper.

Практическая работа №2. Разработка и отладка модуля вычисления площади геометрической фигуры

Цель работы

Овладение навыками модульного тестирования.

Общие сведения

Модульное тестирование - это тестирование программы на уровне отдельно взятых модулей, функций или классов. Цель модульного тестирования состоит в выявлении локализованных в модуле ошибок в реализации алгоритмов, а также в определении степени готовности системы к переходу на следующий уровень разработки и тестирования. Модульное тестирование проводится по принципу "белого ящика", то есть основывается на знании внутренней структуры программы, и часто включает те или иные методы анализа покрытия кода.

Модульное тестирование обычно подразумевает создание вокруг каждого модуля определенной среды, включающей заглушки для всех интерфейсов тестируемого модуля. Некоторые из них могут использоваться для подачи входных значений, другие для анализа результатов, присутствие третьих может быть продиктовано требованиями, накладываемыми компилятором и сборщиком.

На уровне модульного тестирования проще всего обнаружить дефекты, связанные с алгоритмическими ошибками и ошибками кодирования алгоритмов, типа работы с условиями и счетчиками циклов, а также с использованием локальных переменных и ресурсов. Ошибки, связанные с неверной трактовкой данных, некорректной реализацией интерфейсов, совместимостью, производительностью и т.п. обычно пропускаются на уровне модульного тестирования и выявляются на более поздних стадиях тестирования.

Задание

1) "Мозговая атака": Этапы:

- Получить вопрос (задание) для обсуждения.

- Задать вопросы относительно не понятных моментах ввопросе (задании).

- Высказать свои мысли по данному вопросу (заданию).

- Записать все прозвучавшие высказывания с уточнениями.

- По окончанию прочитать все, что было записано.

- Обсудить все варианты ответов.

- Выяснить, как можно использовать полученные результаты при выполнении данной лабораторной работы.

2) "Работа в команде"Этапы:

- Разбиться на команды.

- Реализовать полученный вопрос (задание), согласно технологии TDD.

- Представить результаты.

Практическая работа №3. Разработка и отладка модуля сортировки элементов массива

Цель работы

Овладение навыками интеграционного тестирования.

Общие сведения

Интеграционное тестирование называют еще тестированием архитектуры системы. С одной стороны, это название обусловлено тем, что интеграционные тесты включают в себя проверки всех возможных видов взаимодействий между программными модулями и элементами, которые определяются в архитектуре системы - таким образом, интеграционные тесты проверяют полноту взаимодействий в тестируемойреализации системы. С другой стороны, результаты выполнения интеграционных тестов - один из основных источников информации для процесса улучшения и уточнения архитектуры системы, межмодульных и межкомпонентных интерфейсов.Т.е., с этой точки зрения, интеграционные тесты проверяют корректностьвзаимодействия компонент системы.

В результате проведения интеграционного тестирования и устранения всех выявленных дефектов получается согласованная и

целостная архитектура программной системы, т.е. можно считать, что интеграционное тестирование - это тестирование архитектуры и низкоуровневых функциональных требований.

Интеграционное тестирование, как правило, представляет собой итеративный процесс, при котором проверяется совокупность модулей, возрастающая от итерации к итерации. В интеграционном тестирование выделяют три методы выполнения: восходящее тестирование; монолитное тестирование; нисходящее тестирование.

Задание

Согласно варианту провести один из методов интеграционного тестирования.

Практическая работа №4. Разработка и отладка модуля обработки элементов массива

Цель работы: приобрести практические навыки проведения автоматизированного тестирования и использования программ для автоматизированного тестирования на примере программы Selenium IDE.

4.1 Краткие теоретические сведения

Автоматизация тестирования (test automation) – набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполне- ния некоторых задач в процессе тестирования.

Преимущества автоматизации тестирования.

1 Повторяемость – все написанные тесты всегда будут выполняться одинаково, т.е. исключен «человеческий фактор». Тестировщик не пропустит тест по неосторожности и ничего не напутает в результатах.

2 Быстрое выполнение – автоматизированному скрипту не нужно сверяться с инструкциями и документациями, это сильно экономит время выполнения.

3 Меньшие затраты на поддержку – когда автоматические скрипты уже написаны, на их поддержку и анализ результатов требуется, как правило, меньшее время, чем на проведение того же объема тестирования вручную.

4 Отчеты – автоматически рассылаемые и сохраняемые отчеты о результатах тестирования.

5 Выполнение без вмешательства – во время выполнения тестов инженер-тестировщик может заниматься другими полезными делами или тесты могут выполняться в нерабочее время (этот метод предпочтительнее, т. к. нагрузка на локальные сети ночью снижена).

Недостатки автоматизации тестирования.

1 Повторяемость – все написанные тесты всегда будут выполняться одинаково. Это одновременно является и недостатком, т. к. тестировщик, выпол- няя

тест вручную, может обратить внимание на некоторые детали и, проведя не- сколько дополнительных операций, найти дефект. Скрипт этого сделать не может. 2 Затраты на поддержку – несмотря на то что в случае автоматизирован-

ных тестов они меньше, чем затраты на ручное тестирование того же функционала, они все же есть. Чем чаще изменяется приложение, тем они выше.

3 Большие затраты на разработку – разработка автоматизированных тестов это сложный процесс, т. к. фактически идет разработка приложения, которое тестирует другое приложение. В сложных автоматизированных тестах также есть фреймворки, утилиты, библиотеки и прочее. Все это нужно тестировать и отлаживать, а это требует времени.

4 Стоимость инструмента для автоматизации – в случае, если используется лицензионное ПО, его стоимость может быть достаточно высока. Свободно распространяемые инструменты, как правило, отличаются более скромным функционалом и меньшим удобством работы.

5 Пропуск мелких ошибок автоматический скрипт может пропускать мелкие ошибки, на проверку которых он не запрограммирован. Это могут быть неточности в позиционировании окон, ошибки в надписях, которые не прове- ряются, ошибки контролов и форм, с которыми не осуществляется взаимодей- ствие во время выполнения скрипта.

Что автоматизировать при тестировании?

1 Труднодоступные места в системе (бэкенд процессы, логирование файлов, запись в БД).

2 Часто используемая функциональность, риски от ошибок в которой доста- точно высоки. Автоматизировав проверку критической функциональности, мож- но гарантировать быстрое нахождение ошибок, а значит и быстрое их решение.

3 Рутинные операции, такие как переборы данных (формы с большим количеством вводимых полей). Заполнение полей различными данными и их про- верку после сохранения.

4 Валидационные сообщения. Заполнение полей некорректными данными и проверку на появление той или иной валидации.

5 Длинные end-to-end-сценарии.

6 Проверка данных, требующих точных математических расчетов. 7Проверка правильности поиска данных.

Задание

Записать скрипт в Selenium IDE, сохранить в файл с расширением html и отправить почтой результат с указанием в теме письма текста «Selenium webinar».

После записи скрипта требуется:

– убедиться, что присутствуют необходимые команды для синхронизации;

– убедиться, что добавлены проверки, необходимые для тестирования тогоили иного функционала;

– несколько раз выполнить скрипт и убедиться в его работоспособности.

Требования к скрипту:

должен содержать действия с одним из сайтов, доступных в общем доступе (варианты в пункте «Список возможных сайтов»);

количество команд в скрипте – от 5;

скрипт может быть открыт на другой машине и запуститься без ошибок. Если для определённых действий требуется аутентификация, можно использовать тестовые логин-пароль, которые прямо используются в скрипте.

должно содержаться несколько проверок: и assert, и verify (в сопроводительном письме желательно указать, почему в одном случае используется assert, а в другом – verify).

Список возможных сайтов: www.tut.by; www.vkontakte.ru;

www.onliner.by.Повышенная сложность: www.facebook.com;

www.gmail.com.

Практическая работа №5. Разработка и отладка модуля шифрования записей текстового файла

Цель работы

Выполните слияние документов, которые изображены на схеме, и получите письма приглашения на олимпиаду.

Общие сведения

Слияние документов - это объединение основного документа, содержащего постоянную часть информации, и источника данных, содержащих переменную часть. Примером слияния документов может быть персонализация писем. Текст делового письма постоянный, например, сообщение участникам математической олимпиады. Это основной документ. Такое письмо нужно выслать участникам олимпиады. Переменным является Фамилия И.О. участника, его адрес, набранные баллы. Данные об участниках представляют собой источник данных (список). Слияние проходит по следующей схеме.

Источник данных (список)

Фамилия	**Имя**	**Отчество**	**Индекс**	**Адрес**	**Сумма_баллов**
Петров	Иван	Сергеевич	220015	г. Минск ул. Я. Мавра д.23 кв.12	25
Сергеев	Петр	Иванович	220088	г. Минск ул. Ленина, д.34 кв. 112	30

Основной докуме

<<Индекс>>
<<Адрес>>

Уважаемый << Фамилия >> << Имя>> << Отчество >>!

Сообщаем, что Вы, участвуя в математической олимпиаде, набрали <<Сумма_баллов>> баллов.

Оргкомитет

Результат слияния

220015
г. Минск ул. Я. Мавра д.23 кв.12

Уважаемый Петров Иван Сергеевич!

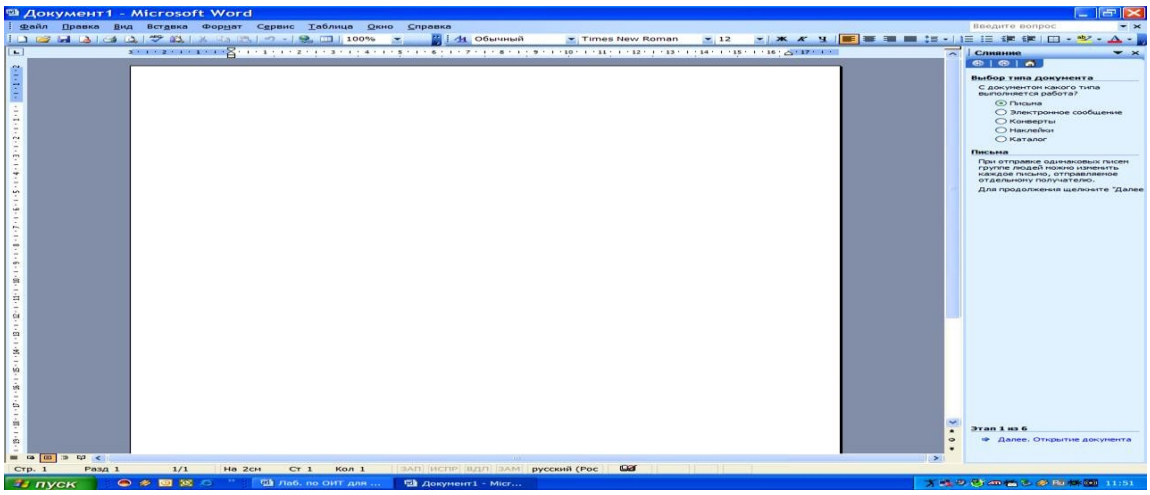
Сообщаем, что Вы, участвуя в математической олимпиаде, набрали 25 баллов.

Оргкомитет

220080
г. Минск ул. Ленина, д.34 кв. 112
Уважаемый Сергеев Петр Иванович!
Сообщаем, что Вы, участвуя в математической олимпиаде, набрали 30 баллов.
Оргкомитет

В результате слияния основного документа и источника данных (списка) для каждого участника из списка готовится письмо. В итоге получается несколько писем одинакового содержания.

Слияние документов выполняется в диалоговом окне **«Слияние»**, вызываемом командой **Сервис/Письма и рассылки/Слияние**.



Работа по слиянию документов состоит из шести этапов:

- выбор типа документа (письма, электронное сообщение, конверты, наклейки, каталог);
- выбор документа (текущий документ, шаблон, существующий документ);
- выбор получателей (создание списка, использование существующего списка, контакты Outlook);
- создание документа (основной документ с полями слияния);
- просмотр полученных документов (результат слияния);
- завершение слияния.

Кроме этого, пользователь может вносить изменения в основной документ и в список источника данных, т.е. возвращаться к любому этапу.

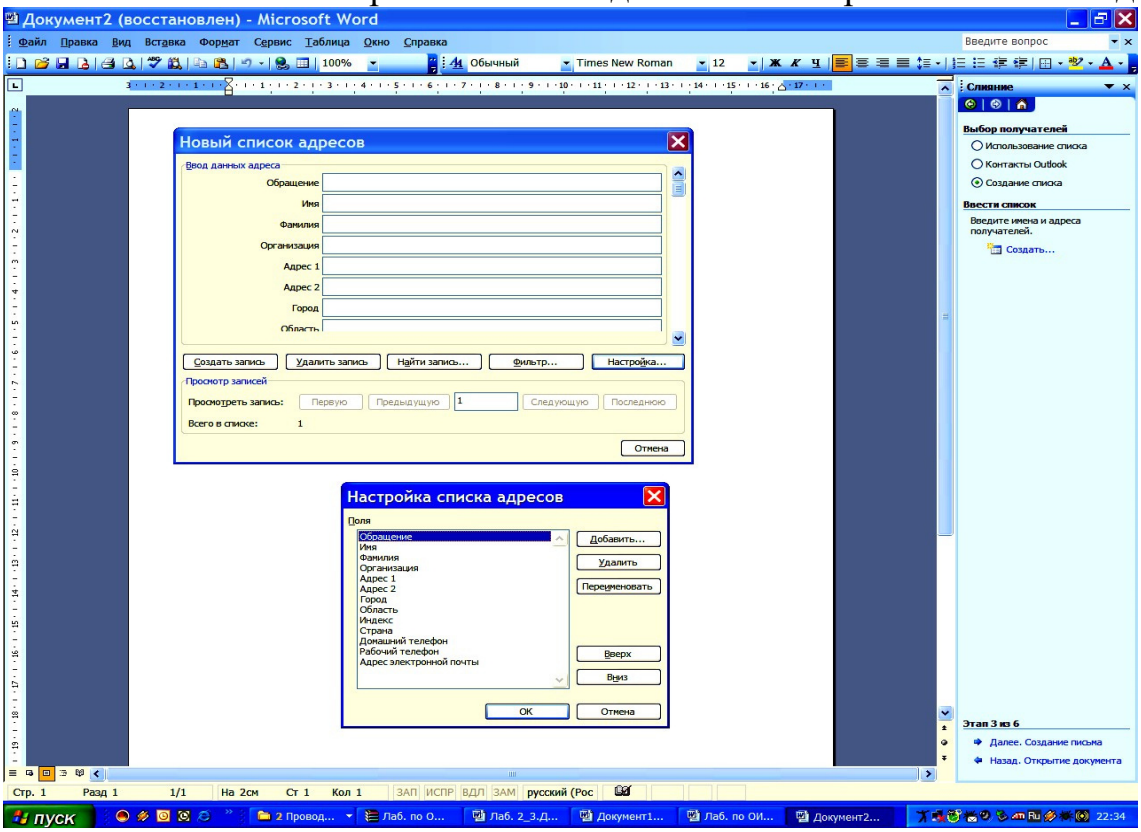
Задание

Выполните команду Word **Файл/Создать.**

Запустите команду **Сервис/Письма и рассылки/Слияние**.

Выполняйте последовательно этапы друг за другом, используя кнопку **Вперед** (***Далее)***

Этап 1. Выбор типа документа – *письма*.
Этап 2. Выбор документа – *текущий документ*.
Этап 3. Выбор получателей – *создание списка* (нажмите кнопку ***Создать***). В окне ***Новый список адресов*** нажмите кнопку ***Настройка***. Используя кнопки настройки ***Добавить, Удалить, Переименовать,*** создайте нужный список получателей, **заполните 5 записей** произвольными данными и сохраните источник данных вфайле.



Этап 4. **Создание письма (*основного документа)***.
Подготовьте основной документ, который изображен на схеме.
Поля отмеченные << >> выбирайте из закладки ***Другие элементы*** или

воспользуйтесь кнопкой ***Вставить поле слияния***

Этап 5. **Просмотр писем.**
Просмотрите полученные письма.
Если нужно внести изменения в список или текст письма, вернитесь к соответствующему этапу.
Этап 6. Завершить слияние.
Запишите созданные письма в новый документ (команда – изменить часть писем)

Практическая работа №6. Разработка и отладка модуля для генерации конечной последовательности случайных чисел и символов

Цель работы

Овладение навыками системного тестирования.

Общие сведения

Системное тестирование - один из самых сложных видов тестирования. На этапе системного тестирования проводится не только функциональное тестирование, но и оценка характеристик качества системы - ее устойчивости, надежности, безопасности и производительности. На этом этапе выявляются многие проблемы внешних интерфейсов системы, связанные с неверным взаимодействием с другими системами, аппаратным обеспечением, неверным распределением памяти, отсутствием корректного освобождения ресурсов и т.п.

После завершения системного тестирования разработка переходит в фазу приемо-сдаточных испытаний (для программных систем, разрабатываемых на заказ) или в фазу альфа- и бета- тестирования (для программных систем общего применения).

Системное тестирование проводится в несколько фаз, на каждой из которых проверяется один из аспектов поведения системы, т.е. проводится один из типов системного тестирования. Все эти фазы могут протекать одновременно или последовательно. Следующий раздел посвящен рассмотрению особенностей каждого из типовсистемного тестирования на каждой фазе.

Виды системного тестирования:

1) функциональное тестирование;
2) тестирование производительности;
3) нагрузочное или стрессовое тестирование;
4) тестирование конфигурации;
5) тестирование безопасности;
6) тестирование надежности и восстановления после сбоев;
7) тестирование удобства использования.

Задание

Согласно варианту провести несколько видов системного тестирования.

Практическая работа №7. Разработка, отладка и оптимизация модуля управления движением объекта по двум координатам

Цель работы

С помощью формульного редактора Equation Editor наберите формулу:

$$\chi^2 = \sum_{i=1}^{n} \left(\frac{X_i - \mu}{\sigma} \right)$$

Общие сведения

Вставка формул в редакторе WORD осуществляется с помощью формульного редактора.

Вызов формульного редактора Equation Editor из Word можно осуществить следующей последовательностью действий:

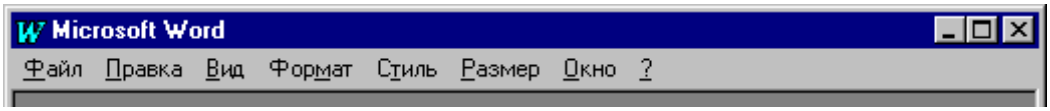
- поместите курсор в то место, где должна быть вставлена формула;
- в меню "вставка" выберите команду "обьект";
- выберите закладку "создание";
- В окне "тип обьекта" выберите "Microsoft Equation 3.0 (2.0)";

Более удобный вызов редактора математических формул выполняется при помощи кнопки редактора формул, которая помещается на панель инструментов. Размещение кнопки на панели выполняется следующим образом: через меню **Вид/Панели инструментов/Настройка** на вкладке **Команды** отыскивается вписке категорий пункт **Вставка**, в котором выбирается кнопка **Редактор формул** иперетаскивается на любую панель инструментов.

В результате вызова **Редактора формул** на экране появляется панель инструментов, состоящая из двух рядов пиктограмм:

В первом ряду расположено 10 пиктограмм, за каждой из которых находится группа символов (математические оп ерации, греческие символы и т. д.) Во втором ряду находятся пиктограммы для вызова шаблонов наиболее распространенных структурных формул (матрицы, суммы и т. д.). Кроме того главное меню **Word** заменяется на меню редактора математических формул.

Дадим краткую характеристику меню редактора математических формул:



Меню **Файл** содержит обычные для этого пункта команды работы с файлами, печати документа и т. п.

Меню **Правка** содержит команды редактирования, которые применяются для формул.

Меню **Вид** содержит команды задания масштаба отображения формул, управления панелями инструментов, обновления экрана.

Меню **Формат** содержит команды выравнивания формул, изменения макета матриц, установления расстояний между элементами формул.

Меню **Стиль** содержит команды, задающие вид шрифта для математических символов, для текста, для функций и т.д.

Меню **Размер** содержит команды, определяющие размеры символов и индексов в формулах.

Иногда необходимо редактировать ранее набранную формулу. Для этого следует дважды щелкнуть мышью в поле формулы. При этом активизируется редактор формул. Нужный элемент формулы выделяется мышью. В формулу можно добавлять новые элементы, удалять их или изменять.

Задание

- Вызовите формульный редактор;
- В палитре шаблонов выберите третий слева шаблон с индексами;
- В открывшемся списке выберите левый в первом ряду;
- Наберите греческое "Хи" (в палитре символов второе поле справа - греческие символы).

Обратите внимание на различный вид курсоров. Вводимый символ вставляется в позицию, определяемую вертикальной чертой курсора!

- Подведите курсор в поле верхнего индекса и нажмите 2;
- Введите "=" после X;
- — из палитры шаблонов выберите знак суммы с верхним и нижним индексами, и введите индексы;
- Выберите из палитры шаблонов объект с верхним индексом (первый в четвёртом ряду);
- Выберите шаблон со скобками;
- Выберите шаблон для дроби;
- — Выберите шаблон с нижним индексом, введите Хи, переведите курсор в следующую позицию (стрелкой -> или шелчком мыши), наберите "-", затем "m"
- В знаменателе введите "σ"
- В месте верхнего индекса наберите 2;
- Выйдите из редактора формул, щёлкнув левой кнопкой мыши вне поля редактирования.
- Сохраните формулу в файле.

Практическая работа №8. Разработка, отладка и оптимизация модуля отображения элементов двумерного массива

Цель работы

Овладение навыками ручного тестирования и составление тестовых случаев.

Общие сведения

Ручное тестирование заключается в выполнении задокументированной процедуры, где описана методика выполнения тестов, задающая порядок тестов и для каждого теста - список значений параметров, который подается на вход, и список результатов, ожидаемых на выходе. Поскольку процедура предназначена для выполнения человеком, в ее описании для краткости могут использоваться некоторые значения по умолчанию, ориентированные на здравый смысл, или ссылки на информацию, хранящуюся в другом документе.

Описание тестов разрабатывается для облегчения анализа и поддержки тестового набора. Описание может быть реализовано в произвольной форме, но при этом должны выполнять следующие задачи:

1. Анализировать степень покрытия продукта тестами на основании описания тестового набора.
2. Для любой функции тестируемого продукта найти тесты, в которых функция используется.
3. Для любого теста определить все функции и ихсочетания, которые данный тест использует (затрагивает).
4. Понять структуру и взаимосвязи тестовых файлов. 5.Понять принцип построения системы автоматизации тестирования

Задание

Подготовить тестовый случай, выполнить и задокументироватьрезультаты.

Практическая работа №9. Разработка, отладка и оптимизация модуля выполнения операций реляционной алгебры над множествами

Цель работы

Составить таблицу расчета доходов фирмы в абсолютном и процентном отношении и диаграмму роста доходов на основе данных о доходах фирмы.

Рост уровня доходов фирмы в абсолютном и процентном отношении

Месяцы	Уровень доходов фирмы в 1998 году, млн.руб.	Уровень доходов фирмы в 1999 году, млн.руб.	Рост уровня доходов фирмы в 1999 году в%
январь	180	200	
февраль	195	210	
март	200	230	
апрель	213	245	
май	240	270	
июнь	254	275	
июль	260	281	
август	265	290	
сентябрь	280	300	
октябрь	290	315	
ноябрь	300	323	
декабрь	325	330	
Всего:			

Задание

1. Составить таблицу расчета доходов фирмы: определить тип, размер и стиль шрифтов для заголовков строк и столбцов: Times New Roman Cyr, размер 12, стиль полужирный; для остального текста - Times New Roman Cyr, размер 10, стиль обычный;
2. Вычислить рост уровня доходов фирмы в процентном отношении в каждом месяце 1999 года по отношению к январю 1999 года (3-й столбец таблицы);
   =(Ci-C$3)/C$3 где Ci – адрес ячейки i-го месяца графы Уровень доходов фирмы в 1999 году, C$3 – абсолютный адрес ячейки Уровень доходов фирмы за январь 1999 года;
3. Вычислить суммарный уровень доходов фирмы за 1999 и 1998 годы, результаты поместить в последней строке второго и третьего столбца соответственно;
4. Вычислить среднее значение роста уровня доходов в процентах, результат поместить в последней строке четвертого столбца;
5. Построить диаграмму зависимости уровня доходов фирмы за 1999 и 1998 годы по месяцам в виде гистограммы;
6. Построить диаграмму зависимости уровня доходов фирмы в процентном отношении в виде линейного графика;
7. Построить совмещенную диаграмму (тип **нестандартная/график|гистограмма 2)** по данным полученной таблицы (второй, третий и четвертый столбцы);

Рассмотреть другие типы диаграмм, освоить редактирование элементов диаграмм.

Практическая работа №10. Разработка, отладка и оптимизация модуля для арифметических операций

Цель работы

Дан список сотрудников фирмы, содержащий паспортные данные (фамилию, имя, отчество, дату рождения, дату зачисления в состав фирмы). По этому списку составить список, содержащй следующие данные (фамилию и инициалы, возраст, рабочий стаж в фирме).

Задание

1. Составьте таблицу сотрудников фирмы, содержащий следующие данные:

Список сотрудников фирмы					
№ п/п	Фамилия	Имя	Отчество	Дата рождения	Дата зачисления
1.	Макаров	Сергей	Петрович	23.05.40	05.09.90
…	…	…	…	…	…

2. Изучите календарные функции **СЕГОДНЯ(), ГОД(), ДОЛЯГОДА(), МЕСЯЦ().**
3. Постройте другую таблицу

Список сотрудников фирмы			
№ п/п	Фамилия И.О.	Возраст	Стаж
1.	Макаров С.П.	58	8
…	…	…	…

4. Для получения данных в графе “Фамилия И.О.” можно применить формулу
=Фамилия&" "&ЛЕВСИМВ(Имя;1)&"."&ЛЕВСИМВ(Отчество;1)&"."

В приведенной формуле **Фамилия, Имя, Отчество** – это имена соответствующих столбцов или адреса ячеек с соответствующей информацией.

Для получения данных в графе “Возраст” можно применить формулу
=ГОД(СЕГОДНЯ())-ГОД(Дата_рождения)
Для получения данных в графе “Стаж” можно применить формулу
=ОТБР(ДОЛЯГОДА(Дата_зачисления;СЕГОДНЯ();1))
Для определения числа месяцев можно применить функцию МЕСЯЦ.
Для определения возраста в днях можно применить формулу
=СЕГОДНЯ()-Дата_рождения+1.

Практическая работа №11. Отладка оптимизация модулей инструментальными средствами

Цель работы

На основании следующей таблицы:

Менеджер	Месяц	Продукты	Доход	Расход	Прибыль	Регион
Иванов	январь	мясо	100,00	50,00		Страны СНГ
Иванов	февраль	мясо	100,00	50,00		Россия
Иванов	февраль	мясо	100,00	50,00		Россия
Иванов	апрель	мясо	100,00	50,00		Россия
Иванов	апрель	мясо	100,00	50,00		Россия
Петров	январь	мясо	100,00	50,00		Страны СНГ
Петров	февраль	мясо	100,00	50,00		Страны СНГ
Петров	февраль	мясо	100,00	50,00		Страны СНГ
Петров	апрель	мясо	100,00	50,00		Страны СНГ
Петров	апрель	мясо	100,00	50,00		Страны СНГ
Сидоров	май	рыба	100,00	50,00		Страны СНГ
Сидоров	январь	рыба	100,00	50,00		Россия
Иванов	февраль	рыба	100,00	50,00		Россия
Иванов	март	молоко	200,00	20,00		Россия
Петров	март	молоко	300,00	30,00		Страны СНГ
Сидоров	март	молоко	150,00	100,00		Страны СНГ

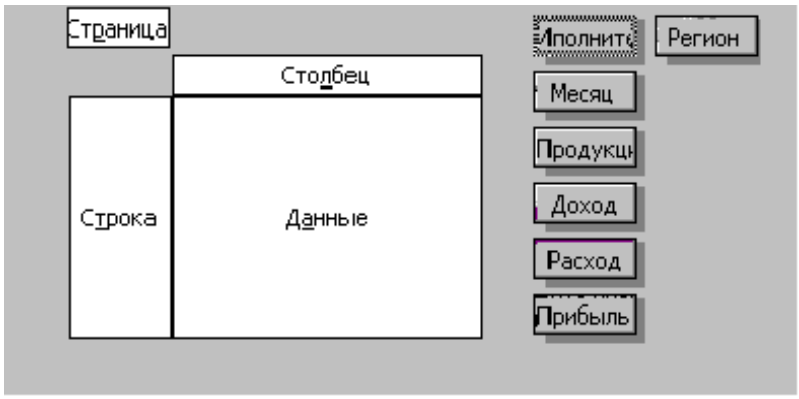
Построить таблицу, показывающую объем прибыли полученной от продажи разных видов продукции разными исполнителями по месяцам в разрезе регионов;

Общие сведения

Сводные таблицы предназначены для обобщения (объединения, переработки) информации, хранящейся в базе данных. Они также позволяют отображать табличные данные в виде двух мерной или трехмерной таблицы. Кроме того, с их помощью можно вывести промежуточные итоги с любым уровнем детализации.

Сводная таблица может быть создана на основании данных находящихся:

- в списке или базе данных Microsoft Excel;
- во внешнем источнике данных;
- в нескольких диапазонах консолидации;



– в другой сводной таблице.

Каждая сводная таблица состоит из 4 областей: страница, строка, столбец, данные.

Кроме того, всегда имеются кнопки с названиями полей соответствующей базы данных, которые расположены рядом с макетом сводной таблицы или на панели инструментов. Для получения нужной сводной таблицы необходимо перетащить одну или несколько кнопок с названиями полей в нужную область. Назначение областей следующее:

– **Строка**. Уникальные значения полей, помещенных в эту область, используются в качестве заголовков строк в сводной таблице. Если в эту область помещено одно поле, то количество строк в сводной таблице (без учета итогов) равно числу уникальных значений этого поля.

– **Столбец**. Уникальные значения полей, помещенных в эту область, используются в качестве заголовков столбцов в сводной таблице. Если в эту область помещено одно поле, то количество столбцов в сводной таблице (без учета итогов) равно числу уникальных значений этого поля.

– **Данные**. Значения полей, помещенных в эту область, используются для заполнения ячеек сводной таблицы итоговыми данными (суммирование, подсчет количества, вычисление среднего значения и т. д.).

– **Страница**. Уникальные значения полей, помещенных в эту область, и элемент «все» используются для построения раскрывающихся списков. В поле страницы можно выбрать только одно значение в каждом из списков. В области данных будут отображены итоговые данные, для выбранного значения. Использование этого элемента сводной таблицы позволяет, в некоторой мере, реализовать отображение трехмерной таблицы.

Задание

Скопируйте в буфер обмена таблицу в редакторе Word.

Вставьте таблицу на рабочий лист Excel лист и оформите данные в виде списка.

Рассчитайте значение поля «Прибыль», записав соответствующую формулу.

Сделайте текущей любую ячейку построенного списка.

Выполните команды **Данные** и **Сводная таблица.**

Установите флажок – **В списке или базе данных Microsoft Excel;**

Укажите диапазон, содержащий построенный список. Если список был построен правильно, нужный диапазон будет выбран автоматически.

Перетащите кнопки «Продукция» и «Менеджер» в область «**Строка**». При этом

важен порядок перетаскивания – поле «Менеджер» будет вложенным по отношению к полю «Продукция». Затем в область «**Столбец**» перетащите кнопку «**Месяц**» и в область страниц – кнопку «**Регион**». В область данных перетащите кнопку «Прибыль».

Укажите место размещения сводной таблицы.

Построенная сводная таблица будет иметь следующий вид:

	A	B	C	D	E	F	G	H
1	Регион	(Все)						
2								
3	Сумма по полю Приб		Месяц					
4	Продукция	Менеджер	янв	фев	мар	апр	май	Общий итог
5	молоко	Иванов			180			180
6		Петров			270			270
7		Сидоров			50			50
8	молоко Всего				500			500
9	мясо	Иванов	50	100		100		250
10		Петров	50	100		100		250
11	мясо Всего		100	200		200		500
12	рыба	Иванов		50				50
13		Сидоров	50				50	100
14	рыба Всего		50	50			50	150
15	Общий итог		150	250	500	200	50	1150
16								

Тема 2. Отладка и тестирование программного продукта на уровне модулей

Практическая работа №1. Тестирование «белым ящиком»

Цель работы

Овладение навыками документирования результатов тестирования.

Общие сведения

Каждый дефект, обнаруженный в процессе тестирования, должен быть задокументирован и отслежен. При обнаружении нового дефекта его заносят в базу дефектов. При занесении нового дефекта рекомендуется указывать, как минимум, следующую информацию:

1) Наименование подсистемы, в которой обнаружен дефект.
2) Версия продукта (номер build), на котором дефект был найден.
3) Описание дефекта.
4) Описание процедуры (шагов, необходимых для

воспроизведения дефекта).

5) Номер теста, на котором дефект был обнаружен.

6) Уровень дефекта, то есть степень его серьезности с точкизрения критериев качества продукта или заказчика.

Тестовый отчет обновляется после каждого цикла тестирования и должен содержать следующую информацию для каждого цикла:

1) Перечень функциональности в соответствии с пунктами требований, запланированный для тестирования на данном цикле, и реальные данные по нему.

2) Количество выполненных тестов – запланированное и реально исполненное.

3) Время, затраченное на тестирование каждой функции, и общее время тестирования.

4) Количество найденных дефектов.

5) Количество повторно открытых дефектов.

6) Отклонения от запланированной последовательности действий, если таковые имели место.

7) Выводы о необходимых корректировках в системе тестов, которые должны быть сделаны до следующего тестового цикла.

Задание

Задокументировать результаты тестирования. Для выполнения работы использовать тестовые случае из лабораторной работы №5.

Практическая работа №2. Тестирование «черным ящиком»

Цель работы: изучить элементы плана тестирования; приобрести навыки разработки плана тестирования.

1.1 Краткие теоретические сведения

Тест-план, или план тестирования (test plan), – часть проектной докумен- тации, описывающая и регламентирующая процесс тестирования.

1.1.1 Ключевые секции тест-плана.

1.1.1.1 Перечень работ.

Эта секция включает перечень функциональных областей приложений, ко- торые будут подвергаться тестированию.

Здесь же может быть перечень компонентов или функциональности, кото- рые не будут тестироваться по тем или иным причинам.

1.1.1.2 Критерии качества.

Одна из наиболее важных секций, содержащая перечень критериев, по ко- торым оценивается текущее и финальное качество продукта.

1.1.1.3 Оценка рисков.

Риск (risk) – сочетание вероятности наступления события и последствий, вызванных этим событием.

В секции приводится перечень рисков, которые могут (вполне вероятно) возникнуть в процессе работы с проектом. По каждому риску даётся оценка представляемой им угрозы и приводятся варианты выхода из ситуации.

1.1.1.4 Документация.

Здесь приводится полный перечень используемой документации, а также указывается, кто и когда должен её готовить, кому передавать и т.п.

1.1.1.5 Стратегия тестирования.

Стратегия тестирования подразумевает описание процесса тестирования с

точки зрения применяемых методов, подходов, инструментальных средств и т. п.

1.1.1.6 Ресурсы.

В данной секции перечисляются необходимые для успешного проведения тестирования ресурсы:

– программные;
– аппаратные;
– человеческие;
– временные;
– финансовые.

1.1.1.7 Метрики.

Секция, в которой приводятся числовые характеристики показателей качества, способы их оценки, формулы и т. п.

1.1.1.8 Расписание и ключевые точки.

Самая простая для понимания секция. Фактически это календарь, в кото- ром указано, что и к какому моменту должно быть сделано.

Задание

1 Приведите по пять примеров тестов для графического редактора

Photoshop, которые относились бы: а) к уровню Smoke Test;

б) к уровню Critical Path Test;

в) к уровню Extended Test.

2 Приведите пример риска, который может быть отмечен при планировании тестовых испытаний. Дайте рекомендации по недопущению такой ситуа- ции и выходу из неё в случае возникновения.

3 Перечислите основные секции тестового плана и дайте краткое пояснение того, что размещается в каждой из них.

Практическая работа №3. Модульное тестирование

Цель работы: приобрести практические навыки создания тестов и тест- кейсов; научиться создавать тест-кейсы для приложений.

2.1 Краткие теоретические сведения

2.1.1 Рекомендации по разработке тестов.

Начинайте с простых очевидных тестов. Затем переходите к более сложным тестам. Помните о граничных условиях.

Если остаётся время, занимайтесь исследовательским тестированием.

Последовательность разработки и выполнения тестов:

- простые позитивные;
- простые негативные;
- сложные позитивные;
- сложные негативные.

2.1.2 Оформление тест-кейсов.

Общие идеи по разработке тест-кейсов приведены на рисунке 2.1.

Задание

На основе представленного ниже набора требований сформируйте для разрабатываемых приложений:

- смоук-тест;
- чек-лист для теста критического пути;
- тест критического пути (насколько хватит времени – расписывайте идеи из чек-листа в полноценные тесты).

Требования к разрабатываемому приложению 1

Приложение должно выполнять математические вычисления.

1 Приложение должно работать под всеми версиями ОС Windows.

2 Приложение должно быть максимально похоже на стандартный кальку- лятор Windows (рисунок 2.2) за исключением некоторых особенностей.

3 Несколько приложений должны иметь возможность работать одновременно. 4 При запуске приложения должно отображаться окно со стандартными для калькулятора кнопками и полем ввода и отображения данных.

5 Для начала вычислений пользователь должен нажать кнопку «Начать».

6 Приложение должно позволять легко сохранять вычисления в выбран- ном пользователем формате.

7 Опционально предусматривается поддержка нескольких языков.

8 Приложение должно позволять выполнять вычислениясразу же после запуска.

9 Скорость вычислений должна быть максимально высокой.

10 Приложение должно позволять выполнять следующие операции: сложение, умножение, вычитание и деление чисел.

11 Приложение должно позволять строит графики простых функций.

12 Приложение должно запрашивать подтверждение («Результат не сохра- нён. Выйти?») в случае, если пользователь не сохранил результаты работы.

Требования к разрабатываемому приложению 2

1 Приложение должно работать под версиями ОС Windows7 и Windows 8. 2 Несколько приложений должны иметь возможность работать одновременно, т. е. можно открыть несколько калькуляторов и вести в них невзаимосвязанные вычисления.

3 При запуске приложения должно отображаться окно с кнопками калькулятора (рисунок 2.2) и полем отображения данных.

4 Данные в приложение могут вводиться как с помощью кнопок приложе- ния, так и с помощью клавиатуры.

5 Приложение должно позволять сохранять вычисления во внешний файл с расширением, задаваемым пользователем.

6 Должна быть предусмотрена поддержка английского и русского языков.

Отображается тот язык, который выбран в ОС по умолчанию.

7 Вычисления должны производиться со скоростью не более 1 с.

8 Приложение должно позволять выполнять следующие операции: сложе- ние, умножение, вычитание и деление чисел, взятие квадратного корня, возве- дение в степень, вычисление процентов, ввод отрицательного числа.

Практическая работа №4. Интеграционное тестирование

Цель работы: приобрести практические навыки составления документации, используемой при тестировании приложений на примере отчета о тестировании.

3.1 Краткие теоретические сведения

Отчёт о результатах тестирования (test result report, TRR) – часть те- стовой документации, включающая в себя описание процесса тестирования, суммарную информацию о протестированных за подотчётный период билдах, информацию о деятельности тестировщиков, а также некоторые статисти- ческие данные.

Цель написания TRR – предоставление лицам, заинтересованным в проек- те, полной и объективной информации о текущем состоянии качества проекта. Эта информация выражается в конкретных фактах и цифрах.

Обычно TRR предоставляется для ознакомления всей проектной коман- де и заказчику.

Тестировщики не заинтересованы в приукрашивании отчётов и часто об- ладают более полной информацией о текущем состоянии качества продукта, чем какая бы то ни было другая часть проектной команды.

3.1.1 Структура отчёта о результатах тестирования.

Команда тестировщиков (test team).

В этой части TRR перечисляются все задействованные в процессе тестиро- вания сотрудники с указанием занимаемой должности и роли на проекте в под- отчётный период.

Описание процесса тестирования (testing process description).

В этой части TRR даётся краткое описание того, как происходило тестиро- вание: какие использовались методы, техники, инструментальные средства

и т. п.

Краткое описание (summary).

В этой части TRR даётся краткое описание того, какие билды были проте-

стированы, есть ли в качестве приложения прогресс или регресс, есть ли какие- либо проблемы, требующие внимания руководства.

Краткое описание – важная часть отчёта, т. к. менеджеру проекта прихо- дится просматривать огромное количество документации, и он часто принимает решение о необходимости более детального изучения отчёта как раз на основе краткого описания.

Расписание (testing timetable).

В данном разделе отчёта приводится детализированное описание того, какая работа и на протяжении какого времени выполнялась каждым тестировщиком.

Рекомендации (recommendations).

В этой части TRR следует подчеркнуть те важные моменты, на которые следует обратить внимание руководству или лидерам проектных команд. Здесь также, возможно, будет дана рекомендация на передачу проекта заказчику («передачу в продакшн»).

Статистика по ошибкам (bugs statistics).

Здесь приводится сводная таблица, содержащая информацию об ошибках, с которыми команде тестировщиков приходилось иметь дело в подотчётный период.

Список новых ошибок (new bugs found).

Здесь приводится список ошибок, обнаруженных командой тестировщиков за подотчётный период. Список ошибок легко извлечь из баг-трекинговой системы.

Статистика по всем ошибкам (all bugs statistics).

Здесь приводится сводная таблица, содержащая информацию об ошибках, с которыми команде тестировщиков приходилось иметь дело за всё время рабо- ты с проектом. Статистика по всем ошибкам также отражается в виде графика.

Задание

1 Разработать отчет о тестировании по заданию преподавателя. 2 Выполнить тест для проверки знаний.

Тест для проверки знаний

1 Отчёт о результатах тестирования – это:

а) разновидность отчёта об ошибке;

б) часть тестовой документации, включающая в себя описание про- цесса тестирования;

в) диаграмма с указанием распределения дефектов по их важности;

г) отчёт, подготавливаемый лидером команды разработчиков для ли- дера команды тестировщиков.

2 К целям написания отчёта о результатах тестирования относятся:

а) стимулирование команды разработчиков;

б) демонстрация преимуществ проекта перед конкурирую- щими проектами;

в) предоставление заказчику экономической информации о проекте;

г) предоставление лицам, заинтересованным в проекте, полной и объ- ективной информации о текущем состоянии качества проекта.

3 Периодичность выпуска отчётов о результатах тестирования:

а) ничем не определяется;

б) отсутствует. Отчёт готовится один раз в конце проекта;

в) определяется набором критериев, установленных в фирме для дан- ного вида документации;

г) определяется законодательными актами и стандартами.

4 К основным разделам отчёта о результатах тестирования относятся:

а) шаги по воспроизведению; б) идентификатор;

в) расписание;

г) рекомендации.

5 В разделе «Описание процесса тестирования» отчёта о результатах тестирования приводится:

а) список группы тестировщиков; б) список найденных дефектов;

в) краткое описание того, как происходило тестирование: какие использовались методы, техники, инструментальные средства и т. п.;

г) подробное описание процесса автоматизации тестирования, вклю- чая перечень тест-кейсов, журналы выполнения тестов и т. п.

6 В разделе «Краткое описание» отчёта о результатах тестирования приводится:

а) краткое описание мнения команды тестировщиков о перспективах дальнейшего сотрудничества с данным заказчиком;

б) краткое описание того, какие билды были протестированы, есть ли в качестве приложения прогресс или регресс, есть ли какие-либо проблемы, требующие внимания руководства;

в) краткий перечень рекомендаций по закупке нового оборудования; г) краткое описание процесса разработки программного средства за

подотчётный период.

7 Отчёт о результатах тестирования необходим:

а) менеджеру проекта;

б) лидеру команды разработчиков;

в) системному администратору филиала; г) заказчику.

8 Финальный отчёт о результатах тестирования :

а) такого отчёта нет;

б) отчёт о результатах тестирования, создаваемый в конце работы с проектом;

в) ещё одно название обычного отчёта о результатах тестирования; г) список найденных за весь период тестирования ошибок.

Практическая работа №5. Тестирование «белым ящиком»

Цель работы: приобрести практические навыки проведения автоматизированного тестирования и использования программ для автоматизированного тестирования на примере программы Selenium IDE.

4.1 Краткие теоретические сведения

Автоматизация тестирования (test automation) – набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполне- ния некоторых задач в процессе тестирования.

Преимущества автоматизации тестирования.

6 Повторяемость – все написанные тесты всегда будут выполняться одно-образно, т.е. исключен «человеческий фактор». Тестировщик не пропустит тест по неосторожности и ничего не напутает в результатах.

7 Быстрое выполнение – автоматизированному скрипту не нужно сверять-ся с инструкциями и документациями, это сильно экономит время выполнения.

8 Меньшие затраты на поддержку – когда автоматические скрипты уже написаны, на их поддержку и анализ результатов требуется, как правило, меньшее время, чем на проведение того же объема тестирования вручную.

9 Отчеты – автоматически рассылаемые и сохраняемые отчеты о результа-тах тестирования.

10 Выполнение без вмешательства – во время выполнения тестов инженер-тестировщик может заниматься другими полезными делами или тесты могут выполняться в нерабочее время (этот метод предпочтительнее, т. к. нагрузка на локальные сети ночью снижена).

Недостатки автоматизации тестирования.

1 Повторяемость – все написанные тесты всегда будут выполняться одно-образно. Это одновременно является и недостатком, т. к. тестировщик, выпол- няя

тест вручную, может обратить внимание на некоторые детали и, проведя не- сколько дополнительных операций, найти дефект. Скрипт этого сделать не может. 2 Затраты на поддержку – несмотря на то что в случае автоматизирован-

ных тестов они меньше, чем затраты на ручное тестирование того же функционала, они все же есть. Чем чаще изменяется приложение, тем они выше.

6 Большие затраты на разработку – разработка автоматизированных тестов это сложный процесс, т. к. фактически идет разработка приложения, которое тестирует другое приложение. В сложных автоматизированных тестах также есть фреймворки, утилиты, библиотеки и прочее. Все это нужно тестировать и отлаживать, а это требует времени.

7 Стоимость инструмента для автоматизации – в случае, если используется лицензионное ПО, его стоимость может быть достаточно высока. Свободно распространяемые инструменты, как правило, отличаются более скромным функционалом и меньшим удобством работы.

8 Пропуск мелких ошибок автоматический скрипт может пропускать мелкие ошибки, на проверку которых он не запрограммирован. Это могут быть неточности в позиционировании окон, ошибки в надписях, которые не прове- ряются, ошибки контролов и форм, с которыми не осуществляется взаимодей- ствие во время выполнения скрипта.

Что автоматизировать при тестировании?

7 Труднодоступные места в системе (бэкенд процессы, логирование файлов, запись в БД).

8 Часто используемая функциональность, риски от ошибок в которой доста- точно высоки. Автоматизировав проверку критической функциональности, мож- но гарантировать быстрое нахождение ошибок, а значит и быстрое их решение.

9 Рутинные операции, такие как переборы данных (формы с большим количеством вводимых полей). Заполнение полей различными данными и их про- верку после сохранения.

10 Валидационные сообщения. Заполнение полей некорректными данными и проверку на появление той или иной валидации.

11 Длинные end-to-end-сценарии.

12 Проверка данных, требующих точных математических расчетов. 7 Проверка правильности поиска данных.

Задание

Записать скрипт в Selenium IDE, сохранить в файл с расширением html и отправить почтой результат с указанием в теме письма текста «Selenium webinar».

После записи скрипта требуется:

– убедиться, что присутствуют необходимые команды для синхронизации;

– убедиться, что добавлены проверки, необходимые для тестирования того или иного функционала;

– несколько раз выполнить скрипт и убедиться в его работоспособности.

Требования к скрипту:

должен содержать действия с одним из сайтов, доступных в общем до- ступе (варианты в пункте «Список возможных сайтов»);

количество команд в скрипте – от 5;

скрипт может быть открыт на другой машине и запуститься без ошибок. Если для определённых действий требуется аутентификация, можно использо- вать тестовые логин-пароль, которые прямо используются в скрипте.

должно содержаться несколько проверок: и assert, и verify (в сопроводи- тельном письме желательно указать, почему в одном случае используется assert, а в другом – verify).

Список возможных сайтов: www.tut.by; www.vkontakte.ru; www.onliner.by.

Повышенная сложность: www.facebook.com; www.gmail.com.

Тема 3. Документирование

Практическая работа №1. Оценка сложности алгоритмов сортировки.

Цель работы: изучение требований к создаваемому программному продукту, разработка технического задания

Программа выполнения работы

1. Изучить нормативные документы по разработке технического задания на разработку программного продукта.
2. Разработать техническое задание на программный продукта по заданному варианту.

Содержание отчета

Техническое задание на программный продукт

Методические указания

Техническое задание представляет собой документ, в котором сформулированы основные цели разработки, требования к программ- ному продукту, определены сроки и этапы разработки и регламенти- рован процесс приемно-сдаточных испытаний. В разработке техниче- ского задания участвуют как представители заказчика, так и предста- вители исполнителя. В основе этого документа лежат исходные тре- бования заказчика, анализ передовых достижений техники, результа- ты выполнения научно-исследовательских работ, предпроектных ис- следований, научного прогнозирования и т. п.

Основные факторы, определяющие характеристики разрабатывае- мого программного обеспечения:

- исходные данные и требуемые результаты, которые определяют *функции* программы или системы;
- среда функционирования (программная и аппаратная); может быть задана, а может выбираться для обеспечения параметров, указанных в техническом задании;
- возможное взаимодействие с другим программным обеспече- нием или специальными техническими средствами - также может

быть определено, а может выбираться исходя из набора выполняемыхфункций.

Разработка технического задания выполняется в следующей по- следовательности. Прежде всего, устанавливают набор выполняемых функций, а также перечень и характеристики исходных данных. Затемопределяют перечень результатов, их характеристики и способы пред-ставления.

Далее уточняют среду функционирования программного обеспече- ния: конкретную комплектацию и параметры технических средств, версию используемой операционной системы и, возможно, версии и параметры другого установленного программного обеспечения, с ко- торым предстоит взаимодействовать будущему программному про- дукту.

В случаях, когда разрабатываемое программное обеспечение соби- рает и хранит некоторую информацию или включается в управление каким-либо техническим процессом, необходимо также четко регла- ментировать действия программы в случае сбоев оборудования и энергоснабжения.

На техническое задание существует стандарт ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению». В соответствии с этим стандартом техническое задание должно содер- жать следующие разделы:

- введение;
- основания для разработки;
- назначение разработки;

- требования к программе или программному изделию;
- требования к программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки.

При необходимости допускается в техническое задание включать приложения.

Рассмотрим более подробно содержание каждого раздела.

Введение должно включать наименование и краткую характеристи- ку области применения программы или программного продукта, а также объекта (например, системы) в котором предполагается их ис- пользовать. Назначение введения - продемонстрировать актуальность данной разработки и показать, какое место эта разработка занимает в ряду подобных.

Раздел *Основания для разработки* должен содержать наименова- ние документа, на основании которого ведется разработка, организа- ции, утвердившей данный документ, и наименование или условное обозначение темы разработки. Таким документом может служить план, приказ, договор и т. п.

Раздел *Назначение разработки* должен содержать описание функ- ционального и эксплуатационного назначения программного продук- та с указанием категорий пользователей.

Раздел *Требования к программе или программному изделию* должен включать следующие подразделы:

- требования к функциональным характеристикам;
- требования к надежности;
- условия эксплуатации;
- требования к составу и параметрам технических средств;

- требования к информационной и программной совместимости;
- требования к маркировке и упаковке;
- требования к транспортированию и хранению;
- специальные требования.

Наиболее важным из перечисленных выше является подраздел *Требования к функциональным характеристикам.* В этом разделе должны быть перечислены выполняемые функции и описаны состав, характеристики и формы представления исходных данных и результа-тов. В этом же разделе при необходимости указывают критерии эф- фективности: максимально допустимое время ответа системы, мак- симальный объем используемой оперативной и/или внешней памяти и др.

Примечание. Если разработанное программное обеспечение не будет выполнять указанных в техническом задании функций, то оно считается не соответствующим техническому заданию, т. е. непра- вильным с точки зрения критериев качества. Универсальность буду- щего продукта также обычно специально не оговаривается, но подра- зумевается.

В подразделе *Требования к надежности* указывают уровень на- дежности, который должен быть обеспечен разрабатываемой систе- мой и время восстановления системы после сбоя. Для систем с обыч- ными требованиями к надежности в этом разделе иногда регламенти- руют действия разрабатываемого продукта по увеличению надежно- сти результатов (контроль входной и выходной информации, создание резервных копий промежуточных результатов и т. п.).

В подразделе *Условия эксплуатации,* указывают особые требо- вания к условиям эксплуатации: температуре окружающей среды, от- носительной влажности воздуха и т. п. Как правило, подобные требо- вания формулируют, если разрабатываемая система будет эксплуати- роваться в нестандартных условиях или использует специальные внешние устройства, например для

хранения информации. Здесь же указывают вид обслуживания, необходимое количество и квалифика- ция персонала. В противном случае допускается указывать, что тре- бования не предъявляются.

В подразделе *Требования к составу и параметрам технических средств* указывают необходимый состав технических средств с ука- занием ихосновных технических характеристик: тип микропроцессо- ра, объем памяти, наличие внешних устройств и т. п. При этом часто указывают два варианта конфигурации: минимальный и рекомендуе- мый.

В подразделе *Требования к информационной и программной совместимости* при необходимости можно задать методы решения, оп-ределить язык или среду программирования для разработки, а также используемую операционную систему и другие системные и пользо- вательские программные средства, с которым должно взаимодейство-вать разрабатываемое программное обеспечение. В этом же разделе при необходимости указывают, какую степень защиты информации необходимо предусмотреть.

В разделе *Требования к программной документации* указывают необходимость наличия руководства программиста, руководства поль-зователя, руководства системного программиста, пояснительной за- писки и т. п. На все эти типы документов также существуют ГОСТы.

В разделе *Технико-экономические показатели* рекомендуется указывать ориентировочную экономическую эффективность, предпо- лагаемую годовую потребность и экономические преимущества по сравнению с существующими аналогами.

В разделе *Стадии и этапы разработки* указывают стадии раз- работки, этапы и содержание работ с указанием сроков разработки и исполнителей.

В разделе *Порядок контроля и приемки* указывают виды испытаний и общие требования к приемке работы.

В приложениях при необходимости приводят: перечень научно-исследовательских работ, обосновывающих разработку; схемы алго-

ритмов, таблицы, описания, обоснования, расчеты и другие докумен- ты, которые следует использовать при разработке.

В зависимости от особенностей разрабатываемого продукта раз- решается уточнять содержание разделов, т. е. использовать подразде- лы, вводить новые разделы или объединять их.

В случаях, если какие-либо требования, предусмотренные тех- ническим заданием, заказчик не предъявляет, следует в соответст- вующем месте указать «Требования не предъявляются».

Разработка технического задания - процесс трудоемкий, тре- бующий определенных навыков. Наиболее сложным, как правило, яв- ляется четкое формулирование основных разделов: введения, назна- чения и требований к программному продукту.

Практическая работа №2. Оценка сложности алгоритмов поиска.

Цель работы: оценка показателей связности и сцепления модульной программной системы

Программа выполнения работы

1. Изучить методические указания по расчету характеристик модульной программной системы и показателей ее сложности.
2. Вычислить характеристики модульности и сложности программы для заданного варианта.

Методически указания

Модуль - фрагмент программного текста, являющийся строи- тельным блоком для физической структуры системы. Как правило, модуль состоит из интерфейсной части и части-реализации.

Модульность — свойство системы, которая может подвергаться декомпозиции на ряд внутренне связанных и слабо зависящих друг от друга модулей. По определению Г. Майерса, модульность — свойст- во ПО, обеспечивающее интеллектуальную возможность создания сколь угодно сложной программы.

Связность модуля (Cohesion) — это мера зависимости его частей. Связность — внутренняя характеристика модуля. Чем выше связ- ность модуля, тем лучше результат проектирования, то есть тем «черней» его ящик (капсула, защитная оболочка модуля), тем меньше «ручек управления» на нем находится и тем проще эти «ручки».

Для измерения связности используют понятие силы связности (СС).

Существует 7 типов связности:

1. Связность по совпадению (СС=0). В модуле отсутствуют явно выраженные внутренние связи.

2. Логическая связность (СС=1). Части модуля объединены по принципу функционального подобия. Например, модуль состоит из разных подпрограмм обработки ошибок. При использовании такого модуля клиент выбирает только одну из подпрограмм.

Недостатки:

- сложное сопряжение;
- большая вероятность внесения ошибок при изменении сопряжения ради одной из функций.

3. Временная связность (СС=3). Части модуля не связаны, но необходимы в один и тот же период работы системы.

Недостаток: сильная взаимная связь с другими модулями, отсю-да — сильная чувствительность внесению изменений.

4. Процедурная связность (СС=5). Части модуля связаны порядком выполняемых ими действий, реализующих некоторый сценарий поведения.

5. Коммуникативная связность (СС=7). Части модуля связаны по данным (работают с одной и той же структурой данных).

6. Информационная (последовательная) связность (СС=9). Выходные данные одной части используются как входные данные в другой части модуля.

7. Функциональная связность (СС=10). Части модуля вместе реализуют одну функцию.

Отметим, что типы связности 1,2,3 — результат неправильного Если

принадлежат к одной категории, то уровень связности — по совпаде- нию.Конец алгоритма.

Возможны более сложные случаи, когда с модулем ассоциируются несколько уровней связности. В этих случаях следует применять одноиз двух правил:

- правило параллельной цепи. Если все действия модуля имеют несколько уровней связности, то модулю присваивают самый сильный уровень связности;
- правило последовательной цепи. Если действия в модуле имеют разные уровни связности, то модулю присваивают самый слабый уровень связности.

Например, модуль может содержать некоторые действия, которые связаны процедурно, а также другие действия, связные по совпаде- нию. В этом случае применяют правило последовательной цепи и в целом модуль считают связным по совпадению.

Сцепление модулей. Сцепление (Coupling) — мера взаимозави- симости модулей по данным. Сцепление — внешняя характеристика модуля, которую желательно уменьшать.

Количественно сцепление измеряется степенью сцепления (СЦ).

Выделяют 6 типов сцепления.

1. Сцепление по данным (СЦ=1). Модуль А вызывает модуль В.

Все входные и выходные параметры вызываемого модуля — про- стые элементы данных.

2. Сцепление по образцу (СЦ=3). В качестве параметров используются структуры данных.
3. Сцепление по управлению (СЦ=4). Модуль А явно управляет функционированием модуля В (с помощью флагов или переключателей), посылая ему управляющие данные.
4. Сцепление по внешним ссылкам (СЦ=5). Модули А и В ссыла-

ются на один и тот же глобальный элемент данных.

5. Сцепление по общей области (СЦ=7). Модули разделяют одну и ту же глобальную структуру данных.
6. Сцепление по содержанию (СЦ=9). Один модуль прямо ссылается на содержание другого модуля (не через его точку входа). Например, коды их команд перемежаются друг с другом.

Сложность программной системы. В простейшем случае сложность системы определяется как сумма мер сложности ее моду- лей. Сложность модуля может вычисляться различными способами.

М. Холстед предложил меру длины N модуля:

$$N \approx n_1 log_2 (n_1) + n_2 log_2 (n_2),$$

где n_1 — число различных операторов, n_2 — число различных опе- рандов.

В качестве второй метрики М. Холстед рассматривал объем V мо- дуля (количество символов для записи всех операторов и операндов текста программы):

$$V = N \times log_2 (n_1 + n_2).$$

Вместе с тем известно, что любая сложная система состоит из элементов и системы связей между элементами и что игнорировать внутрисистемные связи неразумно.

Том МакКейб при оценке сложности ПС предложил исходить из топологии внутренних связей. Для этой цели он разработал метрику цикломатической сложности:

$$V(G) = E\text{-}N + 2,$$

где E — количество дуг, а.N — количество вершин в управляющем графе ПС. Таким образом, при комплексной оценке сложности ПС необходимо рассматривать меру сложности модулей, меру сложностивнешних связей (между модулями) и меру сложности внутреннихсвязей (внутри модулей).

Практическая работа №3. Оформление документации на программные средства с использованием инструментальных средств.

Цель работы: построение диаграммы классов UML

Программа выполнения работы

1. Изучить назначение и методику разработки диаграммы классов.
2. Изучить технологию создания диаграмм классов UML в Rational Rose.
3. Разработать UML диаграмму классов для заданного варианта.

Методически указания

- Диаграмма *классов* является основным логическим представлением модели и содержит детальную информацию о внутреннем устройстве объектно-ориентированной программной системы или, используя современную терминологию, об архитектуре программной системы. Активизировать рабочее окно диаграммы *классов* можно несколькими способами:

- окно диаграммы *классов* появляется по умолчанию в рабочем окне диаграммы после создания нового проекта;

- щелкнуть на кнопке с изображением диаграммы *классов* на стандартной панели инструментов;

- раскрыть логическое представление (Logical View) в браузере проекта и дважды щелкнуть на пиктограмме Main (Главная);

- выполнить операцию главного меню: Browse →Class Diagram (Обзор →Диаграмма *классов*).

- При этом появляется новое окно с чистым рабочим листом диаграммы

классов и специальная панель инструментов, содержащая кнопки с изображением графических примитивов, необходимых для разработки диаграммы *классов* (табл. 4.1). Назначение отдельных кнопок панели можноузнать также из всплывающих подсказок. На

специальной панели инструментов по умолчанию присутствует толь- ко часть пиктограмм элементов, которые могут быть использованы для построения диаграммы *классов*. Добавить кнопки с пиктограмма- ми других графических элементов таких как, например, отношения агрегации и композиции, шаблон, *класс* бизнес-сущность, *управляю- щий класс*, или удалить ненужные кнопки можно с помощью на- стройки специальной панели инструментов. Соответствующее диало- говое окно настройки специальной панели инструментов для диа- граммы *классов* можно вызвать аналогично другим панелям с помо- щью операции контекстного меню **Customize** (Настройка) при пози- ционировании курсора на специальной панели инструментов.

Таблица 4.1.

Назначение кнопок специальной панели инструментовдля диаграммы классов

Графическое изображение	Всплывающая подсказка	Назначение кнопки
	Selection Tool	Превращает изображение курсора в форму стрелки для последующего выделения эле- ментов на диаграмме
	Text Box	Добавляет на диаграмму текстовую область
	Note	Добавляет на диаграмму примечание
	Anchor Note to Item	Добавляет на диаграмму связь примечания с соответствующим графическим элемен- том диаграммы

	Class	Добавляет на диаграмму *класс*
	Interface	Добавляет на диаграмму *интерфейс*
	Unidirectional Association	Добавляет на диаграмму направленную *ассоциацию*
	Association Class	Добавляет на диаграмму *ассоциацию класс*
	Package	Добавляет на диаграмму пакет
	Dependency or Instantiates	Добавляет на диаграмму отношение зависимости
	Generalization	Добавляет на диаграмму отношение обобщения

Для построения диаграммы классов рассматриваемой модели банкомата следует описанным выше способом добавить оставшиеся классы и *ассоциации*, а также специфицировать стереотипы, атрибу- ты и операции этих классов. С этой целью следует выполнить сле- дующие действия:

1. Для класса IИнтерфейс Банка добавить операцию: прове-рить идентификатор карточки (идентификатор карточки: Integer) с квантором видимости public. В качестве типа возвращаемого результата для этой операции следует выбрать тип Boolean (логический), а в качестве целочисленного аргумента задать идентификатор карточки. Для задания аргумента необходимо перейти на вкладку Detail (Подробно) окна спецификации свойств даной операции и после добавления аргумента с помощью операции контекстного меню Insert ввести имя аргумента и его тип Integer в соответствующие поля ввода.
2. Для класса IИнтерфейс Банка добавить операцию: открыть счет клиента (идентификатор карточки: Integer) с квантором видимости public. В качестве целочисленного аргумента этой операции следует задать идентификатор карточки.
3. Для класса IИнтерфейс Банка добавить операцию: прове-рить баланс клиента (идентификатор карточки: Integer, введенная сумма наличных: Currency) с квантором видимости public. В качестве типа возвращаемого результата для этой операции следует выбрать тип Boolean (логический). В качестве первого целочисленного аргу- мента этой операции следует задать идентификатор карточки, а в ка- честве второго аргумента - введенная сумма наличных с типом Currency (Денежный).

4. Для класса IИнтерфейс Банка добавить операцию: уменьшить счет клиента (идентификатор карточки: Integer, введенная сумма наличных: Currency) с квантором видимости public. В качестве типа возвращаемого результата для этой операции следует выбрать тип Boolean (логический). В качестве первого целочисленного аргумента этой операции следует задать идентификатор карточки, а в качестве второго аргумента - введенная сумма наличных с типом Currency (Денежный).

5. Для класса Устройство чтения карточки добавить операцию: прочитать идентификатор карточки() с квантором видимости public.

В качестве типа возвращаемого результата для этой операции следует выбрать тип Integer (целочисленный), а в секцию документации дан- ной операции следует ввести поясняющий текст: «Вызывается после того, как кредитная карточка вставлена в Устройство чтения карточ- ки».

1. Для класса Устройство чтения карточки добавить операцию: прочитать ПИН-код() с квантором видимости public. В качестве типа возвращаемого результата для этой операции следует выбрать тип Integer (целочисленный), а в секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как кредитная карточка вставлена в Устройство чтения карточки».

2. Для класса Устройство чтения карточки добавить операцию: вернуть кредитную карточку() с квантором видимости public. В секцию документации данной операции следует ввести поясняющий текст: «Вызывается после завершения транзакции».

3. Для класса Устройство чтения карточки добавить операцию: блокировать кредитную карточку() с квантором видимости public. В секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как установлен факт утраты кредитной карточки владельцем».

4. Добавить класс с именем Экран Банкомата, для которого выбрать стереотип boundary. Данный класс также находится на границе моделируемой системы, на что и указывает этот стереотип. В секцию документации данного класса следует ввести поясняющий текст: «Устанавливается на банкомате».

5. Для класса Экран Банкомата добавить операцию: показать меню опций() с квантором видимости public.

6. Для класса Экран Банкомата добавить операцию: показать меню снятия суммы() с квантором видимости public.

7. Добавить класс с именем Клавиатура Банкомата, для которого выбрать стереотип boundary. В секцию документации данного класса следует ввести поясняющий текст: «Устанавливается на банкомате».

8. Для класса Клавиатура Банкомата добавить операцию: ввести ПИН-код() с квантором видимости public. В качестве типа возвращаемого результата для этой операции следует выбрать тип Integer, а в секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как клиент ввел значение ПИН-кода с клавиатуры».

9. Для класса Клавиатура Банкомата добавить операцию: ввести тип транзакции() с квантором видимости public. В качестве типа возвращаемого результата для этой операции следует выбрать тип Boolean (логический), а в секцию документации данной операции следует ввести поясняющий текст: «Возвращает значение Истина, если клиент выбирает снятие наличных, и значение Ложь, если клиент выбирает получение справки о состоянии счета».

10. Для класса Клавиатура Банкомата добавить операцию: ввести сумму снятия наличных() с квантором видимости public. В качестве типа возвращаемого результата для этой операции следует выбрать тип Currency (Денежный), а в секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как клиент ввел значение снимаемой суммы с клавиатуры».

11. Добавить класс с именем Устройство выдачи наличных, для которого выбрать стереотип boundary. В секцию документации данного класса следует ввести поясняющий текст: «Устанавливается на банкомате».

12. Для класса Устройство выдачи наличных добавить операцию: выдать наличные() с квантором видимости public. В секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как выполнено снятие запрошенной клиен- том суммы со счета».

13. Добавить класс с именем Принтер Банкомата, для которого выбрать стереотип boundary. В секцию документации данного класса следует ввести поясняющий текст: «Устанавливается на банкомате».

14. Для класса Принтер Банкомата добавить операцию: распечатать чек() с квантором видимости public. В секцию документации данной операции следует ввести поясняющий текст: «Вызывается по дополнительному запросу клиента».

15. Добавить направленную *ассоциацию* от класса Контроллер Банкомата к классу Устройство чтения карточки. В качестве *кратности* концов этой *ассоциации* установить значение 1.

16. Добавить направленную *ассоциацию* от класса Контроллер Банкомата к классу Принтер Банкомата. В качестве *кратности* концов этой *ассоциации* установить значение 1.

17. Добавить направленную *ассоциацию* от класса Контроллер Банкомата к классу Клавиатура Банкомата. В качестве *кратности* концов этой *ассоциации* установить значение 1.

18. Для класса Клавиатура Банкомата добавить операцию: ввести тип транзакции() с квантором видимости public. В качестве типа возвращаемого результата для этой операции следует выбрать тип Boolean (логический), а в секцию документации данной операции следует ввести поясняющий текст: «Возвращает значение Истина, если клиент выбирает снятие наличных, и значение Ложь, если клиент выбирает получение справки о состоянии счета».

19. Для класса Клавиатура Банкомата добавить операцию: ввести сумму снятия наличных() с квантором видимости public. В качестве типа возвращаемого результата для этой операции следует выбрать тип Currency (Денежный), а в секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как клиент ввел значение снимаемой суммы с клавиатуры».

20. Добавить класс с именем Устройство выдачи наличных, для которого выбрать стереотип boundary. В секцию документации данного класса следует ввести поясняющий текст: «Устанавливается на банкомате».

21. Для класса Устройство выдачи наличных добавить операцию: выдать наличные() с квантором видимости public. В секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как выполнено снятие запрошенной клиен- том суммы со счета».

22. Добавить класс с именем Принтер Банкомата, для которого выбрать стереотип boundary. В секцию документации данного класса следует ввести поясняющий текст: «Устанавливается на банкомате».

23. Для класса Принтер Банкомата добавить операцию: распечатать чек() с квантором видимости public. В секцию документации данной операции следует ввести поясняющий текст: «Вызывается по дополнительному запросу клиента».

24. Добавить направленную *ассоциацию* от класса Контроллер Банкомата к классу Устройство чтения карточки. В качестве *кратности* концов этой *ассоциации* установить значение 1.

25. Добавить направленную *ассоциацию* от класса Контроллер Банкомата к классу Принтер Банкомата. В качестве *кратности* концов этой *ассоциации* установить значение 1.

26. Добавить направленную *ассоциацию* от класса Контроллер Банкомата к классу Клавиатура Банкомата. В качестве *кратности* концов этой *ассоциации* установить значение 1.

Список рекомендуемой литературы

1. Майерс Гленфорд Дж. Искусство тестирования программ. -М.: Финансы и статистика, 2018. - 176 с.
2. Липаев В.В. Тестирование компонентов и комплексовпрограмм. - М.: Синтег, 2019. - 399 с.
3. Винниченко И. В. Автоматизация процессов тестирования:производственно-практическое издание. - СПб. : Питер, 2019. - 202 с.
4. Бек К. Экстремальное программирование: разработка черезтестирование. - СПб. : Питер, 2018. - 224 с.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к проведению практических занятий для студентов
направления *38.02.01*
«Экономика и бухгалтерский учет (по отраслям)»

Составители:
Составитель Д.Г. Сергиенко, преподаватель СПК

В авторской редакции

Компьютерный набор Д.Г. Сергиенко
Подписано к изданию 00.09.2021.
Уч.-изд. л. 1,1.

ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14

Программа обсуждена на заседании методического совета СПК
«19» 03 2021 года. Протокол № 7,
Председатель методического совета СПК

Сергеева С. И. .
(Ф.И.О., подпись)
Программа одобрена на заседании педагогического совета СПК
«26»03 2021 года. Протокол № 7.
Председатель педагогического совета СПК

Облиенко А.В. .
(Ф.И.О., подпись)