

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Кубанский государственный университет»
(ФГБОУ ВО «КубГУ»)



МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ по дисциплине
«Автоматизация тестирования программного обеспечения»
для бакалавров направлений подготовки 01.03.02 «Прикладная математика и
информатика» 02.03.03 «Математическое обеспечение и администрирование
информационных систем» профиль «Технология программирования», 09.03.03
«Прикладная информатика» профиль «Прикладная информатика в экономике»

Краснодар,
2023

Таблица 1. План лабораторных работ

<i>№</i>	<i>Тема лабораторной работы</i>	<i>Срок выполнения (учебные недели)</i>
1	Примитивный поиск дефектов в ПО. Тестирование формы входа	1-2
2	Построение баг-репорта	3-4
3	Чек-лист. Построение классов эквивалентности. Тестирование граничных значений	5-6
4	Тестирование таблиц решений. Попарное тестирование	7-8
5	Сценарное и исследовательское тестирование	9-10
6	Отчеты о тестировании	11-12
7	Чек-лист для тестирования веб-форм	13-14
8	Тестирование строки поиска	15-16

Требования к содержанию отчетов о выполнении лабораторных работ

В процессе изучения курса «Автоматизация тестирования программного обеспечения» должны быть подготовлены и сданы восемь отчетов о выполнении лабораторных работ (см. таблицу 1). Лабораторные работы выполняются в соответствии с заданием.

Отчет должен содержать заголовок, тему лабораторной работы, задание, описание хода выполнения работы, необходимые прикладные материалы (схемы, скриншоты и т.п.) и выводы по работе.

Отчет сдается в электронном виде на паре в установленные сроки.

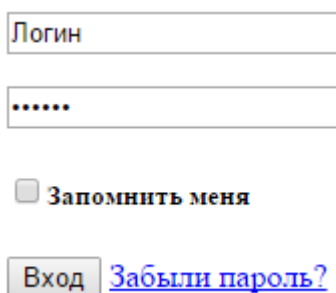
Лабораторная работа 1

Примитивный поиск дефектов в ПО

Задание 1. Написать программу, определяющую тип треугольника по трем его сторонам. Протестировать всевозможные дефекты разработанной другим студентом группы программы по заданию 1.

Тестирование формы входа

Рассмотрим часто встречающуюся форму с двумя полями: логин и пароль.



Логин

.....

☐ Запомнить меня

Вход [Забыли пароль?](#)

Под логином может быть email, номер телефона или ник.

— найти форму входа должно быть легко

Она должна быть на виду или доступна в один клик. Пользователь должен сразу понять, что этот сайт имеет форму входа. Иначе пользователь просто не поймет, или пропустит и не зарегистрируется. Тогда мы потеряем потенциального клиента.

— убедитесь в том, что форма входа имеет поля для логина и пароля, имеет кнопку для входа и ссылку «забыли пароль?»

Формы бывают разные. Обычно это два поля: логин и пароль. Ссылка на регистрацию нужна, если ее нет отдельно на главной странице. Потому что искать ее будут именно на форме входа (если не смогут найти). Ссылку «забыли пароль?» также будут искать на форме входа. Она не должна быть отдельно, чтобы пользователь не тратил время на ее поиски. Он сталкивается с проблемой «вспомнить пароль» именно на форме входа, где пытается его подобрать.

— где находится курсор после загрузки формы входа?

— форма входа содержит ссылку на регистрацию?

— убедитесь в том, что пользователь может войти в систему с действительным логином и паролем

— как отображаются ошибки?

— должно быть видно сообщение об ошибке если введены некорректные данные

Сообщения об ошибках могут отображаться всплывающим окном или появляться текстом рядом с полем.

— сообщение об ошибке должно быть показано

— убедитесь в том, что нельзя авторизоваться с неверными логином и паролем

— убедитесь в том, что есть предупреждение о незаполненном поле

В форме два поля и оба они являются важными, обязательными. И если одно из них пустое, тогда это должно как то выделяться: окрашивать рамку в красный цвет, выводить предупреждение, все что угодно, но только не проходить дальше.

— введите действительный логин и неправильный пароль

— введите неправильный логин и правильный пароль

— оставьте поля пустыми и нажми enter или кликни кнопку вход

— поле логин пустое, поле пароль — правильные данные

— поле логин с правильными данными, поле пароль — пустое

— введите неправильные логин и пароль

— есть ли предупреждение, что превышен лимит символов в поле логин или в поле пароль?

У каждого поля может быть ограничение на количество введенных символов. Другое дело отображается ли оно и если да, то как?

Возьмем поле «логин».

Здравый смысл подсказывает, что это ограничение должно быть одинаково как для формы входа, так и для формы регистрации. Мы можем посмотреть в форме регистрации, какие ограничения стоят для поля «логин».

Например, в форме регистрации для поля логин ограничение не больше 40 символов.

Тогда даже если в форме входа логин принимает больше 40 символов — действительным данным это никак не помешает.

Максимум неудобства, если пользователь ошибется и введет лишний символ в свой 40 символьный логин. В этом случае желательно чтоб поле «логин» из формы входа имело столько же ограничений, сколько и поле «логин» из формы регистрации. Когда в поле логин (формы входа) стоит ограничение меньше 40 символов, могут возникнуть трудности с действительными данными. (40 символов рассматриваются условно. Нет никакого правила определяющего максимальное количество символов для логина).

Если есть максимальное ограничение на количество символов в поле пароль и это никак не сообщается — могут быть проблемы. Пользователь может думать, что ввел один пароль, а на самом деле используется его укороченная версия. Например, форма может обрезать последние символы. Или просто не дать их ввести. Но пользователь об этом не догадается, так как обычно символы в поле пароль маскируются звездочками.

— убедитесь в том, что существует ограничение на общее число неудачных попыток.

Необходима защита от подбора пароля. Самый распространенный способ — ограничить число неудачных попыток. Их число может быть разным. После которого аккаунт и\или возможность входа блокируется на некоторое время.

— Есть ли чувствительность регистра у полей логина и пароля?

Чувствительность к регистру (различает ли поле маленькие буквы от больших) в поле для логина, например, быть не должно. Поле не должно считать разными usernames login и LOGIN. А вот для поля пароль это необходимо.

— поле пароль маскирует символы?(обычно звездочками)

— есть ли минимальная длина у поля пароль?

Надежность пароля можно рассматривать как длину символов (чем длиннее пароль тем тяжелее его подобрать) и количество вариантов для одного символа (чем больше вариантов, тем больше комбинации и больше времени понадобится чтобы проверить их все). Поэтому чаще требуют чтобы пароль содержал и цифры и буквы, верхний и нижний регистр, спецсимволы — чтобы максимально расширить количество вариантов для одного символа.

- Арабские цифры = 10 вариантов
- Латинский алфавит = 26 вариантов

- Латинский алфавит только с нижним регистром (или только с верхним) и цифры = 36 вариантов
- Латинский алфавит с верхним и нижним регистром = 52 варианта
- Латинский алфавит с верхним и нижним регистром + цифры = 62 варианта
- Латинский алфавит с верхним и нижним регистром + цифры + спецсимволы и без пробела = 94 варианта
- Латинский алфавит с верхним и нижним регистром + цифры + спецсимволы + пробел (все символы таблицы ASCII) = 95 вариантов
- Символы расширенной таблицы ASCII - 218 вариантов

— логин может содержать буквы, цифры и точку.

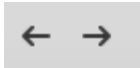
— логин не должен содержать спецсимволы

— пользователь должен осознавать, авторизирован он или нет

Система должна показывать, авторизирован пользователь или нет. Когда пользователь авторизирован, ему не должна показываться форма входа или форма регистрации. Иначе он подумает, что не авторизирован.

— нажатие кнопки логута переводит на форму входа?

— как реагирует система на кнопки браузера «вперед» и «назад» после авторизации? А как после логута?

Стрелками браузера  удобно пользоваться если нужно вернуться на несколько страниц назад, а потом обратно вперед. Но такой переход по

страницам не должен быть переходом из одного состояния в другое. Если пользователь уже авторизован, то возвращение на страницу назад не должно возвращать его в состояние неизвестного посетителя. Так же и при выходе из аккаунта (логаут).

— убедитесь в том, что после авторизации нажав кнопку в браузере «назад» пользователь не разлогинится обратно

— кнопка назад в браузере не должна возвращать не авторизованного пользователя обратно авторизованным

— проверьте работу чекбокса «запомнить меня»(если есть)

Если на странице входа в систему предусмотрена возможность сохранения имени пользователя и пароля, с помощью флажка, то пользователю не должно быть предложено повторно ввести учетные данные.

— копираст разрешен для поля логина, но не желателен для поля пароль

Почему запрещается копираст для поля пароль? Иногда бывает риск зацепить лишний пробел при копировании из файла в поле для пароля.

Задание 2.

Протестируйте форму входа некоторого мобильного приложения, подготовьте чек-лист тестирования.

Лабораторная работа 2

Построение баг-репорта

Структура баг-репорта.

Шапка	
Короткое описание (Summary)	Короткое описание проблемы, явно указывающее на причину и тип ошибочной ситуации.
Проект (Project)	Название тестируемого проекта
Компонент приложения (Component)	Название части или функции тестируемого продукта
Номер версии (Version)	Версия на которой была найдена ошибка
Серьезность (Severity)	Наиболее распространена пятиуровневая система градации серьезности дефекта: <ul style="list-style-type: none">• S1 Блокирующий (Blocker)• S2 Критический (Critical)• S3 Значительный (Major)• S4 Незначительный (Minor)• S5 Тривиальный (Trivial)
Приоритет (Priority)	Приоритет дефекта: <ul style="list-style-type: none">• P1 Высокий (High)• P2 Средний (Medium)• P3 Низкий (Low)
Статус (Status)	Статус бага. Зависит от используемой процедуры и жизненного цикла бага (bug workflow and life cycle)
Автор (Author)	Создатель баг репорта
Назначен на (Assigned To)	Имя сотрудника, назначенного на решение проблемы

Окружение	
ОС / Сервис Пак и т.д. / Браузера + версия / ...	Информация об окружении, на котором был найден баг: операционная система, сервис пак, для WEB тестирования - имя и версия браузера и т.д.
...	
Описание	
Шаги воспроизведения (Steps to Reproduce)	Шаги, по которым можно легко воспроизвести ситуацию, приведшую к ошибке.
Фактический Результат (Result)	Результат, полученный после прохождения шагов к воспроизведению
Ожидаемый результат (Expected Result)	Ожидаемый правильный результат
Дополнения	
Прикрепленный файл (Attachment)	Файл с логами, скриншот или любой другой документ, который может помочь прояснить причину ошибки или указать на способ решения проблемы

Градация Серьезности дефекта (Severity)

S1 Блокирующая (Blocker)

Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы.

S2 Критическая (Critical)

Критическая ошибка, неправильно работающая ключевая бизнес логика, дыра в системе безопасности, проблема, приведшая к временному падению сервера или приводящая в нерабочее состояние некоторую часть системы, без возможности решения проблемы, используя другие входные точки. Решение проблемы необходимо для дальнейшей работы с ключевыми функциями тестируемой системой.

S3 Значительная (Major)

Значительная ошибка, часть основной бизнес логики работает некорректно. Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки.

S4 Незначительная (Minor)

Незначительная ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.

S5 Тривиальная (Trivial)

Тривиальная ошибка, не касающаяся бизнес логики приложения, плохо воспроизводимая проблема, малозаметная посредством пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта.

Градация Приоритета дефекта (Priority)

P1 Высокий (High)

Ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.

P2 Средний (Medium)

Ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.

P3 Низкий (Low)

Ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения.

Порядок исправления ошибок по их приоритетам:

High -> Medium -> Low

Требования к количеству открытых багов

Следующий подход к определению требований к количеству открытых багов:

- Наличие открытых дефектов P1, P2 и S1, S2, считается неприемлемым для проекта. Все подобные ситуации требуют срочного решения и идут под контроль к менеджерам проекта.
- Наличие строго ограниченного количества открытых ошибок P3 и S3, S4, S5 не является критичным для проекта и допускается в выдаваемом приложении. Количество же открытых ошибок зависит от размера проекта и установленных критериев качества.

Все требования к открытым ошибкам оговариваются и документируются на этапе принятия решения о качестве разрабатываемого продукта.

Баг репорт - это технический документ и в связи с этим хотим отметить, что язык описания проблемы должен быть техническим. Должна использоваться правильная терминология при использовании названий элементов пользовательского интерфейса (editbox, listbox, combobox, link, text area, button, menu, popup menu, title bar, system tray и т.д.), действий пользователя (click link, press the button, select menu item и т.д.) и полученных результатах (window is opened, error message is displayed, system crashed и т.д.).

Требования к обязательным полям баг репорта

Отметим, что обязательными полями баг репорта являются: **короткое описание** (*Bug Summary*), **серьезность** (*Severity*), **шаги к воспроизведению** (*Steps to reproduce*), **результат** (*Actual Result*), **ожидаемый результат** (*Expected Result*). Ниже приведены требования и примеры по заполнению этих полей.

Короткое описание

Название говорит само за себя. В одном предложении вам надо уместить смысл всего баг репорта, а именно: коротко и ясно, используя правильную терминологию сказать что и где не работает. Например:

1. Приложение зависает, при попытке сохранения текстового файла размером больше 50Мб.

2. Данные на форме "Профайл" не сохраняются после нажатия кнопки "Сохранить".

В дополнение предлагаем изучить *Принцип "Где? Что? Когда?"*:

"В чем этот принцип состоит?"

Составьте предложение, в котором факты дефекта изложены в следующей последовательности:

- *Где?: В каком месте интерфейса пользователя или архитектуры программного продукта находится проблема. Причем, начинайте предложение с существительного, а не предлога.*

- *Что?: Что происходит или не происходит согласно спецификации или вашему представлению о нормальной работе программного продукта. При этом указывайте на наличие или отсутствие объекта проблемы, а не на его содержание (его указывают в описании). Если содержание проблемы варьируется, все известные варианты указываются в описании.*

- *Когда?: В какой момент работы программного продукта, по наступлению какого события или при каких условиях проблема проявляется.*

Почему последовательность должна быть именно такой? В таком виде незнакомые дефекты удобнее сортировать по sumtagu как показывает практика (ведь, скорее всего, именно среди дефектов других инженеров будет производиться поиск дубликатов). Если вы другого мнения - придумайте свою последовательность, но она должна стать единой для всех без исключения членов проекта, иначе вы не добьетесь необходимого результата."

Серьезность

В двух словах можно отметить, что если проблема найдена в ключевой функциональности приложения и после ее возникновения приложение становится полностью недоступно, и дальнейшая работа с ним невозможна, то она **блокирующая**. Обычно все блокирующие проблемы находятся во время первичной проверки новой версии продукта (**Build Verification Test**, **Smoke Test**), т.к. их наличие не позволяет полноценно проводить тестирование. Если же

тестирование может быть продолжено, то серьезность данного дефекта будет **критическая**.

На счет **значительных**, **незначительных** и **тривиальных** ошибок вопрос достаточно прозрачный и на наш взгляд не требует лишних объяснений.

Шаги к воспроизведению / Результат / Ожидаемый результат

Очень важно четко описать все шаги, с упоминанием всех вводимых данных (имени пользователя, данных для заполнения формы) и промежуточных результатов.

Например:

Шаги к воспроизведению

1. Войдите в системы: Пользователь Тестер1, пароль xxxXXX

--> Вход в систему осуществлен

2. Кликните линк Профайл

--> Страница Профайл открылась

3. Введите Новое имя пользователя: Тестер2

4. Нажмите кнопку Сохранить

Результат

На экране появилась ошибка. Новое имя пользователя не было сохранено

Ожидаемый результат

Страница профайл перегрузилась. Новое значение имени пользователя сохранено.

Основные ошибки при написании багов репортов

Недостаточность предоставленных данных

Не всегда одна и та же проблема проявляется при всех вводимых значениях и под любым вошедшим в систему пользователем, поэтому настоятельно рекомендуется вносить все необходимые данные в баг репорт

Определение серьезности

Очень часто происходит либо завышение, либо занижение серьезности дефекта, что может привести к неправильной очередности при решении проблемы.

Язык описания

Часто при описании проблемы используются неправильная терминология или сложные речевые обороты, которые могут ввести в заблуждение человека, ответственного за решение проблемы.

Отсутствие ожидаемого результата

В случаях, если вы не указали, что же должно быть требуемым поведением системы, вы тратите время разработчика, на поиск данной информации, тем самым замедляете исправления дефекта. Вы должны указать пункт в требованиях, написанный тест кейс или же ваше личное мнение, если эта ситуация не была документирована.

Задание. Выполнить поиск багов и оформление баг-репорта.

Рассмотреть сайты с целью поиска багов: <https://www.orange-elephant.ru/>, <https://flowwow.com/krasnodar/>, <https://sadvod-opt.com/> и др.

Лабораторная работа 3

Оформление чек-листов. Построение классов эквивалентности.

Тестирование граничных значений

Чек-лист.

Контрольный список/лист проверок - это список проверок, которые помогают тестировщику протестировать приложение или отдельные функции. Основная цель чек-листа состоит в том, чтобы вы не забыли проверить всё, что планировали. Классический чек-лист состоит из:

- 1-й столбец: заголовки тест-кейсов, структурированные по разделам/функционалу, или любые определенные составителем пункты;
- 2-й столбец для отметки: pass/fail;
- 3-й столбец опционально под заметки.

Если чек-лист используется еще и для наглядного отображения хода тестирования (а-ля test run), 2-й столбец может иметь опции: пусто (еще не проверялось)/успех/ошибка/пропущено или заблокировано (например, другим дефектом).

Задание 1.

Напишите чеклист для тестирования API добавляющее атрибут пользователю.

<pre>{ "userId": "string", "clientType": "string", "attributeId": 0, "attributeData": "string", }</pre>	<pre>{ "userId": "1234567", "clientType": "Юр", "attributeId": 13, "attributeData": "Москва" }</pre>
---	--

Построение классов эквивалентности.

Рассмотрим ипотечную компанию "Гуфи" (ИКГ). Она будет выдавать ипотеки людям с доходами от \$1000 до \$83333 в месяц. Те, у кого доходы ниже \$1000 в месяц, не имеют на это права. Тем, у кого доходы выше \$83333 в месяц, не нужна ипотека, т.к. они просто рассчитываются наличными.

Для корректного входного значения можно было бы выбрать \$1342 в месяц. Для некорректных - \$123 в месяц и \$90000 в месяц.

1) Непрерывные классы эквивалентности.



Если входное состояние принимает дискретные значения в пределах диапазона допустимых значений, то обычно существует один корректный и два некорректных класса. ИКГ выдаст ипотеку для покупки от одного до пяти домов. Ноль или меньше домов, а также шесть и больше — не корректные входные значения. Также это не могут быть ни дробные, ни десятичные значения, такие как $2\frac{1}{2}$ или 3,14159.

2) Дискретные классы эквивалентности



Для корректного входного значения можно выбрать два дома. Некорректными могут быть -2 и 8. ИКГ может выдавать ипотеки только человеку. Они не выдают ипотеки компаниям, трастам, партнерствам или другим легальным организациям.

3) Классы эквивалентности с одним возможным значением.



Для верных входных данных будем использовать "человек". Для неверных можно использовать "корпорация" или "траст" или любую другую случайную строку. Следует создать как минимум один тестовый сценарий с неверными данными.

ИКГ выдают ипотеки под квартиры, таунхаусы и частные дома. Они не выдают ипотеки под дома на двух хозяев, передвижные дома, домики на дереве или другие типы жилья.

4) Классы эквивалентности с несколькими возможными значениями.



Для верных входных данных можно выбрать значение "квартира", "таунхаус" или "частный дом". Несмотря на то, что правило указывает выбрать один тест из класса эквивалентности, более комплексный подход заключается в создании тестового сценария на каждое значение в классе эквивалентности. Это имеет смысл, если список верных значений мал. Но, если у нас список из пятидесяти городов, то все зависит от рисков организации, если тестировщик пропустит что-либо жизненно важное.

Создавать отдельные тесты для каждого отдельного класса эквивалентности всех входных данных, вводимых в систему неэффективно. Удобнее создавать тестовые сценарии, которые будут проверять некоторое число полей ввода одновременно. Например, мы можем создать один тестовый сценарий для следующих комбинаций входных данных:

Таблица 1. Тестовый сценарий для верных входных данных.

Ежемесячный доход	Количество жилых помещений	Заявитель	Вид жилья	Результат
\$5000	2	Человек	Квартира	корректное значение

Но использовать такой подход для неверных данных опасно.

Таблица 2. Тестовый сценарий неверных данных. Это плохой подход.

Ежемесячный доход	Количество жилых помещений	Заявитель	Вид жилья	Результат
\$100	8	Партнерство	Домик на дереве	некорректное значение

Если система принимает эти данные как верные, то ясно, что система не проверяет все четыре поля для ввода как следует. Но если система отклонит эти данные как неверные, то тестировщик не сможет понять, какое поле было отклонено.

Задание 2. Разработать набор тест-кейсов, различающихся на одно недопустимое значение. Причем, входные значения (как корректные, так и некорректные) могут варьироваться.

Задание 3.

Для веб-сайта Регистрационной системы Государственного Университета определите классы эквивалентности и подходящие тест-кейсы для следующего:

- a. Индекс – шесть цифр.
- b. Регион – строка 15 символов (область, край, республика).
- c. Населенный пункт – строка 12 символов (город).
- d. Фамилия – пятнадцать символов (включая алфавитные символы, точку, дефис, апостроф, пробел и цифры).
- e. Идентификатор пользователя – восемь символов, как минимум два из которых не алфавитные (число, спецсимвол, непечатаемый символ).
- f. Идентификатор студента – восемь символов. Первые два представляют собой выпускающую кафедру, а последние шесть являются уникальным шестизначным номером.

Допустимые сокращения кафедр: ИТ (информационных технологий), ПМ (прикладной математики), ММ (математического моделирования), ВТ (вычислительных технологий), АД (анализа данных).

Тестирование граничных значений

Рассмотрим обработку заявок на вакансии в зависимости от возраста соискателя. Эти правила:

- от 0 до 16 - не принимаются

- от 16 до 18 - могут быть приняты только на неполный рабочий день
- от 18 до 55 - могут быть приняты как сотрудники на полный рабочий день
- от 55 до 99 - не принимаются

Возникают проблемы на границах - это "края" каждого класса. Возраст "16" входит в два различных класса эквивалентности (как и "18", и "55"). Первое правило гласит не нанимать шестнадцатилетних. Второе правило гласит, что шестнадцатилетние могут быть наняты на неполный рабочий день.

Возможно, что тестируемая организация имеет в виду следующее:

- от 0 до 15 - не принимаются
- от 16 до 17 - могут быть приняты только на неполный рабочий день

28

- от 18 до 54 - могут быть приняты как сотрудники на полный рабочий день
- от 55 до 99 - не принимаются

Как рассматривать возраст -3 и 101?

В этом примере интересными значениями на границах или вблизи них являются $\{-1, 0, 1\}$, $\{15, 16, 17\}$, $\{17, 18, 19\}$, $\{54, 55, 56\}$ и $\{98, 99, 100\}$. Другие значения, например $\{-42, 1001, \text{FRED}, \%\$ \# @\}$ могут быть включены в зависимости от предусловий документации модуля.

Для создания тест-кейсов для каждого граничного значения выберите одну точку на границе, одну точку чуть ниже границы и одну точку чуть выше границы.

Задание 4. Определите для ипотечной компании "Гуфи" из лаб.№3 набор тест-кейсов, содержащих комбинации допустимых (на границе) и недопустимых (за границей) значений.

Задание 5. Для веб-сайта Регистрационной системы Государственного Университета определите границы и подходящие тестовые граничные значения для следующего:

- Индекс – шесть цифр.
- Сначала рассмотрите почтовый индекс только с точки зрения цифр. Затем определите минимальный и максимальный корректные почтовые индексы в РФ.

Дополнительно, определите формат минимального и максимального допустимых значений почтовых кодов для Беларуси.

- Фамилия – пятнадцать символов (включая алфавитные символы, точек, дефисы, апострофы, пробелы и числа). Дополнительно создайте несколько очень сложных фамилий. Можете ли вы определить "правила" для корректных фамилий?

- Идентификатор пользователя – восемь символов, как минимум два из которых являются не буквой (число, спецсимвол, непечатаемый символ).

- Идентификатор курса – три буквенных символа, представляющие факультет последующим шестизначным числом, которое является уникальным идентификационным номером курса.

Возможные факультеты:

- PHY - физика
- EGR - инжиниринг
- ENG - английский
- LAN - иностранные языки
- CHM - химия
- MAT - математика
- PED - физкультура
- SOC – социология

Лабораторная работа 4

Тестирование таблиц решений. Парное тестирование

Таблицы решений представляют собой комплекс бизнес-правил, основанных на заданных условиях.

Таблица 1. Основной формат таблицы решений

Правило 1 Правило 2 ... Правило r

Условия

Условие-1

Условие-2

...

Условие-m

Действия

Действие-1

Действие-2

...

Действие-n

Пример. Компания по автострахованию дает скидку водителям, которые состоят в браке и/или хорошо учатся. Давайте начнем с **условий**. Следующая таблица решений имеет два условия, каждое из которых может принимать значение "Да" или "Нет".

Таблица 2. Таблица решений с двумя бинарными условиями

	Правило 1	Правило 2	Правило 3	Правило 4
Условия				
Состоит в браке?	Да	Да	Нет	Нет
Хороший студент?	Да	Нет	Да	Нет

Каждое правило является причиной "запуска" действия. Каждое правило может задать действие, уникальное для этого правила, или правила могут иметь общие действия.

Таблица 3. Добавление единственного действия в таблицу решений

	Правило 1	Правило 2	Правило 3	Правило 4
Условия				
Состоит в браке?	Да	Да	Нет	Нет
Хороший студент?	Да	Нет	Да	Нет
Действия				
Скидка (\$)	60	25	50	0

Для каждого правила с помощью таблицы решений можно указать более одного действия. Эти правила могут быть уникальными или быть общими.

Таблица 4. Таблица решений с несколькими действиями.

	Правило 1	Правило 2	Правило 3	Правило 4
Условия				
Условие-1	Да	Да	Нет	Нет
Условие-2	Да	Нет	Да	Нет
Действия				
Действие-1	Выполнить X	Выполнить Y	Выполнить X	Выполнить Z
Действие-2	Выполнить A	Выполнить B	Выполнить B	Выполнить B

В такой ситуации выбрать тесты просто: каждое правило (вертикальная колонка) становится тест-кейсом. Условия указывают на входные значения, а действия - на ожидаемые результаты. В предыдущих примерах используются простые бинарные условия, но условия могут быть более сложными.

Таблица 5. Таблица решений со сложными условиями.

	Правило 1	Правило 2	Правило 3	Правило 4
Условия				
Условие-1	0-1	1-10	10-100	100-1000
Условие-2	<5	5	6 или 7	>7

Действия

Действие-1	Выполнить X	Выполнить Y	Выполнить X	Выполнить Z
Действие-2	Выполнить A	Выполнить B	Выполнить B	Выполнить B

В этой ситуации выбор тестов становится чуть более сложным - каждое правило (вертикальная колонка) становится тест-кейсом, но при этом должны быть выбраны значения, удовлетворяющие условиям.

Выбирая необходимые значения, мы создаем следующие тест-кейсы.

Таблица 6. Примерные тест-кейсы.

ID тест-кейса	Условие 1	Условие 2	Ожидаемый результат
Тест-кейс 1	0	3	Выполнить X / Выполнить A
Тест-кейс 2	5	5	Выполнить Y / Выполнить B
Тест-кейс 3	50	7	Выполнить X / Выполнить B
Тест-кейс 4	500	10	Выполнить Z / Выполнить B

Если тестируемая система имеет сложные бизнес-правила, а у бизнес-аналитиков или проектировщиков нет документации этих правил, то тестировщикам следует собрать эту информацию и представить её в виде таблицы решений. Причина проста: представляя поведение системы в такой полной и

компактной форме, тест-кейсы могут быть созданы непосредственно из таблицы решений.

При тестировании для каждого правила создаётся как минимум один тест-кейс. Если состояния этого правила бинарные, то должно быть достаточно одного теста для каждого сочетания. С другой стороны, если состояние является диапазоном значений, то тестирование должно учитывать и нижнюю, и высшую границы диапазона. Таким образом, мы объединяем идею тестирования граничных значений с тестированием таблиц решений.

Для каждого правила создаётся как минимум один тест-кейс.

Чтобы создать тестовую таблицу, просто измените заголовки строк и столбцов.

Таблица 7. Таблица принятия решений преобразуется в таблицу тест-кейсов.

	Тест-кейс 1	Тест-кейс 2	Тест-кейс 3	Тест-кейс 4
Входные значения				
Условие-1	Да	Да	Нет	Нет
Условие-2	Да	Нет	Да	Нет
Ожидаемые результаты				
Действие-1	Выполнить X	Выполнить Y	Выполнить X	Выполнить Z
Действие-2	Выполнить A	Выполнить B	Выполнить B	Выполнить B

Задание 1. Для веб-сайта Регистрационной системы Государственного Университета для добавления нового студента следует ввести имя, адрес и телефонную информацию в верхней части экрана и нажать Enter. Студент будет добавлен в базу данных и система вернет новый СтудентID. Чтобы изменить или удалить студента, введите СтудентID, выберите радио-кнопку "Удалить" или "Изменить" и нажмите Enter.

Предложите таблицу решений: набор правил, условий (например, данные введены?) и действий (например, добавление нового студента)

Задание 2. При назначении материальной помощи в Регистрационной системе Государственного Университета необходимо определить понятие родителя.

(Пример) Шаг 4: кого рассматривать в качестве родителя на этом шаге?

Ответьте на все вопросы четвертого шага, чтобы определить, кого нужно рассматривать в качестве родителя при заполнении формы.

Вы сирота, находитесь или находились (до 18 лет) под опекой? Если да, то пропустите четвертый шаг. Если ваши родители живут вместе и женаты друг на друге, ответьте на вопросы про них. Если один из ваших родителей вдовец или живет один, ответьте на вопросы только об этом родителе. Если ваш овдовевший родитель снова женился и женат до сих пор, ответьте на вопросы об этом родителе и о человеке, на котором женат (замужем) ваш родитель. Если ваши родители разведены или живут раздельно, ответьте на вопросы о том родителе, с которым вы прожили больше за последние 12 месяцев. Если вы не жили с одним из родителей больше чем с другим, ответьте про того родителя, кто оказывал вам большую финансовую поддержку за последние 12 месяцев или в течение последнего года, когда вы получали такую поддержку от одного из родителей. Если этот родитель женился (вышел замуж) на данный момент, ответьте на оставшиеся вопросы этой формы об этом родителе и том человеке, на котором он женат (замужем).

Постройте таблицу решений для тестирования определения понятия родитель.

Попарное тестирование

Программу построения матрицы попарных тестов можно скачать:

<http://www.satisfice.com>

(download allpair)

В Excel следует создать таблицу с вариантами и сохранить с расширением txt (input.txt)

Browser	Client OS	Plug-in	Server	Server OS
IE 5.0	Win 95	None	IIS	Win NT
IE 5.5	Win 98	Real Player	Apache	Win 2000
IE 6.0	Win ME	Media Player	WebLogic	Linux
Netscape 6.0	Win NT			
Netscape 6.1	Win 2000			
Netscape 7.0	Win XP			
Mozilla 1.1				
Opera 7				

Команда запуска:

allpairs input.txt> output.txt

Задание 3.

Определите набор тестов с попарными комбинациями.

1. Банк создал новую систему обработки данных, готовую к тестированию.

У этого банка есть разные клиенты - обычные клиенты, важные клиенты, юр.лица и физ.лица; различные виды счетов - сберегательные, ипотечные кредиты, потребительские кредиты и коммерческие кредиты; плюс отделения банка работают в разных штатах, с разной спецификой проведения фин. операций - Калифорния, Невада, Юта, Айдахо, Аризона и Нью-Мехико.

2. В объектно-ориентированной системе объект класса "A" может передать сообщение, содержащее параметр P, объекту класса "X". Классы "B", "C" и "D" унаследованы от класса "A", поэтому тоже могут послать сообщение. Классы "Q", "R", "S" и "T" унаследованы от "P", поэтому могут быть переданы как параметры. Классы "Y" и "Z" унаследованы от класса "X" и могут получать данные.

Лабораторная работа 5

Сценарное и исследовательское тестирование

Сценарное тестирование предполагает разделение задачи на этапы (подготовка, выполнение, завершение и т.д.) и последовательное выполнение этих этапов.

Особенности этого подхода:

- четкое понимание того, какие функции покрыты тестами;
- уверенность, что все задокументированные тест-кейсы будут пройдены в срок;
- быстрое и легкое подключение нового специалиста на проект благодаря наличию подробных сценариев.

Пример. Тестируется почта, подробное сценарное тестирование может выглядеть так:

- *Перейдите на стартовую страницу*
- *Нажмите на кнопку «Новый»*
- *В поле «Кому» напишите j.cannegieter@squerist.nl*
- *Нажмите на кнопку «Файл»*
- *Выберите документ «Тест1»*
- *В поле «Тема» напишите «Тестовое вложение <дата>»*
- *Откройте почтовый ящик j.cannegieter*
- *Проверьте, доставлено ли сообщение и вложение.*

Для этого же примера сценарий общего тестирования может выглядеть так:

- *Создайте и отправьте письмо с вложением одному получателю*
- *Проверьте, доставлено ли сообщение и вложение*
- *<https://habr.com/ru/companies/jugru/articles/341852/>*

Задание. Сценарное тестирование.

Открыть сайт интернет-магазина (например, Спортмастер). Необходимо описать сценарий тестирования (выбрать три задачи из списка ниже). Например, в

соответствии с пунктом 8: поиск горнолыжной мужской куртки 52 размера, с ограничением стоимости. Оформить отчет.

1. Ориентирование в категориях каталога. Вариант 1
2. Ориентирование в категориях каталога. Вариант 2
3. Фильтрация по заданным параметрам
4. Сброс фильтров
5. Сортировка по заданному параметру
6. Изучение карточки товара в каталоге
7. Изучение страницы с подробным описанием товара
8. Поиск товара по заданным параметрам
9. Сравнение товаров
10. Ориентирование в подборках товаров
11. Добавление товара в избранное
12. Работа с изображением товара
13. Ориентирование в стоимости товара
14. Взаимодействие с чужими отзывами
15. Считываемость доступного кол-ва товара
16. Обращение в поддержку
17. Заказ в один клик
18. Добавление товаров в корзину
19. Удаление товара из корзины
20. Очистка корзины
21. Применение промокодов
22. Поиск ближайшей точки по карте

23. Понимание условий возврата товаров
24. Понимание условий программы лояльности
25. Понимание Push/SMS о статусе заказа
26. Считываемость баннеров
27. Перелистывание карусели
28. Отмена заказа
29. Треккинг заказа
30. Ориентирование в стоимости опций и заказа
31. Считываемость рекомендуемых товаров
32. Поиск заданной точки по карте
33. Оформление доставки
34. Смена города/региона
35. Работа с кликабельными тэгами
36. Оставление фидбека о товаре
37. Взаимодействие с видео о товаре
38. Передача ссылок и шеринг информации о товаре
39. Использование списка избранного
40. Повторение прошлых заказов
41. Редактирование профиля
42. Редактирование состава заказа в корзине
43. Изменений опций заказа
44. Выбор способа оплаты
45. Выбор способа получения товара
46. Считываемость экранов/сообщений об ошибке

- 47. Взаимодействие с всплывающими подсказками
- 48. Регистрация
- 49. Поиск товаров, похожих на просматриваемый
- 50. Считываемость продвигаемых товаров
- 51. Оформление самовывоза
- 52. Свободный поиск товара
- 53. Работа с поисковой строкой
- 54. Оплата заказа
- 55. Поиск товара через каталог

Лабораторная работа 6

Отчеты о тестировании

Отчет о тестировании — документ, включающий в себя результаты работ по тестированию и содержащий информацию, достаточную для соотнесения текущей ситуации с тест-планом и принятия необходимых управленческих решений.

Основные принципы составления отчета о тестировании:

- **Подробность:** необходимо подробно описывать проделанную работу. Не стоит использовать абстрактные слова, формы и выражения.
- **Технический стиль:** используйте технический стиль в отчете.
- **Стандарт:** используйте общепринятый на проекте шаблон отчета.

Типы отчетов

Выделяют два типа отчетов — высокоуровневые и низкоуровневые. Высокоуровневые отчеты отличаются тем, что предоставляют общую оценку о качестве продукта, а низкоуровневые — содержат более детальную информацию о тестировании.

К задачам **высокоуровневых отчетов** относятся:

- сбор, агрегация и предоставление в удобной для восприятия форме объективной информации о результатах работы;
- оценка текущего статуса тестирования и прогресс;
- подсвечивание существующих проблем;
- прогноз развития ситуации и фиксация рекомендаций по устранению дефектов.

К задачам **низкоуровневых отчетов** относятся:

- оценка объема и качества выполненных работ;
- информирование о текущих проблемах;
- сравнение текущего прогресса с планом;
- полная информация о текущем состоянии качества проекта.

Содержание отчета о тестировании

Хоть в каждой компании свои требования, свой формат, свои цели, типичный шаблон сводного отчета о тестировании обычно содержит:

Информацию о проекте — раздел, в котором кратко описывается содержание отчета. Например, название проекта, фичи, тестируемая версия и т.д.

Команда тестировщиков — список специалистов, задействованных в тестировании проекта, с указанием закрепленного за каждым участником тестируемого функционала.

Описание процесса тестирования — в этом разделе кратко описывается то, как происходил процесс тестирования, а именно какие использовались методы, техники, инструменты.

Краткое описание — в данном разделе дается краткое описание того, что было протестировано (версия, номер сборки), что тормозит процесс тестирования (критичные дефекты, недоработанная функциональность, пробелы в требованиях).

Расписание — указываются временные сроки тестирования функционала.

Статистика по дефектам — количество дефектов, обнаруженных за отчетный период, с указанием степени важности, срочности, места обнаружения и прочего.

Список новых дефектов — список дефектов, которые обнаружили в отчетном периоде, с указанием их основных атрибутов.

Список устраненных дефектов — список дефектов, которые были устранены командой разработки в отчетном периоде.

Качество объекта тестирования — экспертное заключение об уровне качества продукта.

Рекомендации — выводы по результатам тестирования и рекомендации по повышению его эффективности в будущем.

Приложения — графики, таблицы, диаграммы, демонстрирующие числовые характеристики процесса тестирования.

Пример отчета о тестировании сайта



Оглавление

Краткое описание.....	3
Сведения о ходе испытаний.....	4
Выводы по результатам испытаний.....	9
Приложение.....	10

Краткое описание

В ходе тестирования был применен метод 50 тест - кейсов, который проверяет работу сайта apple.com/ru/

Суть тестирования заключается в проверке работы всех обязательных функций сайта и его корректное отображение во всех актуальных браузерах.

Сведения о ходе испытаний

Таблица 1. Тест-кейсы, пройденные в ходе тестирования сайта <https://www.apple.com/ru/>

№ Тест-кейса	Название проверки	Результат	Замечания и рекомендации
Кроссбраузерное тестирование			
1	Приложение корректно отображается и функционирует во всех актуальных браузерах (IE, Edge, Chrome, Firefox, Safari,...).	Отрицательно	Отсутствует эффект скроллинга в браузере IE
2	Приложение корректно отображается и функционирует во всех актуальных версиях браузеров.	Отрицательно	Отсутствует эффект скроллинга в браузере IE
3	Корректная обработка тайм-аутов	Положительно	
4	Работоспособность при удалении cookies во время использования приложения.	Положительно	
5	Работоспособность при удалении cookies после использования приложения.	Положительно	
Работа с формами			
6	Обязательные поля должны быть заполнены до подтверждения ввода данных.	Положительно	
7	Обязательные поля должны быть явно обозначены.	Положительно	
8	Если пользователь нажал кнопку подтверждения ввода данных с незаполненными обязательными полями, эти поля должны быть подсвечены и отображено сообщение об ошибке.	Положительно	

№ Тест-кейса	Название проверки	Результат	Замечания и рекомендации
9	Сообщение об ошибке не отображается, если не заполнены необязательные поля.	Положительно	
10	Текстовые поля допускают ввод специальных символов.	Положительно	
11	При вводе количества знаков меньше минимального отображается сообщение об ошибке.	Положительно	
12	При вводе количества знаков больше максимального отображается сообщение об ошибке.	Положительно	
13	Если обязательное поле заполнено не полностью, отображается сообщение об ошибке.	Положительно	
14	Сообщение об ошибке отображается при попытке ввести недопустимые символы (например, ввод текста в поле, принимающее только числа).	Положительно	
Авторизация и работа с профилем			
15	Создайте новый аккаунт.	Положительно	
16	Авторизуйтесь с существующим аккаунтом.	Положительно	
17	Восстановите/сбросьте забытый пароль.	Положительно	
18	При вводе неверного пароля отображается сообщение об ошибке и ссылка на страницу восстановления пароля.	Положительно	
19	Отредактируйте профиль пользователя.	Положительно	

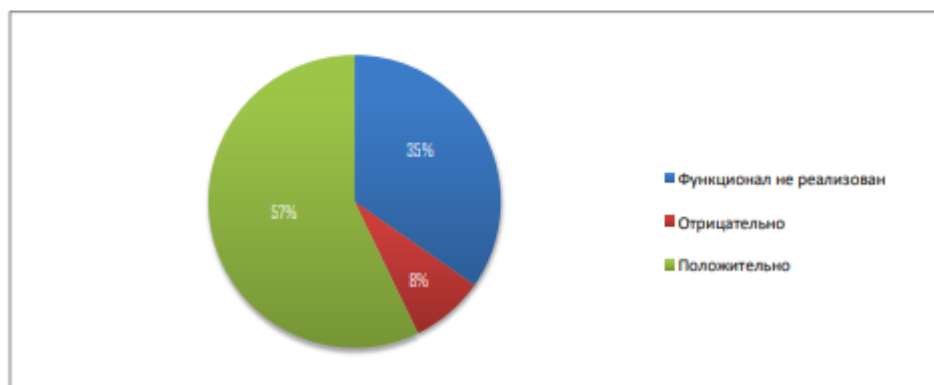
№ Тест-кейса	Название проверки	Результат	Замечания и рекомендации
20	Зарегистрируйте нового пользователя, используя аккаунт социальных сетей.	Функционал не реализован	
21	Авторизуйтесь, используя аккаунт социальных сетей.	Функционал не реализован	
22	Загрузите в профиль аватар/картинку.	Функционал не реализован	
23	Отредактируйте аватар/картинку.	Функционал не реализован	
24	Удалите аватар/картинку.	Функционал не реализован	
25	При регистрации нового пользователя ему приходит подтверждение по электронной почте.	Положительно	
26	При оформлении подписки на уведомления пользователю приходит подтверждение по электронной почте.	Отрицательно	
27	При смене пароля пользователю приходит подтверждение по электронной почте	Положительно	
Навигация			
28	При выполнении действий (например, выход из аккаунта) и манипуляциях с данными (например, удаление файла) отображается сообщение с подтверждением.	Отрицательно	

№ Тест-кейса	Название проверки	Результат	Замечания и рекомендации
29	Корректность работы сортировки.	Функционал не реализован	
30	Корректность работы фильтрации.	Положительно	
31	Корректность работы кнопок и ссылок в меню навигации.	Положительно	
32	Отображение страницы с ошибкой 404, если пользователь указал неверный путь к файлу или URL.	Положительно	
Числа и даты			
33	Осуществляется проверка на корректность дат	Положительно	
34	Корректная обработка високосного года, отсутствует сообщение об ошибке	Положительно	
35	Числовые поля не допускают ввода текста, отображается сообщение об ошибке.	Положительно	
36	При попытке ввода отрицательных чисел, отображается сообщение об ошибке, если предусмотрен ввод строго положительных чисел.	Положительно	
37	При попытке деления на ноль отображается сообщение об ошибке.	Функционал не реализован	
38	Поля специальных форматов (например, процентные, денежные) отображаются корректно.	Положительно	
39	Всплывающие уведомления об ошибке и предупреждения работают корректно.	Положительно	

№ Тест-кейса	Название проверки	Результат	Замечания и рекомендации
Интеграция с социальными сетями			
40	Корректное отображение кнопок социальных сетей.	Функционал не реализован	
41	При нажатии на кнопку "поделиться" отображается всплывающее окно, в котором доступна функция редактирования сообщения перед отправкой.	Функционал не реализован	
42	Корректное отображение окна авторизации с помощью аккаунта социальных сетей.	Функционал не реализован	
43	Корректное отображение аватара пользователя из профиля социальных сетей.	Функционал не реализован	
44	Пользователь может оставлять комментарии на вашем сайте, используя аккаунт социальных сетей.	Функционал не реализован	
Загрузка и скачивание файлов			
45	Файлы успешно загружаются.	Функционал не реализован	
46	Слишком тяжелые файлы не загружаются, и отображается сообщение об ошибке.	Функционал не реализован	
48	Файлы ошибочного формата не загружаются, и отображается сообщение об ошибке.	Функционал не реализован	

№ Тест-кейса	Название проверки	Результат	Замечания и рекомендации
49	Файлы недопустимого формата не загружаются, и отображается сообщение об ошибке.	Функционал не реализован	
50	Файлы успешно скачиваются.	Функционал не реализован	

Рисунок 1. Диаграмма результатов выполнения тест - кейсов.



Выводы по результатам испытаний

В данном разделе представлены выводы по результатам испытаний, а также общая статистика по дефектам, которые были выявлены в результате тестирования сайта <https://www.apple.com/ru/>. В период испытаний был выявлено 4 дефекта. Список дефектов приведен в Приложении.

Приложение

Таблица 2. Список дефектов

Порядковый №	№ Кейса	Краткое описание	Состояние	Критичность
1	1	Отсутствует эффект скроллинга в браузере IE	Обнаружена	Низкая
2	2	Отсутствует эффект скроллинга в браузере IE	Обнаружена	Низкая
3	26	Письмо с подтверждением не приходит	Обнаружена	Высокая
4	28	Пользователь разлогинивается без подтверждения действия	Обнаружена	Низкая

Задание

По некоторому сайту (например, <https://www.orange-elephant.ru/>, <https://flowwow.com/krasnodar/>, <https://sadvod-opt.com/> и др.) составить отчет о тестировании.

Лабораторная работа 7

Тестирование форм (полей ввода)

Используйте классы эквивалентности, граничные значения для тестирования форм. Для тестирования форм есть плагин для Chrome и Firefox — **BugMagnet**. После установки кликните правой кнопкой мыши по любому полю формы и появится возможность выбора значений для заполнения из огромного списка данных.

Таблица верификации полей

Тип данных	Корректный ввод	Некорректный ввод
Положительные целые числа	<ul style="list-style-type: none">— только числа— максимально возможное значение (N)— числа внутри промежутка $(N + 1) / 2$	<ul style="list-style-type: none">— число больше максимально возможного (N + 1)— дробные числа— отрицательные числа— строковые значения— числа + строковые значения— числа + специальные символы— Unicode (например U+0000, U+0001)
Строки	<ul style="list-style-type: none">— только символы— только числа— только специальные символы— числа + символы— числа + спецсимволы— символы + спецсимволы	—
Даты	<ul style="list-style-type: none">— проверить, что при выборе появляется datepicker— проверить, что поле нельзя изменить с клавиатуры— проверить, что значение поля можно скопировать, но вставить нельзя— проверить, что при выборе значения в datepicker, оно появляется в поле— проверить, что в феврале 29 дней в високосных годах— проверить, что в феврале 28 дней не в високосных годах	—

Чек-лист для веб-форм

Ниже приведены основные элементы чек-листа.

1. Расположение формы на экране до заполнения, после неправильного заполнения, после правильного заполнения.
2. Сохранение/изменение URL при отправке данных из формы.
3. Отправка формы с пустыми обязательными полями.
4. Отправка формы с некорректно заполненными полями (числа вне допустимого диапазона, длина строк превышает допустимую и т.д.)
5. Подача формы с «нелогичными данными» (например, дата рождения в будущем).
6. Отправка формы с полями, содержащими спецсимволы.
7. Восстановление значений полей после отправки формы с некорректно заполненными полями.
8. Информативные сообщения об ошибках.
9. Функциональная группировка полей формы. Информативные этикетки, подсказки.
10. Эффективность и удобство использования формы с отключенным JavaScript.
11. + использовать контрольные списки для каждого стандартного поля (текстовое, числовое, дата и т. д.)

Пример чек-листа для тестирования веб-форм.

Extra High <ul style="list-style-type: none"><input type="checkbox"/> Сохранение формы.<input type="checkbox"/> Форма сохраняется в веб-формах (админ-панели) или SQL-таблицах.<input type="checkbox"/> Изменение адреса отправки.<input type="checkbox"/> E-mail, на который приходят данные из веб-формы, можно менять в административной панели.	Extra high <ul style="list-style-type: none"><input type="checkbox"/> Save the form.<input type="checkbox"/> The form is saved in web forms (admin panels) or SQL tables.<input type="checkbox"/> Change the shipping address.<input type="checkbox"/> E-mail, which receives data from the web form, can be changed in the administrative panel. High <ul style="list-style-type: none"><input type="checkbox"/> Relevance of the shipping address.<input type="checkbox"/> A real e-mail address of the person
---	--

<p>High</p> <ul style="list-style-type: none"> □ Актуальность адреса отправки. □ Прописан реальный e-mail лица, отвечающего за обработку заявок. □ Отправка формы. □ Данные из заполненной формы отправляются администратору на e-mail. □ Отправка уведомления пользователю. □ Опционально. Пользователь получает уведомление на свой e-mail об успешно полученной заявке и последующих действиях, которые от него требуются. <p>Навигация</p> <ul style="list-style-type: none"> □ Предусмотрены плейсхолдеры (placeholder) для полей. □ Если названия полей не подписаны, то внутри полей выводится подсказка, которая исчезает при внесении текста. □ Прописан атрибут autocomplete для полей, поддерживающих это значение. <p>Атрибут autocomplete подставляет ранее введенные пользователем данные в поле, если функция не отключена в браузере.</p> <ul style="list-style-type: none"> □ Правильная работа многошаговых форм. □ Навигация рядом с формой показывает текущий этап и количество оставшихся шагов. □ Многошаговые формы корректно работают при навигации посредством кнопок «Вперед» и «Назад» в браузере. <p>Валидация</p>	<p>responsible for processing applications has been registered.</p> <ul style="list-style-type: none"> □ Submit the form. □ Data from the completed form is sent to the administrator by e-mail. □ Send notification to user. □ Optional. The user receives a notification to his e-mail about the successful receipt of the application and the subsequent actions that are required of him. <p>Navigation</p> <ul style="list-style-type: none"> □ Placeholders are provided for the fields. □ If the field names are not signed, a hint is displayed inside the fields, which disappears when the text is entered. □ The autocomplete attribute is registered for fields supporting this value. <p>The autocomplete attribute substitutes previously entered user data in the field, if the function is not disabled in the browser.</p> <ul style="list-style-type: none"> □ Correct work of multi-step forms. □ Navigation next to the form shows the current stage and the number of remaining steps. □ Multi-step forms work correctly when navigating using the "Forward" and "Back" buttons in the browser. <p>Validation</p> <ul style="list-style-type: none"> □ For numeric values from a certain range, there are limiters for the minimum and maximum number of characters. □ Check for dates, times, and other similar characteristics. □ For fields that require files to be loaded, the accept attribute is specified, which determines the type of documents to be loaded. □ For fields whose validation passes through a regular expression, the pattern
---	---

<ul style="list-style-type: none"> □ Для числовых значений из определенного диапазона прописаны ограничители минимального и максимального количества символов. □ Проверьте это для дат, времени и прочих подобных характеристик. □ Для полей, предполагающих загрузку файлов, прописан атрибут assert, определяющий тип загружаемых документов. □ Для полей, валидация которых проходит через регулярное выражение, прописан атрибут pattern. □ Требуемый формат данных, которые должен ввести пользователь, очевиден для него. □ Доступна инструкция по формату вводимых данных на человеческом языке. □ Пользователь не видит регулярного выражения как подсказки к действию. □ Сообщения об ошибках понятны обычным пользователям и логичны. <p>Прочее</p> <ul style="list-style-type: none"> □ Форма запрашивает у пользователя только необходимые данные. <p>Откройте форму, визуально убедитесь, что требуется внести только необходимый минимум информации.</p> <ul style="list-style-type: none"> □ Если все поля обязательны для заполнения, рядом с их названиями не выводятся звездочки — символ *. <p>Откройте форму и убедитесь в этом визуально. Желательно наличие поясняющего текста об обязательном заполнении всех полей.</p>	<p>attribute is specified.</p> <ul style="list-style-type: none"> □ The required data format that the user must enter is obvious to him. □ Available instructions on the format of the input data in the human language. □ The user doesn't see the regular expression as a hint to the action. □ Error messages are understandable to ordinary users and logical. <p>Other</p> <ul style="list-style-type: none"> □ The form prompts the user for only the necessary data. <p>Open the form, visually make sure that you only need to enter the minimum information.</p> <ul style="list-style-type: none"> □ If all fields are required, asterisks are not displayed next to their names - the * symbol. <p>Open the form and see it visually. It is desirable to have an explanatory text on the mandatory filling of all fields.</p> <ul style="list-style-type: none"> □ For an authorized user, all known data about the visitor is automatically inserted in the form fields. <p>Make sure visually that the information specified by the user in the profile is automatically displayed in the fields of the forms requesting this data.</p> <ul style="list-style-type: none"> □ The text multiline field changes the height when entering a voluminous message, or a scrollbar appears on the right side to view the entire contents. <p>Open a form with a multiline text field, enter as many characters as possible into it.</p> <ul style="list-style-type: none"> □ A valid TYPE attribute is registered in the form fields, informing the browser of the form element type. □ The types of dates, time, phone numbers, ranges, url, e-mail, numbers are correct.
--	--

<p>□ Для авторизованного пользователя в поля формы автоматически подставляются все известные о посетителе данные.</p> <p>Убедитесь визуально, что указанная пользователем в профиле информация автоматически выводится в полях форм, запрашивающих эти данные.</p> <p>□ Текстовое многострочное поле при вводе объемного сообщения изменяет высоту либо в правой части появляется скроллбар для просмотра всего содержимого.</p> <p>Откройте форму с текстовым многострочным полем, введите в него максимально большое количество символов.</p> <p>□ В полях формы прописан корректный атрибут TYPE, сообщающий браузеру тип элементов формы.</p> <p>□ Правильно указаны типы дат, времени, телефонов, диапазонов, url, e-mail, чисел.</p> <p>□ Во время отправки формы на медленном канале пользователь не может менять в ней данные.</p> <p>Важно. Действительно для аjax-форм.</p> <p>Low</p> <p>□ Вывод подсказок и ошибок сделан с анимационным эффектом. Опционально и зависит от проекта.</p> <p>□ Кнопка отправки данных неактивна, пока не активирован чекбокс «Согласиться с правилами», «Пользовательское соглашение».</p> <p>□ Кнопка отправки данных неактивна, пока все введенные данные не прошли положительную валидацию.</p>	<p>□ While the form is submitted on a slow channel, the user cannot change the data in it.</p> <p>Important. Valid for AJAX forms.</p> <p>Low</p> <p>□ Display hints and errors made with animation effect. Optional and depends on the project.</p> <p>□ Send data button is inactive until the "Accept rules", "User agreement" checkbox is not activated.</p> <p>□ Send data button is inactive until all entered data have passed a positive validation.</p> <p>Open the form with fields for input, enter incorrect data, check whether the button is active.</p> <p>□ If the data has not passed a positive validation, when you hover the cursor on the button to send data, an informational message is displayed.</p> <p>Open the form, enter incorrect data, hover the cursor on the button to send data, check whether the message is displayed.</p>
---	---

<p>Откройте форму с полями для ввода, введите некорректные данные, проверьте, активна ли кнопка.</p> <p>□ Если данные не прошли положительную валидацию, при наведении курсора на кнопку для отправки данных выводится информационное сообщение.</p> <p>Откройте форму, введите некорректные данные, наведите курсор на кнопку отправки данных, проверьте, выводится ли сообщение.</p>	
---	--

Задание 1.

Протестируйте поля форм, подготовьте отчет о тестировании.

Задание 2.

Протестируйте веб-формы, предварительно разработав чек-лист.

Подготовьте заполненные чек-листы и отчет по тестированию.

Лабораторная работа 8

Тестирование строки поиска

Что надо проверить, тестируя поиск:

1. Поиск ищет по всем полям, указанным в ТЗ
2. Поиск НЕ ищет по тем полям, которые НЕ указаны в ТЗ
3. Релевантность выдачи — то, что я ищу, в начале списка, или в конце?
4. Учитывается ли контекст поиска — ищу я по всему сайту или только разделу игрушек
5. Регистронезависимость поиска — найдет ли «Платье», если я ввела «платье»?
6. Ищет ли по включению или полному соответствию — «ту» найдет мне «туфли»?
7. Найдет ли 2 слова из одного поля? В любом порядке введенные?
8. Найдет ли 2 слова из разных полей?
9. Ошибка в вводе (исправляются ли опечатки, ищет ли похожее)
10. Исправляет ли система неправильную раскладку?
11. Ищет ли на разных языках? А если сразу на двух попробовать?
12. Поиск со спецсимволами работает?
13. А с эмоджи? Не упадет система при вводе какашки?
14. Тримаются ли открывающие и закрывающие пробелы
15. Пустое поле / только пробелы в поле
16. Нижнюю границу (от скольких символов ищет)
17. Верхнюю произвольную границу — указанную в ТЗ
18. Верхнюю границу на выходе
19. Поиск технологической границы — ввести «войну и мир», 100 млн символов.

Рассмотрим отдельно каждый пункт и выясним, как и зачем его проверять.

1. Поиск ищет по всем полям, указанным в ТЗ

Сначала надо проверить основное — то, что поиск вообще работает. Что он ищет по всем тем полям, по которым должен.

В любом чек-листе надо думать о приоритетах. Сначала — самое важное. Как в чек-листе в целом, так и внутри каждого блока проверок, постепенно идем от важного к неважному. От позитива к негативу.

А что самое важное в поиске? Для этого думаем, зачем его вообще делают. Чтобы искать. По чему? По каким-то полям / признакам. По каким? Узнаем и проверяем, ищет он по ним или нет.

2. Поиск НЕ ищет по тем полям, которые НЕ указаны в ТЗ

Необходимо убедиться, что поиск НЕ ищет там, где не должен.

3. Релевантность выдачи

Здесь важно проверить приоритет поисковой выдачи. Понятно, что туда может попасть что-то «не то». Чем больше у нас данных и чем больше полей, по которым поиск возможен, тем больше вариантов на каждый запрос.

4. Контекст поиска

Откуда я вызывается поиск? С главной страницы сайта или из конкретного раздела?

В интернет-магазинах обычно куча разных товаров, поэтому контекст поиска очень важен, чтобы выдавать релевантные товары.

А ещё контекст очень важен в мессенджерах. Искать вообще везде, во всех 100500 чатах, или только в одном?

.

5. Регистронезависимость поиска

Обычно поиск регистронезависимый, и это логично. Так что проверяем:

1. нижний регистр
2. ВЕРХНИЙ РЕГИСТР
3. ЗоЛоТуЮ СеРеДиНу

6. Ищет ли по включению или полному соответствию

Поиск по включению — это когда можно ввести только часть слова. Например, «ко» вместо «котик». Это очень удобно во всяких чатах. Если поиск работает по полному совпадению, то нужно перебирать все падежи. Если он работает по включению, мне достаточно написать только часть слова.

Необходимо уточнить — как должно работать? А потом проверить:

- Поиск по полному соответствию в одном слове
- Поиск по частичному соответствию в одном слове — совпадение в начале слова / середине / конце (вспоминаем про классы эквивалентности и граничные значения)

7. Два слова из одного поля

А что будет, если у нас не одно слово, а два или более? Причем 2 слова у нас может быть:

1. В искомом поле
2. В поисковом запросе

И это будут разные тесты! Например, название товара: «Игровой набор».

Тесты при этом:

1. Игровой → то есть в поле несколько слов, а мы ввели одно из них, найдётся?
2. Игровой набор → в поиске тоже несколько слов

Но что будет, если в поиске мы изменим порядок слов?

- Набор игровой → найдется ли он в таком формате? Или нужно прям четкое совпадение?

8. Два слова из разных полей

Если мы проверяем несколько слов в поиске, то тут тоже возможны варианты:

1. Они все из разных полей — например, мы ищем по цвету + названию + полу: «красная майка женская».
2. Они из одного поля — как выше с игровым набором. И в этом случае намного интереснее изменить порядок слов и проверить поведение системы =)

Важно понимать, что это разные тесты. И стоит проверить и тот вариант, и другой.

9. Опечатки

Как система работает с опечатками? Найдет ли похожее слово?

Кранный галстук → Красный галстук

Если система работает с опечатками, то как:

- 1 неправильную букву исправит / 2 и более
- 1 пропущенную букву исправит / 2 и более

Это, конечно, зависит от длины искомого слова, но тогда исправляются ли опечатки в коротких словах?

10. Неправильная раскладка

Типичный пример опечатки — пользователь забыл изменить раскладку на клавиатуре и напечатал английскими символами русский текст. Ищем «котик» но вводим «`tjnbr`».

Если система русскоязычная и пользователь ввел данные на английском, по которым ничего не ищется, надо проверить — не забыл ли он переключить раскладку? Если по переключенной раскладке поиск работает, показываем эти результаты.

11. Другой язык

Что, если в системе есть данные и на русском, и на английском? Или даже смешанный вариант: «Сухой корм Purina ONE». Найдет ли система и по русскому алфавиту, и по английскому?

12. Спецсимволы

Это стандартный подход пуля для всех текстовых полей:

- Русский алфавит
- Английский
- Спецсимволы
- Эмоджи
- Перемешал

И поэтому приоритет у таких проверок неважный.

1. Спецсимвол есть в искомом поле.

Поэтому проверить нужно все спецсимволы. Можно: создать товар / искомый объект сразу с набором спецсимволов:

~!@#%&*()_+{|:’>?<Ё!”№;%;?*()_+/Ъ,/,.;’[]\|

2. Пользователь может забыть переключить раскладку и вот уже вместо «Хлеб» он вводит «{kt,»

3. Пользователь может случайно вставить в поле какой-то текст со спецсимволами. Эти ошибки тоже надо предусмотреть.

13. Эмоджи

Эмоджи это не совсем спецсимвол. Система может не искать по спецсимволам, но при этом обрабатывать эмоджи. Следовательно, классы эквивалентности разные.

14. Тримаются ли открывающие и закрывающие пробелы

Метод trim() удаляет пробельные символы с начала и конца строки.

То есть вводим « тест », а система преобразует строку в «тест» и потом уже передаёт дальше, в нашем случае — функции поиска.

Проверка на трим более логична при заполнении и сохранении полей — введено случайно « Ольга» в имя, а потом поиск не находит, требуя точного совпадения. И тогда логично системе перед сохранением сделать трим.

15. Пустое поле

Фактически это проверка на ноль. А ноль — это отдельный класс эквивалентности, который часто приносит баги, поэтому его надо проверять.

Если мы оставили поисковую строку пустой, то есть варианты:

- Система выводит всю базу
- Система выводит пустоту
- Кнопка поиска просто не срабатывает / вообще заблокирована

до ввода символов (что сомнительно, впрочем, да и зачем?)

16. Пробелы в поле

Ещё один вариант тестирования нуля — ввести в поле ТОЛЬКО пробелы. Как система их обрабатывает? В данном тесте мы просто собираем информацию о том, как работает система. Потому что почти любое поведение (разве что кроме ошибки) можно считать нормальным.

17. Нижняя граница

Пустое поле (ноль) — мы уже проверили. А дальше думаем, какая у наших данных будет нижняя граница?

Павел Абдюшев приводит замечательный пример с писателем и поэтом Эдгаром ПО. Это пример короткой реальной фамилии, по которой могут искать. Попробуем найти там книгу «Ворон».

Сначала пробуем по фамилии:

ПО

Если книги автора найти не получается — то проблемы с приоритетами в выборке. Поиск по ФИО автора должен быть более релевантным, чем совпадение какого-то слова в описании.

18. Верхняя произвольная граница

Есть ли ограничение в строке поиска? Если есть, то оно обычно в разумных пределах — 100 / 500 / 1000 символов. И если пользователь вводит больше, то значение обрезается до максимального.

Иногда верхнюю границу просто не ставят. И это тоже нормально, это не баг, который надо срочно исправить. Просто верхней произвольной границы у нас не будет.

19. Верхняя граница на выходе

Прошлый пункт — о том, как подать много на вход. А что будет, если «много» не на входе, а на выходе? Или «где-то посередине», то есть там, где поиск идёт?

- Вернулось много данных по поиску (распространенный запрос)
- Много данных находится в самой системе / базе данных

Сколько времени займет поиск? И пройдет ли он вообще? Может на поиск установлен тайм-аут в 1 минуту. И при плохом интернете / большом объеме данных будут проблемы.

20. Поиск технологической границы

Для поиска технологической границы мы вбиваем в строку поиска **ОЧЕНЬ БОЛЬШОЕ** значение. 1000 символов — не поиск технологической границы. Введите 100 миллионов символов или главу «Войны и мира».

Задание.

Выполнить в строке поиска некоторого интернет-магазина набор тестов. Оформить отчет.