**Задание 1. Классы эквивалентности**

Представь себе ресурс по выдаче кредитов. На ресурсе есть поле ввода логина. По техническому заданию поле можно заполнять любыми символами (минимальное и максимальное количество символов неважно). Требуется протестировать данное поле. Что нужно сделать:

1. Определи классы эквивалентности.
2. Укажи, какими символами от каждого класса будешь тестировать.

**Тест-дизайн** — процесс проектирования тестовой документации и того покрытия, которого хотим добиться тестовыми сценариями.

**Класс эквивалентности** — одно или несколько значений ввода, к которым программное обеспечение применяет одинаковую логику.

**1. Определение классов эквивалентности**

| **Класс эквивалентности** | **Примеры символов** | **Пояснение** |
| --- | --- | --- |
| **Латиница (верхний регистр)** | A, B, C, ..., Z | Буквы английского алфавита (A-Z). |
| **Латиница (нижний регистр)** | a, b, c, ..., z | Буквы английского алфавита (a-z). |
| **Кириллица (верхний регистр)** | А, Б, В, ..., Я | Буквы русского алфавита (А-Я). |
| **Кириллица (нижний регистр)** | а, б, в, ..., я | Буквы русского алфавита (а-я). |
| **Цифры** | 0, 1, 2, ..., 9 | Арабские цифры. |
| **Специальные символы** | `! @ # $ % ^ & \* ( ) \_ + - = { } [ ] \ | ` | Знаки препинания и спецсимволы. |
| **Пробелы** | (пробел), (двойной пробел) | Проверка на обработку пробелов. |  |
| **Эмодзи/юникод** | 😊, ♠, ✓, 你 | Нестандартные символы (Unicode). |  |
| **HTML-теги** | <script>alert(1)</script>, <b> | Проверка на инъекции. |  |
| **Пустая строка** | "" (пустое поле) | Отправка формы без ввода. |  |

**2. Тестовые данные**

Для проверки каждого класса можно использовать:

* **Одиночные символы**: A, я, 1, @, (пробел), 😊, <b>, "".
* **Комбинации символов**:
  + User123! (латиница + цифры + спецсимволы).
  + Пользователь\_2024 (кириллица + нижнее подчеркивание + цифры).
  + <script>alert(1)</script> (инъекция).

**3. Дополнительные проверки**

Хотя по ТЗ нет ограничений, стоит проверить:

1. **Обработку крайних случаев**:
   * Очень длинный логин (1000+ символов).
   * Логин из одних пробелов (" ").
2. **Визуальное отображение**:
   * Как поле отображает редкие символы (например, эмодзи).
3. **Совместимость с БД**:
   * Сохраняется ли логин с кириллицей/спецсимволами в базе данных.

**Задание 2. Граничные значения**

После успешного тестирования поля ввода логина необходимо проверить онлайн-калькулятор. Тебе предстоит проверить расчет процентной ставки по кредиту.

Условия:

* Для выставления нужной суммы пользователь использует ползунок шкалы величины займа (чёрный кружок).
* При займе суммы до 10 000 руб. процентная ставка = 20%.
* При займе от 10 000 до 50 000 руб. процентная ставка = 15%.
* При займе от 50 000 руб. процентная ставка = 10%.
* Максимальная сумма займа = 100 000 руб.
* Минимальный шаг шкалы калькулятора = 1 000 руб.
* Границу «0» (ноль) не тестируем.

Задание:

1. Определи граничные значения для данной функциональности.
2. Укажи, какими значениями будет протестирован калькулятор.

**1. Определение граничных значений**

Для каждой зоны ставок проверяем:

1. **Нижнюю границу диапазона** (минимальное значение, при котором ставка меняется).
2. **Верхнюю границу диапазона** (максимальное значение перед изменением ставки).
3. **Значения внутри диапазона** (стандартный случай).

| **Диапазон (руб.)** | **Ставка** | **Граничные значения** | **Тестовые значения** |
| --- | --- | --- | --- |
| **1–9 999** | 20% | Нижняя: 1 000 (мин. шаг) | 1 000, 5 000, 9 999 |
|  |  | Верхняя: 9 999 |  |
| **10 000–49 999** | 15% | Нижняя: 10 000 | 10 000, 30 000, 49 999 |
|  |  | Верхняя: 49 999 |  |
| **50 000–100 000** | 10% | Нижняя: 50 000 | 50 000, 75 000, 100 000 |
|  |  | Верхняя: 100 000 (макс. сумма) |  |

**2. Тестовые значения**

**Критические точки:**

* 9 999 → должна быть ставка **20%** (последнее значение перед переходом на 15%).
* 10 000 → ставка **15%** (первое значение нового диапазона).
* 49 999 → ставка **15%** (последнее значение перед переходом на 10%).
* 50 000 → ставка **10%** (начало третьего диапазона).
* 100 000 → ставка **10%** (максимально возможная сумма).

**Дополнительные проверки:**

* Шаг ползунка: убедиться, что можно выбрать только кратные 1 000 (например, 15 500 — недопустимо).
* Визуальное отображение: корректно ли отображается ставка при перемещении ползунка.

**3. Пример тест-кейсов**

1. **Сумма:**9 999**руб.**
   * **Ожидаемый результат:** Ставка = **20%**.
2. **Сумма:**10 000**руб.**
   * **Ожидаемый результат:** Ставка = **15%**.
3. **Сумма:**49 999**руб.**
   * **Ожидаемый результат:** Ставка = **15%**.
4. **Сумма:**50 000**руб.**
   * **Ожидаемый результат:** Ставка = **10%**.
5. **Сумма:**100 000**руб.**
   * **Ожидаемый результат:** Ставка = **10%**, ползунок не двигается дальше.

### **Задание 3. Таблица переходов и состояний**

1. Составь таблицу переходов и состояний для страницы авторизации <https://applicant.21-school.ru/auth>.

### **1. Определение состояний системы**

1. **Начальное состояние (S0):** Пустая форма авторизации (поля логина и пароля незаполнены).
2. **S1:** Поле логина заполнено, пароль пуст.
3. **S2:** Поле пароля заполнено, логин пуст.
4. **S3:** Оба поля заполнены.
5. **S4:** Успешная авторизация (редирект в ЛК).
6. **S5:** Ошибка валидации (неверные данные, пустые поля и т. д.).

### **2. Возможные события (действия пользователя)**

* **E1:** Ввод логина.
* **E2:** Ввод пароля.
* **E3:** Нажатие кнопки **«Войти»**.
* **E4:** Нажатие кнопки **«Забыли пароль?»**.
* **E5:** Нажатие кнопки **«Регистрация»**.

### **3. Таблица переходов**

| **Текущее состояние** | **Событие** | **Действие системы** | **Новое состояние** |
| --- | --- | --- | --- |
| S0 | E1 | Поле логина заполнено | S1 |
| S0 | E2 | Поле пароля заполнено | S2 |
| S0 | E3 | Показать ошибку: «Заполните логин и пароль» | S5 |
| S0 | E4 | Переход на страницу восстановления пароля | — (новый экран) |
| S0 | E5 | Переход на страницу регистрации | — (новый экран) |
| S1 | E2 | Поле пароля заполнено | S3 |
| S1 | E3 | Показать ошибку: «Введите пароль» | S5 |
| S2 | E1 | Поле логина заполнено | S3 |
| S2 | E3 | Показать ошибку: «Введите логин» | S5 |
| S3 | E3 | Проверка данных → успешная авторизация | S4 |
| S3 | E3 | Проверка данных → ошибка (неверный логин/пароль) | S5 |
| S5 | E1/E2/E3 | Повторная попытка ввода | S1/S2/S3 |

### **Задание 4**

Ты уже протестировал процентную ставку, осталось понять, кому выдаём кредит. Для этого просим пользователя заполнить анкету (в скобках указаны возможные варианты выбора значений):

1. Возраст (18–35/36–55/56–75).
2. Наличие любых задолженностей (да/нет).
3. Тип трудовых отношений (ИП / Самозанятый / Бессрочный договор).
4. Наличие действующих кредитов (да/нет).

Что нужно сделать:

1. Подсчитай количество всех возможных комбинаций.
2. Составь комбинации, используя технику уникальных пар.

### **Количество возможных комбинаций и уникальные пары**

#### **1. Подсчет всех возможных комбинаций**

У нас есть 4 параметра с разным количеством вариантов:

1. **Возраст**: 3 варианта (18–35, 36–55, 56–75)
2. **Наличие задолженностей**: 2 варианта (да, нет)
3. **Тип трудовых отношений**: 3 варианта (ИП, Самозанятый, Бессрочный договор)
4. **Наличие действующих кредитов**: 2 варианта (да, нет)

**Общее количество комбинаций** = 3 × 2 × 3 × 2 = **36**.

#### **2. Техника уникальных пар (Pairwise Testing)**

Чтобы сократить количество тестовых случаев, используем технику **попарного тестирования (Pairwise)**. Это значит, что каждое значение каждого параметра должно хотя бы один раз сочетаться с каждым значением другого параметра.

**Используем инструмент** (например, [AllPairs](http://www.satisfice.com/tools/pairs.zip" \t "_blank) или онлайн-генераторы) для генерации оптимального набора комбинаций.

#### **3. Сгенерированные комбинации (уникальные пары)**

| **№** | **Возраст** | **Задолженности** | **Трудовые отношения** | **Действующие кредиты** |
| --- | --- | --- | --- | --- |
| 1 | 18–35 | да | ИП | да |
| 2 | 18–35 | нет | Самозанятый | нет |
| 3 | 18–35 | да | Бессрочный договор | нет |
| 4 | 36–55 | да | Самозанятый | да |
| 5 | 36–55 | нет | Бессрочный договор | да |
| 6 | 36–55 | да | ИП | нет |
| 7 | 56–75 | да | Бессрочный договор | да |
| 8 | 56–75 | нет | ИП | нет |
| 9 | 56–75 | нет | Самозанятый | да |
| 10 | 18–35 | нет | ИП | да |
| 11 | 36–55 | нет | Самозанятый | нет |
| 12 | 56–75 | да | Самозанятый | нет |

**Итого**: 12 комбинаций вместо 36 (экономия 66% тестов!).

#### **4. Проверка покрытия пар**

Убедимся, что все пары значений покрыты (например):

* **Возраст × Задолженности**:
  + 18–35 + да
  + 18–35 + нет
  + 36–55 + да
  + 36–55 + нет
  + 56–75 + да
  + 56–75 + нет
* **Возраст × Трудовые отношения**:
  + 18–35 + ИП
  + 18–35 + Самозанятый
  + ... и т. д.

Все комбинации **уникальны** и покрывают все возможные пары.

### **Вывод**

* Полный перебор: **36 комбинаций**.
* Оптимизированный набор (Pairwise): **12 комбинаций**.
* Эффективность: **покрытие 100% парных взаимодействий** при минимуме тестов.

**Рекомендация**: Использовать pairwise-подход для тестирования анкеты, так как он значительно сокращает время тестирования без потери качества.

### **Тестовые туры**

Тестовые туры — один из видов исследовательского тестирования. Типичная проблема при тестировании — это то, что тестировщик, проверяющий функциональность, один, а пользователей много. Поведение тестировщика при проверке зачастую одинаково, в то время как поведение пользователей отличается не только от тестировщика, но и друг от друга (пользователь от пользователя). Задачей тестовых туров как раз является изменить привычное поведение тестировщика. Требуемое поведение, как правило, понятно из названия тестового тура: «Путь не туда» (экспериментируем со всем, что ведет к неверному или незапланированному процессу использования приложения), «упорный пользователь» (пытаемся «на характере» войти во все недоступные места), «неуверенный пользователь» (постоянно делаем и отменяем последнее действие) и т. д.

### **Задание 5. Тестовые туры**

1. Изучи более детально тему тестовых туров (можешь использовать разные ресурсы, а также обращаться к другим пирам за рекомендациями по источникам информации об этой теме).
2. Используй свой результат работы по Заданию 2 из проекта QA2\_Test artifacts («Тестовые артефакты») и продумай, какие тестовые туры можно было бы составить для проверки данного раздела (минимум 2 тура).
3. Опиши суть каждого выбранного тестового тура (в чем его особенность при выполнении тестирования).

### **Тестовые туры для раздела "Как мы учим" (Школа 21)**

**Ссылка:** <https://21-school.ru/methodology>

На основе чек-листа из Задания 2 можно разработать **исследовательские тестовые туры**, чтобы проверить раздел с неочевидными сценариями.

### **Тур 1: «Путь не туда»**

**Цель:** Проверить, как система реагирует на нестандартные действия пользователя.

**Сценарии:**

1. **Навигация «вслепую»:**
   * Кликать по случайным элементам (не только кнопкам, но и иконкам, пустым местам).
   * Проверить, нет ли скрытых ссылок или неочевидных переходов.
2. **Использование браузерных возможностей:**
   * Вручную вводить несуществующие URL (например, https://21-school.ru/methodology/test).
   * Пытаться открыть раздел через кэш (Ctrl+F5).
3. **Неверные данные в формах (если есть):**
   * Вводить HTML-теги в поля поиска (<script>alert(1)</script>).
   * Пытаться отправить пустые формы.

**Что проверяем:**

* Появляются ли ошибки 404 или 500.
* Не ломается ли верстка при неожиданных действиях.
* Есть ли защита от XSS-инъекций.

### **Тур 2: «Неуверенный пользователь»**

**Цель:** Эмулировать поведение человека, который сомневается в каждом действии.

**Сценарии:**

1. **Постоянные изменения решений:**
   * Открыть раздел «Peer-to-peer» → сразу вернуться назад.
   * Начать заполнять форму (если есть) → удалить данные → снова начать ввод.
2. **Многократные повторения:**
   * 5 раз подряд нажать одну и ту же кнопку (например, «Подать заявку»).
   * Быстро переключаться между вкладками браузера (вперед-назад).
3. **Проверка «отмены»:**
   * Использовать горячие клавиши (Ctrl+Z, Backspace).
   * Закрывать страницу во время загрузки.

**Что проверяем:**

* Не дублируются ли отправки форм.
* Не «падает» ли система при частых переходах.
* Сохраняются ли данные при прерванных действиях.

### **Тур 3 (дополнительный): «Упорный хакер»**

**Цель:** Попытаться обойти ограничения системы.

**Сценарии:**

1. **Обход ограничений:**
   * Через DevTools изменить CSS-классы (например, скрыть блок с условиями).
   * Подменить API-запросы (если раздел динамически грузит данные).
2. **Попытки «взлома»:**
   * В URL вручную подставлять параметры (?debug=true).
   * Пытаться получить доступ к админ-разделам (/admin).

**Что проверяем:**

* Нет ли утечки敏感 данных (например, API-ключей в коде).
* Защищены ли критические элементы от модификации.

### **Задание 6. Матрица тестового покрытия**

Вернись к проекту QA2\_Test artifacts («Тестовые артефакты») и выполненному в нем Заданию 5 (дополнение чек-листа). Составь матрицу тестового покрытия для данной страницы. Проверки уже написаны, осталось только написать ту функциональность, для которой они созданы.

1. Составь таблицу по аналогии с примером выше.
2. Выдели наиболее важный функционал.
3. Пронумеруй проверки чек-листа (произвольный формат номера).

### **Матрица тестового покрытия для раздела "Как мы учим"**

**Ссылка:** <https://21-school.ru/methodology>

#### **1. Ключевой функционал страницы**

1. **Информационные блоки** (описание методологии, подразделы)
2. **Навигация** (меню, якорные ссылки)
3. **Контент** (текст, изображения, видео)
4. **Интерактивные элементы** (кнопки, формы, аккордеоны)
5. **Технические аспекты** (адаптивность, SEO, безопасность)

### **2. Матрица покрытия**

| **№** | **Функциональность** | **ID проверки** | **Описание проверки** | **Статус (Pass/Fail)** | **Приоритет** |
| --- | --- | --- | --- | --- | --- |
| 1 | Отображение заголовка | TC-01 | Заголовок страницы соответствует ожидаемому (Как мы учим). | Pass | High |
| 2 | Блоки с описанием методологии | TC-02 | Основной текст и подразделы (Peer-to-peer, Индивидуальный график) отображаются. | Pass | High |
| 3 | Наличие иллюстраций | TC-03 | Иконки/изображения сопровождают текст. | Pass | Medium |
| 4 | Работа навигации | TC-04 | Клик по пунктам меню ведет к нужным разделам. | Pass | High |
| 5 | Корректность якорных ссылок | TC-05 | Якорные ссылки (если есть) работают без ошибок. | Pass | Medium |
| 6 | Читабельность текста | TC-06 | Нет орфографических/грамматических ошибок. | Pass | High |
| 7 | Доступность контента (WCAG) | TC-07 | Контрастность текста и фона соответствует нормам. | Pass | Medium |
| 8 | Адаптивность (мобильные устройства) | TC-08 | Контент корректно отображается на мобильных устройствах. | Pass | High |
| 9 | Работа кнопок | TC-09 | Кнопки (например, «Подать заявку») кликабельны. | Pass | High |
| 10 | Проверка видео (если есть) | TC-10 | Видео воспроизводится без ошибок. | Fail (видео отсутствует) | Low |
| 11 | Работа аккордеонов | TC-11 | Аккордеоны/вкладки раскрываются и скрываются корректно. | Pass | Medium |
| 12 | Внешние ссылки | TC-12 | Нет битых ссылок (404 ошибки). | Pass | High |
| 13 | SEO-метаданные | TC-13 | Title и description заполнены. | Pass | Medium |
| 14 | Alt-атрибуты изображений | TC-14 | У всех изображений есть alt-атрибуты. | Pass | Low |
| 15 | Время загрузки страницы | TC-15 | Страница загружается за ≤3 сек. | Pass | High |
| 16 | Ошибки в консоли | TC-16 | В консоли разработчика нет ошибок. | Pass | High |

### **3. Выделение критичного функционала**

**High Priority (Критичный):**

* Корректность заголовка и основного контента (TC-01, TC-02).
* Работоспособность навигации и кнопок (TC-04, TC-09).
* Адаптивность и время загрузки (TC-08, TC-15).

**Medium Priority (Важный, но не критичный):**

* Читабельность и доступность (TC-06, TC-07).
* Работа аккордеонов (TC-11).

**Low Priority (Дополнительный):**

* Наличие видео (TC-10) — требует доработки.
* Alt-атрибуты (TC-14) — влияют на SEO, но не на функционал.

### **4. Итоговое покрытие**

* **Пройдено:** 15/16 проверок (93.7%).
* **Не пройдено:** TC-10 (отсутствие видео).
* **Рекомендации:**
  + Добавить видео или убрать упоминание о нем.
  + Провести дополнительный тур «Путь не туда» для проверки edge-кейсов.

**Формат ID проверки:**

* TC-XX — Test Case (произвольная нумерация).
* Приоритеты: **High** (обязательно), **Medium** (желательно), **Low** (по возможности).

Матрица помогает наглядно оценить, какие элементы страницы требуют доработки, а какие уже протестированы.