# Проект №1 — Введение

Привет, участник Школы 21! Сфера IT для тебя новая, и иногда может показаться, что ничего не понятно. Это нормально (и не только для IT-сферы). Справиться с любой сложностью поможет корректно составленный вопрос. В каждой непонятной ситуации формулируй вопрос. Как только удалось сформировать правильный вопрос, считай, что полдела сделано. С этим вопросом отправляйся (в порядке приоритета):

* в поисковик,
* к одногруппнику / другу / в общий чат группы.

Главное — в курсе нет ничего, что тебе не под силу.

Этот проект является первой частью нашего большого курса по ручному тестированию. Сегодня мы с тобой познакомимся с тем, как устроено тестирование, зачем оно нужно и как его проводить. А начнём, пожалуй, с основных понятий 🤫!

## Instructions

Хотим дать ещё несколько подсказок, перед тем как ты приступишь к курсу 🙃

Все отчёты по результатам выполнения задания тебе придется оформлять в файлах с расширением .md. Markdown (отсюда и расширение md) — это облегчённый язык разметки. Подробно про него ты можешь прочитать [тут](https://ru.wikipedia.org/wiki/Markdown), если ещё с ним не знаком. Даже этот курс написан в файлах с расширением .md! Все картинки, выделения текстовой информации, гиперссылки, блоки программного кода — в общем, всё то, что не было доступно в .txt, есть в markdown.

Все созданные отчёты и файлы тебе нужно будет загрузить в папку src/ в корне проекта (обязательно в ветку develop). Если они уже созданы, то пересоздавать или удалять их не нужно (просто отредактируй этот файл).

А ещё мы подготовили список дополнительной литературы, который позволит тебе лучше освоиться в материале этого и последующих блоков:

* Святослав Куликов — "Тестирование программного обеспечения. Базовый курс. (3-е издание)"
* Роман Савин — "Тестирование Дот Ком"
* Гленфорд Майерс — "Искусство тестирования программ"

## Contents

1. [Chapter I](https://edu.21-school.ru/project/62652/task#chapter-i)  
   1.1. [Общая инструкция](https://edu.21-school.ru/project/62652/task#%D0%BE%D0%B1%D1%89%D0%B0%D1%8F-%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D0%B8%D1%8F)
2. [Chapter II](https://edu.21-school.ru/project/62652/task#chapter-ii)  
   2.1. [Тестирование и тестировщики](https://edu.21-school.ru/project/62652/task#%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D0%B8-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D1%89%D0%B8%D0%BA)  
   2.2. [QA, QC](https://edu.21-school.ru/project/62652/task#qa-qc)  
   3.3. [Задание №1](https://edu.21-school.ru/project/62652/task#%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5-1-%D0%B2%D0%B8%D0%B4%D1%8B-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)
3. [Chapter III](https://edu.21-school.ru/project/62652/task#chapter-iii)  
   3.1. [Уровни тестирования](https://edu.21-school.ru/project/62652/task#%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B8-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)  
   3.2. [Стандарты](https://edu.21-school.ru/project/62652/task#%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D1%8B)  
   3.3. [Задание №2](https://edu.21-school.ru/project/62652/task#%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5-2-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D0%B2-%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B5)
4. [Chapter IV](https://edu.21-school.ru/project/62652/task#chapter-iv)  
   4.1. [Модальные окна](https://edu.21-school.ru/project/62652/task#%D0%BC%D0%BE%D0%B4%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5-%D0%BE%D0%BA%D0%BD%D0%B0)  
   4.2. [Чек-боксы](https://edu.21-school.ru/project/62652/task#%D1%87%D0%B5%D0%BA-%D0%B1%D0%BE%D0%BA%D1%81%D1%8B)  
   4.3. [Задание №3](https://edu.21-school.ru/project/62652/task#%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5-3-%D0%BC%D0%BE%D0%B4%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5-%D0%BE%D0%BA%D0%BD%D0%BE-%D0%B0%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8)

## Chapter I

**Общая инструкция**

Методология Школы 21 может быть не похожа на тот образовательный опыт, который случался с тобой ранее. Её отличает высокий уровень автономии: у тебя есть задача, ты должен её выполнить. По большей части тебе нужно будет самому добывать знания для её решения. Второй важный момент — это peer-to-peer обучение. В образовательном процессе нет менторов и экспертов, перед которыми ты защищаешь свой результат. Ты это делаешь перед такими же учащимися, как и ты сам. У них есть чек-лист, который поможет им качественно выполнить приемку вашей работы.

Роль Школы 21 заключается в том, чтобы обеспечить через последовательность заданий и оптимальный уровень поддержки такую траекторию обучения, при которой ты не только освоишь hard skills, но и научишься самообучаться.

* Не доверяй слухам и предположениям о том, как должно быть оформлено ваше решение. Этот документ является единственным источником, к которому стоит обращаться по большинству вопросов;
* твое решение будет оцениваться другими учащимися;
* подлежат оцениванию только те файлы, которые ты выложил в GIT (ветка develop, папка src);
* в твоей папке не должно быть лишних файлов — только те, что были указаны в задании;
* не забывай, что у вас есть доступ к интернету и поисковым системам;
* обсуждение заданий можно вести и в Rocket.Chat;
* будь внимателен к примерам, указанным в этом документе — они могут иметь важные детали, которые не были оговорены другим способом;
* и да пребудет с тобой Сила!

**Chapter II**

**Тестирование и тестировщики**

Что же такое тестирование? **Тестирование** — это процесс оценки программного продукта на предмет его соответствия поставленным требованиям. Под **требованиями** в свою очередь мы будем понимать спецификацию того, что должно быть реализовано.

Зачем проводится тестирование? Попробуем сформулировать это так:

* чтобы повысить вероятность того, что приложение, предназначенное для тестирования, будет работать правильно при любых обстоятельствах;
* чтобы повысить вероятность того, что приложение, предназначенное для тестирования, будет соответствовать всем описанным требованиям;
* чтобы иметь представление о состоянии продукта на данный момент;
* чтобы выявить ситуации, в которых поведение программы является неправильным или не соответствующим спецификации.

Соответственно, **тестировщиком** называют человека, который осуществляет это тестирование, проверяя какой-либо программный продукт.

**QA, QC**

Часто в сфере тестирования можно встретить такие понятия, как Quality Assurance (обеспечение качества) и Quality Control (контроль качества). Давай разберёмся, что это такое и чем они отличаются от обычного тестирования.

**Quality Control (QC)** — обеспечивает проверку продукта не только на соответствие требованиям, но и на соответствие заранее согласованному уровню качества продукта и готовность к выпуску продукта в эксплуатацию.

**Quality Assurance (QA)** — это не только и не столько тестирование, будь то ручное или автоматизированное, это набор методик и подходов, предназначенных для выявления и контроля всех несоответствий качества программных продуктов. То есть QA включает в себя QC. QA должно обеспечивать методы и технологии всех участников процесса создания ПО, чтобы в итоге получить качественный веб-продукт.

Тогда можно сказать, что тестирование является одной из техник контроля качества, включающая в себя активности по планированию работ (Test Management), проектированию тестов (Test Design), выполнению тестирования (Test Execution) и анализу полученных результатов (Test Analysis).

Если это представить графически, то мы получим следующее изображение:

**Задание №1. Виды тестирования**

В этом задании тебе нужно найти информацию про различные виды тестирования. Существует множество оснований, по которым можно классифицировать виды тестирования ПО — уровень, функциональность, позитивность и другие. Сейчас мы ограничимся следующими:

По уровню тестирования:

* модульное тестирование;
* интеграционное тестирование;
* системное тестирование.

По степени автоматизации:

* ручное тестирование;
* полуавтоматизированное тестирование;
* автоматизированние тестирование.

Создай документ exercise1.md, в котором дай определения вышеперечисленным шести видам тестирования. Каждое новое определение пиши с новой строки.

**Chapter III**

**Уровни тестирования**

Существуют четыре уровня тестирования:

* Модульное (оно же компонентное) тестирование;
* Интеграционное тестирование;
* Системное тестирование;
* И ещё один уровень тестирования, который ты ещё не рассматривал в предыдущем задании, — **приемочное тестирование** — тестирование, проводимое при сдаче продукта заказчику. Целью приемочного тестирования является определение готовности программного продукта к использованию конечными пользователями. Набор тест-кейсов и сценариев разрабатывается на основании требований к данному приложению.

**Стандарты**

Многие организации во всем мире разрабатывают и внедряют различные стандарты для повышения качества своих программ. Рассмотрим самые используемые из них:

* **ISO / IEC 9126** (рассматриваются различные аспекты для определения качества программного приложения)
* **ISO / IEC 9241-11** (рассматривается степень, в которой продукт может использоваться указанными пользователями для достижения определенных целей)
* **ISO / IEC 25000: 2005** (рекомендации по требованиям и оценке качества программного обеспечения — *SQuaRE*)
* **ISO / IEC 12119** (о пакетах программного обеспечения, поставляемых клиенту)
* **IEEE 829** (стандарт для формата документов, используемых на разных этапах тестирования программного обеспечения)
* **IEEE 1008** (стандарт для модульного тестирования)
* **IEEE 1059** (руководство по планам проверки и валидации программного обеспечения)
* **IEEE 1061** (методология определения требований к качеству, метрики)
* **BS 7925-1** (словарь терминов, используемых при тестировании программного обеспечения)
* **BS 7925-2** (стандарт для тестирования компонентов программного обеспечения)

Полную информацию о них ты всегда можешь найти в интернете!😊

**Задание №2. Тестирование в разработке**

Создай файл exercise2.md. Пользуясь дополнительной литературой, а также информацией из интернета, опиши в этом файле: каким образом, на каких стадиях жизненного цикла ПО и кем может проводиться каждый из описанных выше уровней тестирования. Озаглавь свои идеи заголовком # Уровни тестирования.

**Chapter IV**

Конечно, в сфере IT есть множество своих терминов, которые рядовому тестировщику необходимо знать. В ходе курса мы потихоньку будем разбираться с каждым из них, а сейчас начнём разговор с самых простых и базовых вещей — модальные окна (модалки) и чек-боксы.

**Модальные окна**

Модальное (или всплывающее) окно — это элементы интерфейса, которые открываются поверх текущей страницы, требующие отклика пользователя. Самым простым примером модального окна является окно подтверждения. Например, ты хочешь удалить какое-нибудь сообщение в чате, тогда "всплывёт" модальное окно с просьбой подтвердить действие. Пример более сложных модальных окон — это окна авторизации и регистрации, которые требуют не только нажатия на одну из предложенных кнопок, но также заполнения текстовых полей.

**Чек-боксы**

Чек-бокс тебе может быть знаком под другим названием — флажок/галочка. Флажок, флаговая кнопка, чекбокс, галочка — это элемент графического пользовательского интерфейса, позволяющий пользователю управлять параметром с двумя состояниями — включено и отключено (галочка стоит, либо галочки нет).

**Задание №3. Модальное окно авторизации**

Перейди на сайт <https://sbermarket.ru/>, потом найди на сайте модальное окно авторизации. Подобная форма есть во всех приложениях и веб-сервисах. Рассмотрим ситуацию, когда нет никакой документации (требований и технического задания) к данной форме. В файле exercise3.md опиши, за что отвечает каждая кнопка и каждый чек-бокс. К чему приводит нажатие на каждую функциональную часть модального окна? Что происходит, когда ты наводишь курсор на какую-нибудь кнопку?

В md-файле не забывай добавлять заголовки к своему тексту, чтобы разделить его по функционалу приложения. Например:

## Закрытие модального окна

Чтобы закрыть модальное окно, необходимо либо нажать на иконку крестика в правой верхней части модального окна, либо нажать кнопку Esc на клавиатуре.

**Double-check**

Перед загрузкой выполненного проекта в репозиторий перепроверь наличие всех необходимых файлов, которые требовалось создать во время его выполнения:

exercise1.md

exercise2.md

exercise3.md

💡 [Нажми здесь](https://forms.gle/EeP2Q2t7ZezVet6W6) **чтобы отправить обратную связь по проекту**.