|  |  |
| --- | --- |
|  | **Тестовый документ** |

## **Аннотация теста**

|  |  |
| --- | --- |
| **Название проекта** | ClassLibrary1 |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Савицкий Даниил |
| **Дата(ы) теста** | 02.07.22, 03.07.22 |

## **Расшифровка тестовых информационных полей:**

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Название тестируемого проекта |
| **Рабочая версия** | Версия проекта/программного обеспечения (первый тест считается 1.0). |
| **Имя тестирующего** | Имя того, кто проводил тесты |
| **Дата(ы) теста** | Дата(ы) проведения тестов – это один или несколько дней. Если тесты проводились в более протяженный период времени, нужно отметить отдельную дату для каждого теста. |
| **Тестовый пример #** | Уникальный ID для каждого тестового примера. Следуйте некоторым конвенциям, чтобы указать типы тестов. Например,‘TC\_UI\_1′ означает‘user interface test case #1′ ( ТС\_ПИ\_1: тестовый случай пользовательского интерфейса#1) |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Насколько важен каждый тест. Приоритет тестирования для бизнес-правил и функциональных тестовых случаев может быть средним или высоким, в то время как незначительные случаи пользовательского интерфейса могут иметь низкий приоритет. |
| **Заголовок/название теста** | Название тестового случая. Например, Подтвердите страницу авторизации с действительным именем пользователя и паролем. |
| **Краткое изложение теста** | Описание того, что должен достичь тест. |
| **Этапы теста** | Перечислите все этапы теста подробно. Запишите этапы теста в том порядке, в котором они должны быть реализованы. Предоставьте как можно больше подробностей и разъяснений. Пронумерованный список – хорошая идея. |
| **Тестовые данные** | Перечислите/опишите все тестовые данные, используемые для данного тестового случая. Так, фактические используемые входные данные можно отслеживать по результатам тестирования. Например, Имя пользователя и пароль для подтверждения входа. |
| **Ожидаемый результат** | Каким должен быть вывод системы после выполнения теста? Подробно опишите ожидаемый результат, включая все сообщения/ошибки, которые должны отображаться на экране. |
| **Фактический результат** | Каким должен быть фактический результат после выполнения теста? Опишите любое релевантное поведение системы после выполнения теста. |
| **Предварительное условие** | Любые предварительные условия, которые должны быть выполнены до выполнения теста. Перечислите все предварительные условия для выполнения этого тестового случая. |
| **Постусловие** | Каким должно быть состояние системы после выполнения теста? |
| **Статус**  *(Зачет/Незачет)* | Если фактический результат не соответствует ожидаемому результату, отметьте тест как неудачный. В ином случае обновление пройдено. |
| **Примечания/комментарии** | Используйте эту область для любых дополнительных заметок/комментариев/вопросов. Эта область предназначена для поддержки вышеуказанных полей (например, если есть некоторые особые условия, которые не могут быть описаны в любом из вышеуказанных полей, или если есть вопросы, связанные с ожидаемыми или фактическими результатами). |

## **Тестовый пример #1:**

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Check\_HasUpperCase\_ReturnsTrue |
| **Краткое изложение теста** | Инициалы должны состоять только из заглавных букв |
| **Этапы теста** | 1. Ввести фио в переменные 2. Сгенерировать инициалы с помощью функции EnterName(); 3. Получить результат 4. Проверить на заглавные буквы 5. Сравнить фактический и ожидаемый результаты |
| **Тестовые данные** | Фамилия = «вavitskiy» имя = "вaniil"; отчество = "вalentinovich";  Фамилия = «вavitskiy» имя = "Вaniil"; отчество = "Вalentinovich";  Фамилия = «Вavitskiy» имя = "Вaniil"; отчество = "Вalentinovich"; |
| **Ожидаемый результат** | Инициалы состоят только из заглавных букв |
| **Фактический результат** | Инициалы состоят только из прописных букв, условие не выполнено  Не все буквы заглавные, условие не выполнено  Инициалы состоят только из заглавных букв, условие выполнено |
| **Статус** | Незачет  Незачет  Зачет |
| **Предварительное условие** | Должно быть введено фио |
| **Постусловие** | Инициалы подходят для генерации студ билета |
| **Примечания/комментарии** | Инициалы имеют высокий приоритет, т.к без них нельзя составить студенческий билет |

## **Тестовый пример #2:**

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Check\_Year4Sym\_ReturnsTrue |
| **Краткое изложение теста** | Год должен состоять из 4 цифр |
| **Этапы теста** | 1. Ввести год поступления в переменные 2. Проверить его на количество цифр 3. С помощью StudCheck() внесем год в билет 4. Сравнить фактический и ожидаемый результаты |
| **Тестовые данные** | 2019 219 20190  2020 |
| **Ожидаемый результат** | Год состоит из 4 цифр |
| **Фактический результат** | 4 цифры, условие выполнено 3 цифры, условие не выполнено  5 цифр, условие не выполнено  4 цифры, условие выполнено |
| **Статус** | Зачет Незачет Незачет  Зачет |
| **Предварительное условие** | Должен быть введен год поступления |
| **Постусловие** | Год не может состоять из более чем 4 цифр / менее чем из 4 цифр |
| **Примечания/комментарии** | Год поступления имеет высокий приоритет, т.к без него нельзя составить студенческий билет |

## **Тестовый пример #3:**

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_3 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Check\_Group3Sym\_ReturnsTrue |
| **Краткое изложение теста** | Группа студента может состоять только из 3 цифр |
| **Этапы теста** | 1. Ввести группу студента в переменные 2. Проверить его на количество цифр 3. С помощью StudCheck() внесем группу в билет 4. Сравнить фактический и ожидаемый результаты |
| **Тестовые данные** | 175  195  19  1815 |
| **Ожидаемый результат** | Группа студента состоит из 3 цифр |
| **Фактический результат** | 3 цифры, условие выполнено  3 цифры, условие выполнено  2 цифры, условие не выполнено  4 цифры, условие не выполнено |
| **Статус** | Зачет  Зачет  Незачет  Незачет |
| **Предварительное условие** | Должна быть введена группа студента |
| **Постусловие** | Группа не может состоять из более или менее чем 3 цифр |
| **Примечания/комментарии** | Группа студента имеет высокий приоритет, т.к без неё нельзя составить студенческий билет |

## **Тестовый пример #4:**

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_4 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Check\_IsNull\_ReturnsTrue |
| **Краткое изложение теста** | В студенческом билете должны присутствовать все предыдущие данные (год.группа.фио) |
| **Этапы теста** | 1. Ввести все данные в переменные 2. Сгенерировать фио при помощи EnterName() 3. Внести данные в билет при помощи StudCheck() 4. Провести проверку на заполнение билета 5. Сравнить фактический и ожидаемый результаты |
| **Тестовые данные** | 2019.195.SDV  195.SDV  2019.SDV |
| **Ожидаемый результат** | В билете должны содержаться год поступления, группа студента, инициалы студента |
| **Фактический результат** | Введены год группа и фио, условие выполнено  Введены группа и фио, условие не выполнено  Введены год и фио, условие не выполнено |
| **Статус** | Зачет  Незачет  Незачет |
| **Предварительное условие** | Должны быть введены данные студента для билета |
| **Постусловие** | Не один из моментов не должен оставаться пустым |
| **Примечания/комментарии** | Высокий приоритет, т.к без этой проверки нельзя составить студенческий билет |