**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчет по лабораторной работе №6**

**Изучение процесса тестирования ПО методом «белого ящика»**

Студент: Власов Вячеслав и Панов Матвей

Группа: ИСПП-31

Преподаватель: Садовский Роман

Архангельск 2025

**1 Цель работы**

1.1 Изучить и освоить процесс тестирования методом «белого ящика»

**2 Протокол тестирования**

**1. Идентификатор теста:** BS-001

**2. Название теста:** Тестирование класса Book

**3. Цель теста:** Проверить корректность работы элементов класса: свойства Title и Author, конструктор для создания объекта, переопределённый метод ToString() выводящий информацию об экземпляре.

**4. Предусловия:**

* Свойства на чтения и запись Title и Authors должны быть реализованы,
* Создание объекта класса Book, при помощи конструкторов,
* Вывод информации, метод должен выводить информацию об экземпляре.

**5. Описание тестируемого класса:**

Конструктор принимает два параметра title и author, присваивая их свойствам, метод ToString() переопределён таким образом, что выводит информацию об объекте.

class Book

{

public string Title { get; set; }

public string Author { get; set; }

public Book(string title, string author)

{

Title = title;

Author = author;

}

public override string ToString()=> $"\"{Title}\" by {Author}";

}

**6. Шаги выполнения теста:**

* Шаг 1: Создать экземпляр класса, передав два параметра.
* Шаг 2: Вызвать свойства, сравнить их с передаваемыми значениями.
* Шаг 3: Вызвать метод ToString(), выводимые значения должны совпадать с передаваемыми.

**7. Ожидаемый результат:**

* При создании объекта с параметрами “Война и Мир”, “Толстой”. Метод выводит значение “Война и Мир by Толстой”.

**8. Фактический результат:**

* При создании объекта с параметрами “Война и Мир”, “Толстой”. Метод выводит значение “Война и Мир by Толстой” (тест пройден)

**9. Примечания:**

Класс работает корректно, переопределённый метод вывод верную информацию

**10. Статус:**

Пройден

**1. Идентификатор теста:** BS-002

**2. Название теста:** Тестирование ветвления в методе отображения всех книг.

**3. Цель теста:** Проверить корректность ветвления в методе ShowBooks, в зависимости от количества сохранённых книг в библиотеке.

**4. Предусловия:**

* Метод ShowBooks() должен быть реализован.
* Метод ведёт перебор всех книг циклом for, в том случае если они есть в библиотеке, если их нет, то выводиться соответствующая информация.

**5. Описание тестируемого класса:**

Метод проверяет количество объектов в списке, если их ноль, сообщается что книг нет, если их больше чем ноль, то выводит все объекты в списке поочередно.

static void ShowBooks(List<Book> library)

{

if (library.Count == 0)

Console.WriteLine("В библиотеке нет книг.");

else

{

Console.WriteLine("\nСписок книг в библиотеке:");

for (int i = 0; i < library.Count; i++)

Console.WriteLine($"{i + 1}. {library[i]}");

}

}

**6. Шаги выполнения теста:**

* Шаг 1: Передаём в метод список объектов Book.
* Шаг 2: Проверить что выведет метод, если в списке нет объектов.
* Шаг 3: Проверить что выведет метод, если в списке одна или более книг.

**7. Ожидаемый результат:**

* Если в списке 0 объектов, то выведется сообщение "В библиотеке нет книг."
* Если в списке есть объекты, выведется сообщение “Список книг в библиотеке”. “i. Title by Author”, где i – номер книги, Title – название, Author – автор.

**8. Фактический результат:**

* В списке 0 объектов, выводится сообщение "В библиотеке нет книг." (пройден)
* В списке 2 объекта, вывели:

*Список книг в библиотеке:*

*1. "чирик" by Чикр*

*2. "Номих" by хома*

**9. Примечания:**

Метод работает корректно, ветвление обрабатывается по правилам

**10. Статус:**

Пройден

**1. Идентификатор теста:** BS-003

**2. Название теста:** Тестирование добавления объектов в список

**3. Цель теста:** Проверить добавление объектов Book при помощи метода AddBook().

**4. Предусловия:**

* Метод AddBook() должен быть реализован.
* Метод запрашивает «Название Книги» и «Имя Автора», после добавления уведомляет об этом.

**5. Описание тестируемого класса:**

AddBook() принимает список, запрашивает название и имя автора. Добавляет новую книгу в список.

static void AddBook(List<Book> library)

{

Console.Write("Введите название книги: ");

string title = Console.ReadLine();

Console.Write("Введите автора книги: ");

string author = Console.ReadLine();

library.Add(new Book(title, author));

Console.WriteLine("Книга добавлена.");

}

**6. Шаги выполнения теста:**

* Шаг 1: Передаём в метод AddBook() список объектов Book.
* Шаг 2: Вводим значения для нового объекта
* Шаг 3: Проверяем добавляет ли метод объекты в список при помощи метода ShowBook().

**7. Ожидаемый результат:**

* Вводим название книги “Десять негритят” и имя автора ”Агата Кристи”. После просмотра списка книг должны увидеть “Десять негритят by Агата Кристи”.

**8. Фактический результат:**

* Вводим название книги “Десять негритят” и имя автора ”Агата Кристи”. После просмотра списка книг видим “Десять негритят by Агата Кристи” (тест пройден).

**9. Примечания:**

Метод работает корректно объекты добавляются в список.

**10. Статус:**

Пройден

**1. Идентификатор теста:** BS-004

**2. Название теста:** Проверка удаления книг из списка, а также ветвления

**3. Цель теста:** Удостоверить что метод для удаления книг из списка корректно обрабатывает условия RemoveBook()

**4. Предусловия:**

* Метод RemoveBook() должен быть реализован
* Проверка настроена

**5. Описание тестируемого класса:**

RemoveBook() принимает список объектов Book, проверяет есть ли в нём какие-либо объекты. Если есть, запрашивает номер книги для удаления, если такая книга существует, то удаляет её.

static void RemoveBook(List<Book> library)

{

ShowBooks(library);

if (library.Count == 0) return;

Console.Write("Введите номер книги для удаления: ");

if (int.TryParse(Console.ReadLine(), out int index) && index > 0 && index <= library.Count)

{

library.RemoveAt(index - 1);

Console.WriteLine("Книга удалена.");

}

else

Console.WriteLine("Неверный ввод.");

}

**6. Шаги выполнения теста:**

* Шаг 1: Передаём в метод список объектов Book.
* Шаг 2: Проверить, что произойдёт, если нет объектов.
* Шаг 3: Проверить, что произойдёт если ввести номер несуществующий в списке.
* Шаг 4: Проверить, что произойдёт, если ввести номер существующей книги.
* Шаг 5: Поверить удаление при помощи метода ShowBooks()

**7. Ожидаемый результат:**

* Вводим список без элементов, должны получить ничего.
* Вводим список книг, выбираем существующую книгу
* Вводим список книг, выбираем несуществующую книгу

**8. Фактический результат:**

* Ввели список без элементов, получили ничего. (пройден)
* Вводим список книг, выбираем существующую книгу, удаляется один элемент (пройден)
* Вводим список книг, выбираем несуществующую книгу, метод сообщает, что такой книги нет

**9. Примечания:**

Метод работает корректно, проверки на количество и существование книги в списке работает корректно, метод удаляет объекты.

**10. Статус:**

Пройден

**1. Идентификатор теста:** BS-005

**2. Название теста:** Проверка конструкции switch-case

**3. Цель теста:** Проверить, что конструкция правильно обрабатывает передаваемое значение и вызывает нужные методы.

**4. Предусловия:**

* Конструкция должна быть реализована
* Передаётся int значение, в зависимости от числа вызывать разные методы: 1 – AddBook(); 2 – RemoveBook(); 3 – ShowBook(); 4 – Выход из цикла; default - "Неверный выбор. Пожалуйста, попробуйте снова."

**5. Описание тестируемого класса:**

Принимает int значение, и в зависимости от него вызываются разные части кода.

switch (choice)

{

case "1":

AddBook(library);

break;

case "2":

RemoveBook(library);

break;

case "3":

ShowBooks(library);

break;

case "4":

running = false;

break;

default:

Console.WriteLine("Неверный выбор. Пожалуйста, попробуйте снова.");

break;

}

**6. Шаги выполнения теста:**

* Шаг 1: передаём число диапазоне (1-4), проверяем какой метод выводится
* Шаг 2: передаём число вне диапазона (1-4), выводится значение по умолчанию

**7. Ожидаемый результат:**

* Вводим число 1, должен вызваться метод AddBook().
* Вводим число 2, должен вызваться метод RemoveBook ().
* Вводим число 3, должен вызваться метод ShowBooks ().
* Вводим число 4, должен прекращаться цикл.
* Вводим число 5, должно выводиться "Неверный выбор. Пожалуйста, попробуйте снова."

**8. Фактический результат:**

* Вводим число 1, вызывается метод AddBook(). (пройден)
* Вводим число 2, вызывается метод RemoveBook (). (пройден)
* Вводим число 3, вызывается метод ShowBooks (). (пройден)
* Вводим число 4, прекращается цикл. (пройден)
* Вводим число 5, выводиться "Неверный выбор. Пожалуйста, попробуйте снова." (пройден)

**9. Примечания:**

Конструкция работает корректно, вводимые значение соответствуют вызываемым частям кода.

**10. Статус:**

Пройден

**3 Контрольные вопросы**

3.1 Как минимизировать количество тест-кейсов при тестировании?

Для этого можно применять тестирования классами эквивалентности, граничными значениями или попарное тестирование.

3.2 Каковы этапы тестирования по методу белого ящика?

Основные этапы тестирования по методу белого ящика следующие:

- анализ программного кода,

- определение возможных маршрутов программы,

- подбор передаваемых данных, чтобы пройти по всем путям программы, определение ожидаемых результатов для входных данных,

- выполнение тестов,

- анализ результатов.

**4 Вывод**

Изучили и освоили процесс тестирования методом «белого ящика».