Пенза 2018

**ОТЧЕТ**

по лабораторной работе №4

«Основы процесса тестирования»

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Цель работы

Выполнили:

студенты группы 15ВВ1

Буряков А.В.

Рахматуллина Р.Х.

Николайчев В.А.

Принял:

Карамышева Н.С.

Трокоз Д.А.

Тестирование – процесс оценки качества ПО, направленный на выявление отличий между его реально существующими и требуемыми свойствами (дефектов).

Тестирование реализуется с помощью следующих методов:

* поиск и документирование дефектов качества;
* проверка выполнения основных предположений и требований на корректных примерах;
* проверка функционирования в соответствии со спецификацией;
* проверка требований к результатам выполнения.

Можно выделить несколько основных уровней тестирования.

1. *Unit testing* (модульное)

На данном этапе тестируются все новые или измененные минимальные блоки приложения, в качестве которых могут выступать, например, функции или классы в программе. При этом проверяется логика работы, правильность производимых вычислений, обработка ошибок.

1. *Integration testing* (интеграционное)

При тестировании проверяется внутренняя целостность взаимосвязанных блоков программы (модуля) и взаимодействие (интерфейсы) с другими модулями программы, файлами данных и т.п.

1. *Build/Release testing* (сборочное)

Конструкцией (build) называется часть программного обеспечения, которая удовлетворяет некоторому набору требований из числа всех требований к приложению. Этот этап имеет место в тех случаях, когда размеры программы диктуют необходимость ее разбиения на такие крупные блоки.

### Листинг проекта с тестами

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using DB;

using System.Windows.Forms;

namespace DB.Tests

{

[TestClass]

public class UnitTest\_case2

{

[TestMethod]

public void TestMethod\_case2\_1()

{

MainForm mainFormTest = new MainForm();

int returned = mainFormTest.SearchParts("Все", "saasdasdmlkanqwneqwnenjnkjasdsandjnsadasdklmsadlkamsdlkmaskd alksdlkamsdkaslkd askdnlasdalkdsnsalkdqelkqnwknlqewnnewqknqlwe");

Assert.AreEqual(-1, returned);

}

[TestMethod]

public void TestMethod\_case2\_2()

{

MainForm mainFormTest = new MainForm();

int returned = mainFormTest.SearchParts("Все", "-\\,,-\*0\_ёЁ~+?&)||//");

Assert.AreEqual(-1, returned);

}

[TestMethod]

public void TestMethod\_case2\_3()

{

MainForm mainFormTest = new MainForm();

int returned = mainFormTest.SearchParts("Все1", "Запчасть");

Assert.AreEqual(-2, returned);

}

[TestMethod]

public void TestMethod\_case2\_4()

{

MainForm mainFormTest = new MainForm();

int returned = mainFormTest.SearchParts("Все", "");

Assert.AreEqual(-3, returned);

}

}

[TestClass]

public class UnitTest\_case3

{

[TestMethod]

public void TestMethod\_case3\_1()

{

MainForm mainForm = new MainForm();

AddDealsForm addDealsForm = new AddDealsForm(mainForm);

DateTime returnedData = addDealsForm.dateTimePicker.Value;

DateTime actualData = System.DateTime.Now;

Assert.AreEqual(actualData.Date, returnedData.Date);

}

[TestMethod]

public void TestMethod\_case3\_2()

{

MainForm mainForm = new MainForm();

AddDealsForm addDealsForm = new AddDealsForm(mainForm);

addDealsForm.buttonSetCurrentDate.PerformClick();

DateTime returnedData = addDealsForm.dateTimePicker.Value;

DateTime actualData = System.DateTime.Now;

Assert.AreEqual(actualData.Date, returnedData.Date);

}

}

}

# Результаты по Тест-Кейсам из Отчета №3:

|  |  |
| --- | --- |
| ***Тестовый случай 1*** | *Проверка названий элементов управления* |
| Действия по проверке требования | Визуально проверить заголовки, кнопки и прочие элементы программы. |
| Результат | Корректные названия |
| Выполнение требования | **Выполнено** |
| Комментарий |  |

|  |  |
| --- | --- |
| ***Тестовый случай 2*** | *Проверка блока поиска [Unit]* |
| Действия по проверке требования | 1. Ввести длинную строку в поле поиска. 2. Ввести строку со спецсимволами. 3. Выполнить поиск по колонке, которой нет в таблице. 4. Выполнить поиск с пустыми входными данными. |
| Результат | Отсутствие совпадений, корректная обработка поиска. |
| Выполнение требования | **Выполнено** |
| Комментарий | 1.  2.  Скопируем ее несколько раз:  3.  Unit тестирование:  [TestMethod]  public void TestMethod\_case2\_1()  {  MainForm mainFormTest = new MainForm();  int returned = mainFormTest.SearchParts("Все", "saasdasdmlkanqwneqwnenjnkjasdsandjnsadasdklmsadlkamsdlkmaskd alksdlkamsdkaslkd askdnlasdalkdsnsalkdqelkqnwknlqewnnewqknqlwe");  Assert.AreEqual(-1, returned);  }  [TestMethod]  public void TestMethod\_case2\_2()  {  MainForm mainFormTest = new MainForm();  int returned = mainFormTest.SearchParts("Все", "-\\,,-\*0\_ёЁ~+?&)||//");  Assert.AreEqual(-1, returned);  }  [TestMethod]  public void TestMethod\_case2\_3()  {  MainForm mainFormTest = new MainForm();  int returned = mainFormTest.SearchParts("Все1", "Запчасть");  Assert.AreEqual(-2, returned);  }  [TestMethod]  public void TestMethod\_case2\_4()  {  MainForm mainFormTest = new MainForm();  int returned = mainFormTest.SearchParts("Все", "");  Assert.AreEqual(-3, returned);  } |
|  | Результаты выполнения тестов: |

|  |  |
| --- | --- |
| ***Тестовый случай 3*** | *Корректное выставление даты по кнопке «Сегодня» [Unit]* |
| Действия по проверке требования | 1. Проверить соответствует ли дата текущей при открытии формы 2. Проверить соответствует ли дата при нажатии на кнопку Сегодня |
| Результат | Совпадение дат |
| Выполнение требования | **Выполнено** |
| Комментарий | Сам контрол является системным компонентом, его проверять не нужно    Результаты выполнения тестов:  [TestMethod]  public void TestMethod\_case3\_1()  {  MainForm mainForm = new MainForm();  AddDealsForm addDealsForm = new AddDealsForm(mainForm);  DateTime returnedData = addDealsForm.dateTimePicker.Value;  DateTime actualData = System.DateTime.Now;  Assert.AreEqual(actualData.Date, returnedData.Date);  }  [TestMethod]  public void TestMethod\_case3\_2()  {  MainForm mainForm = new MainForm();  AddDealsForm addDealsForm = new AddDealsForm(mainForm);  addDealsForm.buttonSetCurrentDate.PerformClick();  DateTime returnedData = addDealsForm.dateTimePicker.Value;  DateTime actualData = System.DateTime.Now;  Assert.AreEqual(actualData.Date, returnedData.Date);  } |

|  |  |
| --- | --- |
| ***Тестовый случай 4*** | *Ввод неверных данных в числовые поля «Количество запчастей» (Добавление записи на склад) и «Цена» (добавление запчасти).* |
| Действия по проверке требования | 1. Ввести число более чем 2 147 483 647 2. Ввести отрицательное число 3. Ввести последовательность из чисел и букв |
| Результат | Корректное сообщение о неправильном формате вводимых данных или отсутствие возможности такого ввода |
| Выполнение требования | **Выполнено** \ **Не выполнено** |
| Комментарий | Функции написаны схожим образом и проходят одинаковую проверку  **Число должно влезать в Int32 и являться неотрицательным**  1.  2.  3. |

|  |  |
| --- | --- |
| ***Тестовый случай 5*** | *Проверка правильности отображения ошибок введенных данных в окнах добавления записей* |
| Действия по проверке требования | 1. Вводить отсутствующие данные в таблицу и проверять отображение ошибок 2. Вводить частично правильные данные и проверять. 3. Ввести сначала одни, затем другие данные. Проверить среагировала ли программа на изменения. |
| Результат | Корректное отображение ошибок соответственно введенным данным. |
| Выполнение требования | **Выполнено** |
| Комментарий | Проверить всплывающие подсказки  1.  2.  3.  4.  5-6 |

|  |  |
| --- | --- |
| ***Тестовый случай 6*** | *Проверка доступности кнопки для всех окон добавления записи* |
| Действия по проверке требования | Пока в окне добавления записи показываются ошибки (красные кресты с примечаниями), это сигнал того что введенные пользователем значения не могут быть внесены в БД т.к. не соответствуют требованиям типов данных. |
| Результат | Если все ошибки скрыты, в окне доступен клик по кнопке добавления записи. Иначе если хотя бы один крест показан, кнопка недоступна. |
| Выполнение требования | **Выполнено** |
| Комментарий | См. тест-кейс №5 |

|  |  |
| --- | --- |
| ***Тестовый случай 7*** | *Удаление связанных записей* |
| Действия по проверке требования | 1. Удалить деталь, которая ссылается на склад. 2. Удалить поставщика этой детали на складе. |
| Результат | Корректная ошибка удаления, предлагающая сначала удалить зависимые записи. |
| Выполнение требования | **Выполнено** |
| Комментарий | Историю цен можно удалять полностью, т.к. текущая цена запчасти (она же последняя измененная цена) всегда содержится в таблице «Детали». |

**Вывод**

Мы провели тестирование созданного приложения, научились использовать модульные тесты для проверки модулей, отладили программу для соответствия ожидаемым результатам тестов.