МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

|  |  |  |
| --- | --- | --- |
| ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ: |  |  |

ПРЕПОДАВАТЕЛЬ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Е. В. Павлов |
| (должность, уч. степень, звание) |  | (подпись, дата) |  | (инициалы, фамилия) |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

«СОСТАВЛЕНИЕ ВАРИАНТОВ ТЕСТИРОВАНИЯ

И МЕТРИКА ПОКРЫТИЯ ТРЕБОВАНИЙ»

ПО КУРСУ: «МЕТРОЛОГИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

РАБОТУ ВЫПОЛНИЛ:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4831 |  |  |  |  |  | В.М. Денисенко |
|  |  |  | (подпись) |  | (дата отчёта) |  | (инициалы, фамилия) |

Санкт-Петербург 2020

1. Цель работы

Целью данной работы является изучение способов тестирования функций приложения и оценка плотности покрытия тестами функциональных требований.

1. Задание на лабораторную работу

Разработать консольное приложение в соответствии с индивидуальным вариантом задания и представленной концепцией продукта.

Составить тест кейсы для тестирования функций данного приложения и построить матрицу соответствия требований для анализа тестового покрытия.

Выполнить тестирование приложения по составленным тест кейсам.

Реализовать не менее 7-ми профильных функций приложения и составить как минимум 10-ть тест кейсов к данным функциям (тест кейсы должны осуществлять проверку сформулированных функциональных требований приложения).

|  |
| --- |
| Вариант задания:  120. Приложение для генерации ответов |

1. Спецификация требований к программному обеспечению

Данное приложение представляет собой адаптацию под сформулированную концепцию продукта и предназначено для того, чтобы генерировать случайные ответы.

Функциональные требования к приложению представлены в таблице 1 (каждому требованию соответствует уникальный идентификатор):

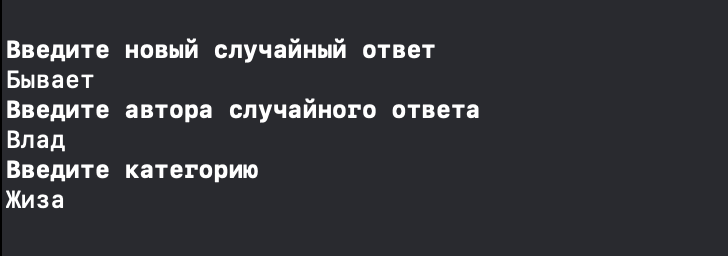
Таблица 1 — Функциональные требования

|  |  |
| --- | --- |
| FR-01 | Добавление нового ответа в базу данных |
| FR-02 | Генерация случайного ответа |
| FR-03 | Количество ответов в базе данных |
| FR-04 | Информация о приложении |
| FR-05 | Вывод всех ответов , которые находятся в базе данныз |
| FR-06 | Удаление ответа из базы данных |
| FR-07 | Просмотр ответов по выбранной категории |
| FR-08 | Изменение данных(ответ, автор, категория) |
| FR-09 | Завершение программы в штатном режими |
| FR-10 | Сохранение данных в файл формата txt. Имя и путь к файлу необходимо указать в качестве имени константы Href |

1. Тест дизайн и анализ тестового покрытия

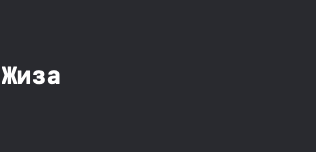
|  |  |  |
| --- | --- | --- |
| ID и наименование: | TC-1 Добавление нового ответа | |
| Ссылка на требование: | FR-01 | |
| Дата создания / Автор: | 17.03.2020 / Владислав | |
| Предусловия: отсутствует | | |
| Шаги теста: | | Ожидаемый результат: |
| 1.Выбрать в главном меню пункт:  “2-добавить новый ответ”  2 Заполнить предложенные поля | | 1.Приложение добавляет новый ответ в базу данных. |

Фактический результат:



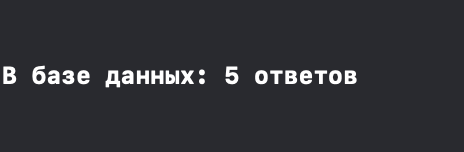
|  |  |  |
| --- | --- | --- |
| ID и наименование: | TC-2 Генерация случайного ответа | |
| Ссылка на требование: | FR-02 | |
| Дата создания / Автор: | 17.03.2020 / Владислав | |
| Предусловия: Наличие хотя бы одного отзыва в базе данных | | |
| Шаги теста: | | Ожидаемый результат: |
| 1.Выбрать в главном меню пункт:  “5-получить случайный ответ” | | 1.Приложение выводит случайный ответ |

Фактический результат:



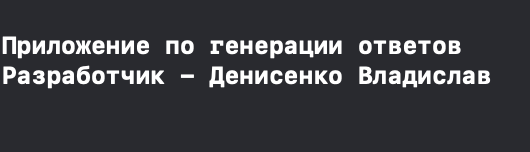
|  |  |  |
| --- | --- | --- |
| ID и наименование: | TC-3 Количество ответов в базе данных | |
| Ссылка на требование: | FR-03 | |
| Дата создания / Автор: | 17.03.2020 / Владислав | |
| Предусловия: отсутствует | | |
| Шаги теста: | | Ожидаемый результат: |
| 1.Выбрать в главном меню пункт:  “8-получить количество ответов в базе данных” | | 1.Приложение выводит количество ответов в базе данных |

Фактический результат:



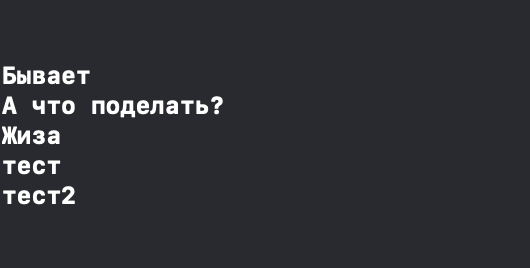
|  |  |  |
| --- | --- | --- |
| ID и наименование: | TC-4 Информация о приложении | |
| Ссылка на требование: | FR-04 | |
| Дата создания / Автор: | 17.03.2020 / Владислав | |
| Предусловия: отсутствует | | |
| Шаги теста: | | Ожидаемый результат: |
| 1.Выбрать в главном меню пункт:  “9-информация о программе” | | 1. Приложение выводит информацию о программе. |

Фактический результат:



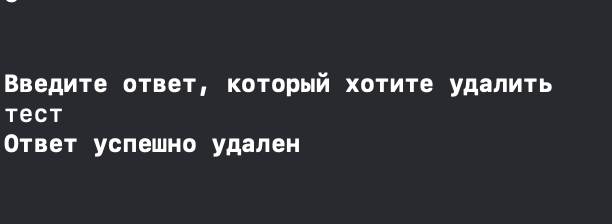
|  |  |  |
| --- | --- | --- |
| ID и наименование: | TC-5 Вывод всех ответов | |
| Ссылка на требование: | FR-05 | |
| Дата создания / Автор: | 17.03.2020 / Владислав | |
| Предусловия: наличие хотя бы одного ответа | | |
| Шаги теста: | | Ожидаемый результат: |
| 1.Выбрать в главном меню пункт:  “1-вывод списка всех ответов” | | 1. Приложение выводит все ответы, которые хранятся в базе данных. |

Фактический результат:



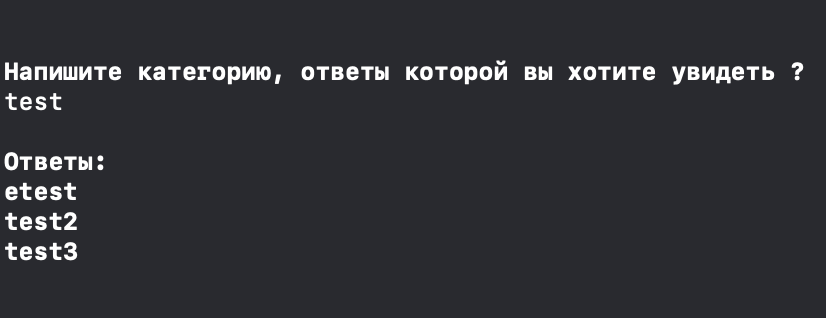
|  |  |  |
| --- | --- | --- |
| ID и наименование: | TC-6 Удаление ответа | |
| Ссылка на требование: | FR-06 | |
| Дата создания / Автор: | 17.03.2020 / Владислав | |
| Предусловия: наличие хотя бы одного ответа | | |
| Шаги теста: | | Ожидаемый результат: |
| 1.Выбрать в главном меню пункт:  “3-удалить выбранный ответ  2. Написать ответ, который хотим удалить | | 1. Приложение удаляет фотографию из файла. |

Фактический результат:



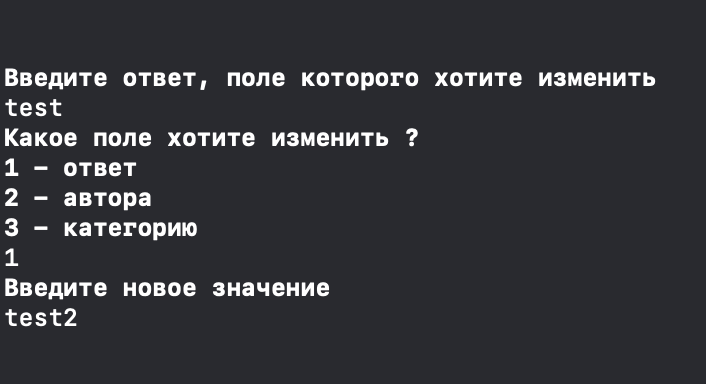
|  |  |  |
| --- | --- | --- |
| ID и наименование: | TC-7 Просмотр ответов по выбранной категории | |
| Ссылка на требование: | FR-07 | |
| Дата создания / Автор: | 17.03.2020 / Владислав | |
| Предусловия: наличие хотя бы одного ответа | | |
| Шаги теста: | | Ожидаемый результат: |
| 1.Выбрать в главном меню пункт:  “6-вывод всех ответов по заданной категории”  2. Написать категорию, ответы которой вы хотите увидеть | | 1. Вывод всех ответов заданной категории |

Фактический результат:



|  |  |  |
| --- | --- | --- |
| ID и наименование: | TC-8 Изменение данных | |
| Ссылка на требование: | FR-08 | |
| Дата создания / Автор: | 17.03.2020 / Владислав | |
| Предусловия: наличие хотя бы одного ответа | | |
| Шаги теста: | | Ожидаемый результат: |
| 1.Выбрать в главном меню пункт:  “4-изменить поле какого-то ответа”  2. Выбрать необходимое поле | | 1.Успешное изменение поля ответа |

Фактический результат:



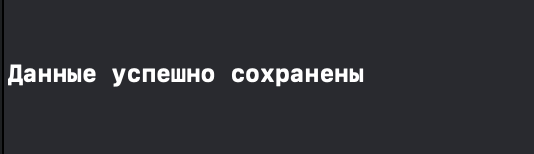
|  |  |  |
| --- | --- | --- |
| ID и наименование: | TC-9 Завершение программы | |
| Ссылка на требование: | FR-09 | |
| Дата создания / Автор: | 17.03.2020 / Владислав | |
| Предусловия: отсутствует | | |
| Шаги теста: | | Ожидаемый результат: |
| 1.Выбрать в главном меню пункт:  “10-выход” | | 1.Завершение программы |

Фактический результат:



|  |  |  |
| --- | --- | --- |
| ID и наименование: | TC-10 Сохранение данных | |
| Ссылка на требование: | FR-10 | |
| Дата создания / Автор: | 17.03.2020 / Владислав | |
| Предусловия: отсутствует | | |
| Шаги теста: | | Ожидаемый результат: |
| 1. Выбрать в главном меню пункт:  “7-сохранить данные в базе данных” | | 1.Данные успешно сохранены в базе данных |

Фактический результат:



Выполним анализ тестового покрытия при помощи построения матрицы соответствия требований:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Требование | FR-01 | FR-02 | FR-03 | FR-04 | FR-05 | FR-06 | FR-07 | FR-08 | FR-09 | FR-10 |
| Тест кейсы | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| TC-1 | ✓ |  |  |  |  |  |  |  |  |  |
| TC-2 |  | ✓ |  |  |  |  |  |  |  |  |
| TC-3 |  |  | ✓ |  |  |  |  |  |  |  |
| TC-4 |  |  |  | ✓ |  |  |  |  |  |  |
| TC-5 |  |  |  |  | ✓ |  |  |  |  |  |
| TC-6 |  |  |  |  |  | ✓ |  |  |  |  |
| TC-7 |  |  |  |  |  |  | ✓ |  |  |  |
| TC-8 |  |  |  |  |  |  |  | ✓ |  |  |
| TC-9 |  |  |  |  |  |  |  |  | ✓ |  |
| TC-10 |  |  |  |  |  |  |  |  |  | ✓ |

Таким образом, из данной матрицы следует, что для всех функциональных требований приложения есть хотя бы по одному тесту, соответственно тестовое покрытие функциональных требований в данном случае составляет 100%.

Выводы по работе

В результате выполнения данной работы было разработано консольное приложение для генерации ответов. Составлено необходимое количество тест кейсов, в соответствии с которыми выполнено тестирование приложения, и построена матрица соответствия требований для анализа тестового покрытия.

Анализ покрытия показывает, что составленные тесты покрывают 100% функциональных требований, однако нельзя говорить о том, что само приложение было полностью протестировано, так как выбранная метрика тестового покрытия работает только с требованиями и не учитывает конечную реализацию, соответственно вопрос о дефектах в коде остается открытым. В тоже время по результатам тестирования серьёзных расхождений со спецификацией выявлено не было.

В соответствии с предложенной концепцией продукта был сформулирован ряд нефункциональных требований, одним из которых является защита от очевидно ошибочных/неверных действий пользователя. Данное требование было реализовано лишь частично посредством следующих проверок:

* если встречается неизвестный символ, то приложение вылетает;
* поля ограничены длинной string;
* приложение проверяет наличие используемых файлов.

Очевидно, что представленный контроль данных для ввода не учитывает изобретательность пользователя и оставляет некоторые критические моменты без внимания, в частности можно отметить следующее:

* нет контроля дублирующей информации (можно добавить два или более одинаковых поля);
* нет защиты от редактирования данных в файлах вручную (приложение не проверяет целостность и корректность данных в файлах, соответственно при выводе информации может возникнуть ситуация, которую возможно интерпретировать как дефект);

В фактических результатах тестирования приведены только скриншоты реакции приложения через интерфейс пользователя, скриншоты изменения данных в файлах не приводятся в виду их малой информативности.

Таким образом, можно заключить, что выполненная работа соответствует поставленной задаче и отвечает всем сформулированным в методических указаниях требованиям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Виггерс, Карл. Разработка требований к программному обеспечению = Software Requirements: пер. с англ.; 3-е издание, дополненное / Карл Виггерс, Джой Битти — М.: Издательство «Русская редакция», 2014. — 736 с.: ил.
2. Куликов, С. С. Тестирование программного обеспечения. Базовый курс /  
   С. С. Куликов — Минск: Четыре четверти, 2017 — 312 с.
3. Traceability matrix [Электронный ресурс]: From Wikipedia, the free encyclopedia. — Wikipedia, 2020 — URL: <https://en.wikipedia.org/wiki/Traceability_matrix>   
   (дата обращения: 18.02.2020)
4. Работа с файлами в C++. Библиотека fstream [Электронный ресурс]. — PureCodeCpp, 2014-2020 — URL: <https://purecodecpp.com/archives/2751>  
   (дата обращения: 18.02.2020)

ПРИЛОЖЕНИЕ А

Исходный код программы

#include <iostream>

#include <string>

#include <vector>

#include <fstream>

#include <ctime>

using namespace std;

string const Href = "";

struct ansers{

string anser;

string auther;

string category;

};

class RandomAnser{

private:

vector<ansers> allAnser;

void addAnser();

void removeAnser();

void editAnser();

void randomAnser();

void printCategoryList();

void countAnser();

void printAllAnser();

void readData();

void info();

void saveData();

void menu();

public:

void start();

};

int main() {

RandomAnser randomAnser;

randomAnser.start();

return 0;

}

void RandomAnser::addAnser(){

ansers newElemet;

bool check = true;

cout << "Введите новый случайный ответ" << endl;

cin.ignore();

getline(cin, newElemet.anser);

for (int i=0;i<allAnser.size();i++){

if (newElemet.anser == allAnser[i].anser){

cout << "Такой ответ уже есть" << endl;

check = false;

}

}

if (check){

cout << "Введите автора случайного ответа" << endl;

getline(cin, newElemet.auther);

cout << "Введите категорию" << endl;

getline(cin, newElemet.category);

allAnser.push\_back(newElemet);

}

}

void RandomAnser::removeAnser(){

string remove;

bool check = true;

cout << "Введите ответ, который хотите удалить" << endl;

cin.ignore();

getline(cin, remove);

for (int i=0;i<allAnser.size();i++){

if (remove == allAnser[i].anser){

allAnser.erase(allAnser.begin()+i);

cout << "Ответ успешно удален" << endl;

check = false;

}

}

if (check){

cout << "Не удалось найти такой ответ" << endl;

}

}

void RandomAnser::editAnser(){

string remove;

bool check = false;

int value;

int index = -1;

cout << "Введите ответ, поле которого хотите изменить " << endl;

cin.ignore();

getline(cin, remove);

for (int i=0;i<allAnser.size();i++){

if (remove == allAnser[i].anser){

check = true;

index = i;

break;

}

}

if (check){

cout << "Какое поле хотите изменить ?" << endl;

cout << "1 - ответ" << endl;

cout << "2 - автора" << endl;

cout << "3 - категорию" << endl;

cin >> value;

if (value>0 && value<4){

string newValue;

cout << "Введите новое значение" << endl;

cin.ignore();

getline(cin, newValue);

switch (value) {

case 1:

allAnser[index].anser = newValue;

break;

case 2:

allAnser[index].auther = newValue;

break;

case 3:

allAnser[index].category = newValue;

break;

default:

cout << "error" << endl;

break;

}

}else{

cout << "error" << endl;

}

}else{

cout << "Такого ответа нету" << endl;

}

}

void RandomAnser::randomAnser(){

srand(unsigned(time(0)));

if (allAnser.size()>0){

int randomIndex = rand() % allAnser.size();

cout << allAnser[randomIndex].anser << endl;

}else{

cout << "error" << endl;

}

}

void RandomAnser::printCategoryList(){

cout << "Напишите категорию, ответы которой вы хотите увидеть ?" << endl;

string line;

cin.ignore();

getline(cin, line);

bool check = false;

cout << endl << "Ответы:" << endl;

for (int i=0;i<allAnser.size();i++){

if(allAnser[i].category == line){

cout << allAnser[i].anser << endl;

check = true;

}

}

if (!check){

cout << "По вашей категории ничего не найдено" << endl;

}

}

void RandomAnser::countAnser(){

cout << "В базе данных: " << allAnser.size() << " ответов" << endl;

}

void RandomAnser::printAllAnser(){

for (int i=0;i<allAnser.size();i++){

cout << allAnser[i].anser << endl;

}

}

void RandomAnser::info(){

cout << "Приложение по генерации ответов" << endl;

cout << "Разработчик - Денисенко Владислав" << endl;

}

void RandomAnser::readData(){

ifstream file(Href);

if (!file){

cout << "файл не найден" << endl;

}else{

string data;

while (getline(file,data)) {

ansers readData;

string line;

int count = 0;

for (int i=0;i<data.length();i++){

if (data[i]==' '){

switch (count) {

case 0:

readData.anser = line;

break;

case 1:

readData.auther = line;

break;

case 2:

readData.category = line;

allAnser.push\_back(readData);

break;

default:

break;

}

count++;

line = "";

}else{

line+= data[i];

}

}

}

file.close();

}

}

void RandomAnser::saveData(){

ofstream file(Href);

if (file.is\_open()){

for (int i=0;i<allAnser.size();i++){

string line = allAnser[i].anser + " " + allAnser[i].auther + " " + allAnser[i].category;

file << line << endl;

}

cout << "Данные успешно сохранены" << endl;

}else{

cout << "файл не найден" << endl;

}

}

void RandomAnser::menu(){

int choose = -1;

while (choose!=10) {

//system("cls");

cout << endl << endl;

cout << "Меню" << endl;

cout << "1 - вывод списка всех ответов" << endl;

cout << "2 - добавить новый ответ" << endl;

cout << "3 - удалить выбранный ответ" << endl;

cout << "4 - изменить поле какого-то ответа" << endl;

cout << "5 - получить случайный ответ" << endl;

cout << "6 - вывод всех ответов заданной категории" << endl;

cout << "7 - сохранить данные в базе данных" << endl;

cout << "8 - получить количество ответов в базе данных" << endl;

cout << "9 - информация о программе" << endl;

cout << "10 - выход" << endl;

cin >> choose;

cout << endl << endl;

//system("cls");

switch (choose) {

case 1:

printAllAnser();

break;

case 2:

addAnser();

break;

case 3:

removeAnser();

break;

case 4:

editAnser();

break;

case 5:

randomAnser();

break;

case 6:

printCategoryList();

break;

case 7:

saveData();

break;

case 8:

countAnser();

break;

case 9:

info();

break;

default:

break;

}

}

}

void RandomAnser::start(){

readData();

menu();

}

ПРИЛОЖЕНИЕ Б

test test test

test2 test test

test3 test test

test4 test test

test5 test test

test6 test test