Министерство образования и науки Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ПетрГУ)
Институт математики и информационных технологий

Отчет о тестировании приложения «Список покупок»

Выполнил:
Вершинин И.В.
22608

Петрозаводск
2018

Объект тестирования

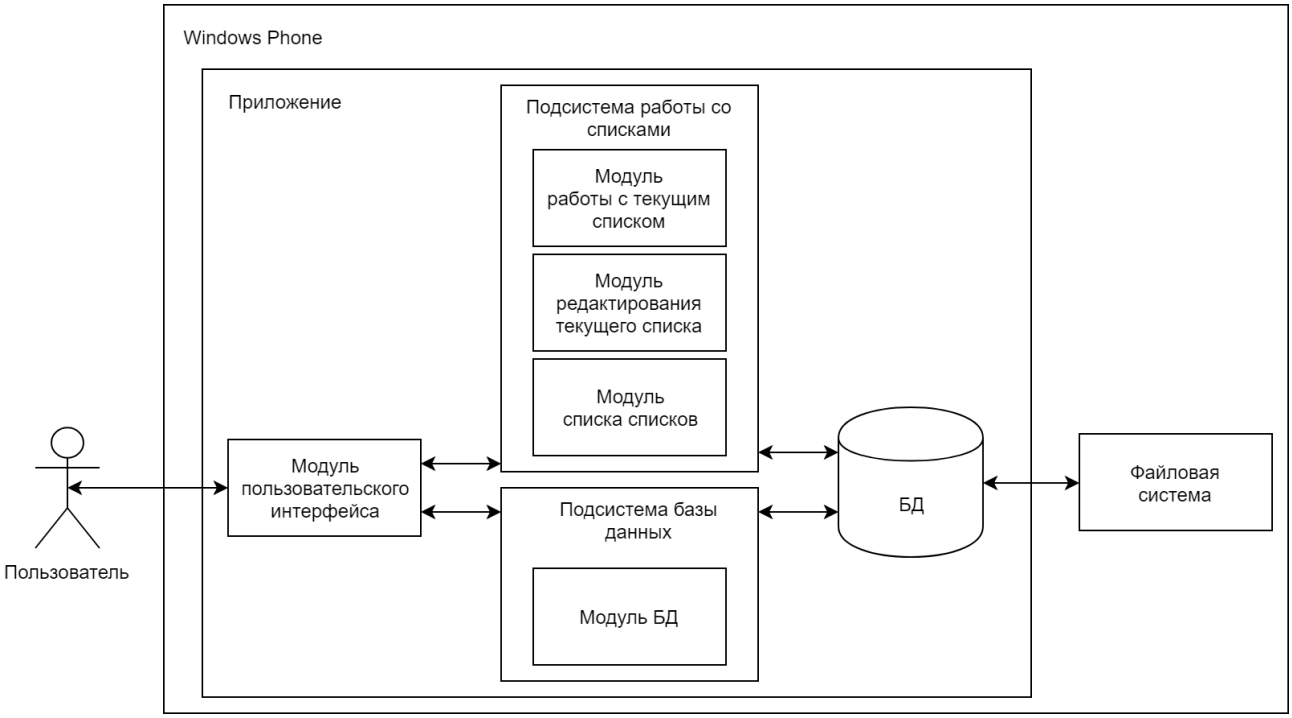
Объектом тестирования выбрано приложение “Список покупок” для платформы Windows Phone, написанное на языке программирования C#. Приложение представляет собой имитатор списка покупок, то есть, используя данное приложение, пользователь вводит в него информацию о том, что он хочет приобрести в магазине и, уже в магазине, вычеркивает или помечает всё, что купил. Приложение сохраняет данные, то есть покупки, которые уже были введены, вследствие чего пользователю необязательно вводить их вновь, если он идет в магазин за теми же позициями. Если пользователь каждую неделю покупает что-то с очень длинным названием, он может выбрать свою покупку из хранимой базы данных продуктов, что освобождает его от необходимости ввода слова целиком.

Перечень функциональностей объекта тестирования

1. Просмотр списка покупок;
2. просмотр списка списков;
3. редактирование списка списков:
    1. добавление нового списка;
    2. удаление более ненужного списка;
    3. изменение названия списка;
4. редактирование списка:
    1. добавление позиции в список;
    2. удаление некоторых позиций по одной;
    3. изменение наименования позиции в списке;
5. возможность отметить приобретенную позицию;
6. хранение базы наименований, ранее введенных пользователем;
7. возможность редактировать базу сохраненных позиций:
    1. удаление позиции из базы данных;
    2. добавление позиции в базу данных вручную.

Стратегия тестирования

Архитектура объекта тестирования



Запуская приложение, пользователь взаимодействует с графическим интерфейсом. Графический интерфейс взаимодействует с подсистемой списков, которая в свою очередь разбита на три модуля: модуль работы с текущим списком, модуль редактирования текущего списка, модуль списка списков. Также интерфейс взаимодействует с подсистемой базы данных, которая содержит модуль БД (отвечает за редактирование базы данных).

Обе подсистемы имеют доступ к базе данных, хранящейся в файловой системе устройства.

Описание модулей системы

Модуль БД (ProdBase)
Модуль реализует работу с базой наименований, в частности, с ее сохранением в память устройства, работу алгоритмов для работы с ней, ее извлечение из памяти устройства.

1. SaveBase(ProdBase *mem) - сохраняет базу в виде документа XML в памяти устройства
    - *mem - любой элемент класса ProdBase с заполненными ссылками
2. ExtractBase(ProdBase *mem) - извлекает базу из документа XML в памяти устройства и создает очередь такой, чтобы была возможность работы с ней, используя конструктор; Если документ XML отсутствует, то он создается приложением в памяти устройства, а очередь создается с пустыми полями.
    - *mem - любой член класса ProdBase
3. last(ProdBase *mem) - возвращает последний элемент очереди
    - *mem - любой член класса ProdBase с заполненными ссылками next и pre

4. first(ProdBase *mem) - возвращает первый элемент очереди
    - *mem - любой член класса ProdBase с заполненными ссылками next и pre
5. fully(ProdBase *mem) - проверяет очередь на заполненность; возвращает 0, если не заполнена, 1, если заполнена
    - *mem - любой член класса ProdBase с заполненными ссылками next и pre
6. noput(ProdBase *mem, string word) - проверяет очередь на наличие соответствующего имени; возвращает 0 - наименование найдено, 1 - наименование не найдено и очередь не заполнена и 2 - наименование не найдено и очередь заполнена
    - *mem - любой член класса ProdBase с заполненными ссылками next и pre
    - word - наименование, наличие которого функция и проверяет
7. autoput(ProdBase *mem, string word) - автоматически вводит в базу наименование при создании списка; если список заполнен, то последний элемент списка удаляется, новое наименование вводится в начало списка, иначе новый элемент вводится в начало списка
    - *mem - любой член класса ProdBase с заполненными ссылками next и pre
    - word содержит наименование, которое вводит функция
8. putter(ProdBase *mem, string word) - вводит наименование в базу, если база не заполнена; если же заполнена, то функция выводит на экран сообщение “База заполнена”; если наименование уже есть в базе, то выводится сообщение “Наименование уже есть в базе”
    - *mem - любой член класса ProdBase с заполненными ссылками next и pre
    - word - наименование, которое функция вводит в начало списка.
9. delete(ProdBase *mem, name) - удаляет наименование из базы
    - *mem - член класса ProdBase
    - name - наименование, что необходимо удалить.
10. message(string word) - вызывает всплывающее окно с сообщением
    - word – сообщение

Модуль работы с текущим списком (CurrentList)
Модуль реализует работу с текущим списком, в частности, реализуется инициализация текущего списка, действия при отметке той или иной позиции в списке.

1. showList(String listName) - выводит все элементы списка на экран
    - listName - название текущего списка
2. getSize(String listName) - получает количество элементов списка
    - listName - название текущего списка
3. counter() - считает количество отмеченных позиций

Модуль редактирования текущего списка (CurrentListEdit)
Модуль реализует редактирование текущего списка, в частности, реализуется изменение названия позиции в списке, удаление и/или добавление позиций в список.

1. showListEdit(String listName) - выводит все элементы списка на экран в виде для редактирования
    - listName - название списка для редактирования
2. addElement(String listName) - добавляет элемент в список
    - listName - название списка для редактирования
3. removeElement(String listName, int position) - удаляет элемент из списка
    - listName - название списка для редактирования
    - param position - номер позиции, подлежащей удалению
4. saveList(String listName) - сохраняет изменения

- listName - название списка для редактирования

Модуль списка списков (ListOfLists)
Модуль реализует работу со всеми списками, в частности, реализуется проверка на наличие списков, создание списка по умолчанию при отсутствии сохраненных списков, изменение названия списка и удаление списков по одному.

1. checkFile() - проверяет наличие файла со всеми списками; если его не существует, то создает его
2. showLists() - выводит на экран названия всех списков, хранящихся в файле
3. createList() - создание пустого списка
4. changeListName(String listName) - меняет название списка
    - listName - название списка
5. deleteList(String listName) - удаляет список
    - listName - название списка, подлежащего удалению

Стратегия блочного тестирования

Тестированию подлежат следующие методы:

- SaveBase
- ExtractBase
- last
- first
- fully
- noput
- putter
- delete
- message
- getSize
- checkFile

Методы будут тестироваться с помощью выделения отдельных строк кода с использованием специальных тестовых наборов данных.

Стратегия интеграционного тестирования

Интеграционное тестирование будет проведено для следующих взаимодействий:

- first → fully → noput
- first → last → noput → autoput
- first → noput → putter

Методы будут тестироваться с помощью выделения отдельных строк кода с использованием специальных тестовых наборов данных.

Стратегия специального тестирования

В виде специального тестирования будет использовано нагрузочное тестирование на базу данных приложения в следующих методах:

- SaveBase

- fully
- noput
- autoput
- putter
- addElement
- createList

Для имитации недостатка места на устройстве будет использоваться эмулятор платформы Windows Phone с соответствующими настройками.

Стратегия аттестационного тестирования

Тестирование будет проводиться с помощью эмулятора платформы Windows Phone в соответствие с функциональными требованиями к приложению.

Во всех тестах подразумевается следующее:

- текущий список не выбран;
- список списков заполнен.

Критерий прохождения тестирования

Тест считается успешно пройденным, если ожидаемый результат совпадает с фактическим. Если тест завершился неудачей, то производится заключение о найденной ошибке. Тестирование считается пройденным, если во время его прохождения не выявлено критических ошибок, влияющих на функциональность приложения, а процент не пройденных тестов меньше 3% от общего количества.

Критерий приостановки тестирования

Тестирование должно быть приостановлено, если количество не пройденных тестов превысит 10% от их общего количества, а также при обнаружении критических ошибок, влияющих на функциональность приложения.

Критерий возобновления тестирования

Тестирование возобновляется после исправления ошибок, выявленных при предыдущем цикле тестирования.

Детальный план тестов

План блочного тестирования

Тест B-1 (Модуль ProdBase, метод SaveBase)
Тип теста: общий
Описание: проверяет сохранение базы наименований в файле ProdBase.txt в памяти устройства
Входные данные: тестовый набор данных 1, tester[30]; наличие файла ProdBase.txt
Ожидаемый результат: перезаписывает файл ProdBase.txt значениями, которые находятся в полях tester[i].name от i = 0 до 30; в файле ProdBase.txt будет “i хлеб” + ‘\n’ от i=0 до 30

Тест B-2 (Модуль ProdBase, метод SaveBase)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: NULL
Ожидаемый результат: срабатывает NullException, приложение завершает свою работу

Тест B-3 (Модуль ProdBase, метод ExtractBase)
Тип теста: общий
Описание: проверяет извлечение базы наименований из памяти устройства при наличии файла ProdBase.txt, заполненного строками “i + хлеб \n” при i от 0 до 30
Входные данные: тестовый набор данных 1, mem1; наличие файла ProdBase.txt
Ожидаемый результат: memi.name = “i + хлеб \n” при i от 0 до 30

Тест B-4 (Модуль ProdBase, метод ExtractBase)
Тип теста: общий
Описание: проверяет извлечение базы наименований из памяти устройства при отсутствии файла ProdBase.txt (в следствии первого запуска или ручного удаления)
Входные данные: тестовый набор данных 1, mem1; отсутствие файла ProdBase.txt
Ожидаемый результат: memi.name = “” при i от 0 до 30; на устройстве создается пустой файл ProdBase.txt

Тест B-5 (Модуль ProdBase, метод ExtractBase)
Тип теста: негативный
Описание: проверяет работу исключения NoDataException, когда файл ProdBase.txt есть на устройстве, но открыть файл для чтения нельзя
Входные данные: тестовый набор данных 1, tester[30]; файл ProdBase.txt недоступен для чтения
Ожидаемый результат: срабатывает NoDataException и приложение завершает свою работу

Тест B-6 (Модуль ProdBase, метод ExtractBase)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: NULL

Ожидаемый результат: срабатывает NullException и приложение завершает свою работу

Тест B-7 (Модуль ProdBase, метод last)
Тип теста: общий
Описание: проверяет возвращение последнего элемента класса ProdBase (тот у которого поле next = NULL)
Входные данные: тестовый набор данных 1, mem1
Ожидаемый результат: возвращает mem30

Тест B-8 (Модуль ProdBase, метод last)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: NULL
Ожидаемый результат: срабатывает NullException и приложение завершает свою работу

Тест B-9 (Модуль ProdBase, метод first)
Тип теста: общий
Описание: проверяет возвращение первого элемента класса ProdBase (тот у которого поле pre = NULL)
Входные данные: тестовый набор данных 1, mem10
Ожидаемый результат: возвращает mem1

Тест B-10 (Модуль ProdBase, метод first)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: NULL
Ожидаемый результат: срабатывает NullException и приложение завершает свою работу

Тест B-11 (Модуль ProdBase, метод fully)
Тип теста: общий
Описание: проверяет, что база наименований не заполнена (количество наименований <30)
Входные данные: тестовый набор данных 2, mem1
Ожидаемый результат: при выводе значения, которое вернул метод, на экран, получаем 0, что свидетельствует о том, что база наименований не заполнена

Тест B-12 (Модуль ProdBase, метод fully)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: NULL
Ожидаемый результат: срабатывает NullException и приложение завершает свою работу

Тест B-13 (Модуль ProdBase, метод noput)
Тип теста: общий
Описание: проверяет наличие строки в базе наименований
Входные данные: тестовый набор данных 1, mem1; строка “ 5 хлеба”

Ожидаемый результат: при выводе значения, которое вернул метод, на экран, получаем 0, что свидетельствует о том, что строка есть в базе наименований

Тест B-14 (Модуль ProdBase, метод noput)
Тип теста: общий
Описание: проверяет отсутствие строки в базе наименований
Входные данные: тестовый набор данных 2, mem1; строка “ нет хлеба”
Ожидаемый результат: при выводе значения, которое вернул метод, на экран, получаем 1, что свидетельствует о том, что строки нету в базе наименований

Тест B-15 (Модуль ProdBase, метод noput)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: NULL, строка “хлеб”
Ожидаемый результат: срабатывает NullException и приложение завершает свою работу

Тест B-16 (Модуль ProdBase, метод noput)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: тестовый набор данных 1, tester[30], NULL
Ожидаемый результат: срабатывает NullException и приложение завершает свою работу

Тест B-17 (Модуль ProdBase, метод autoput)
Тип теста: общий
Описание: проверяет ввод наименования в базу наименований в начало базы, когда наименования в базе нет и база содержит <30 наименований
Входные данные: тестовый набор данных 2, mem1; строка “нет хлеба”
Ожидаемый результат: mem29.name = “ нет хлеба”. mem29.pre = NULL; mem29.next = mem1; memi.next = memi+1 i от 1 до 28. mem28.next = mem30
memi.pre = memi-1 i от 2 до 28. mem1.pre =mem29 mem30.pre = mem28

Тест B-18 (Модуль ProdBase, метод autoput)
Тип теста: общий
Описание: проверяет перенаправление наименования в начало базы наименований, когда наименования в базе есть
Входные данные: тестовый набор данных 1, mem10; строка “ 29 хлеба”
Ожидаемый результат: mem29.name = “ 29 хлеба”. mem29.pre = NULL; mem29.next = mem1; memi.next = memi+1 i от 1 до 27. mem28.next = mem30
memi.pre = memi-1 i от 2 до 28. mem1.pre = mem29 mem30.pre = mem28

Тест B-19 (Модуль ProdBase, метод autoput)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: NULL, строка “хлеб”
Ожидаемый результат: срабатывает NullException и приложение завершает свою работу

Тест B-20 (Модуль ProdBase, метод autoput)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: тестовый набор данных 1, mem10; NULL
Ожидаемый результат: срабатывает NullException и приложение завершает свою работу

Тест B-21 (Модуль ProdBase, метод putter)
Тип теста: общий
Описание: проверяет ввод наименования в базу наименований, когда база не заполнена и аналогичных наименований в базе нет
Входные данные: тестовый набор данных 2, mem1; строка “ нет хлеба”
Ожидаемый результат: mem29.name = “нет хлеба”. mem29.pre = NULL; mem29.next = mem1; memi.next = memi+1 i от 1 до 28. mem28.next = mem30
memi.pre = memi-1 i от 2 до 28. mem1.pre = mem29 mem30.pre = mem28

Тест B-22 (Модуль ProdBase, метод putter)
Тип теста: общий
Описание: проверяет вывод сообщения о том, что наименование в базе уже есть, если база не заполнена, но обнаружено совпадение строки-аргумента с другой в базе наименований при попытке добавить в неё новое наименование
Входные данные: тестовый набор данных 2, mem1; строка “3 хлеба”
Ожидаемый результат: на экран выводится сообщение “Наименование уже есть в базе”

Тест B-23 (Модуль ProdBase, метод putter)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: NULL, строка “хлеб”
Ожидаемый результат: срабатывает NullException и приложение завершает свою работу

Тест B-24 (Модуль ProdBase, метод putter)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: тестовый набор данных 1, tester[30]; NULL
Ожидаемый результат: срабатывает NullException и приложение завершает свою работу

Тест B-25 (Модуль ProdBase, метод delete)
Тип теста: общий
Описание: проверяет удаление наименования из базы наименований
Входные данные: тестовый набор данных 2, mem1; строка “3 хлеба”
Ожидаемый результат: строки “3 хлеба” больше нет в базе наименований

Тест B-26 (Модуль ProdBase, метод message)
Тип теста: общий
Описание: выводит сообщение на экран
Входные данные: строка “привет”
Ожидаемый результат: строка “привет” отобразится на экране

Тест B-27 (Модуль CurrentList, метод getSize)
Тип теста: общий
Описание: проверяет корректность возвращаемого значения количества элементов списка
Входные данные: строка “ужин”
Ожидаемый результат: метод возвращает число, равное количеству элементов в списке – 30

Тест B-28 (Модуль CurrentList, метод getSize)
Тип теста: негативный
Описание: проверяет работу исключения NullException
Входные данные: NULL
Ожидаемый результат: вызывается всплывающее окно с текстом “Ошибка открытия списка”

Тест B-29 (Модуль ListOfLists, метод checkFile)
Тип теста: общий
Описание: проверяет создание файла Lists.txt, если он отсутствует на устройстве
Входные данные: отсутствие файла Lists.txt
Ожидаемый результат: создается файл Lists.txt

План интеграционного тестирования

Тест I-1 (Взаимодействие first → fully →noput)
Тип теста: общий
Описание: проверяет есть ли совпадение, начиная с первой строки, и если нет, то смотрит, заполнена ли база наименований
Входные данные: тестовый набор данных 1, tester[30], строка “0 хлеба”
Ожидаемый результат: метод вернёт 0, что свидетельствует о том, что строка есть в базе наименований

Тест I-2 (Взаимодействие first → fully → noput)
Тип теста: общий
Описание: проверяет есть ли совпадение, начиная с первой строки, и если нет, то смотрит, заполнена ли база наименований
Входные данные: тестовый набор данных 2, tester[30]; строка “ нет хлеба ”
Ожидаемый результат: метод вернёт 1, что свидетельствует о том, что строки нет в базе наименований

Тест I-3 (Взаимодействие first → fully → noput)
Тип теста: общий
Описание: проверяет есть ли совпадение, начиная с первой строки, и если нет, то смотрит, заполнена ли база наименований
Входные данные: тестовый набор данных 1, tester[30]; строка “ нет хлеба”

Ожидаемый результат: метод вернёт 2, что свидетельствует о том, что строки нет в базе наименований и в базе хранится 30 наименований

Тест I-4 (Взаимодействие first → last → noput → autoput)
Тип теста: общий
Описание: проверяет есть ли совпадение и заполнена ли база, затем, если совпадение есть, берёт первый элемент и смещает его относительно совпадения
Входные данные: тестовый набор данных 1, tester[30]; строка “ 28 хлеба”
Ожидаемый результат: tester[28].name = “28 хлеба”. tester[28].pre = NULL; tester[28].next = tester[0]; tester[i].next = tester[i+1] i от 0 до 26. tester[27].next = tester[29]
tester[i].pre = tester[i-1] i от 1 до 27. tester[0].pre = tester[28] tester[29].pre = tester[27]

Тест I-5 (Взаимодействие first → last → noput → autoput)
Тип теста: общий
Описание: проверяет есть ли совпадение и заполнена ли база, затем, если совпадений нет и база не заполнена, берёт первый элемент и ищет следующий за ним пустой элемент
Входные данные: тестовый набор данных 2, tester[30]; строка “нет хлеба”
Ожидаемый результат: tester[28].name = “нет хлеба”. tester[28].pre = NULL; tester[28].next = tester[0]; tester[i].next = tester[i+1] i от 0 до 26. tester[27].next = tester[29]
tester[i].pre = tester[i-1] i от 1 до 27. tester[0].pre = tester[28] tester[29].pre = tester[27]

Тест I-6 (Взаимодействие first → last → noput → autoput)
Тип теста: общий
Описание: проверяет есть ли совпадение и заполнена ли база, затем, если совпадений нет и база заполнена, берёт первый элемент и сдвигает его соответственно последнего
Входные данные: тестовый набор данных 1, tester[30]; строка “нет хлеба”
Ожидаемый результат: tester[29].name = “нет хлеба”. tester[29].pre = NULL; tester[29].next = tester[0]; tester[i].next = tester[i+1] i от 0 до 27. tester[28].next = NULL
tester[i].pre = tester[i-1] i от 1 до 28. tester[0].pre = tester[29]

Тест I-7 (Взаимодействие first → last → noput → autoput)
Тип теста: общий
Описание: проверяет есть ли совпадение и заполнена ли база, затем, если совпадений нет и база заполнена, берёт последний элемент, перезаписывает его значение и переносит его на первое по списку место
Входные данные: тестовый набор данных 1, tester[30]; строка “нет хлеба”
Ожидаемый результат: tester[29].name = “нет хлеба”. tester[29].pre = NULL; tester[29].next = tester[0]; tester[i].next = tester[i+1] i от 0 до 27. tester[28].next = NULL
tester[i].pre = tester[i-1] i от 1 до 28. tester[0].pre = tester[29]

Тест I-8 (Взаимодействие first → last → noput → autoput)
Тип теста: общий
Описание: проверяет есть ли совпадение и заполнена ли база, затем, если совпадение есть, переносит его на первое место списка
Входные данные: тестовый набор данных 1, tester[30]; строка “28 хлеба”

Ожидаемый результат: tester[28].name = “ 28 хлеба”. tester[28].pre = NULL; tester[28].next = tester[0]; tester[i].next = tester[i+1] i от 0 до 26. tester[27].next = tester[29]
tester[i].pre = tester[i-1] i от 1 до 27. tester[0].pre = tester[28] tester[29].pre = tester[27]

Тест I-9 (Взаимодействие first → last → noput → autoput)
Тип теста: общий
Описание: проверяет есть ли совпадение и заполнена ли база, затем, если совпадений нет и база не заполнена, берёт первый элемент и ищет следующий за ним пустой элемент и переносит его на первое место списка
Входные данные: тестовый набор данных 2, tester[30]; строка “нет хлеба”
Ожидаемый результат: tester[28].name = “нет хлеба”. tester[28].pre = NULL; tester[28].next = tester[0]; tester[i].next = tester[i+1] i от 0 до 26. tester[27].next = tester[29]
tester[i].pre = tester[i-1] i от 1 до 27. tester[0].pre = tester[28] tester[29].pre = tester[27]

Тест I-10 (Взаимодействие first → last → noput → autoput)
Тип теста: общий
Описание: проверяет есть ли совпадение и заполнена ли база, затем, если совпадений нет и база заполнена, берёт последний элемент, меняет его значение и переносит на первое место в списке
Входные данные: тестовый набор данных 1, tester[30]; строка “нет хлеба”
Ожидаемый результат: tester[29].name = “нет хлеба”. tester[29].pre = NULL; tester[29].next = tester[0]; tester[i].next = tester[i+1] i от 0 до 27. tester[28].next = NULL
tester[i].pre = tester[i-1] i от 1 до 28. tester[0].pre = tester[29]

Тест I-11 (Взаимодействие first → noput → putter)
Тип теста: общий
Описание: проверяет есть ли совпадение и заполнена ли база, затем, если совпадений нет и база не заполнена, берёт первый элемент и ищет следующий за ним пустой элемент
Входные данные: тестовый набор данных 2, tester[30]; строка “нет хлеба”
Ожидаемый результат: tester[28].name = “нет хлеба”. tester[28].pre = NULL; tester[28].next = tester[0]; tester[i].next = tester[i+1] i от 0 до 27. tester[27].next = tester[29]
tester[i].pre = tester[i-1] i от 1 до 27. tester[0].pre = tester[28] tester[29].pre = tester[27]

Тест I-12 (Взаимодействие first → noput → putter)
Тип теста: общий
Описание: проверяет есть ли совпадение и заполнена ли база, затем, если совпадений нет и база не заполнена, берёт пустой элемент, заносит в него строку word и переносит его на первое место
Входные данные: тестовый набор данных 2, tester[30]; строка “нет хлеба”
Ожидаемый результат: tester[28].name = “нет хлеба”. tester[28].pre = NULL; tester[28].next = tester[0]; tester[i].next = tester[i+1] i от 0 до 27. tester[27].next = tester[29]
tester[i].pre = tester[i-1] i от 1 до 27. tester[0].pre = tester[28] tester[29].pre = tester[27]

Тест I-13 (Взаимодействие first → noput → putter)
Тип теста: общий

Описание: проверяет есть ли совпадение и заполнена ли база, затем, если совпадение есть, выводит сообщение о том, что совпадение есть
Входные данные: тестовый набор данных 2, tester[30]; строка “5 хлеба”
Ожидаемый результат: на экран выводится строка “Такое наименование в базе уже есть!”

Тест I-14 (Взаимодействие first → noput → putter)
Тип теста: общий
Описание: проверяет есть ли совпадение и заполнена ли база, затем, если база заполнена, выводит сообщение об этом
Входные данные: тестовый набор данных 1, tester[30]; Строка “нет хлеба”
Ожидаемый результат: на экран выводится строка “База заполнена”

План специального тестирования

Тест S-1 (Модуль ProdBase, метод SaveBase)
Тип теста: специальный
Описание: проверяет работу исключения NoPlaceException при вызове метода и отсутствии места в памяти устройства
Входные данные: тестовый набор данных 1, tester[30]; заполненная память устройства
Ожидаемый результат: срабатывает NoPlaceException, приложение завершает свою работу

Тест S-2 (Модуль ProdBase, метод fully)
Тип теста: специальный
Описание: проверяет, что база наименований заполнена (количество наименований = 30)
Входные данные: тестовый набор данных 1, mem1
Ожидаемый результат: при выводе значения, которое вернул метод, на экран, получаем 1, что свидетельствует о том, что база наименований заполнена

Тест S-3 (Модуль ProdBase, метод noput)
Тип теста: специальный
Описание: проверяет, что аналогичной строки нет в базе наименований, когда в базе наименований хранится 30 наименований
Входные данные: тестовый набор данных 1, mem1; строка “ нет хлеба”
Ожидаемый результат: при выводе значения, которое вернул метод, на экран, получаем 2, что свидетельствует о том, что строки нету в базе наименований и в базе хранится 30 наименований.

Тест S-4 (Модуль ProdBase, метод autoput)
Тип теста: специальный
Описание: проверяет ввод наименования в начало базы наименований, когда наименования в базе нет и база содержит 30 наименований; последний элемент базы удаляется
Входные данные: тестовый набор данных 1, mem10; строка “нет хлеба”
Ожидаемый результат: mem30.name = “нет хлеба”. mem30.pre = NULL; mem30.next = mem1; memi.next =memi+1 i от 1 до 28. mem29.next = NULL
memi.pre = memi-1 i от 2 до 29. mem1.pre = mem30

Тест S-5 (Модуль ProdBase, метод putter)
Тип теста: специальный
Описание: проверяет вывод сообщения о том, что база заполнена, если база заполнена, при попытке добавить в неё новое наименование
Входные данные: тестовый набор данных 1, mem1; строка “ нет хлеба”
Ожидаемый результат: на экран выводится сообщение “База заполнена”

Тест S-6 (Модуль CurrentListEdit, метод addElement)
Тип теста: специальный
Описание: проверяет работу исключения SizeException при попытке добавления элемента в заполненный список
Входные данные: строка “хлеб”; заполненный список
Ожидаемый результат: срабатывает исключение SizeException; на экран выводится сообщение о том, что список заполнен

Тест S-7 (Модуль ListOfLists, метод createList)
Тип теста: специальный
Описание: проверяет работу исключения SizeException при попытке создать список, когда список списков заполнен
Входные данные: строка “ужин”; заполненный список список
Ожидаемый результат: срабатывает исключение SizeException; на экран выводится сообщение о том, что список списков заполнен

План аттестационного тестирования

Тест А-1
Тип теста: общий
Описание: проверяет возможность просматривать список покупок
Входные данные: запустить приложение, выбрать список из списка списков
Ожидаемый результат: приложение предоставит пользователю экран текущего списка

Тест А-2
Тип теста: общий
Описание: проверяет возможность просматривать список списков.
Входные данные: запустить приложение.
Ожидаемый результат: приложение предоставит пользователю экран списка списков.

Тест А-3
Тип теста: общий
Описание: проверяет возможность добавить новый список
Входные данные: запустить приложение, создать новый с помощью нажатия соответствующей кнопки на экране списка списков, ввести все необходимые данные и сохранить список
Ожидаемый результат: новый список будет отображаться на экране списка списков

Тест А-4
Тип теста: общий
Описание: проверяет возможность удалить более не нужный список
Входные данные: запустить приложение, нажать соответствующую кнопку напротив необходимого списка, удалить список с помощью соответствующей кнопки на экране редактирования списка
Ожидаемый результат: удаленный список больше не будет отображаться на экране списка списков

Тест А-5
Тип теста: общий
Описание: проверяет возможность изменить название списка
Входные данные: запустить приложение, нажать соответствующую кнопку напротив необходимого списка, изменить название списка, сохранить все изменения
Ожидаемый результат: изменится название списка

Тест А-6
Тип теста: общий
Описание: проверяет возможность добавить позицию в список
Входные данные: запустить приложение, нажать соответствующую кнопку напротив необходимого списка, на экране редактирования списка добавить новую позицию, сохранить изменения
Ожидаемый результат: новая позиция будет отображаться на экране текущего списка, если отредактированный список выбран как текущий

Тест А-7
Тип теста: общий
Описание: проверяет возможность удалить некоторые позиции по одной
Входные данные: запустить приложение, нажать соответствующую кнопку напротив необходимого списка, на экране редактирования списка удалить необходимую позицию, сохранить изменения
Ожидаемый результат: удаленная позиция не будет отображаться на экране текущего списка, если отредактированный список выбран как текущий

Тест А-8
Тип теста: общий
Описание: проверяет возможность отметить приобретенную позицию.
Входные данные: запустить приложение, выбрать текущий список из списка списков, поставить галочку напротив необходимой позиции
Ожидаемый результат: на экране текущего списка напротив отмеченной позиции будет отображаться галочка, если данный список выбран как текущий.

Тест А-9
Тип теста: общий
Описание: проверяет возможность удалять позиции из базы данных

Входные данные: запустить приложение, перейти на экран базы данных, на экране базы данных удалить необходимые позиции с помощью соответствующей кнопки, сохранить изменения
Ожидаемый результат: на экране базы данных не отображаются удаленные позиции

Тест А-10
Тип теста: общий
Описание: проверяет возможность добавлять позиции в базу данных вручную
Входные данные: запустить приложение, перейти на экран базы данных, на экране базы данных добавить необходимые позиции с помощью соответствующей кнопки, сохранить изменения.
Ожидаемый результат: на экране базы данных отображаются добавленные позиции

Пример реализации теста

В качестве примера рассмотрим реализацию блочного теста В-1.

Тестовый набор данных имеет следующий вид:

```
ProdBase mem1 = new ProdBase();
ProdBase mem2 = new ProdBase();
ProdBase mem3 = new ProdBase();
ProdBase mem4 = new ProdBase();
ProdBase mem5 = new ProdBase();
ProdBase mem6 = new ProdBase();
ProdBase mem7 = new ProdBase();
ProdBase mem8 = new ProdBase();
ProdBase mem9 = new ProdBase();
ProdBase mem10 = new ProdBase();
ProdBase mem11 = new ProdBase();
ProdBase mem12 = new ProdBase();
ProdBase mem13 = new ProdBase();
ProdBase mem14 = new ProdBase();
ProdBase mem15 = new ProdBase();
ProdBase mem16 = new ProdBase();
ProdBase mem17 = new ProdBase();
ProdBase mem18 = new ProdBase();
ProdBase mem19 = new ProdBase();
ProdBase mem20 = new ProdBase();
ProdBase mem21 = new ProdBase();
ProdBase mem22 = new ProdBase();
ProdBase mem23 = new ProdBase();
ProdBase mem24 = new ProdBase();
ProdBase mem25 = new ProdBase();
ProdBase mem26 = new ProdBase();
ProdBase mem27 = new ProdBase();
ProdBase mem28 = new ProdBase();
ProdBase mem29 = new ProdBase();
ProdBase mem30 = new ProdBase();

        mem1.name = “1 хлеб”;
        mem2.name = “2 хлеб”;
        mem3.name = “3 хлеб”;
        mem4.name = “4 хлеб”;
        mem5.name = “5 хлеб”;
        mem6.name = “6 хлеб”;
```

```
mem7.name = “7 хлеб”;
mem8.name = “8 хлеб”;
mem9.name = “9 хлеб”;
mem10.name = “10 хлеб”;
mem11.name = “11 хлеб”;
mem12.name = “12 хлеб”;
mem13.name = “13 хлеб”;
mem14.name = “14 хлеб”;
mem15.name = “15 хлеб”;
mem16.name = “16 хлеб”;
mem17.name = “17 хлеб”;
mem18.name = “18 хлеб”;
mem19.name = “19 хлеб”;
mem20.name = “20 хлеб”;
mem21.name = “21 хлеб”;
mem22.name = “22 хлеб”;
mem23.name = “23 хлеб”;
mem24.name = “24 хлеб”;
mem25.name = “25 хлеб”;
mem26.name = “26 хлеб”;
mem27.name = “27 хлеб”;
mem28.name = “28 хлеб”;
mem29.name = “29 хлеб”;
mem30.name = “30 хлеб”;
```

Метод SaveBase имеет следующий вид:

```
public async void SaveBase()
{
	BaseFile = await localFolder.CreateFileAsync("ProdBase.txt",
	CreationCollisionOption.ReplaceExisting);
	string res = mem1.name + '\n' + mem2.name + '\n' + mem3.name + '\n' +
	mem4.name + '\n' + mem5.name + '\n' + mem6.name + '\n' + mem7.name + '\n' +
	mem8.name + '\n' + mem9.name + '\n' + mem10.name + '\n' + mem11.name + '\n' +
	mem12.name + '\n' + mem13.name + '\n' + mem14.name + '\n' + mem15.name + '\n'
	+ mem16.name + '\n' + mem17.name + '\n' + mem18.name + '\n' + mem19.name +
	'\n' + mem20.name + '\n' + mem21.name + '\n' + mem22.name + '\n' + mem23.name
	+ '\n' + mem24.name + '\n' + mem25.name + '\n' + mem26.name + '\n' +
	mem27.name + '\n' + mem28.name + '\n' + mem29.name + '\n' + mem30.name;
	await FileIO.WriteTextAsync(BaseFile, res);
}
```

На выходе в файле ProdBase.txt имеем:

```
1 хлеб
2 хлеб
3 хлеб
4 хлеб
5 хлеб
6 хлеб
7 хлеб
8 хлеб
9 хлеб
10 хлеб
11 хлеб
12 хлеб
13 хлеб
14 хлеб
15 хлеб
16 хлеб
17 хлеб
18 хлеб
19 хлеб
20 хлеб
```

```
21 хлеб
22 хлеб
23 хлеб
24 хлеб
25 хлеб
26 хлеб
27 хлеб
28 хлеб
29 хлеб
30 хлеб
```

Что свидетельствует об успешном выполнении теста.

Оценка покрытия тестирования

В качестве метода оценивания покрытия тестирования было выбрано покрытие кода, которое рассчитывается по следующей формуле:

$$\mathbf{Tcov = (Ltc/Lcode) * 100\%}$$

- **Tcov** - тестовое покрытие
- **Ltc** - количество строк кода, покрытых тестами
- **Lcode** - общее количество строк кода

$$\mathbf{Tcov = (2294/ 2646) * 100\% \approx 87\%}$$

Журнал тестирования

Блочное тестирование

№	Дата	Тестировщик	Объект	Попытка	Результат
B-1	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод SaveBase	1	Пройден
B-2	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод SaveBase	1	Пройден
B-3	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод ExtractBase	1	Пройден
B-4	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод ExtractBase	1	Пройден
B-5	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод ExtractBase	1	Пройден
B-6	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод ExtractBase	1	Пройден
B-7	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод last	1	Пройден
B-8	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод last	1	Пройден
B-9	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод first	1	Пройден
B-10	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод first	1	Пройден
B-11	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод fully	1	Пройден
B-12	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод fully	1	Пройден
B-13	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод noput	1	Пройден
B-14	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод noput	1	Пройден
B-15	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод noput	1	Пройден
B-16	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод noput	1	Пройден
B-17	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод autoput	1	Пройден
B-18	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод autoput	1	Пройден
B-19	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод autoput	1	Пройден
B-20	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод autoput	1	Пройден
B-21	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод putter	1	Пройден
B-22	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод putter	1	Пройден
B-23	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод putter	1	Пройден

B-24	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод putter	1	Пройден
B-25	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод delete	1	Пройден
B-26	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод message	1	Пройден
B-27	15.12.2018	Вершинин И.В.	Модуль CurrentList, метод getSize	1	Пройден
B-28	15.12.2018	Вершинин И.В.	Модуль CurrentList, метод getSize	1	Пройден
B-29	15.12.2018	Вершинин И.В.	Модуль ListOfLists, метод checkFile	1	Пройден

Интеграционное тестирование

№	Дата	Тестировщик	Объект	Попытка	Результат
I-1	16.12.2018	Вершинин И.В.	Взаимодействие first → fully →noput	1	Пройден
I-2	16.12.2018	Вершинин И.В.	Взаимодействие first → fully →noput	1	Пройден
I-3	16.12.2018	Вершинин И.В.	Взаимодействие first → fully →noput	1	Пройден
I-4	16.12.2018	Вершинин И.В.	Взаимодействие first → last → noput → autoput	1	Пройден
I-5	16.12.2018	Вершинин И.В.	Взаимодействие first → last → noput → autoput	1	Пройден
I-6	16.12.2018	Вершинин И.В.	Взаимодействие first → last → noput → autoput	1	Пройден
I-7	16.12.2018	Вершинин И.В.	Взаимодействие first → last → noput → autoput	1	Пройден
I-8	16.12.2018	Вершинин И.В.	Взаимодействие first → last → noput → autoput	1	Пройден
I-9	16.12.2018	Вершинин И.В.	Взаимодействие first → last → noput → autoput	1	Пройден
I-10	16.12.2018	Вершинин И.В.	Взаимодействие first → last → noput → autoput	1	Пройден
I-11	16.12.2018	Вершинин И.В.	Взаимодействие first → noput → putter	1	Пройден
I-12	16.12.2018	Вершинин И.В.	Взаимодействие first → noput → putter	1	Пройден
I-13	16.12.2018	Вершинин И.В.	Взаимодействие first → noput → putter	1	Пройден
I-14	16.12.2018	Вершинин И.В.	Взаимодействие first → noput → putter	1	Пройден

Специальное тестирование

№	Дата	Тестировщик	Объект	Попытка	Результат
S-1	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод SaveBase	1	Не пройден (отчет об ошибке №1)
S-2	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод fully	1	Пройден
S-3	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод noput	1	Пройден
S-4	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод autoput	1	Пройден
S-5	15.12.2018	Вершинин И.В.	Модуль ProdBase, метод putter	1	Пройден
S-6	15.12.2018	Вершинин И.В.	Модуль CurrentListEdit, метод addElement	1	Пройден
S-7	15.12.2018	Вершинин И.В.	Модуль ListOfLists, метод createList	1	Пройден

Аттестационное тестирование

№	Дата	Тестировщик	Объект	Попытка	Результат
A-1	17.12.2018	Вершинин И.В.	Ф. Т. 1	1	Пройден
A-2	17.12.2018	Вершинин И.В.	Ф. Т. 2	1	Пройден
A-3	17.12.2018	Вершинин И.В.	Ф. Т. 3.1	1	Пройден
A-4	17.12.2018	Вершинин И.В.	Ф. Т. 3.2	1	Пройден
A-5	17.12.2018	Вершинин И.В.	Ф. Т. 3.3	1	Пройден
A-6	17.12.2018	Вершинин И.В.	Ф. Т. 4.1	1	Пройден
A-7	17.12.2018	Вершинин И.В.	Ф. Т. 4.2	1	Пройден
A-8	17.12.2018	Вершинин И.В.	Ф. Т. 5	1	Пройден
A-9	17.12.2018	Вершинин И.В.	Ф. Т. 7.1	1	Пройден
A-10	17.12.2018	Вершинин И.В.	Ф. Т. 7.2	1	Пройден

Журнал найденных ошибок

№ отчета об ошибке	Дата составления отчета	№ теста	Ожидаемый результат	Фактический результат
1	15.12.2018	S-1	срабатывает NoPlaceException, приложение завершает свою работу	NoPlaceException не срабатывает, приложение зависает

Анализ результатов тестирования

В ходе специального тестирования была найдена одна некритическая ошибка, которая возникает при очень специфичных условиях, а именно при отсутствии памяти на устройстве, что влияет на работу системы в целом, а не только на функционал приложения. Процент не пройденных тестов меньше 3% от общего количества тестов, так что тестирование можно считать успешным.