Министерство образования и науки Российской Федерации

ФГБОУ ВО МГУТУ им. К. Г. Разумовского (ПКУ)

Университетский колледж информационных технологий

**Головоломка «Какуро»**

Листов 9

Исполнитель

Рогулев И.С.

22.02.2015

2015

1. **Объект испытаний**
   1. Объектом испытаний является программа «Головоломка «Какуро»», Предназначенная для решения головоломок «Какуро», установленных на устройстве и для генерации головоломки с размерами от 8х8 до 12х12
2. **Цель испытания**
   * 1. Проверка работоспособности программного продукта.
     2. Проверка соответствия программного продукта требованиям, приведенным в документе «Техническое задание»
     3. Проверка готовности программного продукта к эксплуатации
3. **Требования к программному продукту**
   1. Требования к функциональным характеристикам
      1. Требования к составу выполняемых функций:

Программа должна обеспечивать выполнение следующих функций:

* Генерировать головоломку «Какуро» с размерами от 3x3 до 10x10;
* Давать пользователю возможность выбора головоломок, находящихся на устройстве;
* Создание статистики решений головоломок пользователями;
* Учет времени решений;
* Дать пользователю подсказку в решении головоломки по запросу пользователя;
* Функций окончания игры;
* Функций получения пользователем справки о правилах игры, в любой момент.
  1. **Требования к надежности**
     1. Требования к контролю входной информации
        1. Корректность работы файловых операций
        2. Корректность действий пользователя
        3. Функции проверки пользовательского ввода, которое повлияет на дальнейшее решение головоломки или на работу всей программы в целом.
     2. Другие требования к надежности не применяются
  2. Требования к эксплуатации
     1. Минимальное количество персонала-один человек, владеющий русским языком, обладающий практическими навыками по использованию программ, работающими в оконном режиме, знающий числа и алгебраическую операцию сложения курса математики.
     2. Ввод пользователем числовых значений
     3. Другие специальные требования к эксплуатации не применяются
  3. Требования к параметрам и составу технических средств
     1. Операционная система Windows 7
     2. Процессор Core i3
     3. Манипулятор мышь
     4. Монитор
     5. Жесткий или SSD-диск
     6. Наличие IDE Lazarus v 1.4.2
  4. Требования к информационной и программной совместимости
     1. Требования к защите информации и программ не предъявляются
     2. Другие требования не применяются
     3. Программа должна работать под управлением операционной системы Windows 7

1. **Требования к программной документации**
   1. Состав программной документации
   * Состав программной документации должен включать:
   * Техническое задание
   * Пояснительную записку
   * Текст программы
   * Программу и методику испытаний
   1. Специальные требования к пояснительной записке

Пояснительная записка должна содержать блок-схему (блок-схемы) алгоритма(ов), используемых в программе

* 1. Специальные требования к тексту программы
     1. Требования к исходным кодам изложены в документе: А. И. Глускер «Сборник задач по курсу “Основы алгоритмизации и программирования” [Электронный ресурс] – 2011 /раздел 3.1/ Программа должна быть написана на языке Паскаль и компилироваться в Lazarus v1.4.2

1. **Средства и порядок испытаний**
   1. Технические средства, используемые при проведение испытаний

В состав технический средств входит IBM-совместимый компьютер, включающий:

* Процессор Core i3;
* Клавиатура;
* Мышь;
* Видеокарту;
* Монитор;
* Жесткий или SSD-диск.
* Lazarus IDE v1.4.2
  1. Программные средства, используемые при проведении испытаний. Состав программных средств:
* операционной системы Windows 7, локализованная для работы в Российской Федерации;
* Lazarus v1.4.2 или более (вариант, предназначенный для работы в среде Windows).
  1. Порядок проведения испытаний
     1. Подготовка к проведению испытаний заключается в обеспечении наличия компьютера, и программных средств, установленных на этом компьютере
     2. Ход проведения испытаний документируется в протоколе, где указывается перечень проводимых испытаний, результат каждого испытания и возможно замечания.
     3. Состав испытаний:
        1. Проверка состава программной документации
        2. Проверка требований к программе
        3. Проверка обеспечения требований к программе
        4. Проверка требований к программной документации
        5. Проверка пояснительной записки
        6. Проверка текстов программ

1. **Метод испытаний**
   1. Метод проверки требований к программе

Программа должна выполнять следующие функции:

* Генерировать головоломку «Какуро» с размерами от 8x8 до 12x12;
* Давать пользователю возможность выбора головоломок, находящихся на устройстве;
* Создание статистики решений головоломок пользователями;
* Учет времени решений;
* Дать пользователю подсказку в решении головоломки по запросу пользователя;
* Функций окончания игры;
* Функций получения пользователем справки о правилах игры, в любой момент.
* Функции отображении статистики уровня
* корректность работы файловых операций
* Корректность действий пользователя
* Функции проверки пользовательского ввода, которое повлияет на дальнейшее решение головоломки или на работу всей программы в целом.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Требование | наименование | испытание |
| 1 | Генерировать головоломку «Какуро» с размерами от 8x8 до 12x12; | Проверка генерации головоломки «Какуро» с заданными параметрами высоты и ширины | После запуска программы, из главного меню выбрать пункт «Играть», после выбрать пункт «Сгенерировать»,  Задать параметры ширины и высоты (количество ячеек по высоте и ширине). Если  Головоломка была построена с заданными параметрами то  Функция работает корректно, иначе функция работает не корректно |
| 2 | возможность выбора головоломок, находящихся на устройстве; | Проверка выбора головоломки «Какуро» располагающихся на устройстве | После запуска программы, из главного меню выбрать пункт «Играть», выбрать пункт (Легкий, Нормальный, Тяжелый).  После чего должны появиться уровни, если они не появились, то функция работает не корректно. Если уровни есть, выбираем любой и если идет отображение уровня, то функция работает корректно, иначе функция работает не корректно |
| 3 | Создание и ведение статистики решений головоломок пользователями | Создание и ведение статистики решений головоломок различными пользователями. Результаты будут записаны в файлы | Выполнить все действия пункта, указанного выше.  После завершения уровня должно вывести статистику решения уровня пользователями, и дать возможность ввести свое имя и фамилию. Если условия выполнены, то функция работает корректно, иначе функция работает не корректно |
| 4 | Учет времени решений; | Учет времени решении головоломки, для занесения в статистику и ведения лучших 10 игроков, прошедших уровень | Выполнить действия пункта 2.  Во время игры засечь время.  После истечения 60 секунд, сверить время в программе с засеченным, если время совпадает, то функция работает корректно. Иначе функция работает не корректно. |
| 5 | Дать пользователю подсказку в решении головоломки по запросу пользователя; | По просьбе пользователя, может быть выведена подсказка решения головоломки | Выполнить действия пункта 2.  Нажать на кнопку «Подсказка». После чего будет выведена подсказка в случайном месте уровня. Если все условия выполнены, то функция работает корректно. Иначе функция работает не корректно. |
| 6 | Функций окончания игры; | После решении головоломки, должно выводиться сообщение о конце игры, и выполнены условия пункта 3 | Выполнить действия пункта 2. Пройти любой уровень любой сложности. После прохождения уровня, выведется сообщение о конце игры и будет показана статистика решения уровня разными пользователями, если не появились, то функция работает не корректно.  Затем будет предложено ввести имя и фамилию для статистики, если не появились, то функция работает не корректно. после ввода данных, будет выведено сообщение об успешном введении данных, если не появились, то функция работает не корректно. Если все условия выполнены, то функция работает корректно. |
| 7 | Функций получения пользователем справки о правилах игры, в любой момент | В любой момент выполнении программы пользователю должна даваться возможность, получить справку о правилах игры. (2 вида)  1: из главного меню по пункту «правила»  2: в любое время по нажатию кнопки «правила» в верхней части диалога с пользователем программного продукта | 1. Из главного меню перейти на пункт «Правила». Если произошло отображение правил функция работает корректно.  (2) нажать на кнопку «Правила» в любой момент выполнения программы.   Если произошло отображение правил функция работает корректно.  Если хоть одно из условий не выполнено, , то функция работает не корректно |
| 8 | Функции отображении статистики уровня | По желанию пользователь может узнать статистику уровня. (Всех пользователей, прошедших уровень и их время)  (2 вида)  1: во время выбора уровня  2: по кнопке «Меню» при прохождении уровня | 1. Выполнить все действия пункта 2   Нажать на кнопку «Статистика»   1. Выполнить все действия пункта 2   Выбрать уровень. Нажать на кнопку «Меню»  Если после этого появилась статистика уровня, функция работает корректно, иначе функция работает не корректно |
| 9 | Корректность работы файловых операций | Проверка работы с файлами | Если пункты 2 и 3 будут выполнены корректно. (Должны выполняться чтение и запись файл, без ошибок. Если выведена ошибка, то функция работает не корректно.) то функция работает корректно, иначе функция работает не корректно. |
| 10 | Корректность действий пользователя | Проверка всех компонентов диалога с пользователем |  |
| 11 | Функции проверки пользовательского ввода, которое повлияет на дальнейшее решение головоломки или на работу всей программы в целом. | Проверка ввода пользователем данных | Проверка всевозможных вводимых данных, на соответствие типу, размеру. проверка осуществляется при вводе данных во время игры (водимые значения в игровые поля –цифры),  А также во время ввода имени и фамилии во время учета статистики (буквенные и цифровые значения).  Если все требования соблюдены и операции не вызвали ошибок, то функция работает корректно, иначе функция работает не корректно. |

* 1. Метод проверки требований к составу программной документации
     1. Проверка состава программной документации осуществляется визуально путем сравнения набора предъявленных документов (в форме распечатки или в рукописной форме) списку, приведенному ранее. При этом исходные тексты программ должны быть предоставлены так же и в электронной форме. В случае если набор предъявленных документов соответствует списку, а исходные тексты предоставлены также в электронной форме, то в протокол заносится запись: «Состав программной документации» – соответствует; в противном случае: «Состав программной документации» – не соответствует
  2. Метод проверки требований к пояснительной записке

Проверка состоит из следующих этапов:

* проверка наличия блок-схемы (блок-схем) в пояснительной записке;
* проверка соблюдения требований ГОСТ 19.701-90 для каждой блок-схемы;
* проверка соблюдения локальных стандартов для блок-схем;
* проверка соответствия каждой блок-схемы алгоритму, закодированному в программе. Проверка соблюдения требований ГОСТ 19.701-90 состоит из следующих работ:
* проверка использования только тех символов, которые указаны как применимые к схемам программ в п. 5 ГОСТ 19.701-90;
* проверка соответствия символов их назначению (экспертная оценка лица, проводящего испытания);
* проверка правильности выполнения соединения линий (п. 4.2.3 ГОСТ 19.701-90);
* проверка того, что линии потока управления, выходящие из символа «решение» подписана (п. 4.3.1.2 ГОСТ 19.701-90); Проверка соблюдения локальных стандартов для блок-схем состоит из следующих работ:
* проверка того, что все символы (кроме терминаторов, соединителей, линий и комментариев) имеют одинаковые размеры;
* проверка того, что терминаторы имеют ту же ширину, что и другие символы;
* проверка того, что отношение ширины к высоте составляет 2 к 1 для каждого символа, кроме терминаторов, комментариев и линий
* проверка того, что отношение ширины к высоте составляет 4 к 1 для терминаторов;
* проверка того, что высота соединителей совпадает с высотой терминаторов;
* проверка того, что линии потока управления входят в символ слева или сверху, а выходят снизу или справа;
* проверка того, что подписи к линиям не находятся на самих линиях.   
  Проверка соответствия каждой блок-схемы алгоритму, закодированному в программе, осуществляется путем экспертной оценки лицом, осуществляющим проведение испытаний. В случае, если все вышеприведенные проверки прошли успешно, в протокол заносится запись: «Специальные требования к пояснительной записке» – соответствует; в противном случае «Специальные требования к пояснительной записке» – не соответствует.
  1. Метод проверки требований к исходным кодам

Изложенный ниже метод применяется ко всем файлам, содержащим исходный текст, и входящим в состав программной документации по отдельности. Для каждого файла вносится в протокол запись: «Требования к исходным кодам для файла ####» – соответствует/не соответствует (где вместо #### указывается название файла). Проверка состоит из следующих этапов:

* Наличие комментария в начале файла, содержащего автора работы, номера задания и варианта, краткой формулировки задания (или его части)
* Наличие комментариев к неочевидным действиям (проверяется методом экспертной оценки лицом, осуществляющим испытания)
* Для каждой подпрограммы наличие комментария, содержащего полное описание ее работы, описание всех аргументов и результатов. Достаточность этого комментария для возможности использовать подпрограмму в других программах (без изучения собственно текста подпрограммы).
* Для каждой глобальной переменной указание ее назначения.
* Для всех переменных, кроме переменных цикла, использование «говорящих» названий
* Для всех подпрограмм использование говорящих названий.
* Использование одного оператора на одной строке программы.
* Количество пробелов перед строкой программы должно соответствовать уровню вложенности (по два пробела на уровень вложенности).
* Слова begin и end, соответствующие друг другу, располагаются строго с одной и той же позиции по вертикали.
* Количество строк в подпрограмме и в самой программе (между begin и end) – не более 25 строк.
* Использование модулей для трех и более сходных по назначению подпрограмм.
* Отсутствие в подпрограммах использования глобальных переменных (напрямую).
* Разделение подпрограмм на предназначенные для вычислений (в них не должно быть ввода-вывода) и на предназначенные для ввода-вывода (в них вычисления должны быть только такие, что нужны для ввода-вывода).
* Отсутствие операторов goto, break, continue; процедур halt и exit.
* Проверка того, что вместо явно указанных значений чисел, в тексте программы используются константы.

В случае, если все перечисленные этапы пройдены, то в протокол о соответствии файла требованиям, в противном случае – о несоответствии.

* 1. Метод проверки требований к исходным кодам в части компиляции во Lazarus v1.4.2. Изложенный ниже метод применяется ко всем файлам, содержащим исходный текст, и входящим в состав программной документации по отдельности. Для каждого файла вносится в протокол запись: «Требования к исходным кодам в части компиляции во Lazarus v1.4.2 для файла ####» – соответствует/не соответствует (где вместо #### указывается название файла). Проверка осуществляется путем открытия файла с исходным кодом в среде Lazarus v1.4.2, осуществления компиляции (обе работы делаются в соответствии с документацией к Lazarus v1.4.2). В случае, если компиляция завершилась успешно в протокол заносится запись о соответствии требованиям, в противном случае – о несоответствии.