***Федеральное государственное бюджетное образовательное учреждение высшего образования***

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана***  ***(национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ ИУ

КАФЕДРА Компьютерные системы и сети

**Отчет**

**по лабораторной работе № 2**

**Дисциплина: Технология разработки программных систем**

**Название лабораторной работы: тестирование программного обеспечения**

**Варианты: 3, 10, 5**

Студент гр. ИУ6-42 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузнецов И.А.**

**(Подпись, дата) (И.О. Фамилия)**

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Хорунжина К.С.**

**(Подпись, дата) (И.О. Фамилия)**

Москва, 2018

# Структурный контроль

## Задание

Вар.3. Программа должна формировать массив чисел от 3 до 25 , а затем сортировать элементы массива по возрастанию и исключать повторяющиеся элементы

1. program v3;
2. {$APPTYPE CONSOLE}
3. uses
4. SysUtils;
5. const n=10;
6. var m:array [1..N] of integer;
7. i, j, k, L, b:integer;
8. begin
9. { TODO -oUser -cConsole Main : Insert code here }
10. randomize;
11. L:=N;
12. for i:=1 to L do
13. begin
14. m[i]:=random(25)+3;
15. write(m[i],' ')
16. end;
17. writeln;
18. k:=1;
19. repeat
20. i:=1;
21. repeat
22. if m[i] = m[i+1] then
23. begin
24. for j:=i+1 to L-1 do
25. m[j]:=m[j+1];
26. dec(L);
27. dec(i)
28. end
29. else
30. if m[i] > m[i+1] then +
31. begin {åñëè < - ïî óáûâàíèþ}
32. b:=m[i];
33. m[i]:=m[i+1];
34. m[i+1]:=b
35. end;
36. inc(i); + 1
37. until I<L-1;
38. inc(k);
39. until K>L-1; for i:=1 to L do
40. begin
41. write(m[i],' ')
42. end;
43. readln;
44. end.

## Вопросы для проверки

1. Обращения к данным.
2. Все ли переменные инициализированы?
3. Не превышены ли максимальные (или реальные) размеры массивов и строк?
4. Не перепутаны ли строки со столбцами при работе с матрицами?
5. Присутствуют ли переменные со сходными именами?
6. Используются ли файлы? Если да, то

* При вводе из файла проверяется ли завершение файла?
* Соответствуют ли типы записываемых и читаемых значений?

1. Использованы ли нетипизированные переменные, открытые массивы, динамическая память? Если да, то

* Соответствуют ли типы переменных при "наложении" формата?
* Не выходят ли индексы за границы массивов?

1. Вычисления.
2. Правильно ли записаны выражения (порядок следования операторов)?
3. Корректно ли производятся вычисления неарифметических переменных?
4. Корректно ли выполнены вычисления с переменными различных типов (в том числе с использованием целочисленной арифметики)?
5. Возможно ли переполнение разрядной сетки или ситуация машинного нуля?
6. Соответствуют ли вычисления заданным требованиям точности?
7. Присутствуют ли сравнения переменных различных типов?
8. Передачи управления.
9. Будут ли корректно завершены циклы?
10. Будет ли завершена программа?
11. Существуют ли циклы, которые не будут выполняться из-за нарушения условия входа? Корректно ли продолжатся вычисления?
12. Существуют ли поисковые циклы? Корректно ли отрабатываются ситуации "элемент найден" и "элемент не найден"?
13. Интерфейс.
14. Соответствуют ли списки параметров и аргументов по порядку, типу, единицам измерения?
15. Не изменяет ли подпрограмма аргументов, которые не должны изменяться?
16. Не происходит ли нарушения области действия глобальных и локальных переменных с одинаковыми именами?
17. Логика программы.
18. Правильно ли заданы границы для случайной генерации значений?
19. В правильную ли сторону проводится сортировка? Не перепутаны ли операторы сравнения «<» и «>»?

## Таблица найденных ошибок

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер вопроса** | **Строки подлежащие проверке** | **Результаты проверки** | **Вывод** |
| 1.1 | 7 | Переменные: i, j, k, L, b  инициализированы  Не инициализированы переменные K, I | В результате использования не инициализированных переменных не будет выполняться условие цикла |
| 1.2 | 6 | Массив создается на 10 эл-в. Размер нигде не нарушается. | Массив используется корректно |
| 1.4 | 7,36,38 | Используются переменные со сходными именами, но большего регистра | Данные момент может привести к путанице  При написании программы |
| 2.1 | 12,19,20,21,23,29,39 | Порядок выполнения операторов верный  За исключением, строки 21,где if не имеет закрывающего end | В результате этой ошибки не будет выполнена сортировка |
| 2.3 |  | В программе используются переменные только типа integer. | Операции проводятся корректно |
| 3.1 | 12,19,20,23,39 | Циклы for завершатся корректно. Но циклы repead until не выполнятся из-за некорректных условиях выхода | Следует изменить условие выхода из циклов repeat until |
| 4.1 |  | Массив заполняется функцией randomize, что позволило избежать ошибок с данными | Здесь все выполняется корректно. |
| 5.1 | 14 | Границы заданы верно | Программа выполняется нормально. |

**Вывод**

Структурный контроль позволяет выявить ошибки кодирования, связанные с

человеческим фактором. Также данный метод позволит улучшить стиль

программирования и тем самым повысить технологичность программного

продукта.

Достоинством данного метода является отсутствие необходимости

выполнять программу.

Недостатки: сложность тестирования больших программ;

сложность обнаружения ошибок в логике программы.

**Белый ящик**

Для заданного фрагмента схемы алгоритма подготовить тесты,

используя методы стратегии "белого ящика". Предлагаемые тесты свести в

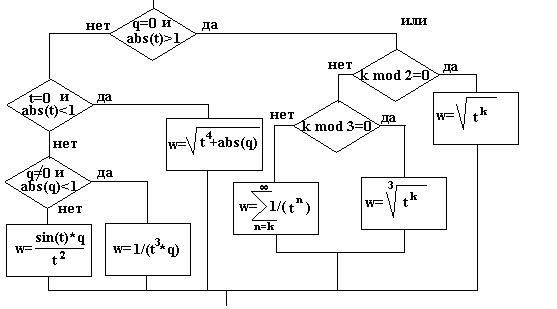
таблицу. Сравнить тесты, предлагаемые различными методами. Сделать

вывод о роли тестирования с использованием стратегии "белого ящика" и

возможностях его применения. Сформулировать его достоинства и

недостатки

Схема алгоритма (Вариант 10)

****

## Покрытие операторов

Позволяет обнаружить ошибки, содержащиеся в операторах. Не позволяет обнаружить ошибки в условиях.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Назначение теста** | **Значения исходных данных** | **Маршрут** | **Ожидаемый результат** |
| 1 | Проверка условия | t = 1,q = 5, | нет, нет, нет | w= (sin(t)\*q)/t^2 |
| 2 | Проверка условия | q = 0.2, t = 1 | нет, нет, да | w= 1/((t^3)\*q) |
| 3 | Проверка условия | t = 0, q = 1, | нет,да |  |
| 4 | Проверка условия | q = 0, t = 2, k = 7 | да, нет, нет |  |
| 5 | Проверка условия | q = 0, t = 2, k = 6 | да, нет, да |  |
| 6 | Проверка условия | q = 0, t = 2, k =4 | да, да |  |

**Достоинства белого ящика:**

Простота применения, возможность выявить наибольшее число

возможных ошибок, позволяет проверить всю структуру программы.

**Недостатки белого ящика:**

Метод требует значительных умственных усилий. Он не применим к

большим и сложным программам, потому что тестирование в этом случае

будет неточным и займет много времени.

**Вывод:**

В данном случае метод тестирования с комбинаторным покрытием

условий является наиболее удобным, потому что он сочетает в себе все

остальные возможные методы (покрытия операторов, покрытия решений,

покрытия условий). Метод применим к достаточно простым программам.

**Часть 3. Тестирование по принципу «черного ящика»**

**Задание:**

Внимательно изучить формулировку задачи, подготовить тесты по

методикам стратегии "черного ящика". Предлагаемые тесты свести в таблицу.

Выполнить тестирование выданного модуля программы. Занести в таблицу

результаты. Сделать вывод о роли тестирования с использованием стратегии

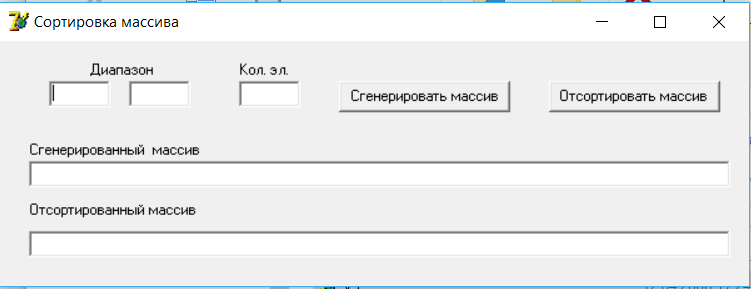
"черного ящика" и возможностях его применения. Сформулировать его

достоинства и недостатки.

**Формулировка задачи (Вариант 5):**

Создать программу, которая позволяет в заданном диапазоне и

количестве с помощью генератора случайных чисел создавать массив, а затем сортировать его.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Назначение**  **Теста** | **Значение**  **исходных**  **данных** | **Ожидаемый**  **результат** | **Реакция**  **программы** | **Вывод** |
| 1 | Проверка ввода  всех корректных  данных | Диапазон:  (1,10)  Кол-во эл-в:10 | Сортировка  сгенерирован-  ного массива | Сгенерированный  массив:  9,5,2,10,7,6,3,1,4,8  Отсортированный  массив:  1,2,3,4,5,6,7,8,9,10 | Тест  пройден |
| 2 | Проверка ввода  всех  некорректных  данных | Диапазон:  (a b)  Кол-во эл-в:  один | Ошибка.  Диапазон и  кол-во  элементов  заданы с  ошибкой | Диапазон задан с  ошибкой! | Тест  пройден  частично |
| 3 | Проверка на  пустой ввод | Диапазон:  Кол-во эл-в: | Ошибка.  Пользователь  не ввел  данные | Диапазон задан с  ошибкой! | Тест  пройден  частично |
| 4 | Проверка на  корректность  диапазона  (положительные  числа) | Диапазон:  (10,1)  Кол-во эл-в:  5 | Ошибка.  Неправильно  введен  диапазон | Сгенерированный  массив:  12,19,14,13,16  Отсортированный  массив:  12,13,14,16 | Тест не  пройден |
| 5 | Проверка на  корректность  диапазона  (отрицательные  числа) | Диапазон:  (-1,-5)  Кол-во эл-в:  5 | Ошибка.  Неправильно  введен  диапазон | Сгенерированный  массив:  1,-1,3,0,2  Отсортированный  массив:  0,-1,1,2,3 | Тест не  пройден |
| 6 | Проверка на  корректность  диапазона (числа  разного знака) | Диапазон:  (50,-4)  Кол-во эл-в:  6 | Ошибка.  Неправильно  введен  диапазон | Сгенерированный  массив:  75,55,67,70,66,99  Отсортированный  массив:  55,66,67,70,75,99 | Тест не  пройден |
| 7 | Проверка на  корректность  диапазона (шаг  диапазона 1) | Диапазон:  (5,6)  Кол-во эл-в:  6 | Сортировка  сгенерирован-  ного массива  из повторяю-  щихся чисел | Зависание  программы | Тест не  пройден |
| 8 | Проверка на  корректность  диапазона (одна  из границ не  число) | Диапазон:  (5,abc)  Кол-во эл-в:  5 | Ошибка.  Неправильно  введен  диапазон | Диапазон задан с  ошибкой! | Тест  пройден |
| 9 | Проверка на  корректность  диапазона  (одинаковые  корректные  границы) | Диапазон:  (55,55)  Кол-во эл-в:  10 | Ошибка.  Неправильно  введен  диапазон | Зависание  программы | Тест не  пройден |
| 10 | Проверка  работы на  больших  граничных  значениях | Диапазон:  (111111111,  222222222)  Кол-во эл-в:  3 | Сортировка  сгенерирован-  ного массива | Сгенерированный  массив:  205093859,  212374757,  200978257  Отсортированный  массив:  200978257,  205093859,  212374757 | Тест  пройден |
| 11 | Проверка  работы на  больших  граничных  значениях на  большом  диапазоне | Диапазон:  (-100000000,  500000000)  Кол-во эл-в:  2 | Сортировка  сгенерирован-  ного массива | Сгенерированный  массив:  210954187,  -929373633  Отсортированный  массив:  210954187,  -929373633 | Тест  пройден  частично |
| 12 | Проверка на  корректность  ввода кол-ва  элементов  (отрицательное) | Диапазон:  (10, 100)  Кол-во эл-в:  -10 | Ошибка. Ввод  некорректных  данных | Сгенерированный  массив:  57,98,  Отсортированный  массив: | Тест не  пройден |
| 13 | Проверка на  корректность  ввода кол-ва  элементов (ноль) | Диапазон:  (10, 100)  Кол-во эл-в:  0 | Ошибка. Ввод  некорректных  данных | Сгенерированный  массив:  89,81,  Отсортированный  массив: | Тест не  пройден |
| 14 | Проверка  работы на  больших  значениях кол-ва  элементов | Диапазон:  (10, 100)  Кол-во эл-в:  1000000000 | Сортировка  сгенерирован-  ного массива | Зависание  программы | Тест не  пройден |
| 15 | Проверка на  ввод нецелых  чисел | Диапазон:  (1.2, 10.2)  Кол-во эл-в:  8 | Сортировка  сгенерирован-  ного массива | Диапазон задан с  ошибкой!  Отсортированный  массив:  0,0,0,0 | Тест  пройден  частично |
| 16 | Проверка на  ввод в поля для  вывода | Сгенериро-  ванный  массив:  5,10, 9,3 | Ошибка. Ввод  данных в поле  вывода | Программа не  реагирует на ввод  в поле вывода | Тест  пройден |

**Таблица тестирования методом «черного ящика»:**

**Достоинства черного ящика:**

Метод прост в использовании из-за тестирования на машине.

Вероятность нахождения ошибки очень велика.

**Недостатки черного ящика:**

Метод позволяет тестировать только уже готовую программу.

**Вывод:**

Тестирование методом «черного ящика» имеет целью выяснение

обстоятельств, в которых поведение программы не соответствует

спецификации.

**Общий вывод**

В ходе лабораторной работы было выяснено, что наибольшее число

ошибок может быть выявлено с помощью тестирования по принципу «черного ящика», что в совокупности с простотой и быстротой метода выделяет его, как лучший. Однако такие методы, как структурный контроль, также необходимо использовать, особенно при изначальном написании кода, чтобы уменьшить количество будущих ошибок. Тестирование по методу «белого ящика» позволяет отдельно отработать ошибко опасные, уязвимые места программы, выявленные на других этапах тестирования.