*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана***  ***(национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**Отчет**

**по лабораторной работе № 2**

**Дисциплина:** Технологии разработки программных систем

**Название лабораторной работы:**

Тестирование программного обеспечения

**Вариант 5**

Студент гр. ИУ6-42  **\_\_\_\_\_\_13.04.2018 А.С. Бурлаков**

(Подпись, дата) (И.О. Фамилия)

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

(Подпись, дата) (И.О. Фамилия)

Москва, 2018

**ВВЕДЕНИЕ**

## 1. Цель работы

## - Знакомство с существующими стратегиями тестирования, приобретение навыков выбора стратегии и разработки тестов для отдельных задач, сравнение и оценка различных методов тестирования и их возможностей.

## ОСНОВНАЯ ЧАСТЬ

1. **Структурный контроль**

Для своего варианта задания выполните **структурный контроль**, используя перечень вопросов в Приложении. В процессе выполнения заполните таблицу. Сделайте общий вывод о роли структурного контроля в процессе создания программы. Сформулируйте его достоинства и недостатки.

{Вариант 4. Программа должна генерировать массив чисел и сортировать данные методом вставки. }

**program** v4;

**const**

N = 10;

**type**

tm = **array** [1..N] **of** integer;

**var**

m: tm;

i, j, k, q, b: integer;

**begin**

{ **TODO** -oUser -cConsole Main : Insert code here }

randomize;

**for** i := 1 **to** N **do begin**

m[i] := random(100);

write(' ', m[i])

**end**;

writeln;

k := 3;

**for** i := k **to** N **do begin**

**for** j := 1 **to** i - 1 **do begin**

**if** (m[i] <= m[j]) **then begin**

b := m[i];

**for** q := i **to** j **do**

m[q] := m[q - 1];

m[j] := b;

**end**;

**end**;

k := k + 1;

**end**;

**for** i := 1 **to** N **do begin**

write(' ', m[i])

**end**;

**end**.

**Перечень вопросов для структурного контроля текста**

1. Обращения к данным.

1. Все ли переменные инициализированы?
2. Не превышены ли максимальные (или реальные) размеры массивов и строк?
3. Не перепутаны ли строки со столбцами при работе с матрицами?
4. Присутствуют ли переменные со сходными именами?
5. Используются ли файлы? Если да, то
   1. При вводе из файла проверяется ли завершение файла?
   2. Соответствуют ли типы записываемых и читаемых значений?
6. Использованы ли нетипизированные переменные, открытые массивы, динамическая память? Если да, то
   1. Соответствуют ли типы переменных при "наложении" формата?
   2. Не выходят ли индексы за границы массивов?

2. Вычисления

1. Правильно ли записаны выражения (порядок следования операторов)?
2. Корректно ли производятся вычисления неарифметических переменных? 3) Корректно ли выполнены вычисления с переменными различных типов (в том числе с использованием целочисленной арифметики)?
3. Возможно ли переполнение разрядной сетки или ситуация машинного нуля?
4. Соответствуют ли вычисления заданным требованиям точности?
5. Присутствуют ли сравнения переменных различных типов?

3. Передачи управления

1. Будут ли корректно завершены циклы?
2. Будет ли завершена программа?
3. Существуют ли циклы, которые не будут выполняться из-за нарушения условия входа? Корректно ли продолжатся вычисления?
4. Существуют ли поисковые циклы? Корректно ли отрабатываются ситуации "элемент найден" и "элемент не найден»?

4. Интерфейс

1. Соответствуют ли списки параметров и аргументов по порядку, типу, единицам измерения?
2. Не изменяет ли подпрограмма аргументов, которые не должны изменяться?
3. Не происходит ли нарушения области действия глобальных и локальных переменных с одинаковыми именами?

|  |  |  |  |
| --- | --- | --- | --- |
| Номер вопроса | Строки, которые надо проверить | Результат проверки | Вывод |
| 1.1 | Вся программа | Все переменные инициализируются и используются в программе | В программе нет лишних объявлений |
| 1.2 | Вся программа | Массив m имеет фиксированный размер N=10, в программе размер не превышается. | Размер массива не превышен. |
| 1.3 | - | - | Матрицы не используются |
| 1.4 | Область var | Все переменные названы по разному | В программе не присутствуют переменные со сходными/одинаковыми именами |
| 1.5 | - | - | Файлы не используются |
| 1.6а | Вся программа | - | Типы переменных одинаковы |
| 1.6б | Вся программа | - | Индексы не выходят за размеры массива |
| 2.1 | - | - | Порядок операторов верен |
| 2.2 | Вся программа | - | Вычисления неарифметических параметров производится корректно |
| 2.3 | - | - | Переполнение разрядной сетки или ситуация машинного нуля невозможны |
| 2.4 | Вся программа | Вычислений нет | - |
| 2.5 | Вся программа | В программе только один тип переменных | - |
| 3.1 | Циклы | - | Циклы будут завершены корректно |
| 3.2 | Вся программа | Программа будет завершена | - |
| 3.3 | Вся программа | - | Таких циклов не существует в программе |
| 3.4 | Вся программа | - | Поисковых циклов нет |
| 4.1 | Вся программа | - | Подпрограмм нет |
| 4.2 | Вся программа | - | Подпрограмм нет |
| 4.3 | Вся программа | - | Не нарушается область действия локальных и глобальных переменных |

Вывод: Структурный контроль позволяет избавиться от ошибок, которые могут привести к «крашу» программы, а именно к полному прекращению её работы. Он позволяет учесть большое количество ошибок, которые описаны в вопросах и в таблице.

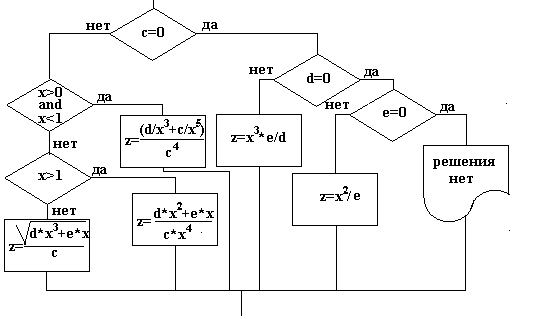
Достоинства:

* Четкие вопросы по программе (Тестирование);
* Простота проверки;
* После удаления ошибок программа практически всегда будет работать.

Недостатки:

* На проверку тратится много времени;
* Вопросы для проверки не учитывают ошибок, связанных с работой алгоритма.

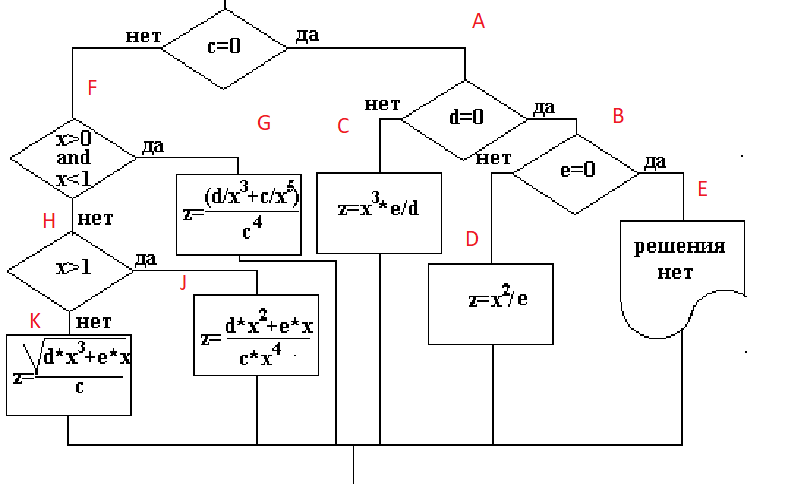
Вариант 8.



Для заданного фрагмента схемы алгоритма подготовьте тесты, используяметоды **стратегии "белого ящика"**. Предлагаемые тесты сведите в таблицу.

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение теста | Значения исходных данных | Ожидаемый результат |
|  |  |  |  |

Сравните тесты, предлагаемые различными методами. Сделайте вывод о роли тестирования с использованием стратегии "белого ящика" и возможностях его применения. Сформулируйте его достоинства и недостатки.



1. Покрытие операторов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение теста | Значения исходных данных | Ожидаемый результат |
| 1 | Достижение блока K | x=1,c=1,d=1,e=1 | sqrt(2) |
| 2 | Достижение блока J | x=2,c=1,d=1,e=1 | 3/8 |
| 3 | Достижение блока G | x=0.5,c=1,d=1,e=1 | 5/32 |
| 4 | Достижение блока C | x=1,c=0,d=1,e=1 | 1 |
| 5 | Достижение блока D | x=1,c=0,d=0,e=1 | 1 |
| 6 | Достижение блока E | x=1,c=0,d=0,e=0 | Решения нет |

1. Покрытие решений

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение теста | Значения исходных данных | Ожидаемый результат |
| 1 | Прохождение путей F, H,K | x=1,c=1,d=1,e=1 | sqrt(2) |
| 2 | Прохождение пути J | x=2,c=1,d=1,e=1 | 3/8 |
| 3 | Прохождение пути G | x=0.5,c=1,d=1,e=1 | 5/32 |
| 4 | Прохождение путей A, C | x=1,c=0,d=1,e=1 | 1 |
| 5 | Прохождение путей B, D | x=1,c=0,d=0,e=1 | 1 |
| 6 | Прохождение пути E | x=1,c=0,d=0,e=0 | Решения нет |

1. Покрытие условий

Выпишем все условия:

x>0, x<1, x<=0, x>=1, x>1, x<=1

c = 0, c <> 0

d = 0, d <> 0

e = 0, e <> 0

Тесты для данных условий:

Точно такие же как и в предыдущих методах.

1. Покрытие решений/условий – хз
2. Комбинаторное покрытие решений

Комбинации:

1. x > 1
2. x = 1
3. x > 0, x < 1
4. x <= 0
5. c = 0
6. c != 0
7. d = 0
8. d != 0
9. e = 0
10. e != 0

Тесты:

1. c = 1, x = 2, d = 0, e = 0 проверяет комбинации 1, 6
2. c = 1, x = 1, d = 0, e = 0 проверяет комбинации 2, 6
3. c = 1, x = 0.1, d = 0, e = 0 проверяет комбинации 3, 6
4. c = 1, x = 0, d = 0, e = 0 проверяет комбинации 4, 6
5. c = 0, x = 2, d = 0, e = 0 проверяет комбинации 5, 7, 9
6. c = 0, x = 2, d = 0, e = 1 проверяет комбинации 5, 7, 10
7. c = 0, x = 2, d = 1, e = 0 проверяет комбинации 5, 8

Вывод: метод белого ящика удобен при небольшом количестве условий. Он позволяет найти ошибки в рассуждениях и понять, что необходимо изменить для устранения ошибок.

Достоинства:

* Проверяет все варианты решений
* Алгоритм рассматривается детально

Недостатки:

* Проверять все пути и условия достаточно долго
* Много методов тестирования этим методом, для полноты тестирования приходится использовать все методы.

Черный ящик

*Задача 7.*

Программа должна вычислять с заданной точностью значение интеграла функции  на введенном с клавиатуры интервале от a до b.

Т.е. это получается

3.1

Правильно – коэффициент – число

Неправильно – что угодно

3.2

Допустим, 1e+999, точность 1e-999