|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **ИУК «Информатика и управление»** |
| **КАФЕДРА** | **ИУК4 «Программное обеспечение ЭВМ,** |
| **информационные технологии»** | |

**ЛАБОРАТОРНАЯ РАБОТА №1**

**«**СТРАТЕГИЯ ЧЕРНОГО ЯЩИКА**»**

**ДИСЦИПЛИНА: «**Тестирование и отладка программного обеспечения**»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-82Б | |  |  | ( | Панина В.С. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Красавин Е.В. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

Калуга, 2021

# Цель: формирование практических навыков составления набора тестовых данных для функционального тестирования.

**Задачи:**

1. рассмотреть два метода формирования тестовых наборов: на основе классов эквивалентности;

2. на основе граничных значений классов эквивалентности.

**Задача 3.1.** Есть программа, которая интерпретирует три целых числа, вводимых с клавиатуры, как длины сторон треугольника и выводит сообщение, о том, какой это треугольник: равносторонний, равнобедренный или неравносторонний. Разработайте тесты (последовательности входных данных и ожидаемые результаты), которые, как вам кажется, будут адекватно проверять эту программу. Рекомендуется запись тестов в виде таблицы:



**Задача 3.2.** В программе «Деканат» переменная, обозначающая количество студентов в группе, может принимать значения от 1 до 30. Составьте тестовый набор для этой переменной.



**Задача 3.3.** В компьютерной обучающей системе тестовые задания для контроля знаний берутся из файла типа .txt, где каждое задание занимает одну строку. Для формирования теста указывается имя файла, количество заданий в тесте и количество вариантов теста (не более 10). Варианты должны различаться не менее чем тремя заданиями. Если заданий в файле недостаточно для реализации этого требования, выдается сообщение «Недостаточно заданий», если файл не найден - «файл отсутствует». Увидеть составленный вариант теста для контроля знаний можно, нажав на соответствующую кнопку «Вариант №».

Составьте тестовые наборы для проверки перечисленных функций. Рекомендуется оформить их в виде таблицы, подобно задаче 3.1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий формирования теста | № теста | Исходные данные | | | Ожидаемый результат |
| Имя файла | Кол-во заданий в тесте | Кол-во вариантов теста |
| Проверка ввода имени несуществующего файла | 1 | somefile.txt | 10 | 3 | Файл отсутствует |
| Проверка пустого значения | 2 |  | 10 | 3 | Файл отсутствует |
| Проверка неправильного расширения файла | 3 | file.file | 10 | 3 | Файл отсутствует |
| Проверка ввода недостаточного кол-ва заданий | 4 | tests.txt | 2 | 2 | Недостаточно заданий |
| Проверка ввода большего кол-ва заданий, чем в файле | 5 | tests.txt | 100 | 2 | Недостаточно заданий |
| Проверка нулевого значения для количества заданий в тесте | 6 | tests.txt | 0 | 1 | Недостаточно заданий |
| Проверка пустого файла | 7 | tests.txt | 3 | 9 | Недостаточно заданий |
| Проверка отрицательного значения количества заданий в тесте | 8 | tests.txt | -1 | 2 | Некорректный ввод |
| Проверка отрицательного значения количества вариантов тесте | 9 | tests.txt | 3 | -1 | Некорректный ввод |
| Проверка нулевого значения для вариантов в тесте | 10 | tests.txt | 1 | 0 | Некорректный ввод |
| Проверка нулевого значения | 11 | tests.txt | 0 | 0 | Некорректный ввод |
| Проверка вещественного значения | 12 | tests.txt | 1,001 | 1,001 | Некорректный ввод |
| Проверка строкового значения | 13 | tests.txt | Три | Девять | Некорректный ввод |
| Корректные данные | 14 | tests.txt | 3 | 9 | Успешное формирование тестов |

**Задача 3.4.** Используя методы эквивалентного разбиения и граничных значений, разработайте программу "Геометрические фигуры" и составьте тестовые наборы для проверки перечисленных ниже функций программы (вариант задается преподавателем). Протестируйте указанную программу.

В программе "Геометрические фигуры" (файл geometry.exe) после ее запуска пользователю предоставляется выбрать вариант геометрической программы. Для выбора пользователь должен нажать на кнопку с названием нужного варианта. Предусмотрены следующие варианты:

**Вариант 1.** Программа определяет тип треугольника. Возможные результаты: прямоугольный, остроугольный, тупоугольный, равнобедренный, правильный (равносторонний)

**Листинг программы:**

def is\_int(\_str):

try:

int(\_str)

return True

except ValueError:

return False

def check\_side():

flag = False

while flag == False:

\_side = (input("Введите сторону треугольника: "))

flag = is\_int(\_side)

if flag and (int(\_side) <= 0):

flag = False

if not flag:

print("Некорректный ввод, введите целое число")

\_side = (input("Введите сторону треугольника: "))

flag = is\_int(\_side)

return int(\_side)

sides = []

a = check\_side()

sides.append(a)

b = check\_side()

sides.append(b)

c = check\_side()

sides.append(c)

print("a = ", a)

print("b = ", b)

print("c = ", c)

if (a + b <= c) or (a + c <= b) or (b + c <= a):

print("Треугольник со сторонами ", a, b, c, " не существует")

else:

if a == b == c:

print("Треуголник со сторонами ", a, b, c, "равносторонний")

elif (a == b) or (b == c) or (a == c):

print("Треуголник со сторонами ", a, b, c, "равнобедренный")

else:

print("Треуголник со сторонами ", a, b, c, "разносторонний")

sides.sort()

if sides[2]\*\*2 == sides[0]\*\*2 + sides[1]\*\*2:

print("Треугольник со сторонами ", a, b, c, " прямоугольный")

if sides[2] \*\* 2 <= sides[0] \*\* 2 + sides[1] \*\* 2:

print("Треугольник со сторонами ", a, b, c, " остроугольный")

if sides[2] \*\* 2 >= sides[0] \*\* 2 + sides[1] \*\* 2:

print("Треугольник со сторонами ", a, b, c, " тупоугольный")

Разработаны тесты и представлены в виде следующей таблицы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий формирования теста | № теста | Исходные данные | | | Ожидаемый результат |
| а | b | c |
| Прямоугольник №1 | 1 | 3 | 4 | 5 | Разносторонний, прямоугольный, остроугольный, тупоугольный |
| Прямоугольник №2 | 2 | 3 | 3 | 4 | Равнобедренный, остроугольный |
| Прямоугольник №3 | 3 | 1 | 1 | 1 | Равносторонний, остроугольный |
| Проверка вещественных значений | 4 | 1,01 | 1,01 | 1,01 | Некорректный ввод данных |
| Проверка нулевых значений | 5 | 0 | 0 | 0 | Некорректный ввод данных |
| Проверка отрицательных значений | 6 | -1 | -1 | -1 | Некорректный ввод данных |
| Проверка текстовых символов | 7 | Один | Два | Три | Некорректный ввод данных |
| Проверка на существование треугольника | 8 | 1 | 2 | 15 | Треугольник не существует |
| Проверка пустого ввода | 9 |  |  |  | Некорректный ввод данных |

**Результаты работы программы:**

**Тест №1:**

a = 3

b = 4

c = 5

Треуголник со сторонами 3 4 5 разносторонний

Треугольник со сторонами 3 4 5 прямоугольный

Треугольник со сторонами 3 4 5 остроугольный

Треугольник со сторонами 3 4 5 тупоугольный

**Тест №2:**

a = 3

b = 3

c = 4

Треуголник со сторонами 3 3 4 равнобедренный

Треугольник со сторонами 3 3 4 остроугольный

**Тест №3:**

a = 1

b = 1

c = 1

Треуголник со сторонами 1 1 1 равносторонний

Треугольник со сторонами 1 1 1 остроугольный

**Тест №4:**

Введите сторону треугольника: 1.01

Некорректный ввод, введите целое число

**Тест №5:**

Введите сторону треугольника: 0

Некорректный ввод, введите целое число

**Тест №6:**

Введите сторону треугольника: -1

Некорректный ввод, введите целое число

**Тест №7:**

Введите сторону треугольника: один

Некорректный ввод, введите целое число

**Тест №8:**

a = 1

b = 2

c = 15

Треугольник со сторонами 1 2 15 не существует

**Тест №9:**

Введите сторону треугольника:

Некорректный ввод, введите целое число

**Выводы:** в результате выполнения лабораторной работы были сформированы практические навыки составления набора тестовых данных для функционального тестирования.