Министерство образования и науки Российской Федерации

ФГБОУ ВО МГУТУ им. Разумовского (ПКУ)

Университетский колледж информационных технологий

***Нахождение площади треугольника***

***по трем сторонам***

Программа и методика испытаний

ПП-26

Листов 7

Исполнитель

Петрова О.П.

02.03.2016

**Объект испытаний**

* 1. Наименование

Нахождение площади треугольника по длинам его сторон(по формуле Герона).

* 1. Область применения

Деятельность, связанная с вычислением площади треугольника по длинам его сторон (по формуле Герона)

* 1. Обозначение программы

ПП-26

1. **Цель испытания**

Проверка соответствия программного продукта требованиям технического задания.

1. **Требования к программе** 
   1. Требования к функциональным характеристикам
      1. Требования к составу выполняемых функций

Программа должна обеспечивать нахождение площади треугольника по длинам его сторон (по формуле Герона) :

S = √p(p- a)(p- b)(p- c)

* + 1. Требования к организации выходных данных

Выходные данные отправляются в стандартный поток вывода и представляют собой вывод сообщения типа “Периметр треугольника равен: “, “Площадь треугольника равна: “; “При данных значениях треугольник не существует!”.

* + 1. Требования к временным характеристикам

Время между окончанием ввода и окончанием вывода результата не должно превышать более 1 секунды.

* 1. Требования к надежности (см. пункт 4.1.1 ТЗ)

Контролировать правильный ввод данных (неотрицательные числа, не буквы и т.д. (см. пункт 4.1.2); не для всех значений a,b,c может существовать треугольник.

3.3 Требования к информационной и программной совместимости (см. пункт 4.5.5 ТЗ)

Требования к информационным структурам на входе и выходе не предъявляются.

1. **Требования к программной документации** 
   1. Состав программной документации

Состав программной документации должен включать: (см. пункт 5.1 ТЗ)

- техническое задание;

- пояснительную записку;

- текст программы;

- текст программы, осуществляющей автоматическое тестирование программы «нахождение площади треугольника по длинам его сторон»;

- программу и методику испытаний;

4.2 Специальные требования к пояснительной записке

Пояснительная записка должна содержать блок-схему алгоритма(-ов), используемых в программе (см. пункт 5.2 ТЗ)

* 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/ (пункт 4.5.3 ТЗ)
     2. Программа должна быть написана на языке Паскаль и компилироваться транслятором Free Pascal 2.6 (пункт 4.5.4 ТЗ)

1. **Средства и порядок испытаний** 
   1. Технические средства, используемые при проведении испытаний (см. пункт 4.4 тз)
   2. Программные средства, используемые при проведении испытаний

В состав программных средств входит:

- лицензионная копия системы Windows XP(любой версии), локализованная для работы в РФ;

- Free Pascal 2.6 или более поздняя версия (вариант, предназначенный для работы в среде Windows).

5.3 Порядок проведения испытаний

5.3.1 Подготовка к проведению испытаний заключается в обеспечении наличия компьютера, описанного в пункте 5.1, и программных средств, указанных в пункте 5.2, установленных на данном компьютере

5.3.2 Ход проведения испытаний документируется в протоколе, где указывается и перечень проводимых испытаний, результат каждого испытания и замечаний

5.3.3 Состав испытаний:

5.3.3.1 Проверка состава программной документации в соответствии с методом, описанным в пункте 6.2

5.3.3.2 Проверка требований к программе

Проверка обеспечения требований к программе (пункт 3) в соответствии с методом, описанным в пункте 6.1

5.3.3.3 Проверка требований к программной документации

5.3.3.3.1 Проверка пояснительной записки (пункт 4.2) в соответствии с методом, описанным в пункте 6.3

5.3.3.3.2 Проверка текстов программы (пункт 4.3.1) в соответствии с методом, описанным в пункте 6.4

5.3.3.3.3 Проверка текстов программы (пункт 4.3.2) в соответствии с методом, описанным в пункте 6.5

6. **Методы испытаний**

6.1 Метод проверки требований к программе

Проверка осуществляется путем запуска программы и сравнения результатов ее запуска с ожидаемыми, при этом осуществляется измерение интервала времени между окончанием ввода и окончанием вывода результатов.

При каждом запуске программы осуществляется ввод трех чисел и визуально контролируется совпадение результатов с ожидаемыми (допускается отклонение не более 0.0001).

Список тестовых примеров:

|  |  |  |
| --- | --- | --- |
| № | Входные данные | Выходные данные |
| 1 | 1 3 0 | Ошибка! Неверные данные! Введите данные заново |
| 2 | 4 -2 2 | Ошибка! Неверные данные! Введите данные заново |
| 3 | a 5 4 | Ошибка! Неверные данные! Введите данные заново |
| 4 | 3 8 9 | Периметр треугольника равен 20  Площадь треугольника равна 11.83 |
| 5 | 4.5 6 5.2 | Периметр треугольника равен 15.7  Площадь треугольника равна 11.35 |
| 6 | 5 13 7 | При данных значениях треугольник не существует |

- В случае, если все результаты тестовых примеров совпадают с контрольными, то в протокол вносится запись: «Требования к составу выполняемых функций»- соответствует; в противном случае- не соответствует.

- В случае, если все результаты тестовых примеров имеют формат, указанный в пункте 3.1.2, то в протокол вносится запись: «Требования к выходным данным»- соответствует; в противном- не соответствует.

- В случае, если результаты примеров №№1-3 совпадают с контрольными,, то в протокол вносится запись: «Требования к надежности»- соответствует; в противном - не соответствует.

- В случае, если длина временного интервала от окончания ввода входных данных до окончания вывода результатов не превышает 1с, то в протокол вносится запись: «Требования к временным характеристикам»- соответствует; в противном - не соответствует.

- В случае, если при выполнении хотя бы одного тестового примера осуществляется запуск программы и ожидается ввод входных данных, то в протокол вносится запись: «Требования к информационной и программной совместимости»- соответствует; в противном - не соответствует.

6.2 Метод проверки требований к составу программной документации

Проверка состава программной документации осуществляется визуально путем сравнения набора предъявленных документов списку, приведенному в пункте 4.1. При этом исходные тексты программ должны быть представлены так же и в электронной форме.

В случае, если набор предъявленных документов соответствует списку, а исходные тексты предоставлены в электронной форме, то в протокол вносится запись: «Состав программной документации» – соответствует; в противном - не соответствует.

6.3 Метод проверки требований к пояснительной записке

Проверка состоит из следующих этапов:

- проверка наличия блок-схемы в пояснительной записке;

- проверка соблюдения требований ГОСТ 19.701-90 для каждой блок-схемы;

- проверка соблюдения локальных стандартов блок-схем;

- проверка соответствия каждой блок-схеме 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 блок-схемам программ в пункте 5 ГОСТ 19.701-90

- проверка соответствия символов их назначению (экспертная оценка лица, проводящего испытания);

- проверка правильности выполнения соединений линий (пункт 4.2.3 ГОСТ 19.701-90);

- проверка того, что линии потока управления, выходящие из символа «решение» подписана (пункт 4.3.2.1 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 отношения 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 справа и снизу;

- проверка того, что подписи к линиям не находятся на самих линиях.

Проверка соответствия каждой блок-схемы алгоритму, закодированному в программу, осуществляется путем экспертной оценки лицом, осуществляющим проведение испытаний.

В случае, если все вышеприведённые проверки прошли успешно, в протокол заносится запись: « Специальные требования к пояснительной записке» - соответствует; в противном случае « Специальные требования к пояснительной записке «- не соответствует.

6.4 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 программы)

- 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

- 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, exit

- Проверка того, что вместо явно указанных значений чисел в тексте программы используются константы.

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

6.5 Метод проверки требований к исходным кодам в части компиляции во Free Pascal 3.6

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 Free Pascal 3.6 для файла ###» - соответствует/ не соответствует (где вместо ###- указывается название файла).

Проверка осуществляется путем открытия файла с исходным кодом в среде Free Pascal 3.6, осуществление компиляции (обе работы делаются в соответствии с документацией к Free Pascal). В случае, если компиляция завершилась успешно в протокол заносится запись о соответствии требованиям, в противном случае – о несоответствии.