

УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите теоретические сведения к лабораторной работе, выполните практическое задание, дайте ответы на контрольные вопросы.

Результаты работы, фотоотчет, предоставить преподавателю на e-mail: r.bigangel@gmail.com **до 20.02.2023.**

Требования к отчету:

Отчет предоставляется преподавателю в электронном варианте и должен содержать:

- название работы, постановку цели, вывод;
- ответы на контрольные вопросы, указанные преподавателем.

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: <https://vk.com/daykini>

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать **ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).**

Лабораторная работа № 29

Тема: «Функциональное тестирование»

Цель работы: провести функциональное тестирование разработанного программного средства в соответствии с заданным вариантом

Теоретическая часть. Тестирование программ

Процесс тестирования состоит из трёх этапов:

1. Проектирование тестов.
2. Исполнение тестов.
3. Анализ полученных результатов.

На первом этапе решается вопрос о выборе некоторого подмножества множества тестов, которое сможет найти наибольшее количество ошибок за наименьший промежуток времени. На этапе исполнения тестов проводят, запуск тестов и отлавливают ошибки в тестируемом программном продукте.

Виды тестов

Функциональные тесты составляются на уровне спецификации, до решения задачи. Будущий алгоритм рассматривается как «черный ящик» - функция с неизвестной (или не рассматриваемой) структурой, преобразующая входы в выходы. Суть функциональных тестов: каким бы способом ни решалась задача, при заданных входных значениях должны получиться соответствующие выходные значения.

Структурные тесты составляются для проверки логики решения, или логики работы уже готового алгоритма. Логика определяется последовательностью операций, их условным выполнением или повторением (т.е. композицией базовых конструкций). Совокупность структурных тестов должна обеспечить проверку каждой из таких конструкций.

Чаще всего совокупность тщательно составленных функциональных тестов покрывает множество структурных тестов.

Приведенные понятия различаются тем, что первое рассматривает программу только с точки зрения входов и выходов, тогда как второе относится к ее структуре; но оба понятия не касаются процесса организации тестирования.

Общая последовательность разработки тестов

Наиболее рациональная процедура заключается в том, что сначала разрабатываются функциональные тесты, а затем – структурные.

Функциональное тестирование (тестирование «черного ящика»)

При функциональном тестировании выявляются следующие категории ошибок:

- некорректность или отсутствие функций;
- ошибки интерфейса;
- ошибки в структурах данных;
- ошибки машинных характеристик (нехватка памяти и др.);
- ошибки инициализации и завершения.

Техника тестирования ориентирована:

- на сокращение необходимого количества тестовых вариантов;

- на выявление классов ошибок, а не отдельных ошибок.

Способы функционального тестирования

Разбиение на классы эквивалентности

Это самый популярный способ. Его суть заключается в разделении области входных данных программы на классы эквивалентности и разработке для каждого класса одного тестового варианта.

Класс эквивалентности – набор данных с общими свойствами, в силу чего при обработке любого набора данных этого класса задействуется один и тот же набор операторов.

Классы эквивалентности определяются по спецификации программы. Тесты строятся в соответствии с классами эквивалентности, а именно: выбирается вариант исходных данных некоторого класса и определяются соответствующие выходные данные.

Самыми общими классами эквивалентности являются классы допустимых и недопустимых (аномальных) исходных данных. Описание класса строится как комбинация условий, описывающих каждое входное данное.

Условия допустимости или недопустимости данных задают возможные значения данных и могут описывать:

- некоторое конкретное значение; определяется один допустимый и два недопустимых класса эквивалентности: заданное значение, множество значений меньше заданного, множество значений больше заданного;
- диапазон значений; определяется один допустимый и два недопустимых класса эквивалентности: множество значений в границах диапазона; множество значений, выходящих за левую границу диапазона; множество значений, выходящих за правую границу диапазона;
- множество конкретных значений; определяется один допустимый и один недопустимый класс эквивалентности: заданное множество и множество значений, в него не входящих.

Такие классы можно описать языком логики, например, языком исчисления предикатов. Описания более сложных условий и соответствующих классов (например, элементы массива должны находиться в некотором диапазоне и при этом массив не должен содержать нулевых элементов) могут быть построены на основании приведенных выше условий.

В примере, приводимом в вопросе 2 темы 3, при построении тестов неформально использовался описанный метод. В методических целях были выделены только основные классы тестов. Кроме того, исходя из условия задачи, были выделены классы эквивалентности внутри класса правильных данных.

Анализ граничных значений

Этот способ построения тестов дополняет предыдущий и предполагает анализ значений, лежащих на границе допустимых и недопустимых данных. Построение таких тестов часто диктуется интуицией.

Основные правила построения тестов:

- если условие правильности данных задает диапазон, то строятся тесты для левой и правой границы диапазона; для значений чуть левее левой и чуть правее правой границы;
- если условие правильности данных задает дискретное множество значений, то строятся тесты для минимального и максимального значений; для значений чуть меньше минимума и чуть больше максимума;
- если используются структуры данных с переменными границами (массивы), то строятся тесты для минимального и максимального значения границ.

Диаграммы причин-следствий

Взаимосвязь классов эквивалентности и соответствующих им действий описывается формально в виде графа на основе автоматного подхода. Граф преобразуется в таблицу решений, столбцы которой в свою очередь преобразуются в тестовые варианты.

Порядок выполнения работы

- 1) По результатам лабораторных работ 1-4 разработать тестовые наборы для функционального тестирования (Приложение 1).
- 2) Провести тестирование программы и представить результаты в виде таблицы. (Приложение 1).
- 3) Выработать рекомендации для корректировки тестируемой программы.
- 4) Представить отчет по лабораторной работе для защиты.

Защита отчета по лабораторной работе

Отчет по лабораторной работе должен быть оформлен согласно требований и состоять из следующих структурных элементов:

1. титульный лист;
2. текстовая часть;

Текстовая часть отчета должна включать пункты:

- условие задачи;
- порядок выполнения
- полученные результаты.

Защита отчета по лабораторной работе заключается в предъявлении преподавателю полученных результатов, демонстрации полученных навыков в ответах на вопросы преподавателя.

Контрольные вопросы

1. Что такое тестирование ПС?
2. Чем тестирование отличается от отладки ПС?
3. Для чего проводится функциональное тестирование?
4. Что такое комплексное тестирование?
5. Каковы правила тестирования программы «как черного ящика»?
6. Как проводится тестирования программы по принципу «белого ящика»?

7.Что такое модульное тестирование?

8.Как осуществляется сборка программы при модульно тестировании?

Шаблон таблицы для представления результатов

Тест (значения для входных данных)	Ожидаемый результат (значения для выходных данных)	Фактический результат (полученные значения выходных данных)	Результат тестирования (успешно/неуспешно)