

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Соловьев Андрей Борисович  
Должность: Директор  
Дата подписания: 27.09.2023 13:12:21  
Уникальный программный ключ:  
с83cc511feb01f5417b9362d2700339df14e123



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)  
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО  
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
В Г. ТАГАНРОГЕ РОСТОВСКОЙ ОБЛАСТИ  
ПИ (филиал) ДГТУ в г. Таганроге**

ЦМК «Прикладная информатика»

**Практикум**

По выполнению практических работ

по профессиональному МДК:

МДК 05.03 Тестирование информационных систем

для специальности 09.02.07 Информационные системы и программирование,

*квалификации*

*«Специалист по информационным системам»*

Составители: А.А. Погорелов

Практикум по выполнению практических работ по профессиональному модулю:

МДК 05.03 Тестирование информационных систем. ПИ (филиала) ДГТУ в г.Таганроге, 2023г.

В практикуме кратко изложены теоретические вопросы, необходимые для успешного выполнения практических работ, рабочее задание и контрольные вопросы для самопроверки.

Предназначено для обучающихся по специальности 09.02.07 «Информационные системы и программирование». Квалификации выпускника: «Специалист по информационным системам»

Ответственный за выпуск:

Председатель ЦМК: \_\_\_\_\_ О.В. Андриян

## ПРАКТИЧЕСКАЯ РАБОТА № 1

### ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

**Цель:** получение навыков проведения функционального тестирования.

**Форма отчета:**

- выполнить задание;
- показать преподавателю;
- ответить на вопросы преподавателя.

**Время выполнения: 2 ч**

#### Теоретические вопросы

Особенности функционального тестирования программного обеспечения(тестирования «черного ящика»).

Ошибки, выявляемые при функциональном тестировании.

Задачи, решаемые при функциональном тестировании.

**Задание № 1.** Опишите методы формирования тестовых наборов при использовании стратегии "черного ящика":

|                                   |  |
|-----------------------------------|--|
| Эквивалентноеразбиение            |  |
| Анализграничныхзначений           |  |
| Анализпричинно-следственныхсвязей |  |
| Предположениеоошибке              |  |

**Задание № 2.** Пусть необходимо выполнить тестирование программы, определяющей точку пересечения двух прямых на плоскости. Попутно, она должна определять параллельность прямой одной из осей координат.

В основе программы лежит решение системы линейных уравнений

$$Ax + By = C \text{ и } Dx + Ey = F.$$

1. Используя метод эквивалентных разбиений, получаем для всех коэффициентов один правильный класс эквивалентности (коэффициент – вещественное число) и один неправильный (коэффициент – не вещественное число). Откуда можно предложить 7тестов:

- 1) все коэффициенты – вещественные числа;
- 2)– 7) поочередно каждый из коэффициентов – не вещественное число.

2. По методу граничных условий можно считать, что для исходных данных граничные условия отсутствуют (коэффициенты – "любые" вещественные числа); для результатов – получаем, что возможны варианты: единственное решение, прямые сливаются (множество решений), прямые параллельны (отсутствиерешений).

Следовательно, можно предложить тесты, с результатами внутри области и с результатами на границе.

3. По методу анализа причинно-следственных связей определяем множество условий.

а) для определения типа прямой;

б) для определения точки пересечения.

Выделяем три группы причинно-следственных связей (определение типа и существования первой линии, определение типа и существования второй линии, определение точки пересечения) и строим таблицы истинности.

К уже имеющимся тестам добавляются:

а) проверки всех случаев расположения обеих прямых – 6 тестов по первой прямой вкладываются в 6 тестов по второй прямой так, чтобы варианты не совпадали, – 6 тестов;

б) выполняется отдельная проверка несовпадения условия  $x\delta = 0$  или  $y = 0$  (в зависимости от того, какой тест был выбран по методу граничных условий) – тест также можно совместить с предыдущими 6 тестами;

4. По методу предположения об ошибке добавим тест, при котором все коэффициенты – нули.

Всего получили 20 тестов по всем четырем методикам. Если еще попробовать вложить независимые проверки, то возможно число тестов можно еще сократить.

**Задание № 3.** Разработать программу определения вида треугольника, заданного длинами его сторон: равносторонний, равнобедренный, прямоугольный, разносторонний.

| Номер теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
|-------------|------------------|--------------------------|---------------------|-------------------|-------|
|             |                  |                          |                     |                   |       |

Предлагаемые тесты свести в таблицу.

**Задание № 4.** Разработать программу решения уравнения  $ax^2 + bx + c = 0$ , где  $a, b, c$  – любые вещественные числа.

Предлагаемые тесты свести в таблицу.

| Номер теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
|-------------|------------------|--------------------------|---------------------|-------------------|-------|
|             |                  |                          |                     |                   |       |

**Задание № 5.** Оформить отчет.

## ПРАКТИЧЕСКАЯ РАБОТА № 2

### ТЕСТИРОВАНИЕ БЕЗОПАСНОСТИ

**Цель:** получение навыков тестирования безопасности информационной системы.

**Форма отчета:**

- выполнить задание;
- показать преподавателю;
- ответить на вопросы преподавателя.

**Время выполнения: 2 ч**

#### Теоретические вопросы

Тестирование восстановления.

Тестирование безопасности.

Технологии тестирования безопасности.

Тестирование безопасности – оценка уязвимости программного обеспечения к различным атакам.

Компьютерные системы очень часто являются мишенью незаконного проникновения. Под проникновением понимается широкий диапазон действий: попытки хакеров проникнуть в систему из спортивного интереса, мсть рассерженных служащих, взлом мошенниками для незаконной наживы. Тестирование безопасности проверяет фактическую реакцию защитных механизмов, встроенных в систему, на проникновение. В ходе тестирования безопасности испытатель играет роль взломщика. Ему разрешено все:

- попытки узнать пароль с помощью внешних средств;
- атака системы с помощью специальных утилит, анализирующих защиты;
- подавление, ошеломление системы (в надежде, что она откажется обслуживать других клиентов);
- целенаправленное введение ошибок в надежде проникнуть в систему в ходе восстановления;
- просмотр несекретных данных в надежде найти ключ для входа в систему.

При неограниченном времени и ресурсах хорошее тестирование безопасности взломает любую систему. Задача проектировщика системы – сделать цену проникновения более высокой, чем цена получаемой в результате информации.

**Задание № 1.** Изучите и опишите одно из средств выявления уязвимостей: Таблица 1. Обзор средств выявления уязвимостей, работающих на уровне кода

| Наименование средства                             | Назначение   | Поддерживаемые языки программирования | Примечание  |
|---|--|---------------------------------------|---|
| <b>Иностранные средства выявления уязвимостей</b> |  |                                       |   |
| Its4  | Статически просматривает исходный код для обнаружения потенциальных уязвимостей защиты | C/c++                                 | Отмечает вызовы потенциально опасных функций, таких, как strcpy/memcpy, и выполняет поверхностный семантический анализ, |

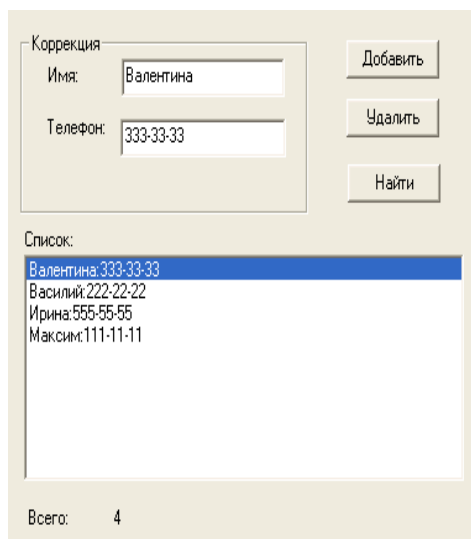
|  |   |                                 |   |
|--|---|---------------------------------|---|
|  |   |                                 | <p>пытаясь оценить, насколько опасен такой код, а также дает советы по его улучшению</p>  |
| Rats(rough auditing tool for security) | <p>Просматривает исходный текст, находя потенциально опасные обращения к функциям</p>     | <p>C/c++, php, perl, python</p> | <p>Использует сочетание проверок надежности защиты от семантических проверок в its4 до глубокого семантического анализа в поисках дефектов, способных привести к переполнению буфера, полученных из tops</p>  |
| Flawfinder                             | <p>Просматривает исходный текст, находя потенциально опасные обращения к функциям</p>     | <p>C/c++</p>                    | <p>Выполняет поиск функций, которые чаще всего используются некорректно, присваивает им коэффициенты риска (опираясь на такую информацию, как передаваемые параметры) и составляет список потенциально уязвимых мест, упорядочивая их по степени риска</p>  |
| Flexelint (pc-lint)                    | <p>Производит семантический анализ исходного кода, анализ потоков данных и управления</p> | <p>C/c++</p>                    | <p>В конце работы выдаются сообщения нескольких основных типов:</p> <ul style="list-style-type: none"> <li>– возможен нулевой указатель – проблемы с выделением памяти (например, нет free() после malloc()) – проблемный поток управления (например, недостижимый код);</li> <li>– возможно переполнение буфера, арифметическое переполнение;</li> </ul> |

| Наименование средства | Назначение  | Поддерживаемые языки программирования | Примечание   |
|-----------------------|---|---------------------------------------|--|
|                       |   |                                       | – предупреждения о плохом и потенциально опасном стиле кода  |
| Parasoftc++ test      | Формирование тестов анализа уязвимостей на уровне метода, класса, файла и проекта   | C++                                   | Генерирует тестовый код, вызывая для его подготовки компилятор visualc++   |
| Coverity              | Используется для выявления и исправления дефектов безопасности и качества в приложениях критического назначения   | C/c++, java                           | Способен с минимальной положительной погрешностью обрабатывать десятки миллионов строк кода, обеспечивая 100- процентное покрытие трассы |
| Klocwork k7           | Предназначен для автоматизированного статического анализа кода, выявления и предотвращения дефектов программного обеспечения и проблем безопасности                         | C/c++, java                           | Выявляет коренные причины недостатков качества и безопасности программного обеспечения   |
| Codesurfer            | Может применяться для поиска ошибок в исходном коде, для улучшения понимания исходного кода   | C/c++                                 | Позволяет проводить анализ указателей, использовать и определять переменные, зависимости данных, строить графы вызовов                   |
| Fxcop                 | Способен обнаружить более 200 недочетов (или ошибок) в следующих областях:<br>– архитектура библиотеки;<br>– правила именования;<br>– производительность;<br>– безопасность | C/c++                                 | Откомпилированный код проверяется с помощью механизмов рефлексии, парсинга msil и анализа графа вызовов                                  |
| Qaudit                | Быстрый анализ исходных файлов на наличие переполнения буфера, ошибок форматной строки, запросов  | C/c++                                 | Написать на интерпретируемом языке perl, прост в использовании   |

|   |  |   |   |
|---|--|---|---|
|   | исполняемых вызовов, переменных среды, и функций, имеющих проблемы защиты  |   |   |
| Российские средства выявления уязвимостей |  |   |   |
| Ак-вс                                     | Автоматизированный анализ исходных текстов, с целью выявления потенциально опасных сигнатур                                | C/c++, java, pascal, c#, php, assembler | Позволяет проводить статический анализ исходных текстов, динамический анализ, имеет базы сигнатур для каждого из поддерживаемых языков программирования |
| Аист-с                                    | Автоматизированный анализ исходных текстов   | C/c++                                   | Позволяет проводить статический анализ исходных текстов   |
| Ксайт                                     | Автоматизированный анализ исходных текстов   | C/c++                                   | Позволяет проводить статический анализ исходных текстов   |
| Уса                                       | Предназначено для выявления потенциально опасных сигнатур  | C/c++, pascal, perl, plm                | Имеет базы сигнатур для каждого из поддерживаемых языков программирования   |
| Viva64                                    | Помогает отслеживать в исходном коде потенциально опасные фрагменты, связанные с переходом от 32-битных систем к 64-битным | C/c++                                   | Помогает писать корректный и оптимизированный код для 64-битных систем  |



**Задание № 2.** Разработать приложение, интерфейс которого представлен на рисунке.



Коррекция

Имя: Валентина

Телефон: 333-33-33

Добавить

Удалить

Найти

Список:

Валентина:333-33-33

Василий:222-22-22

Ирина:555-55-55

Максим:111-11-11

Всего: 4

**Задание № 3.** Добавить в программу форму авторизации по имени и паролю.

### ПРАКТИЧЕСКАЯ РАБОТА № 3

#### НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ, СТРЕССОВОЕ ТЕСТИРОВАНИЕ

**Цель:** получение навыков проведения нагрузочного и стрессового тестирования.

**Форма отчета:**

- выполнить задание;
- показать преподавателю;
- ответить на вопросы преподавателя.

**Время выполнения: 2 ч**

#### Теоретические вопросы

Особенности нагрузочного тестирования.

Особенности стрессового тестирования.

**Задание № 1.** Разработать Компилятор простых арифметических выражений, например  $2+(-5)*(7-8)$ . Вход и выход осуществляются в виде строк.

**Задание № 2.** Разработать тестовый сценарий нагрузочного тестирования. Ответить на вопрос – сколько запросов в секунду может обработать приложение при условии, что они идут последовательно. Построить график зависимости времени ответа от количества параллельных запросов (рассматривать логарифмическую шкалу по основанию два, т.е. 1, 2,4,8,16,32 и т.д. запроса) Ответить на вопрос – какое максимальное количество параллельных запросов может обработать приложение без сбоев.

**Задание № 3.** Оформить отчет.

### ПРАКТИЧЕСКАЯ РАБОТА № 4

## ТЕСТИРОВАНИЕ ИНТЕГРАЦИИ

**Цель:** получение навыков тестирования интеграции.

**Форма отчета:**

- выполнить задание;
- показать преподавателю;
- ответить на вопросы преподавателя.

**Время выполнения: 2 ч**

### Теоретические вопросы

Особенности тестирования интеграции.

Методы интеграционного тестирования.

Нисходящее тестирование интеграции.

Восходящее тестирование интеграции.

Сравнение нисходящего и восходящего тестирования интеграции

**Задание № 1.** Разработать приложение, состоящее из трех модулей:

- 1) главный модуль, считывающий из текстового файла координаты точек на плоскости;
- 2) модуль, содержащий функции расчета расстояния между двумя точками;
- 3) модуль, содержащий функцию, определяющую треугольник с максимальной площадью. **Задание № 2.** Описать этапы нисходящего проектирования разработанного приложения. **Задание № 3.** Описать этапы восходящего проектирования, разработанного приложений. **Задание № 4.** Оформить отчет.

## ПРАКТИЧЕСКАЯ РАБОТА № 5

### КОНФИГУРАЦИОННОЕ ТЕСТИРОВАНИЕ

**Цели:** получение навыков проведения конфигурационного тестирования.

**Форма отчета:**

- выполнить задание;
- показать преподавателю;
- ответить на вопросы преподавателя.

**Время выполнения: 2 ч**

### Теоретические вопросы

Особенности конфигурационного тестирования.

Конфигурационное тестирование (Configuration testing). Проверяется работоспособность при различных конфигурациях, предполагает тестирование работы системы на различных платформах: различных вариантах аппаратной конфигурации, версиях операционной системы и окружения.

**Задание № 1.** Дана структура с именем ZNAK, состоящая из полей:

- фамилия, имя;
- знак Зодиака;
- дата рождения (массив из трех чисел).

Написать программу, которая выполняет следующие действия:

- ввод с клавиатуры данных в массив, состоящий из 8 элементов типа ZNAK, и занесение их в файлданных;
- чтение данных из файла и вывод их наэкран;
- вывод на экран информации о людях, родившихся в месяц, значение которого введено с клавиатуры (если таких нет – вывести об этомсообщение);
- список должен быть упорядочен по знакам Зодиака.

**Задание № 2.** Описать и обосновать итоги тестирования работы разработанного приложения на различных платформах: различных вариантах аппаратной конфигурации, версиях операционной системы иокружения.

## ПРАКТИЧЕСКАЯ РАБОТА № 6 ТЕСТИРОВАНИЕ УСТАНОВКИ

**Цель:** получение навыков тестирования установки.

**Форма отчета:**

- выполнить задание;
- показать преподавателю;
- ответить на вопросы преподавателя.

**Время выполнения: 2 ч**

Теоретические вопросы

Комплексное тестирование приложения.

**Задание № 1.** Разработать приложение, интерфейс которого представлен на рисунке 1.

**Задание № 2.** Провести комплексное тестирование разработанного приложения.

**Задание № 3.** Оформить отчет.

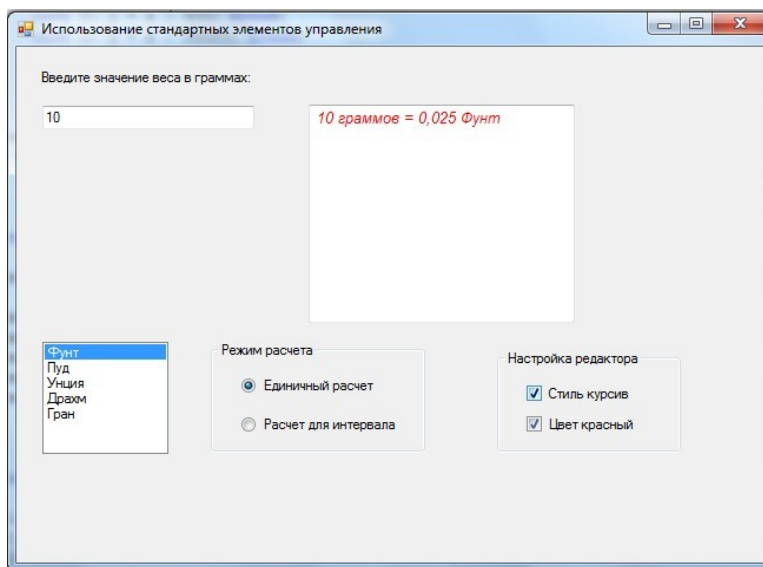


Рисунок 1



## Список литературы

1. *Волкова, В. Н.* Теория информационных процессов и систем : учебник и практикум для академического бакалавриата / В. Н. Волкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2019. — 432 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-05621-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/432843>
2. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук ; под общей редакцией Д. В. Чистова. — Москва : Издательство Юрайт, 2019. — 258 с. — (Профессиональное образование). — ISBN 978-5-534-03173-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/437463>
3. *Зараменских, Е. П.* Информационные системы: управление жизненным циклом : учебник и практикум для среднего профессионального образования / Е. П. Зараменских. — Москва : Издательство Юрайт, 2019. — 431 с. — (Профессиональное образование). — ISBN 978-5-534-11624-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/445765>
4. *Черткова, Е. А.* Компьютерные технологии обучения : учебник для вузов / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 250 с. — (Университеты России). — ISBN 978-5-534-07491-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/437244>
5. *Маркин, А. В.* Программирование на SQL в 2 ч. Часть 2 : учебник и практикум для вузов / А. В. Маркин. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 340 с. — (Высшее образование). — ISBN 978-5-534-12258-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/448191>