Государственное автономное профессиональное образовательное учреждение Чувашской Республики «Чебоксарский экономико-технологический колледж» Министерства образования и молодежной политики Чувашской Республики

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ЛАБОРАТОРНЫМ ЗАНЯТИЯМ**

**ПМ.05 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ**

**МДК. 05.03 ТЕСТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

**Специальность среднего профессионального образования**

**09.02.07 Информационные системы и программирование**

`

Чебоксары 2023

Методические указания содержат комплект материалов для лабораторных занятий по специальности 09.02.07 Информационные системы и программирование. Методические указания ориентированы на закрепление и углубление знаний, отработку профессиональных навыков МДК. 05.03 Тестирование информационных систем. Указания предназначены для преподавателей и студентов специальности 09.02.07 Информационные системы и программирование

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ …………………………………………………………………….……….4](#_Toc482193866)

[ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНЫХ ЗАНЯТИЙ СТУДЕНТОВ ………….………..5](#_Toc482193867)

# ТЕМАТИЧЕСКИЙ ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ…………..………….….....7

ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ ЗАДАНИЙ………………….…...…8

[ЗАКЛЮЧЕНИЕ……………………………………………………………………….....](#_Toc482193870)24

[ЛИТЕРАТУРА……………………………………………………………………..……](#_Toc482193871)25

# 

# ВВЕДЕНИЕ

Методические указания разработаны в соответствии с рабочей программой по МДК. 05.03 Тестирование информационных систем. для студентов специальности 09.02.07 «Информационные системы и программирование».

Методические указания предназначены для организации учебного процесса, а также подготовки и проведению лабораторных занятий и их проверки.

Лабораторные задания предназначены для закрепления теоретического материала по МДК. 05.03 Тестирование информационных систем. и выработки навыков его применения в практических и лабораторных расчетах.

Практические и лабораторные занятия являются важными видами учебной работы студента по учебной дисциплине и выполняются в пределах часов, предусмотренных учебным планом специальности.

Цель данных методических указаний состоит в оказании помощи студентам при проведении лабораторных занятий по изучению данной дисциплины, в формировании готовности к овладению основными умениями, знаниями, а также развитие общих компетенций по специальности.

**ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНЫХ ЗАНЯТИЙ СТУДЕНТОВ**

Практические занятия предназначены для более глубокого изучения дисциплины, развивает мышление, позволяет выявить причинно-следственные связи в изученном материале, решить теоретические и практические задачи. Для формирования у обучающихся умений, приобретение первоначального практического опыта в рамках профессионального модуля по каждому из видов профессиональной деятельности под руководством преподавателей учетных дисциплин.

Целью методических указаний является повышение эффективности учебного процесса, в том числе благодаря выполнению лабораторных заданий, в котором студент становится активным субъектом обучения, что означает:

- способность занимать в обучении активную позицию;

- готовность мобилизовать интеллектуальные и волевые усилия для достижения учебных целей;

- умение проектировать, планировать и прогнозировать учебную деятельность;

- формирование умений использовать справочную документацию и специальную литературу;

- осознание своих потенциальных способностей и развитие исследовательских умений;

- психологическую готовность составить программу действий по саморазвитию, самосовершенствованию и самореализации.

Для достижения этой цели по каждой теме разработаны лабораторные задания с указаниями по их выполнению.

Лабораторные занятия по специальности направлены на формирование у обучающихся умений, приобретение первоначального практического опыта и реализуется в рамках профессиональных модулей ОПОП СПО по основным видам профессиональной деятельности для последующего освоения ими общих и профессиональных компетенций по избранной специальности:

|  |  |
| --- | --- |
| **Код** | **Наименование результата обучения** |
| ВД 5 | Проектирование и разработка информационных систем |
| ПК 5.1 | Собирать исходные данные для разработки проектной документации на информационную систему. |
| ПК 5.2 | Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика |
| ПК 5.3 | Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием |
| ПК 5.4 | Производить разработку модулей информационной системы в соответствии с техническим заданием |
| ПК 5.5 | Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы |
| ПК 5.6 | Разрабатывать техническую документацию на эксплуатацию информационной системы |
| ПК 5.7 | Производить оценку информационной системы для выявления возможности ее модернизации. |
| ОК 1 | Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; |
| ОК 2 | Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; |
| ОК 3 | Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях; |
| ОК 4 | Эффективно взаимодействовать и работать в коллективе и команде; |
| ОК 5 | Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста; |
| ОК 6 | Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения; |
| ОК 7 | Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях |
| ОК 8 | Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности; |
| ОК 9 | Пользоваться профессиональной документацией на государственном и иностранном языках |

# ТЕМАТИЧЕСКИЙ ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Таблица 1 - Тематический план лабораторных занятий

|  |  |
| --- | --- |
| Тема лабораторного занятия | Объем в часах |
| Лабораторное занятие № 1. Тестирование пользовательского интерфейса | 2 |
| Лабораторное занятие № 2. Тестирование баз данных | 2 |
| Лабораторное занятие № 3. Тестирование авторизации | 2 |
| Лабораторное занятие № 4. Разработка тестового сценария проекта | 2 |
| Лабораторное занятие № 5. Разработка тестовых пакетов | 2 |
| Лабораторное занятие № 6. Использование инструментария анализа качества | 2 |
| Лабораторное занятие № 7. Анализ исключительных ситуаций | 2 |
| Лабораторное занятие № 8. Обеспечение обработки исключительных ситуаций | 2 |
| Лабораторное занятие № 9. Функциональное тестирование | 2 |
| Лабораторное занятие № 10. Функциональное тестирование | 2 |
| Лабораторное занятие № 11. Нагрузочное тестирование | 2 |
| Лабораторное занятие № 12. Стрессовое тестирование Тестирование интеграции | 2 |
| Лабораторное занятие № 13. Конфигурационное тестирование Тестирование установки | 2 |

**ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ ЗАДАНИЙ**

**Лабораторное занятие № 1. Тестирование пользовательского интерфейса**

**Цель:** получить навыки разработки тестовых сценариев.

Теоретические вопросы

* Оценка стоимости и причины ошибок в программном обеспечении.
* Виды и методы тестирования.
* Понятие теста.
* Требования к разработке тестовых сценариев.
* Правила разработки тестовых сценариев.

Задание № 1

Написать программу решения квадратного уравнения ах2 + bх + с = 0.

Задание № 2

Найти минимальный набор тестов для программы нахождения веще-ственных корней квадратного уравнения ах2 + bх + с = 0. Решение представлено в таблице.



Таким образом, для этой программы предлагается минимальный набор функциональных тестов, исходя из 7 классов выходных данных.

Заповеди по отладки программного средства, предложенные Г. Майерсом.

*Заповедь 1.* Считайте тестирование ключевой задачей разработки ПС, поручайте его самым квалифицированным и одаренным программистам, нежелательно тестировать свою собственную программу.

*Заповедь 2.* Хорош тот тест, для которого высока вероятность обнаружить ошибку, а не тот, который демонстрирует правильную работу программы.

*Заповедь 3.* Готовьте тесты как для правильных, так и для неправильных данных.

*Заповедь 4.* Документируйте пропуск тестов через компьютер, детально изучайте результаты каждого теста, избегайте тестов, пропуск которых нельзя повторить. Заповедь 5. Каждый модуль подключайте к программе только один раз, никогда не изменяйте программу, чтобы облегчить ее тестирование.

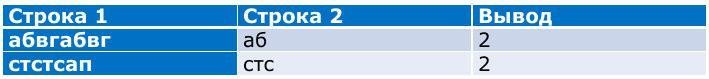
*Заповедь 6.* Пропускайте заново все тесты, связанные с проверкой работы какой-либо программы ПС или ее взаимодействия с другими программами, если в нее были внесены изменения (например, в результате устранения ошибки).

Задание № *3*

Разработайте набор тестовых сценариев (как позитивных, так и негативных) для следующей программы:

Имеется консольное приложение (разработайте самостоятельно). Ему на вход подается 2

строки. На выходе приложение выдает число вхождений второй строки в первую. Например:



Набор тестовых сценариев запишите в виде таблицы, приведенной выше.

Задание № 4

Оформить отчет.

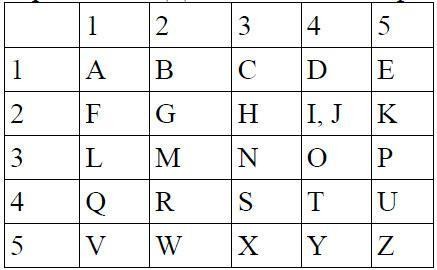
**Лабораторное занятие № 2. Тестирование баз данных Цель:** получить навыки разработки тестовых пакетов.

Теоретические вопросы

* Системные основы разработки требований к сложным комплексам программ.
* Формализация эталонов требований и характеристик комплекса программ.
* Формирование требований компонентов и модулей путем декомпозиции функций комплексов программ.
* Тестирование по принципу «белого ящика».

Задание № 1

В Древней Греции (II в. до н.э.) был известен шифр, называемый "квадрат Полибия". Шифровальная таблица представляла собой квадрат с пятью столбцами и пятью строками, которые нумеровались цифрами от 1 до 5. В каждую клетку такого квадрата записывалась одна буква. В результате каждой букве соответствовала пара чисел, и шифрование сводилось к замене буквы парой чисел. Для латинского алфавита квадрат Полибия имеет вид:



Пользуясь изложенным способом создать программу, которая: а) зашифрует введенный текст и сохранит его в файл;

б) считает зашифрованный текст из файла и расшифрует данный текст.

Задание № 2

Спроектировать тесты по принципу «белого ящика» для программы, разработанной в задании № 1. Выбрать несколько алгоритмов для тестирования и обозначить буквами или цифрами ветви этих алгоритмов. Выписать пути алгоритма, которые должны быть проверены тестами для выбранного метода тестирования. Записать тесты, которые позволят пройти по путям алгоритма. Протестировать разработанную вами программу. Результаты оформить в виде таблиц:

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| … | … | … | … |

Задание № 3

Проверить все виды тестов и сделать выводы об их эффективности

Задание № 4

Оформить отчет.

**Лабораторное занятие № 3. Тестирование авторизации**

**Цель:** получить навыки использования инструментария анализа качества.

Теоретические вопросы

* Общие требования к качеству функционирования сложных программных комплексов.
* Требования к характеристикам качества сложных программных комплексов.
* Требования к эффективности использования ресурсов ЭВМ программным комплексом в реальном времени.
* Проверка корректности функциональных требований к сложным комплексам программ.

Задание № 1

Написать программу, генерирующую массив вещественных чисел в диапазоне от –10 до 10

и определяющую все минимальные положительные элементы.

Задание № 2.

Оценить эффективность разработанной программы:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Исходная программа | | Улучшенная программа | |
| Недостатки | Количественная оценка | Улучшения | Количественная оценка |
| Время |  |  |  |  |
| выполнения |  |  |  |  |
| Оперативная |  |  |  |  |
| память |  |  |  |  |
| Внешняя |  |  |  |  |
| память |  |  |  |  |

Задание № 3

Оценить качество разработанной программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Правильность | Универсальность | Проверяемость | Точность результатов |
| Недостатки |  |  |  |  |
| Оценка |  |  |  |  |

Задание № 4

Оформить отчет.

**Лабораторное занятие № 4, №5. Разработка тестового сценария проекта. Разработка тестовых пакетов**

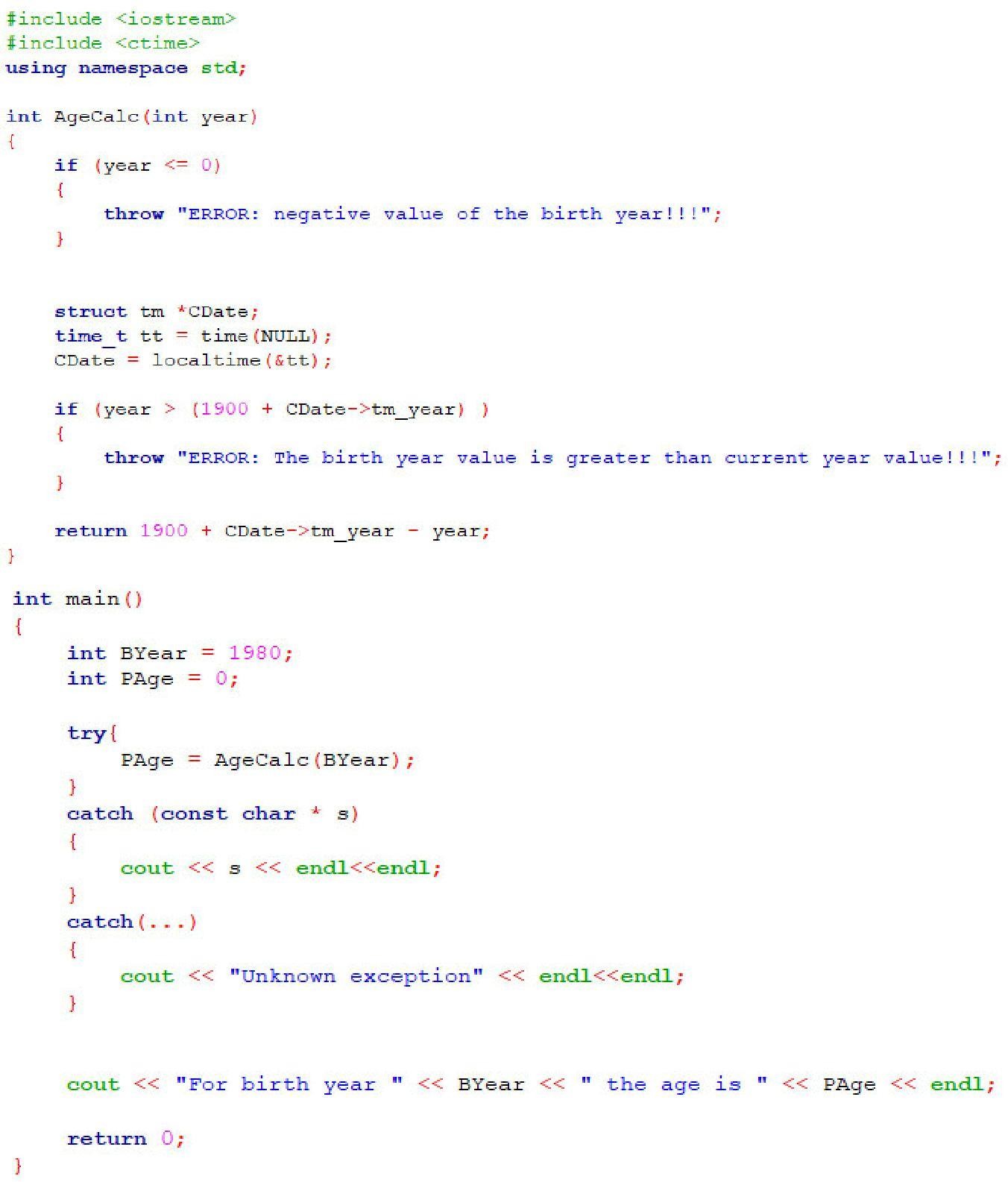
Цели: получение навыков анализа и обеспечения обработки исключительных ситуаций.

Теоретические вопросы

* Исключения в С++.
* Установленные исключения.
* Спецификация исключения.

Задание № 1

Написать программу, в которой обрабатываются следующие исключительные ситуации: "отрицательное значение возраста" и "год рождения больше текущего":



Задание № 2

Составить программу циклического вычисления значений функций, определенных из таблицы вариантов заданий. Значения R должны вводиться с клавиатуры. R1 и R2 –вещественные, R3 –комплексное. Предусмотреть вывод подсказок в виде (например):

Funkciya sin(x)

Q –Vyihod iz programmyi Vvedite chislo ili Q:

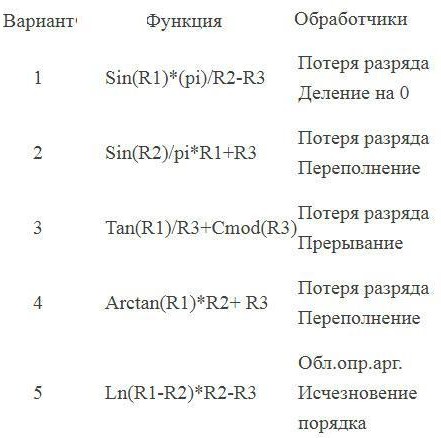
Для вычисления значений функции написать функцию, вычисляющую требуемые по заданию значения. При разработке функции разрешается использовать функции модуля math.h.

Предусмотреть анализ всей введенной информации на ошибки, обработку ошибок реализовать с использованием с использованием обработчиков try… в зависимости от варианта задания. Предусмотреть вывод имени функции, в которой произошла ошибка. Вывод на экран и чтение с клавиатуры организовать при помощи стандартных потоков ввода/вывода/ошибки.

Вывести исходные данные и результат в виде (например):

Sin(R) = rez;

Где rez –результаты вычисления (вещественный).



Задание № 3

Реализуйте класс «очередь» из строк. Реализуйте методы для вставки в очередь удаления.

Породите и обработайте ошибки динамического выделения памяти, переполнения очереди.

Задание № 4

Оформите отчет.

**Лабораторное занятие № 6. Использование инструментария анализа качества**

**Цель:** получение навыков проведения инструментария анализа качества

**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":

|  |  |
| --- | --- |
| Эквивалентное разбиение |  |
| Анализ граничных значений |  |
| Анализ причинно-следственных связей |  |
| Предположение об ошибке |  |

Задание № 2

Пусть необходимо выполнить тестирование программы, определяющей точку пересечения двух прямых на плоскости. Попутно, она должна определять параллельность прямой одной их осей координат.

В основе программы лежит решение системы линейных уравнений Ax + By = C 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 7 тестов:

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

1. 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 (отсутствие решений).

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

1. По методу анализа причинно-следственных связей определяем множество условий. а) для определения типа прямой;

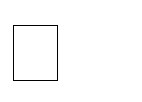
б) для определения точки пересечения.

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 0 или y = 0 (в зависимости



=

от того, какой тест был выбран по методу граничных условий) – тест также можно совместить с предыдущими 6 тестами;

1. По методу предположения об ошибке добавим тест, при которомвсе 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.

Предлагаемые тесты свести в таблицу.

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

Задание № 4

Разработать программу решения уравнения ax2 + 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.

|  |
| --- |
| **Лабораторное занятие № 7, № 8 Анализ исключительных ситуаций** |
| **Обеспечение обработки исключительных ситуаций** |

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

**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, месть 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:

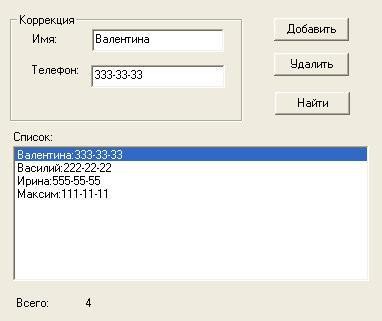
Таблица 1. Обзор средств выявления уязвимостей, работающих на уровне кода

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование средства | Назначение | Поддерживаем ые языки программирова  ния | Примечание |
|  | Иностранные средства выявления уязвимостей | | |
| Its4 | Статически просматривает исходный код для обнаружения потенциальных уязвимостей защиты | С/с++ | Отмечает вызовы потенциально опасных функций, таких, как strcpy/memcpy, и выполняет поверхностный семантический анализ, пытаясь оценить, насколько опасен такой код, а также дает советы по его улучшению |
| Rats(rough auditing tool for security) | Просматривает исходный текст, находя потенциально опасные обращения к  функциям | С/с++, php, perl, python | глубокого семантического анализа в поисках дефектов, способных привести к переполнению буфера,  полученных из mops |
| Flawfinder | Просматривает исходный текст, находя потенциально опасные обращения к функциям | С/с++ | 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 |

|  |  |  |  |
| --- | --- | --- | --- |
| Flexelint (pc- lint) | Производит семантический анализ исходного кода, анализ потоков данных и управления | С/с++ | В конце работы выдаются сообщения нескольких основных типов:   * возможен нулевой указатель * проблемы с выделением памяти (например, нет free() после malloc()) * проблемный поток управления (например, недостижимый код); * возможно переполнение буфера, арифметическое переполнение; * предупреждения о плохом и потенциально опасном стиле кода |
| Parasoft c++ test Coverity | Формирование тестов анализа уязвимостей на уровне метода, класса, файла и проекта Используется для выявления и исправления дефектов безопасности и  качества в  приложениях критического назначения | С++  С/с++, java | Генерирует тестовый код, вызывая для его подготовки компилятор visual c++  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 | С/с++, java | Выявляет коренные причины недостатков качества и безопасности программного обеспечения |
| Codesurfer | Может применяться для поиска ошибок в исходном коде, для улучшения понимания исходного кода | С/с++ | Позволяет проводить анализ указателей, использовать и определять переменные, зависимости данных, строить графы вызовов |
| Fxcop | Способен обнаружить более  200 недочетов (или ошибок) в  следующих областях:   * архитектура библиотеки; * правила   именования;  –  производительность;   * безопасность | С/с++ | 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 исполняемых вызовов, переменных среды, и функций,  имеющих проблемы защиты | С/с++ | Написать на интерпретируемом языке perl, прост в использовании |
|  | Российские средства выявления уязвимостей | | |
| Ак-вс | Автоматизированный анализ исходных текстов, с целью выявления потенциально опасных сигнатур | С/с++, 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 | С/с++ | Позволяет проводить статический анализ исходных текстов |
| Ксаит | Автоматизированный  анализ исходных текстов | С/с++ | Позволяет проводить статический анализ исходных текстов |
| Uca | Предназначено для выявления потенциально  опасных сигнатур | С/с++, 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 | С/с++ | Помогает писать корректный и оптимизированный код для 64- битных систем |

Задание № 2

Разработать приложение, интерфейс которого представлен на рисунке.



Задание № 3

Добавить в программу форму авторизации по имени и паролю.

**Лабораторное занятие № 9, № 10. Функциональное тестирование**

**Цель:** получение навыков функционального тестирования.

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.

**Лабораторное занятие № 11. Нагрузочное тестирование**

**Цель:** получение навыков тестирования интеграции.

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 функию, 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.

**Лабораторное занятие № 12. Стрессовое тестирование Тестирование интеграции**

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

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 файл данных;
* чтение данных из файла и вывод их на экран;
* вывод на экран информации о людях, родившихся в месяц, значе-ние которого введено с клавиатуры (если таких нет – вывести об этом со-общение);
* 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 и окружения.

**Лабораторное занятие № 13. Конфигурационное тестирование Тестирование установки**

**Цель:** получение навыков тестирования установки.

Теоретические вопросы

* Комплексное тестирование приложения.

Задание №1

Разработать приложение, интерфейс которого представлен на рисунке 1.

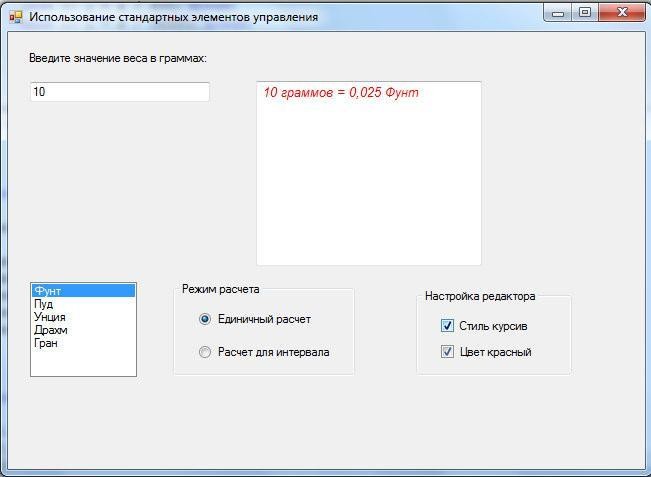


Рисунок 1. Интерфейс приложения

Задание №2

Провести комплексное тестирование разработанного приложения.

Задание №3

Оформить отчет.

**ЗАКЛЮЧЕНИЕ**

Целью выполнения лабораторных занятий студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю специальности, опытом творческой, исследовательской деятельности. Практическая работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами лабораторных занятий являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;

- углубление и расширение теоретических знаний;

- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;

- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;

- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;

- развитие исследовательских умений;

- использование материала, собранного и полученного в ходе лекций, на практических занятиях, в ходе самостоятельной работы при написании курсовой работы, а также для эффективной подготовки к итоговым зачетам и экзаменам.

Контроль и оценка результатов освоения осуществляется преподавателем в процессе проведения лабораторных занятий, а также выполнения студентом индивидуальных заданий, проектов, исследований.

**ЛИТЕРАТУРА**

1. Федорова, Г.И. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие. – Москва: КУРС, 2021. – 336 с.
2. Васильев Р.Б. Управление развитием информационных систем : учебник / Васильев Р.Б., Калянов Г.Н., Левочкина Г.А.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 507 c. — ISBN 978-5-4497-0561-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/94864.html (дата обращения: 13.12.2021). — Режим доступа: для авторизир. пользователей.