

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
ДЛЯ ПРОВЕДЕНИЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ
по дисциплине
Тестирование программного обеспечения

для студентов направления подготовки
01.04.02 Прикладная математика и информатика

Содержание

		Стр.
1.	Введение	
2.	Практическая работа №1 «Методика анализа и оптимизации программного обеспечения»	
3.	Практическая работа №2 «Современные методы разработки качественного программного обеспечения ЭВМ»	
4.	Практическая работа №3 «Способы повышения производительности базовых алгоритмов»	
5.	Литература	
6.	Приложение 1. Оборудование, материалы и программное обеспечение, используемые для организации и проведения Практических работ	
7.	Приложение 2. Содержимое отчета по практической работе ...	
8.	Приложение 3. Правила поведения в компьютерном классе (аудитории, лаборатории)	

Введение

Дисциплина «Тестирование программного обеспечения» относится к вариативной части Б1.В.ДВ.03.01 учебного плана магистерской программы 2021 года и изучается студентами магистратуры специальности 01.04.02 направления «Прикладная математика и информатика» в 1 семестре.

Целью освоения дисциплины является формирование представлений и фундаментальных знаний о современных методах тестирования программного обеспечения.

Основной задачей изучения дисциплины является изучение методов тестирования программного обеспечения.

Формы отчетности: экзамен - 1 семестр.

Содержание дисциплины: основные принципы тестирования компьютерных программ, методику анализа и оптимизации программного обеспечения, современные методы разработки качественного программного обеспечения ЭВМ, способы повышения производительности базовых алгоритмов.

Реализуемые компетенции:

ПК-2	Способен разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение
ПК-3	Способен осуществлять планирование организации разработки и интеграции системного программного обеспечения

Результаты освоения дисциплины.

Знать: основные принципы тестирования компьютерных программ, методику анализа и оптимизации программного обеспечения, современные методы разработки качественного программного обеспечения ЭВМ, способы повышения производительности базовых алгоритмов.

Уметь: применять на практике методы тестирования программного обеспечения ЭВМ

Владеть: навыками использования специализированного программного обеспечения, предназначенного тестирования программ для ЭВМ

Рекомендуемое программное обеспечение:

1. Интегрированная среда разработки программ Lazarus (не ниже 1.4)
2. Операционная система: Windows (не менее XP), Linux Mint не ниже 17.2

Рекомендуемое материально-техническое обеспечение: компьютерный класс (лаборатория, аудитория) с числом исправных ЭВМ по числу студентов группы плюс рабочее место преподавателя и производительностью, достаточной для освоения дисциплины. На всех ЭВМ класса (лаборатории, аудитории) программное обеспечение, выход в интернет для изучения интернет-источников дисциплины (со скоростью доступа, достаточной для комфортной работы).

лабораторная работа №1

Тема: «Методика анализа и оптимизации программного обеспечения»

Цель работы: «закрепление знаний по теоретическим основам анализа и оптимизации программного обеспечения».

Формируемые компетенции (формируется часть компетенций):

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Ход работы

1. Орг. момент.
2. Выбор задания в соответствии с вариантом.
3. Изучение теоретической части (литературы) по теме работы.
4. Изучение техники безопасности (правил поведения в компьютерной лаборатории).
5. Выполнение работы.
 - загрузить шаблон программы.
 - внести в шаблон необходимые изменения.
 - отладить полученную компьютерную программу.
6. Оформление результатов работы.
7. Выводы по проделанной работе.
8. Защита работы.

Теоретическая часть

См. список рекомендуемой литературы.

Оборудование и материалы: см. приложение 1.

Техника безопасности: см. приложение 3.

Задания для практической работы

Задача 1.

Рассчитать число элементарных шагов работы алгоритма.

Алгоритм для анализа взять в соответствии с вариантом.

Задача 2.

Рассчитать объем памяти, требуемый для работы алгоритма.

Алгоритм для анализа взять в соответствии с вариантом.

Задача 3.

Оптимизировать временную сложность алгоритма (уменьшить число шагов путем оптимизации вызовов подпрограмм, циклов и ветвлений, расчетных формул).

Алгоритм для оптимизации взять в соответствии с вариантом.

Задача 4.

Оптимизировать объемы запрашиваемой алгоритмом памяти путем выявления и отказа от неэффективно используемых массивов, оптимизации вызовов подпрограмм, циклов и ветвлений, расчетных формул).

Алгоритм для оптимизации взять в соответствии с вариантом.

Задача 5.

Оптимизировать временную сложность алгоритма путем поиска и переписывания участков кода, пригодных для распараллеливания.

Подпрограмму для анализа и оптимизации взять в соответствии с вариантом.

Задача 6.

Перевести участок кода программы с высокоуровневого языка программирования на язык ассемблера.

Подпрограмму для оптимизации взять в соответствии с вариантом.

Задача 7.

Построить подробный алгоритм работы участка кода программы, записанного на языке ассемблера. Выявить в нем более высокоуровневые алгоритмические структуры. Переписать подпрограмму на высокоуровневом языке.

Подпрограмму для анализа и реконструирования взять в соответствии с вариантом.

Задача 8.

Перевести на язык ассемблера готовую программу (исполняемый модуль) с применением программы-дизассемблера. Нарисовать высокоуровневый алгоритм работы программы. Определить (идентифицировать) алгоритм.

Программу для анализа взять в соответствии с вариантом.

Задача 9.

На основе анализа работы участка кода дизассемблированной программы и идентификации алгоритма его работы, найти и исправить ошибку в машинном коде (исполняемом модуле) без его повторной перекомпиляции.

Программу для анализа и исправления взять в соответствии с вариантом.

Контрольные вопросы

1. Основные принципы анализа компьютерных программ.
2. Основные принципы оптимизации компьютерных программ.
3. Методики анализа и оптимизации программного обеспечения.
4. Правовые аспекты анализа и оптимизации работы готовых программ для ЭВМ.

Список литературы, рекомендуемый к использованию по данной теме **Основная литература:**

1. Макгрегор Дж, Сайкс Д . Тестирование объектно-ориентированного программного обеспечения К: Диасофт, 2012. - 432с.
2. Канер С., Нгуен Енг., Фолк Дж. Тестирование программного обеспечения К: ДияСофт, 2011 - 544с.
3. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Дополнительная литература

1. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб.: Символ-Плюс, 1999. - 304с.
2. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. - 296с.

Методическая литература

1. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Интернет-ресурсы

1. Основы тестирования программного обеспечения
<http://www.intuit.ru/studies/courses/48/48/info>

лабораторная работа №2

Тема: «Современные методы разработки качественного программного обеспечения ЭВМ»

Цель работы: «закрепление знаний по теоретическим основам современных методов разработки качественного программного обеспечения ЭВМ».

Формируемые компетенции (формируется часть компетенций):

УК-2 – «Способность управлять проектом на всех этапах его жизненного цикла».

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Ход работы

1. Орг. момент.
2. Выбор задания в соответствии с вариантом.
3. Изучение теоретической части (литературы) по теме работы.
4. Изучение техники безопасности (правил поведения в компьютерной лаборатории).
5. Выполнение работы.
 - загрузить шаблон программы.
 - внести в шаблон необходимые изменения.
 - отладить полученную компьютерную программу.
6. Оформление результатов работы.
7. Выводы по проделанной работе.
8. Защита работы.

Теоретическая часть

См. список рекомендуемой литературы.

Оборудование и материалы: см. приложение 1.

Техника безопасности: см. приложение 3.

Задания для практической работы

При создании объектно-ориентированных проектов проработать диаграммы классов и объектов моделируемого устройства (процесса) (10-20 классов, 20-100 объектов). Классы и объекты пользовательского интерфейса рекомендуется привести в упрощенном виде (заглушка «устройство ввода/вывода»).

1. Создать ООП-проект, моделирующий электронный калькулятор.
2. Создать ООП-проект, моделирующий электронную записную книжку.
3. Создать ООП-проект, моделирующий часы.
4. Создать ООП-проект, моделирующий автомобиль.
5. Создать ООП-проект, моделирующий самолет.
6. Создать ООП-проект, моделирующий вертолет.

7. Создать ООП-проект, моделирующий корабль.
8. Создать ООП-проект, моделирующий «Умный дом».
9. Создать ООП-проект, моделирующий лифт.
10. Создать ООП-проект, моделирующий телевизор.
11. Создать ООП-проект, моделирующий сотовый телефон.
12. Создать ООП-проект, моделирующий музыкальный центр (техническое устройство).
13. Создать ООП-проект, моделирующий стиральную машину с электронным управлением.
14. Создать ООП-проект, моделирующий микроволновую печь.
15. Создать ООП-проект, моделирующий кухонный комбайн.
16. Создать ООП-проект, моделирующий посудомоечную машину.
17. Создать ООП-проект, моделирующий кухонную электрическую плиту с электронным управлением.
18. Создать ООП-проект, моделирующий токарный станок с электронным управлением.
19. Создать ООП-проект, моделирующий сверлильный станок с электронным управлением.
20. Создать ООП-проект, моделирующий фрезерный станок с электронным управлением.
21. Создать ООП-проект, моделирующий простую наземную экосистему.
22. Создать ООП-проект, моделирующий простую водную экосистему.
23. Создать ООП-проект, моделирующий упрощенное поведение кошки (животное).
24. Создать ООП-проект, моделирующий упрощенное поведение собаки (животное).
25. Создать ООП-проект, моделирующий упрощенное поведение хомяка (животное).
26. Создать ООП-проект, моделирующий упрощенное поведение акулы (животное).
27. Создать ООП-проект, моделирующий упрощенное поведение вороны (животное).
28. Создать ООП-проект, моделирующий развитие куста сирени (растение).
29. Создать ООП-проект, моделирующий процесс горение костра (физический процесс).
30. Создать ООП-проект, моделирующий процессы, происходящие внутри облака (природное явление).
31. Создать ООП-проект, моделирующий грозу (природное явление).
32. Создать ООП-проект, моделирующий копировальный аппарат.
33. Создать ООП-проект, моделирующий ЭВМ фон-Неймановской архитектуры.
34. Создать ООП-проект, моделирующий одноленточную машину Тьюринга.
35. Создать ООП-проект, моделирующий нейрокомпьютер.

36. Создать ООП-проект, моделирующий локальную вычислительную сеть.
37. Создать ООП-проект, моделирующий навигационный спутник.
38. Создать ООП-проект, моделирующий ядерный реактор.
39. Создать ООП-проект, моделирующий продуктовый магазин.
40. Создать ООП-проект, моделирующий процесс работы школы.

Контрольные вопросы

1. Современные методы разработки качественного программного обеспечения ЭВМ.
2. Модульная технология разработки программного обеспечения ЭВМ.
3. Объектно-ориентированная технология разработки программного обеспечения ЭВМ.

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Макгрегор Дж, Сайкс Д . Тестирование объектно-ориентированного программного обеспечения К: Диасофт, 2012. - 432с.
2. Канер С., Нгуен Енг., Фолк Дж. Тестирование программного обеспечения К: ДиаСофт, 2011 - 544с.
3. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Дополнительная литература

1. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб.: Символ-Плюс, 1999. - 304с.
2. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. - 296с.

Методическая литература

1. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Интернет-ресурсы

1. Основы тестирования программного обеспечения
<http://www.intuit.ru/studies/courses/48/48/info>

лабораторная работа №3

Тема: «Способы повышения производительности базовых алгоритмов».

Цель работы: «закрепление знаний по теоретическим основам повышения производительности базовых алгоритмов».

Формируемые компетенции (формируется часть компетенций):

УК-2 – «Способность управлять проектом на всех этапах его жизненного цикла».

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Ход работы

1. Орг. момент.
2. Выбор задания в соответствии с вариантом.
3. Изучение теоретической части (литературы) по теме работы.
4. Изучение техники безопасности (правил поведения в компьютерной лаборатории).
5. Выполнение работы.
 - загрузить шаблон программы.
 - внести в шаблон необходимые изменения.
 - отладить полученную компьютерную программу.
6. Оформление результатов работы.
7. Выводы по проделанной работе.
8. Защита работы.

Теоретическая часть

См. список рекомендуемой литературы.

Оборудование и материалы: см. приложение 1.

Техника безопасности: см. приложение 3.

Задания для практической работы

Составить алгоритмы решения задач в соответствии с выбранным вариантом. Повысить быстродействие критических по времени или памяти блоков.

1. В связном графе из p вершин, заданном матрицей смежности, найти цикл, проходящий через все вершины графа ровно по 1 разу.
2. Известно, что натуральное число A равно произведению двух простых чисел. Найти эти простые числа.
3. Найти цикл минимальной длины, проходящий через все вершины полносвязного графа ровно по 1 разу (граф задан матрицей расстояний между вершинами).
4. Найти натуральный корень сравнения $ax \equiv b \pmod{c}$, где a, b, c — натуральные числа.
5. Заданы натуральные числа a, b, c . Найти натуральные x и y такие, что $ax + by = c$.

6. В связном графе из p вершин, заданном матрицей смежности, найти цепь, проходящую через все вершины графа ровно по 1 разу.
7. Задача о рюкзаке. Дано натуральное число V (емкость рюкзака) и конечное множество предметов, каждый из которых характеризуется двумя натуральными числами: весом (не превышает V) и стоимостью. Найти набор предметов, суммарный вес которых не превосходит V , а суммарная стоимость — максимальная.
8. Граф задан матрицей смежности. Можно ли раскрасить его вершины в три цвета так, чтобы концы каждого ребра были покрашены в разные цвета?
9. Для n точек на плоскости, заданных своими координатами (x, y) найти сеть кратчайшей суммарной длины, соединяющую между собой все точки.
10. Заполнить квадратную таблицу $N \times N$ числами от 1 до N так, чтобы в каждой строке и в каждом столбце таблицы любое число встречалось ровно 1 раз.
11. Вывести на экран все последовательности длины k из n элементов (чисел от 1 до n)
12. Вывести на экран все последовательности длины k , содержащие m элементов из n доступных (чисел от 1 до n).
13. Перечислить (вывести координаты) N ферзей на «шахматном» поле размером $N \times N$, расположенных так, чтобы никакие два ферзя не располагались на одной вертикальной, горизонтальной или диагональной линиях поля.
14. Известно, что натуральное число A равно сумме трех простых чисел. Найти эти простые числа.
15. Дано n чисел $a_1, a_2, \dots, a_n$. Расставить между ними арифметические знаки «+» и «-» так, чтобы в результате расчета полученного выражения получилось заранее заданное число W .

Контрольные вопросы

1. Способы повышения читабельности алгоритмов и исходного кода программ для ЭВМ.
2. Построение профилей (временного и по используемой памяти) работы алгоритма.
3. Методы анализа времени исполнения алгоритма и его подпрограмм.
4. Методы анализа объемов требуемой алгоритму и его подпрограммам памяти.
5. Общие способы оптимизации времени исполнения смешанного алгоритмов.
6. Общие способы оптимизации объема требуемой алгоритму памяти.
7. Оптимизация базовой алгоритмической структуры «следование».
8. Оптимизация базовой алгоритмической структуры «ветвление».
9. Оптимизация базовой алгоритмической структуры «цикл».
10. Низкоуровневое (машинно-ориентированное) программирование и оптимизация программ для ЭВМ.

Список литературы, рекомендуемый к использованию по данной теме
Основная литература:

1. Макгрегор Дж, Сайкс Д . Тестирование объектно-ориентированного программного обеспечения К: Диасофт, 2012. - 432с.
2. Канер С., Нгуен Енг., Фолк Дж. Тестирование программного обеспечения К: ДиаСофт, 2011 - 544с.
3. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Дополнительная литература

1. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб.: Символ-Плюс, 1999. - 304с.
2. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. - 296с.

Методическая литература

1. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Интернет-ресурсы

1. Основы тестирования программного обеспечения
<http://www.intuit.ru/studies/courses/48/48/info>

Литература

Основная литература:

1. Макгрегор Дж, Сайкс Д . Тестирование объектно-ориентированного программного обеспечения К: Диасофт, 2012. - 432с.
2. Канер С., Нгуен Енг., Фолк Дж. Тестирование программного обеспечения К: ДиаСофт, 2011 - 544с.
3. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Дополнительная литература

1. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб.: Символ-Плюс, 1999. - 304с.
2. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. - 296с.

Методическая литература

1. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Интернет-ресурсы

1. Основы тестирования программного обеспечения
<http://www.intuit.ru/studies/courses/48/48/info>

**Оборудование, материалы и программное обеспечение, используемые
для организации и проведения лабораторных работ**

Оборудование:

Компьютерный класс (лаборатория, аудитория) с числом исправных ЭВМ по числу студентов группы плюс рабочее место преподавателя и производительностью, достаточной для освоения дисциплины.

Программное обеспечение:

1. Интегрированная среда разработки программ Lazarus (версия не ниже 1.4)
2. Операционная система: Windows (версия не менее XP), Linux (Mint не ниже 17.2)

Содержимое отчета по практической работе

лабораторная работа считается выполненной студентом и засчитывается преподавателем (с зачислением баллов пропорционально выполненным пунктам отчета, т. е. необязательно, чтобы отчет содержал все пункты) при предоставлении в электронном виде к защите (бумажная распечатка не требуется!!!) аккуратно оформленного файла-отчета в формате .odt (OpenOffice, LibreOffice) или .doc (Microsoft Office), именованный по шаблону (при несоответствии работа не допускается к защите):

ТРО_MAG_PR(номер работы)_V(номер варианта)_(фамилия).odt

Файл-отчет должен иметь следующие элементы в указанном порядке (с сохранением их номера в случае пропуска некоторых пунктов):

0. Отметка об ознакомлении с техникой безопасности (правилами поведения в компьютерном классе (аудитории, лаборатории)).

В данном пункте необходимо наличие записи «С правилами техники безопасности при выполнении лабораторной работы и поведения в компьютерном классе (аудитории, лаборатории) ознакомлен, обязуюсь выполнять _____ (ФИО), _____ дата».

1. Постановка задачи (полное условие в соответствии с выбранным вариантом).

2. Подробная математическая (совокупность формул и поясняющего формулы текста) или информационной (словесное описание алгоритма) модели решения задачи. В данном пункте допустимо наличие экранных снимков расчетов в системах компьютерной алгебры (SciLab, Octave, Maxima и т.п.) или профессиональных графических системах. Использование проприетарных (коммерческих) систем (MathCad, MatLab и т.п.) не рекомендуется.

3. Тестовые наборы исходных данных и соответствующих им правильных результатов для проверки работоспособности программы (в данном пункте **должно быть ручное решение задачи с промежуточными выкладками**). В данном пункте допустимо наличие экранных снимков расчетов в системах компьютерной алгебры (SciLab, Octave, Maxima и т.п.) или профессиональных графических системах. Использование проприетарных (коммерческих) систем (MathCad, MatLab и т.п.) не рекомендуется.

4. Указание имен, типов и назначения всех переменных и сигналов, входящих в математическую или информационную модель.

5. Основной и вспомогательные (если есть) алгоритмы решения задачи (допустимо описание алгоритма на алгоритмическом языке, например, Pascal).

6. Запись полных имен файлов, образующих проект с указанием назначения каждого файла (минимум указать имена и содержимое файлов).

7. Полные исходные тексты компьютерных программ (в каждом файле обязательно наличие информации о разработчике — ФИО, курс, группа, специальность, университет).

8. Экранный снимок или распечатка результата работы компьютерной программы при введенных тестовых данных из пункта 3 отчета. Конкретно в данном пункте необходимо сравнить полученные результаты с эталонным тестовым решением и написать «результат совпадает с эталонным решением».

9. Исследование проекта.

В данном пункте требуется записать несколько наборов входных данных (определить самостоятельно) и соответствующих им результатов. Оценить примерное число шагов и время работы алгоритма решения задачи в общем случае.

10. Выводы о проделанной работе.

Указать возможности, достоинства и недостатки, привести примеры использования.

Правила поведения в компьютерном классе (аудитории, лаборатории)

1. К работе в компьютерном классе допускаются лица, ознакомленные с инструкцией по технике безопасности и охране труда, с правилами поведения и размещения информационных ресурсов.
2. Работа студентов в компьютерном классе разрешается только в присутствии преподавателя (инженера, лаборанта).
3. Во время групповых занятий посторонние лица могут находиться в классе только с разрешения преподавателя.
4. **Перед началом работы необходимо:**
 - убедиться в отсутствии видимых повреждений на рабочем месте;
 - разместить на столе тетради, учебные пособия так, чтобы они не мешали работе на компьютере;
 - принять правильную рабочую позу;
 - если сеанс работы предыдущего пользователя не был завершен, завершить его;
 - ввести регистрационную информацию (при необходимости).
5. **При работе в компьютерном классе категорически запрещается:**
 - находиться в классе в верхней одежде;
 - размещать одежду и сумки на рабочих местах;
 - находиться в классе с едой и напитками;
 - класть книги, тетради и т.п. на клавиатуру;
 - располагаться сбоку или сзади от включенного монитора;
 - присоединять или отсоединять кабели, трогать разъемы, провода и розетки;
 - передвигать компьютеры;
 - открывать системный блок;
 - пытаться самостоятельно устранять неисправности в работе аппаратуры;
 - перекрывать вентиляционные отверстия на системном блоке и мониторе;
 - ударять по клавиатуре, нажимать бесцельно на клавиши;
 - удалять или перемещать чужие файлы;
 - устанавливать и запускать компьютерные игры;
 - использовать Интернет-ресурсы неучебного назначения.
6. **Находясь в компьютерном классе, необходимо:**
 - соблюдать тишину и порядок, выключать мобильные телефоны от громкой связи;
 - выполнять все требования преподавателя, инженера и лаборанта;
 - работать только под своим именем и паролем;
 - соблюдать режим работы (продолжительность непрерывной работы за компьютером не более двух часов с обязательным 10-минутным перерывом и

гимнастикой для глаз; продолжительность интенсивной работы с клавиатурой не более 30 минут с последующей гимнастикой для рук; общая продолжительность работы не более 4 часов в день);

– при появлении рези в глазах, резком ухудшении видимости, невозможности сфокусировать взгляд или навести его на резкость, появлении боли в пальцах и кистях рук, усилении сердцебиения немедленно покинуть рабочее место, сообщить о происшедшем преподавателю и обратиться к врачу;

– после окончания работы завершить все активные программы и корректно завершить сеанс;

– оставить рабочее место чистым.

7. Работая за компьютером, необходимо соблюдать правильную позу:

– расстояние от экрана до глаз 70-80 см (расстояние вытянутой руки);

– вертикально прямая спина;

– плечи опущены и расслаблены;

– ноги на полу и не скрещены;

– локти, запястья и кисти рук на одном уровне;

– локтевые, тазобедренные, коленные, голеностопные суставы под прямым углом.

8. При появлении программных ошибок или сбоях оборудования студент обязан немедленно обратиться к преподавателю (инженеру, лаборанту).

9. В случае порчи или выхода из строя оборудования компьютерного класса по вине пользователя ремонт или замена оборудования производится за счет пользователя.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



Методические указания
по выполнению практических работ по дисциплине
«Тестирование программного обеспечения»
для студентов направления подготовки
01.04.02 «Прикладная математика и информатика»
Направленность (профиль): «Математическое моделирование»

Ставрополь
2019

Содержание

		Стр.
1.	Введение	
2.	Практическая работа №1 «Методика анализа и оптимизации программного обеспечения»	
3.	Практическая работа №2 «Современные методы разработки качественного программного обеспечения ЭВМ»	
4.	Практическая работа №3 «Способы повышения производительности базовых алгоритмов»	
5.	Литература	
6.	Приложение 1. Оборудование, материалы и программное обеспечение, используемые для организации и проведения практических работ	
7.	Приложение 2. Содержимое отчета по практической работе ...	
8.	Приложение 3. Правила поведения в компьютерном классе (аудитории, лаборатории)	

Введение

Дисциплина «Тестирование программного обеспечения» относится к вариативной части Б1.В.ДВ.03.01 учебного плана магистерской программы учебного плана 2019 года и изучается студентами магистратуры специальности 01.04.02 направления «Прикладная математика и информатика» в 4 семестре.

Целью освоения дисциплины является формирование представлений и фундаментальных знаний о современных методах тестирования программного обеспечения.

Основной задачей изучения дисциплины является изучение методов тестирования программного обеспечения.

Формы отчетности: экзамен - 4 семестр.

Содержание дисциплины: основные принципы тестирования компьютерных программ, методику анализа и оптимизации программного обеспечения, современные методы разработки качественного программного обеспечения ЭВМ, способы повышения производительности базовых алгоритмов.

Реализуемые компетенции:

УК-2 – «Способность управлять проектом на всех этапах его жизненного цикла».

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Результаты освоения дисциплины.

Знать: основные принципы тестирования компьютерных программ, методику анализа и оптимизации программного обеспечения, современные методы разработки качественного программного обеспечения ЭВМ, способы повышения производительности базовых алгоритмов.

Уметь: применять на практике методы тестирования программного обеспечения ЭВМ

Владеть: навыками использования специализированного программного обеспечения, предназначенного тестирования программ для ЭВМ

Рекомендуемое программное обеспечение:

1. Интегрированная среда разработки программ Lazarus (не ниже 1.4)
2. Операционная система: Windows (не менее XP), Linux Mint не ниже 17.2

Рекомендуемое материально-техническое обеспечение: компьютерный класс (лаборатория, аудитория) с числом исправных ЭВМ по числу студентов группы плюс рабочее место преподавателя и производительностью, достаточной для освоения дисциплины. На всех ЭВМ класса (лаборатории, аудитории) программное обеспечение, выход в интернет для изучения интернет-источников дисциплины (со скоростью доступа, достаточной для комфортной работы).

Практическая работа №1

Тема: «Методика анализа и оптимизации программного обеспечения»

Цель работы: «закрепление знаний по теоретическим основам анализа и оптимизации программного обеспечения».

Формируемые компетенции (формируется часть компетенций):

УК-2 – «Способность управлять проектом на всех этапах его жизненного цикла».

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Ход работы

1. Орг. момент.
2. Выбор задания в соответствии с вариантом.
3. Изучение теоретической части (литературы) по теме работы.
4. Изучение техники безопасности (правил поведения в компьютерной лаборатории).
5. Выполнение работы.
 - загрузить шаблон программы.
 - внести в шаблон необходимые изменения.
 - отладить полученную компьютерную программу.
6. Оформление результатов работы.
7. Выводы по проделанной работе.
8. Защита работы.

Теоретическая часть

См. список рекомендуемой литературы.

Оборудование и материалы: см. приложение 1.

Техника безопасности: см. приложение 3.

Задания для практической работы

Задача 1.

Рассчитать число элементарных шагов работы алгоритма.

Алгоритм для анализа взять в соответствии с вариантом.

Задача 2.

Рассчитать объем памяти, требуемый для работы алгоритма.

Алгоритм для анализа взять в соответствии с вариантом.

Задача 3.

Оптимизировать временную сложность алгоритма (уменьшить число шагов путем оптимизации вызовов подпрограмм, циклов и ветвлений, расчетных формул).

Алгоритм для оптимизации взять в соответствии с вариантом.

Задача 4.

Оптимизировать объемы запрашиваемой алгоритмом памяти путем выявления и отказа от неэффективно используемых массивов, оптимизации вызовов подпрограмм, циклов и ветвлений, расчетных формул).

Алгоритм для оптимизации взять в соответствии с вариантом.

Задача 5.

Оптимизировать временную сложность алгоритма путем поиска и переписывания участков кода, пригодных для распараллеливания.

Подпрограмму для анализа и оптимизации взять в соответствии с вариантом.

Задача 6.

Перевести участок кода программы с высокоуровневого языка программирования на язык ассемблера.

Подпрограмму для оптимизации взять в соответствии с вариантом.

Задача 7.

Построить подробный алгоритм работы участка кода программы, записанного на языке ассемблера. Выявить в нем более высокоуровневые алгоритмические структуры. Переписать подпрограмму на высокоуровневом языке.

Подпрограмму для анализа и реконструирования взять в соответствии с вариантом.

Задача 8.

Перевести на язык ассемблера готовую программу (исполняемый модуль) с применением программы-дизассемблера. Нарисовать высокоуровневый алгоритм работы программы. Определить (идентифицировать) алгоритм.

Программу для анализа взять в соответствии с вариантом.

Задача 9.

На основе анализа работы участка кода дизассемблированной программы и идентификации алгоритма его работы, найти и исправить ошибку в машинном коде (исполняемом модуле) без его повторной перекомпиляции.

Программу для анализа и исправления взять в соответствии с вариантом.

Контрольные вопросы

1. Основные принципы анализа компьютерных программ.
2. Основные принципы оптимизации компьютерных программ.
3. Методики анализа и оптимизации программного обеспечения.
4. Правовые аспекты анализа и оптимизации работы готовых программ для ЭВМ.

Список литературы, рекомендуемый к использованию по данной теме **Основная литература:**

1. Макгрегор Дж, Сайкс Д . Тестирование объектно-ориентированного программного обеспечения К: Диасофт, 2012. - 432с.
2. Канер С., Нгуен Енг., Фолк Дж. Тестирование программного обеспечения К: ДияСофт, 2011 - 544с.
3. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Дополнительная литература

1. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб.: Символ-Плюс, 1999. - 304с.
2. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. - 296с.

Методическая литература

1. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Интернет-ресурсы

1. Основы тестирования программного обеспечения
<http://www.intuit.ru/studies/courses/48/48/info>

Практическая работа №2

Тема: «Современные методы разработки качественного программного обеспечения ЭВМ»

Цель работы: «закрепление знаний по теоретическим основам современных методов разработки качественного программного обеспечения ЭВМ».

Формируемые компетенции (формируется часть компетенций):

УК-2 – «Способность управлять проектом на всех этапах его жизненного цикла».

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Ход работы

1. Орг. момент.
2. Выбор задания в соответствии с вариантом.
3. Изучение теоретической части (литературы) по теме работы.
4. Изучение техники безопасности (правил поведения в компьютерной лаборатории).
5. Выполнение работы.
 - загрузить шаблон программы.
 - внести в шаблон необходимые изменения.
 - отладить полученную компьютерную программу.
6. Оформление результатов работы.
7. Выводы по проделанной работе.
8. Защита работы.

Теоретическая часть

См. список рекомендуемой литературы.

Оборудование и материалы: см. приложение 1.

Техника безопасности: см. приложение 3.

Задания для практической работы

При создании объектно-ориентированных проектов проработать диаграммы классов и объектов моделируемого устройства (процесса) (10-20 классов, 20-100 объектов). Классы и объекты пользовательского интерфейса рекомендуется привести в упрощенном виде (заглушка «устройство ввода/вывода»).

1. Создать ООП-проект, моделирующий электронный калькулятор.
2. Создать ООП-проект, моделирующий электронную записную книжку.
3. Создать ООП-проект, моделирующий часы.
4. Создать ООП-проект, моделирующий автомобиль.
5. Создать ООП-проект, моделирующий самолет.
6. Создать ООП-проект, моделирующий вертолет.

7. Создать ООП-проект, моделирующий корабль.
8. Создать ООП-проект, моделирующий «Умный дом».
9. Создать ООП-проект, моделирующий лифт.
10. Создать ООП-проект, моделирующий телевизор.
11. Создать ООП-проект, моделирующий сотовый телефон.
12. Создать ООП-проект, моделирующий музыкальный центр (техническое устройство).
13. Создать ООП-проект, моделирующий стиральную машину с электронным управлением.
14. Создать ООП-проект, моделирующий микроволновую печь.
15. Создать ООП-проект, моделирующий кухонный комбайн.
16. Создать ООП-проект, моделирующий посудомоечную машину.
17. Создать ООП-проект, моделирующий кухонную электрическую плиту с электронным управлением.
18. Создать ООП-проект, моделирующий токарный станок с электронным управлением.
19. Создать ООП-проект, моделирующий сверлильный станок с электронным управлением.
20. Создать ООП-проект, моделирующий фрезерный станок с электронным управлением.
21. Создать ООП-проект, моделирующий простую наземную экосистему.
22. Создать ООП-проект, моделирующий простую водную экосистему.
23. Создать ООП-проект, моделирующий упрощенное поведение кошки (животное).
24. Создать ООП-проект, моделирующий упрощенное поведение собаки (животное).
25. Создать ООП-проект, моделирующий упрощенное поведение хомяка (животное).
26. Создать ООП-проект, моделирующий упрощенное поведение акулы (животное).
27. Создать ООП-проект, моделирующий упрощенное поведение вороны (животное).
28. Создать ООП-проект, моделирующий развитие куста сирени (растение).
29. Создать ООП-проект, моделирующий процесс горение костра (физический процесс).
30. Создать ООП-проект, моделирующий процессы, происходящие внутри облака (природное явление).
31. Создать ООП-проект, моделирующий грозу (природное явление).
32. Создать ООП-проект, моделирующий копировальный аппарат.
33. Создать ООП-проект, моделирующий ЭВМ фон-Неймановской архитектуры.
34. Создать ООП-проект, моделирующий одноленточную машину Тьюринга.
35. Создать ООП-проект, моделирующий нейрокомпьютер.

36. Создать ООП-проект, моделирующий локальную вычислительную сеть.
37. Создать ООП-проект, моделирующий навигационный спутник.
38. Создать ООП-проект, моделирующий ядерный реактор.
39. Создать ООП-проект, моделирующий продуктовый магазин.
40. Создать ООП-проект, моделирующий процесс работы школы.

Контрольные вопросы

1. Современные методы разработки качественного программного обеспечения ЭВМ.
2. Модульная технология разработки программного обеспечения ЭВМ.
3. Объектно-ориентированная технология разработки программного обеспечения ЭВМ.

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Макгрегор Дж, Сайкс Д . Тестирование объектно-ориентированного программного обеспечения К: Диасофт, 2012. - 432с.
2. Канер С., Нгуен Енг., Фолк Дж. Тестирование программного обеспечения К: ДиаСофт, 2011 - 544с.
3. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Дополнительная литература

1. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб.: Символ-Плюс, 1999. - 304с.
2. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. - 296с.

Методическая литература

1. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Интернет-ресурсы

1. Основы тестирования программного обеспечения
<http://www.intuit.ru/studies/courses/48/48/info>

Практическая работа №3

Тема: «Способы повышения производительности базовых алгоритмов».

Цель работы: «закрепление знаний по теоретическим основам повышения производительности базовых алгоритмов».

Формируемые компетенции (формируется часть компетенций):

УК-2 – «Способность управлять проектом на всех этапах его жизненного цикла».

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Ход работы

1. Орг. момент.
2. Выбор задания в соответствии с вариантом.
3. Изучение теоретической части (литературы) по теме работы.
4. Изучение техники безопасности (правил поведения в компьютерной лаборатории).
5. Выполнение работы.
 - загрузить шаблон программы.
 - внести в шаблон необходимые изменения.
 - отладить полученную компьютерную программу.
6. Оформление результатов работы.
7. Выводы по проделанной работе.
8. Защита работы.

Теоретическая часть

См. список рекомендуемой литературы.

Оборудование и материалы: см. приложение 1.

Техника безопасности: см. приложение 3.

Задания для практической работы

Составить алгоритмы решения задач в соответствии с выбранным вариантом. Повысить быстродействие критических по времени или памяти блоков.

1. В связном графе из r вершин, заданном матрицей смежности, найти цикл, проходящий через все вершины графа ровно по 1 разу.
2. Известно, что натуральное число A равно произведению двух простых чисел. Найти эти простые числа.
3. Найти цикл минимальной длины, проходящий через все вершины полносвязного графа ровно по 1 разу (граф задан матрицей расстояний между вершинами).
4. Найти натуральный корень сравнения $ax \equiv b \pmod{c}$, где a, b, c — натуральные числа.
5. Заданы натуральные числа a, b, c . Найти натуральные x и y такие, что $ax + by = c$.

6. В связном графе из p вершин, заданном матрицей смежности, найти цепь, проходящую через все вершины графа ровно по 1 разу.
7. Задача о рюкзаке. Дано натуральное число V (емкость рюкзака) и конечное множество предметов, каждый из которых характеризуется двумя натуральными числами: весом (не превышает V) и стоимостью. Найти набор предметов, суммарный вес которых не превосходит V , а суммарная стоимость — максимальная.
8. Граф задан матрицей смежности. Можно ли раскрасить его вершины в три цвета так, чтобы концы каждого ребра были покрашены в разные цвета?
9. Для n точек на плоскости, заданных своими координатами (x, y) найти сеть кратчайшей суммарной длины, соединяющую между собой все точки.
10. Заполнить квадратную таблицу $N \times N$ числами от 1 до N так, чтобы в каждой строке и в каждом столбце таблицы любое число встречалось ровно 1 раз.
11. Вывести на экран все последовательности длины k из n элементов (чисел от 1 до n)
12. Вывести на экран все последовательности длины k , содержащие m элементов из n доступных (чисел от 1 до n).
13. Перечислить (вывести координаты) N ферзей на «шахматном» поле размером $N \times N$, расположенных так, чтобы никакие два ферзя не располагались на одной вертикальной, горизонтальной или диагональной линиях поля.
14. Известно, что натуральное число A равно сумме трех простых чисел. Найти эти простые числа.
15. Дано n чисел $a_1, a_2, \dots, a_n$. Расставить между ними арифметические знаки «+» и «-» так, чтобы в результате расчета полученного выражения получилось заранее заданное число W .

Контрольные вопросы

1. Способы повышения читабельности алгоритмов и исходного кода программ для ЭВМ.
2. Построение профилей (временного и по используемой памяти) работы алгоритма.
3. Методы анализа времени исполнения алгоритма и его подпрограмм.
4. Методы анализа объемов требуемой алгоритму и его подпрограммам памяти.
5. Общие способы оптимизации времени исполнения смешанного алгоритмов.
6. Общие способы оптимизации объема требуемой алгоритму памяти.
7. Оптимизация базовой алгоритмической структуры «следование».
8. Оптимизация базовой алгоритмической структуры «ветвление».
9. Оптимизация базовой алгоритмической структуры «цикл».
10. Низкоуровневое (машинно-ориентированное) программирование и оптимизация программ для ЭВМ.

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Макгрегор Дж, Сайкс Д . Тестирование объектно-ориентированного программного обеспечения К: Диасофт, 2012. - 432с.
2. Канер С., Нгуен Енг., Фолк Дж. Тестирование программного обеспечения К: ДиаСофт, 2011 - 544с.
3. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Дополнительная литература

1. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб.: Символ-Плюс, 1999. - 304с.
2. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. - 296с.

Методическая литература

1. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Интернет-ресурсы

1. Основы тестирования программного обеспечения
<http://www.intuit.ru/studies/courses/48/48/info>

Литература

Основная литература:

1. Макгрегор Дж, Сайкс Д . Тестирование объектно-ориентированного программного обеспечения К: Диасофт, 2012. - 432с.
2. Канер С., Нгуен Енг., Фолк Дж. Тестирование программного обеспечения К: ДиаСофт, 2011 - 544с.
3. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Дополнительная литература

1. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб.: Символ-Плюс, 1999. - 304с.
2. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. - 296с.

Методическая литература

1. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Интернет-ресурсы

1. Основы тестирования программного обеспечения
<http://www.intuit.ru/studies/courses/48/48/info>

**Оборудование, материалы и программное обеспечение, используемые
для организации и проведения практических работ**

Оборудование:

Компьютерный класс (лаборатория, аудитория) с числом исправных ЭВМ по числу студентов группы плюс рабочее место преподавателя и производительностью, достаточной для освоения дисциплины.

Программное обеспечение:

1. Интегрированная среда разработки программ Lazarus (версия не ниже 1.4)
2. Операционная система: Windows (версия не менее XP), Linux (Mint не ниже 17.2)

Содержимое отчета по практической работе

Практическая работа считается выполненной студентом и засчитывается преподавателем (с зачислением баллов пропорционально выполненным пунктам отчета, т. е. необязательно, чтобы отчет содержал все пункты) при предоставлении в электронном виде к защите (бумажная распечатка не требуется!!!) аккуратно оформленного файла-отчета в формате .odt (OpenOffice, LibreOffice) или .doc (Microsoft Office), именованный по шаблону (при несоответствии работа не допускается к защите):

ТРО_MAG_PR(номер работы)_V(номер варианта)_(фамилия).odt

Файл-отчет должен иметь следующие элементы в указанном порядке (с сохранением их номера в случае пропуска некоторых пунктов):

0. Отметка об ознакомлении с техникой безопасности (правилами поведения в компьютерном классе (аудитории, лаборатории)).

В данном пункте необходимо наличие записи «С правилами техники безопасности при выполнении лабораторной работы и поведения в компьютерном классе (аудитории, лаборатории) ознакомлен, обязуюсь выполнять _____ (ФИО), _____ дата».

1. Постановка задачи (полное условие в соответствии с выбранным вариантом).

2. Подробная математическая (совокупность формул и поясняющего формулы текста) или информационной (словесное описание алгоритма) модели решения задачи. В данном пункте допустимо наличие экранных снимков расчетов в системах компьютерной алгебры (SciLab, Octave, Maxima и т.п.) или профессиональных графических системах. Использование проприетарных (коммерческих) систем (MathCad, MatLab и т.п.) не рекомендуется.

3. Тестовые наборы исходных данных и соответствующих им правильных результатов для проверки работоспособности программы (в данном пункте **должно быть ручное решение задачи с промежуточными выкладками**). В данном пункте допустимо наличие экранных снимков расчетов в системах компьютерной алгебры (SciLab, Octave, Maxima и т.п.) или профессиональных графических системах. Использование проприетарных (коммерческих) систем (MathCad, MatLab и т.п.) не рекомендуется.

4. Указание имен, типов и назначения всех переменных и сигналов, входящих в математическую или информационную модель.

5. Основной и вспомогательные (если есть) алгоритмы решения задачи (допустимо описание алгоритма на алгоритмическом языке, например, Pascal).

6. Запись полных имен файлов, образующих проект с указанием назначения каждого файла (минимум указать имена и содержимое файлов).

7. Полные исходные тексты компьютерных программ (в каждом файле обязательно наличие информации о разработчике — ФИО, курс, группа, специальность, университет).

8. Экранный снимок или распечатка результата работы компьютерной программы при введенных тестовых данных из пункта 3 отчета. Конкретно в данном пункте необходимо сравнить полученные результаты с эталонным тестовым решением и написать «результат совпадает с эталонным решением».

9. Исследование проекта.

В данном пункте требуется записать несколько наборов входных данных (определить самостоятельно) и соответствующих им результатов. Оценить примерное число шагов и время работы алгоритма решения задачи в общем случае.

10. Выводы о проделанной работе.

Указать возможности, достоинства и недостатки, привести примеры использования.

Правила поведения в компьютерном классе (аудитории, лаборатории)

1. К работе в компьютерном классе допускаются лица, ознакомленные с инструкцией по технике безопасности и охране труда, с правилами поведения и размещения информационных ресурсов.
2. Работа студентов в компьютерном классе разрешается только в присутствии преподавателя (инженера, лаборанта).
3. Во время групповых занятий посторонние лица могут находиться в классе только с разрешения преподавателя.
4. **Перед началом работы необходимо:**
 - убедиться в отсутствии видимых повреждений на рабочем месте;
 - разместить на столе тетради, учебные пособия так, чтобы они не мешали работе на компьютере;
 - принять правильную рабочую позу;
 - если сеанс работы предыдущего пользователя не был завершен, завершить его;
 - ввести регистрационную информацию (при необходимости).
5. **При работе в компьютерном классе категорически запрещается:**
 - находиться в классе в верхней одежде;
 - размещать одежду и сумки на рабочих местах;
 - находиться в классе с едой и напитками;
 - класть книги, тетради и т.п. на клавиатуру;
 - располагаться сбоку или сзади от включенного монитора;
 - присоединять или отсоединять кабели, трогать разъемы, провода и розетки;
 - передвигать компьютеры;
 - открывать системный блок;
 - пытаться самостоятельно устранять неисправности в работе аппаратуры;
 - перекрывать вентиляционные отверстия на системном блоке и мониторе;
 - ударять по клавиатуре, нажимать бесцельно на клавиши;
 - удалять или перемещать чужие файлы;
 - устанавливать и запускать компьютерные игры;
 - использовать Интернет-ресурсы неучебного назначения.
6. **Находясь в компьютерном классе, необходимо:**
 - соблюдать тишину и порядок, выключать мобильные телефоны от громкой связи;
 - выполнять все требования преподавателя, инженера и лаборанта;
 - работать только под своим именем и паролем;
 - соблюдать режим работы (продолжительность непрерывной работы за компьютером не более двух часов с обязательным 10-минутным перерывом и

гимнастикой для глаз; продолжительность интенсивной работы с клавиатурой не более 30 минут с последующей гимнастикой для рук; общая продолжительность работы не более 4 часов в день);

- при появлении рези в глазах, резком ухудшении видимости, невозможности сфокусировать взгляд или навести его на резкость, появлении боли в пальцах и кистях рук, усилении сердцебиения немедленно покинуть рабочее место, сообщить о происшедшем преподавателю и обратиться к врачу;

- после окончания работы завершить все активные программы и корректно завершить сеанс;

- оставить рабочее место чистым.

7. Работая за компьютером, необходимо соблюдать правильную позу:

- расстояние от экрана до глаз 70-80 см (расстояние вытянутой руки);

- вертикально прямая спина;

- плечи опущены и расслаблены;

- ноги на полу и не скрещены;

- локти, запястья и кисти рук на одном уровне;

- локтевые, тазобедренные, коленные, голеностопные суставы под прямым углом.

8. При появлении программных ошибок или сбоях оборудования студент обязан немедленно обратиться к преподавателю (инженеру, лаборанту).

9. В случае порчи или выхода из строя оборудования компьютерного класса по вине пользователя ремонт или замена оборудования производится за счет пользователя.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



Методические указания
для обучающихся по организации и проведению
самостоятельной работы по дисциплине
«Тестирование программного обеспечения»
для студентов направления подготовки
01.04.02 «Прикладная математика и информатика»
Направленность (профиль):
«Математическое моделирование»

Ставрополь
2019

Содержание

		Стр.
1.	Введение	
2	Общая характеристика самостоятельной работы студента при изучении дисциплины	
3.	План-график выполнения самостоятельной работы	
4.	Методические рекомендации по изучению теоретического материала	
5.	Методические указания (по видам работ, предусмотренных рабочей программой дисциплины)	
6.	Методические указания по подготовке к экзамену	
7.	Список рекомендуемой литературы	
8.	Методические указания по выполнению курсовых работ	

Введение

Дисциплина «Тестирование программного обеспечения» относится к вариативной части Б1.В.ДВ.03.01 учебного плана магистерской программы учебного плана 2019 года и изучается студентами магистратуры специальности 01.04.02 направления «Прикладная математика и информатика» в 4 семестре.

Целью освоения дисциплины является формирование представлений и фундаментальных знаний о современных методах тестирования программного обеспечения.

Основной задачей изучения дисциплины является изучение методов тестирования программного обеспечения.

Формы отчетности: экзамен - 4 семестр.

Содержание дисциплины: основные принципы тестирования компьютерных программ, методику анализа и оптимизации программного обеспечения, современные методы разработки качественного программного обеспечения ЭВМ, способы повышения производительности базовых алгоритмов.

Реализуемые компетенции:

УК-2 – «Способность управлять проектом на всех этапах его жизненного цикла».

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Результаты освоения дисциплины.

Знать: основные принципы тестирования компьютерных программ, методику анализа и оптимизации программного обеспечения, современные методы разработки качественного программного обеспечения ЭВМ, способы повышения производительности базовых алгоритмов.

Уметь: применять на практике методы тестирования программного обеспечения ЭВМ

Владеть: навыками использования специализированного программного обеспечения, предназначенного тестирования программ для ЭВМ

Рекомендуемое программное обеспечение:

1. Интегрированная среда разработки программ Lazarus (не ниже 1.4)
2. Операционная система: Windows (не менее XP), Linux Mint не ниже 17.2

Рекомендуемое материально-техническое обеспечение: компьютерный класс (лаборатория, аудитория) с числом исправных ЭВМ по числу студентов группы плюс рабочее место преподавателя и производительностью, достаточной для освоения дисциплины. На всех ЭВМ класса (лаборатории, аудитории) программное обеспечение, выход в интернет для изучения интернет-источников дисциплины (со скоростью доступа, достаточной для комфортной работы).

Общая характеристика самостоятельной работы студента при изучении дисциплины «Тестирование программного обеспечения»

Практическая работа №1

Тема: «Методика анализа и оптимизации программного обеспечения»

Цель работы: «закрепление знаний по теоретическим основам анализа и оптимизации программного обеспечения».

Задачи: изучение методов тестирования программного обеспечения.

Формируемые компетенции (формируется часть компетенций):

УК-2 – «Способность управлять проектом на всех этапах его жизненного цикла».

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Практическая работа №2

Тема: «Современные методы разработки качественного программного обеспечения ЭВМ»

Цель работы: «закрепление знаний по теоретическим основам современных методов разработки качественного программного обеспечения ЭВМ».

Задачи: изучение методов тестирования программного обеспечения.

Формируемые компетенции (формируется часть компетенций):

УК-2 – «Способность управлять проектом на всех этапах его жизненного цикла».

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Практическая работа №3

Тема: «Способы повышения производительности базовых алгоритмов».

Цель работы: «закрепление знаний по теоретическим основам повышения производительности базовых алгоритмов».

Задачи: изучение методов тестирования программного обеспечения.

Формируемые компетенции (формируется часть компетенций):

УК-2 – «Способность управлять проектом на всех этапах его жизненного цикла».

ПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ПК-3 – «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

План-график выполнения самостоятельной работы

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации.

№ п/п	Вид деятельности студентов	Сроки выполнения	Количество баллов
1	Подготовка к практической работе №1	5	15
3	Подготовка к практической работе №2	11	20
3	Подготовка к практической работе №3	15	20
	Итого за 2 семестр		55
	Итого		55

Оценка 5 (отлично) выставляется студентам, своевременно выполнившим практическую работу, не имеющим пропусков лекционных и практических занятий.

Оценка 4 (хорошо) выставляется студентам, выполнившим практическую работу, возможно с задержкой (до наступления срока защиты следующей практической работы) и имеющим до 30% пропусков лекционных и практических занятий.

Оценка 3 (удовлетворительно) выставляется студентам, выполнившим практическую работу с существенной задержкой и имеющим большое (свыше 30%) количество пропусков лекционных и практических занятий.

Оценка 2 (неудовлетворительно) выставляется студентам, не защитившим практическую работу до зачета, либо вообще потерявшим связь с университетом.

Методические рекомендации по изучению теоретического материала

Практическая работа №1

Тема: «Методика анализа и оптимизации программного обеспечения»

Контрольные вопросы

1. Основных принципы анализа компьютерных программ.
2. Основных принципы оптимизации компьютерных программ.
3. Методики анализа и оптимизации программного обеспечения.
4. Правовые аспекты анализа и оптимизации работы готовых программ для ЭВМ.

Практическая работа №2

Тема: «Современные методы разработки качественного программного обеспечения ЭВМ»

Контрольные вопросы

1. Современные методы разработки качественного программного обеспечения ЭВМ.
2. Модульная технология разработки программного обеспечения ЭВМ.
3. Объектно-ориентированная технология разработки программного обеспечения ЭВМ.

Практическая работа №3

Тема: «Способы повышения производительности базовых алгоритмов».

Контрольные вопросы

1. Способы повышения читабельности алгоритмов и исходного кода программ для ЭВМ.
2. Построение профилей (временного и по используемой памяти) работы алгоритма.
3. Методы анализа времени исполнения алгоритма и его подпрограмм.
4. Методы анализа объемов требуемой алгоритму и его подпрограммам памяти.
5. Общие способы оптимизации времени исполнения смешанного алгоритмов.
6. Общие способы оптимизации объема требуемой алгоритму памяти.
7. Оптимизация базовой алгоритмической структуры «следование».
8. Оптимизация базовой алгоритмической структуры «ветвление».
9. Оптимизация базовой алгоритмической структуры «цикл».
10. Низкоуровневое (машинно-ориентированное) программирование и оптимизация программ для ЭВМ.

Методические указания (по видам работ, предусмотренных рабочей программой дисциплины)

Текущая аттестация студентов проводится преподавателями, ведущими практические работы по дисциплине.

При проверке практической работы, оцениваются правильность ответа, последовательность и рациональность выполнения, точность расчетов.

Максимальное количество баллов студент получает, если оформление отчета соответствует установленным требованиям, а отчет полностью раскрывает суть работы. Основанием для снижения оценки являются:

- незначительные нарушения правил оформления отчета;
- затруднения студентов при ответе на некоторые контрольные вопросы;

Отчет может быть отправлен на доработку в следующих случаях:

- отсутствия каких-либо элементов отчета.
- абсолютное не соответствие требованиям к оформлению.

Практическая работа считается выполненной студентом и засчитывается преподавателем (с зачислением баллов пропорционально выполненным пунктам отчета, т. е. необязательно, чтобы отчет содержал все пункты) при предоставлении в электронном виде к защите (бумажная распечатка не требуется!!!) аккуратно оформленного файла-отчета в формате .odt (OpenOffice, LibreOffice) или .doc (Microsoft Office), именованный по шаблону (при несоответствии работа не допускается к защите):

ТРО_MAG_PR(номер работы)_V(номер варианта)_(фамилия).odt

Файл-отчет должен иметь следующие элементы в указанном порядке (с сохранением их номера в случае пропуска некоторых пунктов):

0. Отметка об ознакомлении с техникой безопасности (правилами поведения в компьютерном классе (аудитории, лаборатории)).

В данном пункте необходимо наличие записи «С правилами техники безопасности при выполнении лабораторной работы и поведения в компьютерном классе (аудитории, лаборатории) ознакомлен, обязуюсь выполнять _____ (ФИО), _____ дата».

1. Постановка задачи (полное условие в соответствии с выбранным вариантом).

2. Подробная математическая (совокупность формул и поясняющего формулы текста) или информационной (словесное описание алгоритма) модели решения задачи. В данном пункте допустимо наличие экранных снимков расчетов в системах компьютерной алгебры (SciLab, Octave, Maxima и т.п.) или профессиональных графических системах. Использование проприетарных (коммерческих) систем (MathCad, MatLab и т.п.) не рекомендуется.

3. Тестовые наборы исходных данных и соответствующих им правильных результатов для проверки работоспособности программы (в данном пункте **должно быть ручное решение задачи с промежуточными выкладками**). В данном пункте допустимо наличие экранных снимков расчетов в системах

компьютерной алгебры (SciLab, Octave, Maxima и т.п.) или профессиональных графических системах. Использование проприетарных (коммерческих) систем (MathCad, MatLab и т.п.) не рекомендуется.

4. Указание имен, типов и назначения всех переменных и сигналов, входящих в математическую или информационную модель.

5. Основной и вспомогательные (если есть) алгоритмы решения задачи (допустимо описание алгоритма на алгоритмическом языке, например, Pascal).

6. Запись полных имен файлов, образующих проект с указанием назначения каждого файла (минимум указать имена и содержимое файлов).

7. Полные исходные тексты компьютерных программ (в каждом файле обязательно наличие информации о разработчике — ФИО, курс, группа, специальность, университет).

8. Экранный снимок или распечатка результата работы компьютерной программы при введенных тестовых данных из пункта 3 отчета. Конкретно в данном пункте необходимо сравнить полученные результаты с эталонным тестовым решением и написать «результат совпадает с эталонным решением».

9. Исследование проекта.

В данном пункте требуется записать несколько наборов входных данных (определить самостоятельно) и соответствующих им результатов. Оценить примерное число шагов и время работы алгоритма решения задачи в общем случае.

10. Выводы о проделанной работе.

Указать возможности, достоинства и недостатки, привести примеры использования.

Методические указания по подготовке к экзамену

Процедура проведения экзамена осуществляется в соответствии с Положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования в СКФУ.

Оценка «отлично» выставляется студенту, если он в течение семестра безукоризненно излагал теоретический материал без наводящих вопросов преподавателя и решал предложенные задачи и разработал алгоритм на языке высокого уровня. Возможны одна - две неточности при освещении второстепенных вопросов или выкладках, которые студент легко исправил после замечания преподавателя

Оценка «хорошо» выставляется студенту, если он в течение семестра показал систематический характер знаний по дисциплине и способность к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности; если допущены один - два недочёта при изложении основного содержания ответа и освещении второстепенных вопросов или выкладках, легко исправленные после замечания преподавателя.

Оценка «удовлетворительно» выставляется студенту, если он в течение семестра допускал погрешности в ответе и при выполнении заданий, недостаточно правильно формулировал основные положения программного материала, нарушал последовательность в его изложении, показал недостаточную сформированность основных умений и навыков при выполнении практических заданий, но обладал необходимыми знаниями для их устранения под руководством преподавателя

Оценка «неудовлетворительно» выставляется студенту, регулярно обнаруживавшему пробелы в знаниях основного учебно-программного материала, допускавшему принципиальные ошибки в выполнении предусмотренных программой практических знаний.

Список рекомендуемой литературы

Основная литература:

1. Макгрегор Дж, Сайкс Д . Тестирование объектно-ориентированного программного обеспечения К: Диасофт, 2012. - 432с.
2. Канер С., Нгуен Енг., Фолк Дж. Тестирование программного обеспечения К: ДиаСофт, 2011 - 544с.
3. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Дополнительная литература

1. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб.: Символ-Плюс, 1999. - 304с.
2. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. - 296с.

Методическая литература

1. Майерс Г. Искусство тестирования программ М.: Финансы и статистика, 2014. -176 с.

Интернет-ресурсы

1. Основы тестирования программного обеспечения
<http://www.intuit.ru/studies/courses/48/48/info>