Министерство образования и науки Российской Федерации

ФГБОУ ВО «Псковский государственный университет»

Колледж ПсковГУ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**по выполнению курсовой работы**

ПМ03 Участие в интеграции программных модулей

МДК03.01 Технология разработки программного обеспечения

для студентов очной / заочной формы обучения

специальности 09.02.03 Программирование в компьютерных системах

Составил: Т.О. Ушарнова,

преподаватель Колледжа ПсковГУ

Псков 2017

Методические указания рассмотрены и рекомендованы к утверждению на заседании цикловой комиссии информационных технологий

протокол № \_\_ от \_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Председатель цикловой комиссии Е.С. Романова

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(ФИО)

(подпись)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Министерство образования и науки РФ

Псковский государственный университет

Колледж ПсковГУ

Утверждаю  
Заместитель директора   
по учебной работе  
\_\_\_\_\_\_\_\_ О.В. Ефимова  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г.

ЗАДАНИЕ

на разработку курсовой работы

*Специальность:* 09.02.03 «Программирование в компьютерных системах »

*Уровень:* Базовый  
*Предмет:* МДК 03.01 Технология разработки программного обеспечения

Тема: ***Разработка программ для решения математических задач с использованием численных методов***

* решение системы уравнений методом Гаусса, Зейделя или итераций;
* вычисление приближённого значения интеграла по формуле прямоугольников, трапеций, Симпсона (парабол), 3/8 или Гаусса;
* решение нелинейного уравнения методом простой итерации, дихотомии(проб), Ньютона (касательных), хорд или комбинированным (хорд и касательных) методом.

*Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Курс \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Группа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Согласовано

протокол заседания цикловой комиссии   
информационных технологий

№ \_\_ от \_\_\_ \_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Председатель цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания " "

Срок окончания комплексного   
курсового проектирования: " "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание выдал (руководитель) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Министерство образования и науки РФ

Псковский государственный университет

Колледж ПсковГУ

Утверждаю  
Заместитель директора   
по учебной работе  
\_\_\_\_\_\_\_\_ О.В. Ефимова  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г.

ЗАДАНИЕ

на разработку курсовой работы

*Специальность:* 09.02.03 «Программирование в компьютерных системах »

*Уровень:* Базовый  
*Предмет:* МДК 03.01 Технология разработки программного обеспечения

Тема: ***Разработка программ для решения задач защиты информации***

* изложение материала по теме «шифрование и дешифрование текста пятью разными методами»
* описание алгоритма программы
* составление блок-схемы, характеризующей работу программы
* составление и испытание программы шифрующей и дешифрующей текст пятью различными методами.

*Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Курс \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Группа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Согласовано

протокол заседания цикловой комиссии   
информационных технологий

№ \_\_ от \_\_\_ \_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Председатель цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания " "

Срок окончания комплексного   
курсового проектирования: " "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание выдал (руководитель) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Методические указания к выполнению курсовой работы студентов по дисциплине «Технология разработки программного обеспечения» предназначены для студентов по специальности 09.02.03 «Программирование в компьютерных системах» (базовой подготовки) в части освоения основного вида профессиональной деятельности (ВПД) «Техник-программист» и соответствующих профессиональных (ПК) и общих компетенций (ОК):

ПК 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

ПК 3.2. Выполнять интеграцию модулей в программную систему.

ПК 3.3. Выполнять отладку программного продукта с использованием специализированных программных средств.

ПК 3.4. Осуществлять разработку тестовых наборов и тестовых сценариев.

ПК 3.5. Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования.

ПК 3.6. Разрабатывать технологическую документацию.

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

Цель методических указаний: оказание помощи обучающимися в выполнении курсовой работы по МДК 03.01. «Технология разработки программного обеспечения» в составе ПМ 03 «Участие в интеграции программных модулей».

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями, обучающийся в ходе освоения профессионального модуля, должен:

**иметь практический опыт:**

* участия в выработке требований к программному обеспечению;
* участия в проектировании программного обеспечения с использованием специализированных программных пакетов;

**уметь***:*

* владеть основными методологиями процессов разработки программного обеспечения;
* использовать методы для получения кода с заданной функциональностью и степенью качества;

**знать***:*

* модели процесса разработки программного обеспечения;
* основные принципы процесса разработки программного обеспечения;
* основные подходы к интегрированию программных модулей;
* основные методы и средства эффективной разработки;
* основы верификации и аттестации программного обеспечения;
* концепции и реализации программных процессов;
* принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения;
* методы организации работы в коллективах разработчиков программного обеспечения;
* основные положения метрологии программных продуктов, принципы построения, проектирования и использования средств для измерений характеристик и параметров программ, программных систем и комплексов;
* стандарты качества программного обеспечения;
* методы и средства разработки программной документации.

ИСХОДНЫЕ ДАННЫЕ

*Стадия разработки № 1. Техническое задание.*

Этап 1.1. Обоснование необходимости разработки программы.

1.1.1. Постановка задачи.

1.1.2. Сбор исходных материалов.

1. Выбор и обоснование критериев эффективности и качества разрабатываемой программы.
2. Обоснование необходимости проведения научно-исследовательских работ.

Этап 1.2. Научно-исследовательские работы.

1.2.1. Определение структуры входных и выходных данных.

1.2.2. Предварительный выбор методов решения задачи.

1.*23.* Обоснование целесообразности применения ранее разработанных программ.

1. Определение требований к техническим средствам.
2. Обоснование принципиальной возможности решения поставленной задачи.

Этап 1.3. Разработка и утверждение технического задания.

1. Определение требований к программе.
2. Разработка технико-экономического обоснования разработки программы.
3. Определение стадий, этапов и сроков разработки программы и документации на неё.
4. Выбор языков программирования.
5. Определение необходимости проведения научно-исследовательских работ на последующих стадиях.
6. Согласование и утверждение технического задания.

*Стадия разработки № 2. Эскизный проект.*

Этап 2.1. Разработка эскизного проекта.

1. Предварительная разработка структуры входных и выходных данных.
2. Уточнение методов решения задачи.
3. Разработка общего описания алгоритма решения задачи.
4. Разработка технико-экономического обоснования.

Этап 2.2. Утверждение эскизного проекта.

1. Разработка пояснительной записки.
2. Согласование и утверждение эскизного проекта.

*Стадия разработки №3. Технический проект.*

Этап З.1. Разработка технического проекта.

1. Уточнение структуры входных и выходных данных.
2. Разработка алгоритма решения задачи.

3.13. Определение формы представления входных и выходных данных.

1. Определение семантики и синтаксиса языка.
2. Разработка структуры программы.
3. Окончательное определение конфигурации технических средств.

Этап 3.2 Утверждение технического проекта.

1. Составление плана мероприятий по разработке и внедрению программы.
2. Разработка пояснительной записки.
3. Согласование и утверждение технического проекта.

*Стадия разработки №4. Рабочий проект.*

Этап 4.1. Разработка программы.

4.1.1. Программирование и отладка программы.

Этап 4.2. Разработка программной документации.

4.2.1. Разработка программных документов в соответствии с требованиями ГОСТов.

Этап 4.3. Испытания программы.

1. Разработка, согласование и утверждение программы и методики испытаний.
2. Проведение предварительных приемо-сдаточных и других видов испытаний.
3. Корректировка программы и программной документации по результатам испытаний.

*Стадия разработки №5. Внедрение.*

Этап 5.1. Подготовка и передача программы.

1. Подготовка и передача программы и программной документации для сопровождения и (или) изготовления.
2. Оформление и утверждение акта о передаче программы на сопровождение и (или) изготовление.

СОСТАВ ПРОЕКТА

*1. Виды программных документов.*

*Компонент* **-** программа, рассматриваемая как единое целое, выполняющая законченную функцию и применяемая самостоятельно или в составе комплекса.

*Комплекс* **-** программа, состоящая из двух или более компонентов и (или) комплексов, выполняющих взаимосвязанные функции, и применяемая самостоятельно или в составе другого комплекса.

1.1. Спецификация.

Состав программы и документация на неё.

1.2. Ведомость держателей подлинников.

Перечень предприятий, на которых хранят подлинники программных документов.

1.3. Текст программы.

Запись программы с необходимыми комментариями.

1.4. Описание программы.

Сведения о логической структуре и функционировании программы.

1.5. Программа и методика испытаний.

Требования, подлежащие проверке при испытании программы, а также порядок и методы их контроля.

1.6. Техническое задание.

Назначение и область применения программы, технические, технико-экономические и специальные требования, предъявляемые к программе, необходимые стадии и сроки разработки, виды испытаний.

1.7. Пояснительная записка.

Схема алгоритма, общее описание алгоритма и (или) функционирования программы, а также обоснование принятых технических и технико-экономических решений.

1.8. Эксплуатационные документы.

Сведения для обеспечения функционирования и эксплуатации программы.

*2. Виды эксплуатационных документов.*

2.1. Ведомость эксплуатационных документов.

Перечень эксплуатационных документов на программу.

2.2. Формуляр.

Основные характеристики программы, комплектность и сведения об эксплуатации программы.

2.3. Описание применения.

Сведения о назначении программы, области применения, применяемых методах, классе решаемых задач, ограничениях для применения, минимальной конфигурации технических средств.

2.4. Руководство системного программиста.

Сведения для проверки, обеспечения функционирования и настройки программы на условия конкретного применения.

2.5. Руководство программиста.

Сведения для эксплуатации программы.

2.6. Руководство оператора.

Сведения для обеспечения процедуры общения оператора с вычислительной системой в процессе выполнения программы.

2.7. Описание языка.

Описание синтаксиса и семантики языка.

2.8. Руководство по техническому обслуживанию.

Сведения для применения тестовых и диагностических программ при обслуживании технических средств.

**Варианты заданий на курсовую работу (решение математических задач с использованием численных методов)**

Задача №1

Разработать алгоритм и составить программу решения системы уравнений методом Гаусса, Зейделя или итераций.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Система уравнений | Точность (ε) | Метод |
|  |  | 0,01 | Гаусса |
|  |  | 0.1 | Зейделя |
|  |  | 0.1 | Итераций |
|  |  | 0,01 | Гаусса |
|  |  | 0.1 | Зейделя |
|  |  | 0.01 | Итераций |
|  |  | 0,1 | Гаусса |
|  |  | 0.1 | Зейделя |
|  |  | 0.001 | Итераций |
|  |  | 0,01 | Гаусса |
|  |  | 0.1 | Зейделя |
|  |  | 0.1 | Итераций |
|  |  | 0,01 | Гаусса |
|  |  | 0.1 | Зейделя |
|  |  | 0.01 | Итераций |
|  |  | 0,01 | Гаусса |
|  |  | 0.1 | Зейделя |
|  |  | 0.1 | Итераций |
|  |  | 0,01 | Гаусса |
|  |  | 0.1 | Зейделя |
|  |  | 0.001 | Итераций |
|  |  | 0,01 | Гаусса |
|  |  | 0.1 | Зейделя |
|  |  | 0.01 | Итераций |
|  |  | 0,01 | Гаусса |
|  |  | 0.1 | Зейделя |
|  |  | 0.1 | Итераций |
|  |  | 0,01 | Гаусса |
|  |  | 0,1 | Зейделя |
|  |  | 0,001 | Итераций |

Задача №2

Разработать алгоритм и составить программу вычисления приближённого значения интеграла по формуле прямоугольников, трапеций или Симпсона (парабол), 3/8, Гаусса.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Интеграл** | **n** | **Формула** |
|  |  | 10 | Прямоугольников |
|  |  | 12 | Трапеций |
|  |  | 5 | Парабол |
|  |  | 10 | 3/8 |
|  |  | 12 | Гаусса |
|  |  | 5 | Прямоугольников |
|  |  | 8 | Трапеций |
|  |  | 12 | Парабол |
|  |  | 4 | 3/8 |
|  |  | 10 | Гаусса |
|  |  | 9 | Прямоугольников |
|  |  | 4 | Трапеций |
|  |  | 8 | Парабол |
|  |  | 12 | 3/8 |
|  |  | 5 | Гаусса |
|  |  | 10 | Прямоугольников |
|  |  | 12 | Трапеций |
|  |  | 5 | Парабол |
|  |  | 8 | 3/8 |
|  |  | 9 | Гаусса |
|  |  | 4 | Прямоугольников |
|  |  | 10 | Трапеций |
|  |  | 9 | Парабол |
|  |  | 4 | 3/8 |
|  |  | 8 | Гаусса |
|  |  | 12 | Прямоугольников |
|  |  | 5 | Трапеций |
|  |  | 10 | Парабол |
|  |  | 12 | 3/8 |
|  |  | 5 | Гаусса |

Задача №3

Разработать алгоритм и составить программу решения нелинейного уравнения методом простой итерации, дихотомии (проб), Ньютона (касательных), хорд или комбинированным (хорд и касательных) методом.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Уравнение** | **Точность** | **Метод** |
|  | lnx+(x+1)3=0 | 0.1 | Простой итерации |
|  | x4-x-1=0 | 0.01 | Дихотомии |
|  | x+sinx=0.25 | 0.01 | Ньютона |
|  | x-cosx=0.25 | 0.01 | Хорд |
|  | x3+x-5=0 | 0.01 | Комбинированный |
|  | x3+2x2+2=0 | 0.1 | Простой итерации |
|  | (x-1)22x=1 | 0.01 | Дихотомии |
|  | x3-3x2+9x-8=0 | 0.01 | Ньютона |
|  | 2x3-3x2-12x-10=0 | 0.01 | Хорд |
|  | 2sinx=x+2 | 0.01 | Комбинированный |
|  | x·2x=1 | 0.01 | Простой итерации |
|  | x4-18x2+6=0 | 0.01 | Дихотомии |
|  | 2x3-3x2-12x-5=0 | 0.01 | Ньютона |
|  | x3-3x2+6x-2=0 | 0.01 | Хорд |
|  | x3-3x2-24x-8=0 | 0.1 | Комбинированный |
|  | x3-3x2+9x-10=0 | 0.01 | Простой итерации |
|  | (x-2)22x=1 | 0.01 | Дихотомии |
|  | 3x-cosx=1 | 0.01 | Ньютона |
|  | 3x=1+cosx | 0.01 | Хорд |
|  | x3+0.2x2+0.5x-1.2=0 | 0.1 | Комбинированный |
|  | x3-2x+2=0 | 0.01 | Простой итерации |
|  | x2-20sinx=0 | 0.01 | Дихотомии |
|  | x3+0.2+0.5x-1=0 | 0.01 | Ньютона |
|  | x3+9x2-21=0 | 0.01 | Хорд |
|  | 2x3+3x2-2=0 | 0.1 | Комбинированный |
|  |  | 0.01 | Простой итерации |
|  | xlg(x+1)=1 | 0.01 | Дихотомии |
|  | 2x-2sinx=1 | 0.01 | Ньютона |
|  | 3x-3sinx=1.5 | 0.01 | Хорд |
|  | x3-3x2+3=0 | 0.1 | Комбинированный |

**Варианты заданий на курсовую работу (методы криптографических преобразований)**

Создать программу, способную зашифровывать и расшифровывать вводимый текст пятью способами:

* + методом Атбаш,
  + шифром Цезаря,
  + квадратом Полибия,
  + методом Гронсфельда,
  + методом Виженера,
  + методом хэширования,
  + другим криптографическим методом.

**Приложения**

1. Содержание и объем разделов курсовой работы.
2. Правила оформления программной документации.
3. Описание численных методов.
4. Описание криптографических методов.

***Приложение 1***

**Содержание и объем разделов курсовой работы**

Пояснительная записка к курсовой работе по разработке программного обеспечения, должна содержать следующие разделы.

1. Введение.
2. Постановка задачи.
3. Разработка алгоритма и программы решаемой задачи.
   1. Математическое теоретическое обоснование.
   2. Алгоритмическая структура.
   3. Блок-схемы модулей.
   4. Программная документация.
      1. Руководство пользователя.
      2. Руководство программиста.
      3. Руководство системного программиста.
      4. Руководства оператора.
      5. Экспериментальное исследование программы.
         1. Анализ результатов экспериментальных проверок.
         2. Экономическая часть.
4. Заключение.
5. Приложения. Листинги программных модулей.
6. Список литературы.
7. Графическая часть:

Постановка задачи     - 1 лист.

Блок-схемы модулей  - 3 листа.

Экспериментальные результаты   - 1 лист.

Экономическая часть      - 1 лист.

***Приложение 2***

**Правила оформления программной документации**

**ГОСТ 19.502-78. Описание применения**

1. Настоящий стандарт устанавливает состав и требования к содержанию программного документа "Описание применения", определенного ГОСТ 19.101-77.

Стандарт полностью соответствует СТ СЭВ 2093-80.

2. Структуру и оформление документа устанавливают в соответствии с ГОСТ 19.105.-78.

Составление информационной части (аннотации и содержания) является обязательным.

3. Текст документа должен состоять из следующих разделов:

- назначение программы;

- условия применения;

- описание задачи;

- входные и выходные данные.

4. В зависимости от особенностей программы допускается вводить дополнительные разделы или объединять отдельные разделы.

5. В разделе "Назначение программы" указывают назначение, возможности программы, ее основные характеристики, ограничения, накладываемые на область применения программы.

6. В разделе "Условия применения" указывают условия, необходимые для выполнения программы (требования к необходимым для данной программы техническим средствам и другим программам, общие характеристики входной и выходной информации, а также требования и условия организационного, технического и технологического характера и т.п.).

7. В разделе "Описание задачи" должно быть описание задачи и методы ее решения.

8. В разделе "Входные и выходные данные" должны быть указаны сведения о входных и выходных данных.

9. В приложение к общему описанию могут быть включены справочные материалы (иллюстрации, таблицы, графики, примеры и т.п.).

**ГОСТ 19.505-79. Руководство оператора**

Настоящий стандарт устанавливает требования к содержанию и оформлению программного документа "Руководство оператора", определенного ГОСТ 19.101-77.

Стандарт полностью соответствует СТ СЭВ 2096-80.

1. Общие положения

1.1. Структура и оформление программного документа устанавливаются в соответствии с ГОСТ 19.105-78.

Составление информационной части (аннотации и содержания) является обязательным.

1.2. Руководство оператора должно содержать следующие разделы:

- назначение программы;

- условия выполнения программы;

- выполнение программы;

- сообщения оператору.

В зависимости от особенностей документа допускается объединять отдельные разделы или вводить новые.

2. Содержание разделов

2.1. В разделе "Назначение программы" должны быть указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

2.2. В разделе "Условия выполнения программы" должны быть указаны условия, необходимые для выполнения программы (минимальный и/или максимальный состав аппаратных и программных средств и т.п.).

2.3. В разделе "Выполнение программы" должна быть указана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузку и управляет выполнением программы, а также ответы программы на эти команды.

2.4. В разделе "Сообщения оператору" должны быть приведены тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия оператора (в случае сбоя, возможности повторного запуска программы и т.п.).

2.5. Допускается содержание разделов иллюстрировать поясняющими примерами, таблицами, схемами, графиками.

2.6. В приложения к руководству оператора допускается включать различные материалы, которые нецелесообразно включать в разделы руководства.

**ГОСТ 19.504-79. Руководство программиста**

Настоящий стандарт устанавливает требования к содержанию и оформлению программного документа "Руководство программиста", определенного ГОСТ 19.101-77.

Стандарт полностью соответствует СТ СЭВ 2095-80.

1. Общие положения

1.1. Структура и оформление программного документа устанавливаются в соответствии с ГОСТ 19.105-78.

Составление информационной части (аннотации и содержания) является обязательным.

1.2. Руководство программиста должно содержать следующие разделы:

- назначение и условия применения программы;

- характеристики программы;

- обращения к программе;

- входные и выходные данные;

- сообщения оператору.

В зависимости от особенностей документа допускается объединять отдельные разделы или вводить новые.

2. Содержание разделов

2.1. В разделе "Назначение и условия применения программы" должны быть указаны назначение и функции, выполняемые программой, условия, необходимые для выполнения программы (объем оперативной памяти, требования к составу и параметрам периферийных устройств, требования к программному обеспечению и т.п.).

2.2. В разделе "Характеристика программы" должно быть приведено описание основных характеристик и особенностей программы (временные характеристики, режим работы, средства контроля правильности выполнения и самовосстанавливаемости программы и т.п.).

2.3. В разделе "Обращение к программе" должно быть приведено описание процедур вызова программы (способы передачи управления и параметров данных и др.).

2.4. В разделе "Входные и выходные данные" должно быть приведено описание организации используемой входной и выходной информации и при необходимости ее кодирования.

2.5. В разделе "Сообщения" должны быть указаны тексты сообщений, выдаваемых программисту или оператору в ходе выполнения программы, описание их содержания и действия, которые необходимо предпринять по этим сообщениям.

2.6. В приложении к руководству программиста могут быть приведены дополнительные материалы (примеры, иллюстрации, таблицы, графики и т.п.).

**ГОСТ 19.503-79. Руководство системного программиста**

Настоящий стандарт устанавливает требования к содержанию и оформлению программного документа "Руководство оператора", определенного ГОСТ 19.101-77.

Стандарт полностью соответствует СТ СЭВ 2094-80.

1. Общие положения

1.1. Структура и оформление программного документа устанавливаются в соответствии с ГОСТ 19.105-78.

Составление информационной части (аннотации и содержания) является обязательным.

1.2. Руководство оператора должно содержать следующие разделы:

- структура программы;

- настройка программы;

- проверка программы;

- дополнительные возможности;

- сообщения системному программисту.

В зависимости от особенностей документа допускается объединять отдельные разделы или вводить новые.

В особых случаях допускается раздел "Дополнительные возможности" не вводить, а в наименованиях разделов опускать слово "программа", или заменять его на "наименование программы".

2. Содержание разделов

2.1. В разделе "Общие сведения о программе" должны быть указаны назначение и функции программы и сведения о технических и программных средствах, обеспечивающих выполнение данной программы.

2.2. В разделе "Структура программы" должны быть приведены сведения о структуре программы, ее составных частях, о связях между составными частями и о связях с другими программами.

2.3. В разделе "Настройка программы" должно быть приведено описание действий по настройке программы на условия конкретного применения (настройка на состав технических средств, выбор функций и др.).

При необходимости приводят поясняющие примеры.

2.4. В разделе "Проверка программы" должно быть приведено описание способов проверки, позволяющих дать общие заключения о работоспособности программы (контрольные примеры, методы прогона, результаты).

2.5. В разделе "Дополнительные возможности" должно быть приведено описание дополнительных разделов функциональных возможностей программы и способов их выбора.

2.6. В разделе "Сообщения системному программисту" должны быть указаны тексты сообщений, выдаваемых в ходе выполнения настройки, проверки программы, а также в ходе выполнения программы, описание их содержания и действий, которые необходимо предпринять по этим сообщениям.

2.7. В приложении к руководству системного программиста могут быть приведены дополнительные материалы (примеры, иллюстрации, таблицы, графики и т.п.).

**Содержание и объем разделов курсовой работы**

К курсовым работам предъявляются высокие требования в плане теоретической проработки материала, поэтому в разделе выбора методов реализации заданной разработки необходимо тщательное изложение рассматриваемых вопросов с математическими выкладками и доказательствами. Объем раздела должен составлять до 30 % объема  всей работы. Особое внимание уделяется разработке и оценке оптимальности  алгоритмов. При этом в разделе должна быть именно разработка алгоритмов, а не их простое описание. Разработка алгоритмов должна вестись с точки зрения их оптимальности по временным, аппаратурным или трудовым затратам. Алгоритмы должны быть обязательно увязаны со всеми остальными подсистемами и проверены путем их моделирования на заданной ЭВМ. Объем разделов, связанных с разработкой алгоритмов и программ, должен составлять 30-40 % объема всей работы.

**Оформление курсовой работы**

Курсовой проект состоит из двух частей - пояснительной записки и графического демонстрационного материала. Пояснительная записка должна содержать 30-40 листов текста с рисунками и схемами, графическая часть - 6-8 листов формата А4.

Пояснительная записка должна содержать в указанной последовательности: титульный лист, задание на курсовой проект, утвержденное заместителем директора по учебной работе, реферат, в котором указываются сведения об объеме пояснительной записки, количестве рисунков и таблиц, приводится перечень ключевых слов, характеризующий основное содержание записки. Перечень содержит от 5 до 15 ключевых слов, приведенных в строку через запятые в именительном падеже, например: микропроцессор, микропроцессорные системы, транслятор. В тексте реферата содержится краткая формулировка задания и полученных результатов. Реферат составляется на русском языке. Далее в пояснительную записку входит: содержание (оглавление), основная текстовая часть, список использованной литературы, приложение в виде спецификаций, чертежей, графиков, таблиц и т.п., включение которых в основную текстовую часть проекта нецелесообразно.

Записка печатается на принтере на одной стороне стандартного листа формата А4 через два интервала. По всем четырем сторонам листа должны оставляться поля. Размер левого поля не менее 30 мм, правого - не менее 10 мм, верхнего - не менее 15 мм и нижнего - не менее 20 мм. При оформлении пояснительной записки рекомендуется пользоваться ГОСТ 7.32-91, весь демонстрационный материал, т.е. листы, предназначенные для защиты (чертежи, схемы, графики), должны быть дублированы в пояснительной записке в виде копий листов.

Текст разделов должен разбиваться на подразделы и пункты. Разделы нумеруются арабскими цифрами. "Введение" и "Заключение" не нумеруются.

Подразделы обозначаются: 1.1, 1.2; пункты обозначаются 1.1.1, 1.1.2; 1.2.1, 1.2.2 и т.д. Номер раздела или подраздела ставится в начале заголовка, номер пункта в начале 1-й строки абзаца, с которого начинается соответствующий пункт. В конце заголовка точку не ставят. В заголовках не допускается переносить слова.

Нумерация страниц должна быть сквозной. Первой страницей является титульный лист. Номер страницы проставляется в правом нижнем углу. На странице 1 (титульный лист) номер не ставят. Заголовки разделов, подразделов с указанием номеров страниц, на которых они помещены, последовательно перечисляют в оглавлении.

Все иллюстрации (фотографии, схемы, чертежи и пр.) именуются рисунками. Рисунки нумеруются в пределах раздела. Каждый рисунок должен сопровождаться содержательной подписью. При ссылке на рисунок указывают его полный номер, например: (рис. 1.2).

Цифровой материал рекомендуется оформлять в виде таблиц. Таблицы нумеруются в пределах раздела. Над правым верхним углом таблицы помещают надпись, например: Таблица 1.2 (вторая таблица первого раздела). Каждая таблица должна иметь содержательный заголовок, который помещается под словом "Таблица" над соответствующей таблицей. Подчеркивать заголовок не следует. Таблицу следует помещать после первого упоминания о ней в тексте. В таблице не допускается ставить кавычки вместо повторяющихся цифр, математических символов.

Если цифровые или иные данные в какой-либо строке таблицы не приводят, то в ней ставят прочерк. В таблице не следует делать графу "№ п/п". При ссылке на таблицу указывают номер (полный), например: (табл. 1.2). При повторной ссылке следует указать: (см. табл. 1.2). Все рисунки и таблицы выполняются в полном соответствии с требованиями ЕСПД.

Формулы, на которые имеются ссылки в тексте, нумеруются в пределах раздела. Номер формулы указывается в скобках около правого поля на уровне нижней строки формулы, например: (1.2) (вторая формула первого раздела). Расшифровка буквенных символов приводится непосредственно под формулой в той последовательности, в какой они даны в формуле. Значение каждого символа числового коэффициента следует давать с новой строки. Первую строку расшифровки начинают со слова "где" (двоеточия после него  не ставят). При ссылке в тексте на формулу необходимо указать ее полный номер в скобках, например: (1.2).

В список литературы заносят все использованные источники (не менее 5-10 наименований, включая зарубежные). Сведения о книгах должны содержать: фамилию и инициалы автора, заглавие книги, место издания, издательство и год издания. Сведения о статье из периодического издания должны включать: фамилию и инициалы автора, заглавие статьи, наименование журнала, наименование серии (если таковая имеется), год выпуска, номер  журнала, страницы, на которых помещена статья.

Графическая часть курсовой работы представляется на листах формата А4.

**Порядок защиты курсовой работы**

Защита курсовых работ происходит на зачетном занятии.

Для защиты обучающемуся предоставляется для доклада не более 15 минут.

В докладе следует изложить постановку задачи, ее актуальность и новизну, главные этапы и результаты работы и четко сформулировать выводы. Далее защищающемуся студенту задаются вопросы по курсовой работе.

***Приложение 3***

**Численные методы**

***Метод половинного деления***

Пусть хо –корень уравнения f(x)=0, отделенный на отрезке , т.е. < 0, , f(x) – непрерывная функция.

Возьмем на отрезке  точку  - середину отрезка . Тогда точка с разделит отрезок на два отрезка  , причем на одном из них функция будет иметь значения разных знаков, в этом отрезке и будет находиться корень уравнения, (или если f(c)=0, то с – корень уравнения). Выбираем тот отрезок, на концах которого функция принимает значения разных знаков, и делим его пополам. Аналогичные действия проводим до тех пор, пока концы отрезка не будут отличаться на заданную точность или значение функции в найденной точке будет равно нулю.

***Метод хорд***

Пусть хо –корень уравнения f(x)=0, отделенный на отрезке , т.е. < 0, , f(x) – непрерывная функция.

Идея метода хорд состоит в том, что на достаточно малом отрезке  дуга кривой заменяется её хордой. В качестве приближенного значения принимается точка пересечения хорды с осью абсцисс. Корень находится в том интервале, на концах которого функция принимает значения разных знаков. Из уравнения хорды можно найти значение точки пересечения с осью абсцисс это и будет первое приближение.

.(1)

Затем вычисляем и берем тот из промежутков,  на концах которого функция принимает значения разных знаков, искомый корень находится в данном промежутке. К этому промежутку применяем формулу (1) и находим второе приближение. И так далее до тех пор пока концы интервала не будут отличаться на заданную точность или значение функции в найденном приближении не будет равно нулю.

***Метод касательных***

Пусть хо –корень уравнения f(x)=0, отделенный на отрезке , т.е. < 0, , f(x) – непрерывная функция.

Идея метода касательных состоит в том, что на достаточно малом отрезке  проводится касательная в том конце отрезка, где  имеют одинаковые знаки. В качестве приближенного значения принимается точка пересечения касательной с осью абсцисс. Из уравнения касательной можно найти значение точки пересечения с осью абсцисс это и будет первое приближение.

 - если касательная проходит в точке b

 - если касательная проходит в точке 

В обоих случаях второе приближение находится по формуле

 Продолжая процесс подобным образом до тех пор, пока приближения не будут отличаться на заданную точность, находим все последующие приближения.

***Комбинированный метод***

Методы хорд и касательных дают приближения с разных сторон, следовательно, целесообразно применять их одновременно, т.к. процесс приближения будет проходить быстрее.

Пусть хо –корень уравнения f(x)=0, отделенный на отрезке , т.е. < 0, , f(x) – непрерывная функция и  конец отрезка где принимают значения одинаковых знаков. Тогда

/если же знаки одинаковы в точке b, то /

Искомый корень заключен в отрезке от  до , для этого отрезка применяются метод хорд и касательных, до тех пор, пока концы отрезка не будут отличаться на заданную точность.

***Метод итераций***

Одним из наиболее важных способов численного решения дифференциальных уравнений является метод итераций (метод последовательного приближения).

Пусть хо –корень уравнения f(x)=0, отделенный на отрезке , т.е. < 0, , f(x) – непрерывная.

Уравнение f(x)=0 заменим уравнением , выберем приближение хо и подставим его в правую часть уравнения . Таким образом, получим первое приближение . Аналогично получаем все последующие приближения, до тех пор, пока концы отрезка не будут отличаться на заданную точность.

Функция находится из соотношения

, где , где  , знак k такой же как знак производной на отделенном отрезке.

***Метод Гаусса***

Данный метод является наиболее распространенным методом. Рассмотрим систему из четырех уравнений с четырьмя неизвестными

(1)

Пусть -ведущий элемент. Разделим коэффициенты первого уравнения системы (1) на ведущий элемент. Получим уравнение:

()

Для исключения из системы (1) первого неизвестного, достаточно из второго уравнения системы вычесть уравнение (), умноженное на элемент , из третьего уравнения этой же системы вычесть уравнение (), умноженное на , из четвертого уравнения системы вычесть уравнение (), умноженное на. В результате получаем систему из трех уравнений

С данной системой проделываются действия, аналогичные действиям с системой (1). Такие действия проделываются до тех пор, пока не останется одно уравнение с одним неизвестным. Затем проделывают обратный ход: из последнего полученного уравнения выражают последнюю неизвестную и подставляют её в предыдущую систему, и так далее до тех пор пока не найдут все неизвестные.

***Метод итераций***

Рассмотрим систему уравнений

Пусть все диагональные элементы .Тогда из первого уравнения выразим первую перемененную, из второго - вторую, из третьего - третью, из четвертого – четвертую.

Считают, что такая система приведена к нормальному виду, её можно переписать в матричной форме:

Решим данную систему методом последовательных приближений. За нулевое приближение принимаем, например, столбец свободных членов

Далее последовательно строим матрицы столбцы

………………..

Если последовательность приближений имеет предел, то этот предел является решением системы.

***Метод Зейделя***

Метод Зейделя является интерпретацией метода итераций. Основная его идея заключается в том, что при вычислении (k+1) приближения неизвестной учитываются уже вычисленные ранее приближения неизвестных.

***Простейшие квадратурные формулы***

Пусть требуется вычислить:

 (1)

где  - непрерывная на  функция. Формулы приближенного вычисления интеграла  называют квадратурными формулами, они имеет вид:



где  - натуральное число. Числа  называются узлами, а числа  - коэффициентами квадратурной формулы.

Введем на  равномерную сетку узлов с шагом 



Таким образом, получим:



Пусть  - значение подынтегральной функции в узле . Рассмотрим простейшие квадратные формулы.

***1. Формулы левых и правых прямоугольников.***

 (2)

***2. Формула трапеций.***

 (3)

***3. Формула парабол (Симпсона), n-четное.***

(4)

Введем обозначения:

 (5)



Тогда квадратурные формулы (2)-(4) запишутся так:

* ***формулы левых и правых прямоугольников:***

 (6)

* ***формула трапеций:***

 (7)

* ***формула Симпсона:***

 (8)

* ***формула 3/8:***

Введем обозначения:

*для вычисления*

***n кратно 3, h =***

***A= y0 + yn, B= y1 + y2 + y4 + y5 +…, C= y3 + y6 + y9 +…***

 (9)

***Приложение 4***

***Атбаш***

Атба́ш (ивр. аъб"щэ) — простой шифр подстановки для иврита.

Правило шифрования

Состоит в замене i-й буквы алфавита буквой с номером n − i + 1, где n — число букв в алфавите.

Пример для латинского алфавита выглядит так:

Исходный текст: abcdefghijklmnopqrstuvwxyz

Зашифрованный текст: ZYXWVUTSRQPONMLKJIHGFEDCBA

История

Происхождение слова «атбаш» объясняется принципом замены букв. Слово аъбщ составлено из букв «алеф», «тав», «бет» и «шин», то есть первой и последней, второй и предпоследней букв еврейского алфавита.

Шифр Атбаш был изобретен Ессеями - иудейской сектой повстанцев. Они разработали множество различных кодов и шифров, которые использовались для сокрытия важных имен и названий, чтобы потом избежать преследования. Знания этих кодов и шифров были потом переданы Гностикам, которые, в свою очередь, передали их Катарам. Позже Орден Тамплиеров завербовал Катарских дворян и перенял знания шифров. Таким образом, шифр был использован на протяжении многих лет, от около 500 до н.э. до 1300 г. н.э. — момента, когда Орден Тамплиеров был распущен.

***Метод Цезаря***

Шифр Цезаря (шифр сдвига) — один из самых простых и наиболее широко известных методов шифрования.

Правило шифрования

Реализует кодирование фразы путем «сдвига» всех букв фразы на определенное число n (в оригинальном шифре Цезаря число n равнялось 3). Если буква кодируемой фразы имеет в алфавите позицию j, то она в "шифровке" будет заменяться буквой, находящейся в алфавите на позиции j + n.

Пример для латинского алфавита выглядит так:

Исходный текст: Съешь же ещё этих мягких французских булок, да выпей чаю.

Зашифрованный текст: Фэзыя йз зьи ахлш пвёнлш чугрщцкфнлш дцосн, жг еютзм ъгб.

История и применение

Согласно «Жизни двенадцати цезарей» Светония, в I в до н. э. Гай Юлий Цезарь во время войны с галлами, переписываясь со своими генералами в Риме, заменял в сообщении первую букву латинского алфавита (А) на четвертую (D), вторую (В) – на пятую (Е), наконец, последнюю – на третью. Цезарь использовал сдвиг на три позиции.

***Квадрат Полибия***

Квадрат Полибия (англ. Polybius square), также известный как шахматная доска Полибия — оригинальный код простой замены.

Правило шифрования

Применительно к современному латинскому алфавиту из 26 букв шифрование по этому квадрату заключалось в следующем. В квадрат размером 5x6 клеток выписываются все буквы алфавита, при этом буквы I,J не различаются (J отождествляется с буквой I). Шифруемая буква заменялась на координаты квадрата, в котором она записана. Так, B заменялась на AB, F на BA, R на DB и т.д. При расшифровании каждая такая пара определяла соответствующую букву сообщения. Ключом такого шифра являлось расположение букв в таблице к примеру 5x5. Начальное расположение букв должно определяться ключом. В современном латинском алфавите 26 букв, следовательно таблица должна состоять из 5 строк и 5 столбцов, так как 25=5\*5 наиболее близкое к 26 число. Но так как английский алфавит всё же содержит 26 букв, мной будет разработана программа из 6 строк и 6 столбцов, содержащая все символы латинского алфавита.

Пример для латинского алфавита выглядит так:

Исходный текст: abcdefghijklmnopqrstuvwxyz

Зашифрованный текст:

DCDDDEDFDGDHEAEBECEDEEEFEGEHFAFBFCDGFEFFFGFHGAGBGCGD

История и применение

Данный вид кодирования изначально применялся для греческого алфавита, но затем был распространен на другие языки. Квадрат Полибия - одна из древнейших систем кодирования, был разработан Полибием (греческий историк, полководец, государственный деятель, III век до н. э.). [4]

***Шифр Гронсфельда***

Шифр Гронсфельда - многоалфавитный шифр сложной замены.

Правило шифрования.

Схема очень напоминает шифр Цезаря, дополненного числовым ключом. При шифровании сообщения его выписывают в строку, а под ним ключ. Если ключ оказался короче сообщения, то его циклически повторяют. Шифровку получают, находя символ в колонке таблицы по букве текста и строке, соответствующей букве ключа.

Пусть нам надо зашифровать слово EXALTATION. Например, берём в качестве ключа число 31415, затем составляем следующую таблицу:

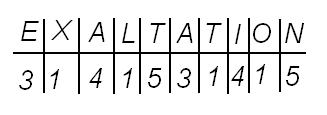


Рисунок - Таблица Гронсфельда с ключом

Получается, что каждой букве соответствует некая цифра, это цифра будет показывать, на сколько позиций будет происходить смещение алфавита для каждой конкретной буквы. Например, покажем, как преобразуется буква E:

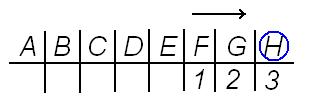


Рисунок - Таблица Гронсфельда. Смещение символа алфавита

То есть букве E соответствует буква H. Таким образом, для всего слова получаем зашифрованный текст: HYEMYDUMPS. Обратное преобразование происходит подобным образом, только каждый раз сдвигаем алфавит в другую сторону.

Исходный текст: EXALTATION

Зашифрованный текст: HYEMYDUMPS

Ключ: 31415

История и применение

Идея использования лозунга (гаммы) без изображения указанных выше таблиц была предложена в 1734 году бельгийцем Хосе де Бронкхором и начальником первого дешифровального отделения в Германии, военным и дипломатом граф ом Гронсфельдом.

***Шифр Виженера***

Шифр Виженера (фр. Chiffre de Vigenиre) — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова. Является простой формой многоалфавитной замены

Правило шифрования

Состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На разных этапах кодировки шифр Виженера использует различные алфавиты из этой таблицы. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.



Рисунок - Таблица Виженера ( tabula recta)

Например, предположим, что исходный текст имеет вид:

ATTACKATDAWN

Человек, посылающий сообщение, записывает ключевое слово («LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста: LEMONLEMONLE

Первый символ исходного текста A зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста X получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом.

Исходный текст: ATTACKATDAWN

Зашифрованный текст: LXFOPVEFRNHR

Ключ: LEMONLEMONLE

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

Если буквы A-Z соответствуют числам 0-25, то шифрование Виженера можно записать в виде формулы:



Формула для шифрония методом Виженера



Формула для дешифрония методом Виженера

История и применение

Первое точное документированное описание многоалфавитного шифра было сформулированно Леоном Баттиста Альберти в 1467 году, для переключения между алфавитами использовался металлический шифровальный диск. Система Альберти переключает алфавиты после нескольких зашифрованных слов. Позднее, в 1518 году, Иоганн Трисемус в своей работе «Полиграфия» изобрел tabula recta — центральный компонент шифра Виженера.

То, что сейчас известно под шифром Виженера, впервые описал Джованни Батиста Беллазо в своей книге La cifra del. Sig. Giovan Battista Bellasо. Он использовал идею tabula recta Трисемуса, но добавил ключ для переключения алфавитов шифра через каждую букву.

**Учебно-методическое и информационное обеспечение**

***Основная***

1. Дадаян А.А. Математика. - М.: Инфра - М, 2015, 552с.
2. Дадаян А.А. Сборник задач по математике. - М.: Инфра - М, 2015 352с.
3. Рудаков А.В. Технология программных продуктов. – М.: Издательский центр «Академия», 2012, 320 с.
4. Орлов С.А. Технологии разработки программного обеспечения: Учебник — СПб.: Питер, 2012, 322 с.
5. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие под ред. Л.Г Гагариной. — М: ИД «ФОРУМ»: ИНФРА-М, 2014, 401 с.

***Дополнительная***

1. Данко П.Е., А.Г. Попов, Т.Я. Кожевникова Высшая математика в упражнениях и задачах - М.: «Мир и Образование», 2015-304с.
2. А. Якобсон, Г. Буч, Дж. Рамбо Унифицированный процесс разработки ПО, М., 2012, 365 с.
3. Леффингуал, Дин, Ундри, Дон Принципы работы с требованиями к ПО. Унифицированный подход. М., 2012, 384 с.
4. Луиза Тампе Введение в тестирование программного обеспечения. М.: Вильямс, 2013, 268 с.
5. Барсуков B.C. Обеспечение информационной безопасности. - М.: ТЭК, 2012, 128 с
6. Завгородний В. И., Комплексная защита информации в компьютерных системах (<http://eusi.narod.ru/lib/savgorodnij/>
7. Боэм В., Браун Д. Характеристика качества программного обеспечения-М.: Мир.2014, 286 с
8. Липаев В.В. Качество программного обеспечения - М: Финансы и статистика.2012, 154 с.
9. Шниер, Толковый словарь компьютерных технологий. М:, 2012, 897 с.

***Электронные ресурсы***

1. http://www.onecomplex.ru/
2. http://www.mathedu.ru/

Составитель:

преподаватель Колледжа ПсковГУ Т.О. Ушарнова