Методические указания по выполнению курсовой работы   
по предмету   
"Алгоритмические языки и программирование"

Оглавление

[Введение 1](#_Toc478395905)

[График выполнения работы 1](#_Toc478395906)

[Критерии оценки 2](#_Toc478395907)

[Техническое задание 2](#_Toc478395908)

[Содержание разделов 2](#_Toc478395909)

[Требования к программному коду 3](#_Toc478395910)

[Содержание и оформление отчета 4](#_Toc478395911)

[Общие сведения 5](#_Toc478395912)

[Структура программного обеспечения 5](#_Toc478395913)

[Структуры данных 5](#_Toc478395914)

[Методика и результаты тестирования 5](#_Toc478395915)

[Приложения 5](#_Toc478395916)

# Введение

Данная курсовая работа предусмотрена учебным планом по специальности и выполняется в третьем семестре. Целью данной работы является закрепление практических навыков по курсу "Алгоритмические языки и программирование, Язык С++", расширение эрудиции в профессиональной области и получение и закрепление навыков самостоятельного решения инженерных задач на всех этапах.

## График выполнения работы

В интересах студентов начать работу как можно раньше и выполнять ее в установленном темпе на протяжение всего семестра, оперативно решая появляющиеся проблемы самостоятельно либо взаимодействуя с руководителем. Выполнение работы начинается с утверждения темы и оформления технического задания. Выбор и утверждение темы должно быть сделаны не позднее 4 недели семестра и не позднее 8 недели - написано и утверждено техническое задание. К 14 неделе должна быть закончена отладка программы и в целом - оформление курсовой работы. Не позднее 16 недели должна состояться защита курсовой работы на кафедре.

## Критерии оценки

Курсовая работа будет оцениваться по следующим параметрам:

* работоспособность: программа должна собираться из исходных текстов, запускаться за заданной платформе, функционировать в соответствии с техническим заданием, при всех допустимых входных воздействиях.
* соответствие содержания отчета утвержденному ТЗ и реализованному программному решению.
* качество оформления отчета и полнота описания разработанного ПО.
* сложность реализованных алгоритмов.

При оценке будут учитываться:

* эффективность реализованных алгоритмов
* архитектура программы (разбиение на модули, интерфейсы, связность)
* оформление исходных текстов и читаемость программы

# Техническое задание

Техническое задание пишется студентом после утверждения темы работы и уточнения требований у преподавателя. Оформленное техническое задание должно быть утверждено и подписано преподавателем в срок оговоренный графиком выполнения работ. Типовое ТЗ содержит следующие разделы:

* Аннотация
* Требования к функциональным характеристикам
* Требования к надежности программы
* Требования к составу и параметрам технических средств
* Порядок контроля и приемки

## Содержание разделов

В разделе "Аннотация" должны быть приведены следующие сведения:

* наименование программы
* назначение программы
* краткая характеристика программы и области ее применения
* используемые при разработке инструменты
* основание для разработки программы

В разделе "Требования к функциональным характеристикам ":

* требования к функциональности (подробное описание сценария использования)
* требования по организации входных и выходных данных, их передача в программу иил из нее
* временные характеристики выполнения программы.

В разделе "Требования к надежности программы":

* степень контроля за корректностью входной и выходной информации
* реакция программы на ошибочные ситуации и т.д.

В разделе "Требования к составу и параметрам технических средств":

* аппаратная и программная платформа для запуска программы
* квалификация пользователя, если применимо.

В разделе "Порядок контроля и приемки" должен быть сформулирован порядок проведения испытаний программы при ее приемке: какие тесты будут проводиться и как будет проверяться правильность результата и корректность работы.

# Требования к программному коду

* Исходные тексты должны быть организованы в упорядоченную структуру файлов и каталогов проекта.
* В корневом каталоге проекта должен находиться файл 'Readme.txt', который должен содержать подробную инструкцию по сборке исполняемой программы из исходных текстов и о порядке ее запуска.
* исходные файлы не должны включать в себя промежуточные результаты сборки, такие как объектные файлы, исходные файлы, автоматически генерируемые с помощью различных инструментов (например, те. что создаются с помощью **moc, uic, lex, yacc, antlr, swig, cmake** и т.д.)
* В начале каждого файла с исходным текстом должна находиться "шапка", кратко описывающее состав и функциональное назначение этого файла, например:

/\* ! \file RadioAltimeterRA4500.h

\* \brief Файл содержит определение класса RA4500 - имитационная модель радиовысотомера RA-4500 вертолета Bell407

\* \author milonov.g (Григорий Милонов)

\* \date 26.10.2016

\*/

Далее в исходных текстах рекомендуется придерживаться стиля оформления комментариев, совместимого с системой doxygen.

Стандартным размером табуляции является 2 или 4 знакоместа. В тексте могут быть использованы как сами символы табуляции, так и соответствующее количество пробелов.

Исходный текст для лучшей читаемости должен быть отформатирован:

* блок должен иметь отступ в соответствии со своим уровнем вложенности
* Фигурные скобки должны располагаться на отдельных строках и парные скобки должны быть на одной колонке
* заголовочные файлы должны иметь сторожей для повторного включения
* Имена констант - только заглавными буквами. Еслиимя константы содержит несколько слов, то эти слова разделяются символами "\_".
* Имена переменных-полей класса и имена параметров функций должны быть явно различимы с помощью префиксов. Например: m\_healthAmount - для переменной поля и healthAmount - для параметра или имени метода.
* Допускается использование case-стиля на выбор разработчика (snake case, camel case и т.д.), главное, чтобы не было совместного использования разных стилей. Рекомендуемым стилем является "camel case".
* Имена идентификаторов должны быть информативными.
* Очень не приветствуется использование в коде различных "магических" чисел. Лучше ввести лишнюю константу, поясняющую семантику данного значения, например:

#define COLUMNS 80

#define BUF\_SIZE 1024

и т.д.

* длина строк не должна превышать 80 символов. При необходимости использовать конструкцию переноса на следующую строку для макроопределений, разбивать строковые литералы на несколько частей и т.д.

# Содержание и оформление отчета

1. Параметры документа с отчетом должны быть следующими: формат бумаги – А4; текст – на одной стороне листа; нумерация страниц – внизу, справа; размеры полей: левое – 35 мм; правое, верхнее и нижнее – 15 мм.

2. Основной текст отчета должен быть отформатирован единообразно. Параметры: шрифт Times New Roman, размер 12; выравнивание абзаца по ширине; межстрочный интервал – 1.5.

3. Формат заголовков 1-го уровня: шрифт Times New Roman, размер 14; жирный; выравнивание по центру. Формат заголовков 2-го уровня: шрифт Times New Roman, размер 12; жирный, курсив; выравнивание по центру.

4. Если в текст отчета помещается исходный код программы, то он форматируется следующим образом: шрифт Courier New, размер 8; выравнивание по левому краю.

5. Иллюстрации (графики, диаграммы, схемы, фотографии и т. д.) вставляются в текст, выравниваются по центру. Подпись должна быть у каждого рисунка и должна формироваться следующим образом: «Рис. », последовательный номер рисунка, точка, пробел, название рисунка. На каждый рисунок в тексте должна быть как минимум одна ссылка.

Описание разработки должно содержать следующие разделы:

### Общие сведения

* наименование программы;
* функциональное назначение и сведения функциональных ограничениях на применение;
* программное и аппаратное обеспечение, необходимое для функционирования программы;
* языки программирования, на которых написана программа;

### Структура программного обеспечения

* алгоритм работы программы;
* используемые методы;
* структура программы с описанием функций составных частей и связи между ними;
* связи программы с другими программами.

### Структуры данных

* характер, организация и предварительная подготовка входных данных;
* формат, описание и способ кодирования входных данных.
* характер и организация выходных данных;
* формат, описание и способ кодирования выходных данных.

### Методика и результаты тестирования

* состав и структуру технических средств, необходимых для проведения тестирования, в том числе требования к аппаратным характеристикам компьютера, таким как объем памяти, тип процессора, подключение к Интернету и т.д.;
* состав и структуру программного обеспечения, необходимого для проведения тестирования, в том числе требования к операционной системе, дополнительному программному обеспечению и т.д.
* программа (последовательность) тестирования, указывающую те свойства программного обеспечения, которые будут проверяться и очередность, в которой эти проверки будут производиться.
* Собственно результаты тестирования: "скриншоты" графических сцен, таблицы входных и выходных данных и т.д.

# Приложения

В приложения могут быть включены такие разделы как

* лист рецензии
* список литературы
* исходные коды программы