

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 01.10.2024 09:47:58
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)
Кафедра программной инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
2024г.



Языки программирования
Методические указания по выполнению курсовой работы
для студентов, обучающихся по направлению 02.03.03 Математическое
обеспечение и администрирование информационных систем

Курск 2024

УДК 681.3(075)

Составитель Л.А. Лисицин

Рецензент

Кандидат технических наук, доцент *Халин Ю.А.*

Языки-программирования [Текст]: методические указания по выполнению курсовой работы по направлению 02.03.03 Математическое обеспечение и администрирование информационных систем / Юго-Зап. гос. ун-т; сост.: Л.А. Лисицин. Курск, 2024. 23 с.: ил. 1. табл. 1. Библиогр. с. 23.

Содержат общую формулировку заданий к курсовой работе, методические указания к содержанию и оформлению пояснительной записки к курсовой работе по языкам программирования на C#.

Методические указания предназначены для студентов обучающихся по направлению 02.03.03 Математическое обеспечение и администрирование информационных систем.

Текст печатается в авторской редакции.

Подписано в печать *9.08* Формат 60x84 1/16.
Усл.печ. л. *1,22* Уч.-изд. л. *1,2*. Тираж *100* экз. Заказ *583*. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Содержание

1 Основные положения	4
2 Цель курсового проектирования	4
3 Тематика курсовой работы	5
4 Задание на курсовое проектирование	5
5 Содержание пояснительной записки к курсовой работе	6
6 Рекомендации по выполнению курсовой работы	8
Список используемых источников	13
Приложение А	14
Приложение Б	18
Приложение В	19
Приложение Г	20

1 Основные положения

Курсовая работа (КР) – учебная (учебно-исследовательская) работа студента, являющаяся результатом самостоятельного углубленного изучения какого-либо вопроса дисциплины, выполненная по определенным требованиям, предъявляемым к ее структуре, содержанию и оформлению, демонстрирующая способность студента применить для решения учебной (учебно-исследовательской)

Курсовой проект (КП) – учебная работа студента, являющаяся результатом самостоятельного решения практической задачи, выполненная по определенным требованиям, предъявляемым к ее структуре, содержанию и оформлению, включающая расчетнографическую (чертежи, макеты, схемы, таблицы и т.п.) и (или) материальную часть (модели, стенды, фотографии, рисунки, изделия и т.п.), демонстрирующая способность студента применить для решения практической задачи знания, умения, навыки и компетенции, приобретенные в процессе изучения дисциплины.

2 Цель курсового проектирования

Цель курсовой работы по дисциплине «Языки программирования» как формы промежуточной аттестации является углубление и закрепление знаний, умений, навыков и компетенций, приобретенных обучающимися при изучении дисциплины, путем применения их к решению поставленной задачи по данной дисциплине, а также оценка уровня их сформированности.

Основными задачами при выполнении КР (КП) являются:

- о закрепление, углубление и систематизация полученных знаний и выработка умения самостоятельно применять их к решению конкретных задач;
- о приобретение и подтверждение наличия навыков исследовательской, расчетной и конструкторской работы;
- о закрепление навыков работы с компьютерной и офисной техникой,
- о использования современных информационных технологий;
- о формирование компетенций, установленных ФГОС ВО и ОП ВО,
- о закреплённых за дисциплиной, и подтверждение владения ими;
- воспитание чувства ответственности за принимаемое решение;

развитие навыков работы с учебной, научной и справочной литературой, нормативно-правовой документацией, периодической печатью, стандартами, типовыми проектами и т.п.;

- о овладение навыками грамотного, ясного и сжатого изложения результатов работы и аргументированной защиты принятых решений и сделанных выводов;
- о формирование научного мировоззрения;
- о приобретение навыков регулярной и ритмичной работы, развитие самостоятельности и инициативы, воспитание сознательного и творческого отношения к труду.

Курсовая работа предполагает выполнение задания повышенной сложности по проектированию, разработке и тестированию программного обеспечения, а также оформление сопутствующей документации.

Важным фактором подготовки специалистов в области информационных технологий является умение программировать, используя современные языки, включающие структурное программирование, объектные возможности, знакомство с основными методами и современными технологиями программирования, а также возможность выполнения работы в подгруппах. Целью такого подхода является развитие способностей к обоснованному принятию самостоятельных решений в ходе проектной работы и получение базового опыта разработки программных продуктов.

3 Тематика курсовой работы

В ходе выполнения курсовой работы студенты должны практически освоить общий методологический подход, используемый при создании программных продуктов на языках высокого уровня, таких как язык C++, C#, JAVA.

4 Задание на курсовое проектирование

Список вариантов задания на курсовое проектирование приведён в Приложении А.

5 Содержание пояснительной записки к курсовой работе

Объём пояснительной записки составляет 20-50 страниц машинописного текста, включая рисунки, таблицы и приложения. Её оформление должно соответствовать требованиям ГОСТ, ЕСКД, ЕСПД, СТУ.

Отчетность по курсовому проектированию состоит из пояснительной записки и созданного программного продукта в виде исполняемого файла.

В содержании перечисляются все структурные элементы КР (КП) в последовательности, в которой они расположены в работе: введение, заголовки всех разделов, подразделов, пунктов (если они имеют наименование) основной части, заключение, список использованных источников и наименования приложений, а также указываются номера страниц, на которых они расположены.

Рекомендуемая структура пояснительной записки:

Титульный лист

Задание на курсовую работу (проект)

Реферат

Содержание

Обозначения и сокращения (при необходимости)

Введение

1 Техническое задание

2 Технический проект

3 Рабочий проект

Заключение

Список используемых источников
Приложения
(при необходимости).

Реферат представляет собой краткое изложение КР (КП).

Реферат размещается на отдельном листе (странице). Рекомендуемый средний объем реферата – 850 печатных знаков. Объем реферата не должен превышать одной страницы.

В качестве заголовка записывается слово «Реферат» (для реферата на иностранном языке – соответствующий иностранный термин).

Реферат должен содержать: с ведения об объеме КР (КП) (количестве страниц), количестве иллюстраций, таблиц, приложений, использованных источниках, графическом материале; перечень ключевых слов;

текст реферата.

Перечень ключевых слов должен включать от 5 до 15 слов или словосочетаний, которые в наибольшей мере характеризуют содержание КР (КП) и обеспечивают возможность информационного поиска. Ключевые слова приводятся в именительном падеже и записываются строчными буквами основного шрифта КР (КП) в строку через запятую.

Реферат должен иметь следующую структуру:

объект исследования или разработки; цель работы;

- метод или методология проведения работы (исследования),
- аппаратура; полученные

результаты и их новизна;

- основные конструктивные, технологические и технико-
- эксплуатационные

характеристики; степень внедрения;

- рекомендации или итоги внедрения результатов работы;

область применения;

- общественно-социальная значимость, экономическая или иная
- эффективность работы; прогнозные предположения о развитии

объекта исследования

- (разработки);

дополнительные сведения (особенности выполнения и оформления работы и т.п.).

Если КР (КП) не содержит сведений по какой-либо из перечисленных структурных частей реферата, данная часть опускается, при этом последовательность изложения сохраняется.

Изложение материала в реферате должно быть кратким, точным и соответствовать требованиям ГОСТ 7.9. Сложных грамматических оборотов следует избегать.

Заголовки печатаются с абзацного отступа и выделяются жирным шрифтом.

Заголовок раздела (подраздела) должен быть отделен от основного текста раздела и от текста предыдущего раздела (подраздела) одинарным междустрочным интервалом 8 мм (1 пустая строка основного текста 14 pt).

Нумерация листов КР (КП) осуществляется в соответствии со следующими требованиями:

- все листы КР (КП) нумеруются арабскими цифрами с соблюдением сквозной нумерации по всему тексту;
- номер страницы ставится в нижней части листа без точки (первым листом является титульный лист, который включается в общее количество страниц, но не нумеруется);
- иллюстрации, таблицы и др., расположенные на отдельных листах, включают в общую нумерацию страниц КР (КП) или ВКР
 - (иллюстрации, таблицы и др., выполненные на листах формата А3, учитываются как одна страница);
- приложения должны иметь общую с остальной частью КР (КП) или ВКР сквозную нумерацию страниц.

Заключение содержит конкретные выводы, которые соотносятся с целью и задачами, поставленными во введении, а также включает предложения и рекомендации по использованию полученных результатов в производственной деятельности.

Список использованных источников содержит сведения об источниках, использованных при выполнении КР (КП).

В приложениях размещаются текст исходного кода программы, графический материал.

6 Рекомендации по выполнению курсовой работы

Основная часть документации состоит из разделов: технического задания, технического проекта, рабочего проекта.

Основным документом, в соответствии с которым выполняется разработка некоторого проекта в любой отрасли, включая проекты по разработке программного обеспечения (ПО), является техническое задание (ТЗ). При разработке ПО техническое задание – технический документ (спецификация), оговаривающий перечень требований к системе и утверждённый как заказчиком/пользователем, так и исполнителем/производителем системы. Спецификация может содержать системные требования, требования к тестированию и др. Техническое задание позволяет обеим сторонам (заказчику и исполнителю) согласовать все необходимые детали реализации ПО, спланировать сроки и этапы выполнения проекта. Кроме того, ТЗ позволяет заказчику требовать от

исполнителя соответствия продукта всем без исключения условиям, оговорённым в ТЗ, а исполнителю на законных основаниях отказаться от выполнения работ, не указанных в ТЗ.

При разработке ТЗ в рамках выполнения курсовой работы предлагается вариант подготовки ТЗ, соответствующий ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.

В соответствии с данным стандартом техническое задание должно содержать следующие разделы:

Введение

В разделе "Введение" указывают наименование, краткую характеристику области применения программы или программного изделия и объекта, в котором используют программу или программное изделие.

1.1 Основания для разработки

В разделе "Основание для разработки" должны быть указаны: документ (документы), на основании которых ведется разработка; организация, утвердившая этот документ, и дата его утверждения; наименование и (или) условное обозначение темы разработки.

1.2 Назначение разработки

В разделе "Назначение разработки" должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.

1.3 Требования к программе или программному изделию

Раздел "Требования к программе или программному изделию" должен содержать следующие подразделы: требования к функциональным характеристикам; требования к надежности; условия эксплуатации; требования к составу и параметрам технических средств; требования к информационной и программной совместимости; специальные требования. В подразделе "Требования к функциональным характеристикам" должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т.п. В подразделе "Требования к надежности" должны быть указаны требования к обеспечению надежного функционирования (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т.п.). В подразделе "Условия эксплуатации" должны быть указаны условия эксплуатации (температура окружающего воздуха, относительная влажность и т.п. для выбранных типов носителей данных),

при которых должны обеспечиваться заданные характеристики, а также вид обслуживания, необходимое количество и квалификация персонала. В подразделе "Требования к составу и параметрам технических средств" указывают необходимый состав технических средств с указанием их основных технических характеристик. В подразделе "Требования к информационной и программной совместимости" должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой. При необходимости должна обеспечиваться защита информации и программ.

1.4 Требования к программной документации

В разделе "Требования к программной документации" должны быть указаны предварительный состав программной документации и, при необходимости, специальные требования к ней.

1.5 Стадии и этапы разработки

В разделе "Стадии и этапы разработки" устанавливают необходимые стадии разработки, этапы и содержание работ (перечень программных документов, которые должны быть разработаны, согласованы и утверждены), а также, как правило, сроки разработки.

1.6 Порядок контроля и приемки

В разделе "Порядок контроля и приемки" должны быть указаны виды испытаний и общие требования к приемке работы. В техническое задание допускается включать приложения. В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

На стадии технического проекта производится дальнейшая функциональная декомпозиция программного изделия. При этом функции, определенные на стадии технического задания, разбиваются на модули. Такие модули проектируются на уровне внешнего описания, т.е. для каждого из них определяется функциональность с точки зрения пользователя.

Пользовательский интерфейс имеет важное значение для любой программной системы и является неотъемлемой ее составляющей, ориентированной, прежде всего, на конечного пользователя. Именно через интерфейс пользователь судит о прикладной программе в целом. Более того, часто решение об использовании прикладной программы пользователь принимает по тому, насколько ему удобен и понятен

пользовательский интерфейс. Вместе с тем, трудоемкость проектирования и разработки интерфейса может быть достаточно велика, и достигать более половины общего времени реализации проекта. Основным предназначением системы является предоставление пользователю необходимой функциональности. Поэтому разработку интерфейса следует реализовать в следующей последовательности:

- определение перечня основных функций системы, которые должны быть отражены в интерфейсе;
- определение перечня окон, их предназначение и общее содержимое;
- разработка диаграмм прецедентов;
- схематичное отображение детального содержимого каждого окна. При разработке диаграммы переходов необходимо следовать, с точки зрения компромисса, двум противоречивым требованиям:
- диаграмма должна быть достаточно полной, чтобы из любой функции (если допустимо предметной областью) можно было бы перейти к любой другой функции (полный граф);
- диаграмма должна быть достаточно простой, не перегруженной множеством возможных переходов и избыточной информацией, непосредственно не требующейся в реализации той или иной функции. Кроме того, при разработке интерфейса пользователя следует придерживаться следующих критериев качества:

- 1) Удобство и интуитивность (привычные названия, возможность самостоятельного изучения и использования функций системы, подсказки, легкость работы с системой).
- 2) Единообразии (предпочтителен стандарт, принятый в операционной системе, недопустимо использование одинаковых функционально, но различных внешне элементов).
- 3) Отсутствие перегруженности (небольшое число объектов на экране – не более 10).
- 4) Устойчивость (по возможности предотвращение некорректных действий пользователя).

Примерное количество страниц в техническом проекте 7 – 10.

Стадия рабочего проекта является заключительной как в конструировании программного изделия, так и в его изготовлении. В процессе конструирования окончательно определяется структура всего программного изделия. Определяется структура каждого фактического

программного модуля в виде схемы его алгоритма. Для всего изделия в целом и для каждого модуля рассчитываются и конструируются тесты.

Тестирование программного продукта входит в более общий процесс верификации, полноценное рассмотрение которого выходит за рамки дисциплины «Программирование на языках высокого уровня». Тем не менее, следует отметить ряд моментов, учёт которых является обязательным даже в том случае, когда речь не идёт о менеджерах проектов или специалистах в области верификации и тестирования.

Во-первых, следует различать процессы тестирования и отладки программного кода. Отладка выполняется программистом с помощью встроенных средств среды разработки и исходя из опыта написания программного кода. В основном отладка сводится к выявлению синтаксических и семантических ошибок в тексте программы. Тестирование – это процесс, требующий планирования и выполнения ряда предварительных процедур, основной из которых является составление набора тестовых примеров, образующих тест-план. Тестовые примеры в большинстве случаев основаны на функциональных требованиях к системе и могут затрагивать различные уровни разработки (модульное тестирование, интеграционное тестирование, системное тестирование).

Примерный график выполнения курсовой работы приведен в Приложении Г.

Тестирование программного продукта входит в более общий процесс верификации, полноценное рассмотрение которого выходит за рамки дисциплины «Программирование на языках высокого уровня». Тем не менее, следует отметить ряд моментов, учёт которых является обязательным даже в том случае, когда речь не идёт о менеджерах проектов или специалистах в области верификации и тестирования.

Во-первых, следует различать процессы тестирования и отладки программного кода. Отладка выполняется программистом с помощью встроенных средств среды разработки и исходя из опыта написания программного кода. В основном отладка сводится к выявлению синтаксических и семантических ошибок в тексте программы. Тестирование – это процесс, требующий планирования и выполнения ряда предварительных процедур, основной из которых является составление набора тестовых примеров, образующих тест-план. Тестовые примеры в

большинстве случаев основаны на функциональных требованиях к системе и могут затрагивать различные уровни разработки (модульное тестирование, интеграционное тестирование, системное тестирование).

Для каждого теста фиксируются:

1. Номер теста (по нумерации тестов в методике тестирования).
2. Входные данные.
3. Ожидаемый результат выполнения теста.
4. Результат выполнения теста.

Список используемых источников

1. Антамошкин, О.А. Программная инженерия. Теория и практика [Электронный ресурс]: учебник / О.А. Антамошкин ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск: Сибирский федеральный университет, 2012. - 247 с. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=363975>.

2. Влацкая, И.В. Проектирование и реализация прикладного программного обеспечения [Электронный ресурс]: учебное пособие / И.В. Влацкая, Н.А. Заельская, Н.С. Надточий; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Оренбургский государственный университет», Кафедра компьютерной безопасности и математического обеспечения информационных систем. - Оренбург : ОГУ, 2015. - 119 с. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=439107>.

3. Соловьев, Н. Системы автоматизации разработки программного обеспечения [Электронный ресурс]: учебное пособие / Н. Соловьев, Е. Чернопрудова ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Оренбургский государственный университет». - Оренбург: ОГУ, 2012. - 191 с. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=270302>.

4. Технология программирования [Электронный ресурс / Ю.Ю. Громов и др. ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2013. - 173 с. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=277802>.

Приложение А

Задания для курсовой работы по дисциплине «Языки программирования»

- 1. Программа поиска кратчайшего пути между двумя населенными пунктами в системе двусторонних дорог.** Описание: даны несколько населенных пунктов, соединенных между собой (произвольным образом) двусторонними дорогами. Для двух любых населенных пунктов найти кратчайший путь между ними. Отобразить решение графически, выделив цветом найденный путь.
- 2. Программа поиска замкнутого пути между двумя населенными пунктами в системе двусторонних дорог.** Описание: даны несколько населенных пунктов, соединенных между собой (произвольным образом) двусторонними дорогами некоторой длины. Найти замкнутый путь (с возвратом в тот же населенный пункт) длиной не более 100 км, проходящий через каждую дорогу ровно 1 раз. Отобразить решение графически, выделив цветом найденный путь.
- 3. Программа поиска кратчайшего пути между двумя населенными пунктами в системе односторонних дорог.** Описание: даны несколько населенных пунктов, соединенных между собой (произвольным образом) односторонними дорогами. Для двух любых населенных пунктов найти кратчайший путь между ними. Отобразить решение графически, выделив цветом найденный путь.
- 4. Программа поиска замкнутого пути между двумя населенными пунктами в системе односторонних дорог.** Описание: даны несколько населенных пунктов, соединенных между собой (произвольным образом) односторонними дорогами некоторой длины. Найти замкнутый путь (с возвратом в тот же населенный пункт) длиной не более 100 км, проходящий через каждую дорогу ровно 1 раз. Отобразить решение графически, выделив цветом найденный путь.
- 5. Программа определения достижимости населенного пункта в системе односторонних дорог.** Описание: даны несколько населенных пунктов, соединенных между собой (произвольным образом) односторонними дорогами некоторой длины. Определить, есть ли населенный пункт, из которого можно добраться до каждого из остальных пунктов, проезжая не более 100 км. Отобразить решение графически, выделив цветом найденный результат.
- 6. Программа определения достижимости населенного пункта в системе двусторонних дорог.** Описание: даны несколько населенных пунктов, соединенных между собой (произвольным образом) двусторонними дорогами некоторой длины. Определить, есть ли населенный пункт, из которого можно добраться до каждого из остальных пунктов, проезжая не более 100 км. Отобразить решение графически, выделив цветом найденный результат.
- 7. Программа определения N-периферии в системе двусторонних дорог.** Описание: даны несколько населенных пунктов, соединенных между собой (произвольным образом) двусторонними дорогами некоторой длины. N-периферией называется множество городов, расстояние от которых до выделенного города (столицы) больше N. Определить N-периферию для заданного N. Отобразить решение графически, выделив цветом найденный результат.
- 8. Программа определения достижимости города в системе односторонних дорог.** Описание: даны несколько городов, соединенных между собой (произвольным образом) односторонними дорогами некоторой длины. Определить, можно ли проехать из города А в город В таким образом, чтобы посетить город С и не проезжать никакой дороги более одного раза. Отобразить решение графически, выделив цветом найденный результат.
- 9. Программа поиска минимальной суммы расстояний между населенными пунктами в системе двусторонних дорог.** Описание: даны несколько населенных пунктов, соединенных между собой (произвольным образом) двусторонними дорогами некоторой длины, причем для любой пары населенных пунктов можно указать соединяющий их путь. Найти такой населенный пункт, для которого сумма расстояний до остальных городов минимальна. Отобразить решение графически, выделив цветом найденный результат.

- 10. Программа поиска пути с минимальными затратами между двумя городами в системе двусторонних дорог.** Описание: даны несколько населенных пунктов, соединенных между собой (произвольным образом) двусторонними дорогами некоторой длины. За проезд каждой дороги взимается некоторая пошлина. Найти путь из города А в город В с минимальной величиной $S+P$, где S - сумма длин дорог, а P - сумма пошлин проезжаемых дорог. Отобразить решение графически, выделив цветом найденный результат.
- 11. Программа определения достижимости города в изменяемой системе двусторонних дорог.** Описание: даны несколько городов, соединенных между собой (произвольным образом) двусторонними дорогами. Определить, можно ли, построив какие-нибудь 3 новые дороги, из заданного города А добраться до всех других городов. Отобразить решение графически, выделив цветом найденный результат.
- 12. Программа определения недостижимости города в изменяемой системе двусторонних дорог.** Описание: даны несколько городов, соединенных между собой (произвольным образом) двусторонними дорогами. Определить, можно ли, закрыв какие-нибудь 3 дороги, добиться того, чтобы нельзя было попасть из города А в город В. Отобразить решение графически, выделив цветом найденный результат.
- 13. Программа поиска пути с минимальными затратами между двумя городами в системе односторонних дорог.** Описание: даны несколько населенных пунктов, соединенных между собой (произвольным образом) односторонними дорогами некоторой длины. За проезд каждой дороги взимается некоторая пошлина. Найти путь из города А в город В с минимальной величиной $S+P$, где S - сумма длин дорог, а P - сумма пошлин проезжаемых дорог. Отобразить решение графически, выделив цветом найденный результат.
- 14. Программа поиска пути с минимальными затратами между двумя городами в системе двусторонних дорог.** Описание: даны несколько городов, соединенных между собой (произвольным образом) двусторонними дорогами некоторой длины. За проезд каждой дороги взимается некоторая пошлина. Найти путь из города А в город В такой, чтобы величина P была минимальна, где P - сумма пошлин проезжаемых дорог. Если таких путей несколько, то выбрать более короткий путь. Отобразить решение графически, выделив цветом найденный результат.
- 15. Программа поиска кратчайшего пути между двумя городами в двух системах односторонних дорог.** Описание: заданы две системы односторонних дорог (железные и шоссейные) с одним и тем же множеством городов. Для двух любых населенных пунктов найти кратчайший путь между ними. Путь может проходить как по железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный путь.
- 16. Программа поиска кратчайшего пути между двумя городами в двух системах двусторонних дорог.** Описание: заданы две системы двусторонних дорог (железные и шоссейные) с одним и тем же множеством городов. Для двух любых населенных пунктов найти кратчайший путь между ними. Путь может проходить как по железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный путь.
- 17. Программа поиска пути с минимальными затратами между двумя городами в двух системах односторонних дорог.** Описание: заданы две системы односторонних дорог (железные и шоссейные) с одним и тем же множеством городов. За проезд каждой дороги путешественник платит некоторую сумму денег. Найти путь из города А в город В с минимальной величиной $S+P$, где S - сумма длин дорог, а P - сумма денег за проезд дорог. Путь может проходить как по железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный результат.
- 18. Программа поиска пути с минимальными затратами между двумя городами в двух системах двусторонних дорог.** Описание: заданы две системы двусторонних дорог (железные и шоссейные) с одним и тем же множеством городов. За проезд каждой дороги путешественник платит некоторую сумму денег. Найти путь из города А в город В с минимальной величиной $S+P$, где S - сумма длин дорог, а P - сумма денег за проезд дорог. Путь может проходить как по

железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный результат.

- 19. Программа поиска пути с минимальными затратами между двумя городами в двух системах двусторонних дорог.** Описание: заданы две системы двусторонних дорог (железные и шоссейные) с одним и тем же множеством городов. За проезд каждой дороги путешественник платит некоторую сумму денег. Найти путь из города А в город В такой, чтобы величина Р была минимальна, где Р - сумма денег за проезд дорог. Если таких путей несколько, то выбрать более короткий путь. Путь может проходить как по железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный результат.
- 20. Программа поиска пути с минимальными затратами между двумя городами в двух системах односторонних дорог.** Описание: заданы две системы односторонних дорог (железные и шоссейные) с одним и тем же множеством городов. За проезд каждой дороги путешественник платит некоторую сумму денег. Найти путь из города А в город В такой, чтобы величина Р была минимальна, где Р - сумма денег за проезд дорог. Если таких путей несколько, то выбрать более короткий путь. Путь может проходить как по железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный результат.
- 21. Программа определения N-периферии в двух системах двусторонних дорог.** Описание: заданы две системы двусторонних дорог (железные и шоссейные) с одним и тем же множеством городов. N-периферией называется множество городов, расстояние от которых до выделенного города (столицы) больше N. Определить N-периферию для заданного N. Путь может проходить как по железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный результат.
- 22. Программа определения N-периферии в двух системах односторонних дорог.** Описание: заданы две системы односторонних дорог (железные и шоссейные) с одним и тем же множеством городов. N-периферией называется множество городов, расстояние от которых до выделенного города (столицы) больше N. Определить N-периферию для заданного N. Путь может проходить как по железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный результат.
- 23. Программа поиска минимальной суммы расстояний между населенными пунктами в двух системах односторонних дорог.** Описание: заданы две системы односторонних дорог (железные и шоссейные) с одним и тем же множеством городов, причем для любой пары населенных пунктов можно указать соединяющий их путь. Найти такой населенный пункт, для которого сумма длин дорог до остальных городов минимальна. Путь может проходить как по железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный результат.
- 24. Программа поиска минимальной суммы расстояний между населенными пунктами в двух системах двусторонних дорог.** Описание: заданы две системы двусторонних дорог (железные и шоссейные) с одним и тем же множеством городов, причем для любой пары населенных пунктов можно указать соединяющий их путь. Найти такой населенный пункт, для которого сумма длин дорог до остальных городов минимальна. Путь может проходить как по железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный результат.
- 25. Программа поиска замкнутого пути между двумя городами в двух системах односторонних дорог.** Описание: заданы две системы односторонних дорог (железные и шоссейные) с одним и тем же множеством городов. Найти замкнутый путь (с возвратом в тот же населенный пункт), проходящий через каждый город ровно 1 раз. Путь может проходить как по железным, так и по шоссейным дорогам. Отобразить решение графически, выделив цветом найденный путь.
- 26. Программа поиска замкнутого пути между двумя городами в двух системах двусторонних дорог.** Описание: заданы две системы двусторонних дорог (железные и шоссейные) с одним и тем же множеством городов. Найти замкнутый путь (с возвратом в тот же населенный пункт),

проходящий через каждый город ровно 1 раз. Путь может проходить как по железным, так и по шоссе́йным дорогам. Отобразить решение графически, выделив цветом найденный путь.

- 27. Программа определения достижимости населенного пункта в двух системах двусторонних дорог.** Описание: заданы две системы двусторонних дорог (железные и шоссе́йные) с одним и тем же множеством городов. Определить, есть ли населенный пункт, из которого можно добраться до каждого из остальных пунктов, проезжая не более 100 км. Путь может проходить как по железным, так и по шоссе́йным дорогам. Отобразить решение графически, выделив цветом найденный результат.
- 28. Программа определения достижимости населенного пункта в двух системах односторонних дорог.** Описание: заданы две системы односторонних дорог (железные и шоссе́йные) с одним и тем же множеством городов. Определить, есть ли населенный пункт, из которого можно добраться до каждого из остальных пунктов, проезжая не более 100 км. Путь может проходить как по железным, так и по шоссе́йным дорогам. Отобразить решение графически, выделив цветом найденный результат.
- 29. Программа поиска кратчайшего пути шахматной фигуры по клеточному полю.** Найти кратчайший путь передвижения шахматной фигуры (коня, ладьи, слона, ферзя, короля) по клеточному полю, соединяющий две указанные его клетки. Отобразить решение графически, выделив цветом найденный результат.
- 30. Программа определения допустимости хода шахматной фигуры.** Описание: задана корректная шахматная позиция. Определить, является ли допустимым ход шахматной фигуры. Если после хода король атакован, то ход считается не допустимым. Отобразить решение графически, выделив цветом найденный результат.
- 31. Программа поиска кратчайшего пути в лабиринте.** Описание: задан лабиринт, составленный из комнат. В каждой из комнат имеется не менее одной и не более трех дверей, соединяющих между собой соседние комнаты. Одна из дверей называется входом в лабиринт, другая – выходом из лабиринта. Найти кратчайший путь от входа в лабиринт к его выходу. Отобразить решение графически, выделив цветом найденный результат.
- 32. Программа поиска пути в лабиринте.** Описание: задан лабиринт, составленный из комнат. В каждой из комнат имеется не менее одной и не более трех дверей, соединяющих между собой соседние комнаты. Одна из дверей называется входом в лабиринт, другая – выходом из лабиринта. Некоторые комнаты объявлены опасными. Найти путь от входа в лабиринт к его выходу, не проходящий через опасные комнаты. Отобразить решение графически, выделив цветом найденный результат.
- 33. Программа поиска пути в лабиринте.** Описание: задан лабиринт, составленный из комнат. В каждой из комнат имеется не менее одной и не более четырех дверей, соединяющих между собой соседние комнаты. Одна из дверей называется входом в лабиринт, другая – выходом из лабиринта. Одна из комнат объявлена комнатой, приносящей удачу. Определить, есть ли путь от входа в лабиринт к его выходу, не проходящий одну и ту же комнату два раза и проходящий через комнату, приносящую удачу. Отобразить решение графически, выделив цветом найденный результат.
- 34. Программа поиска минимального по стоимости пути в лабиринте.** Описание: задан лабиринт, составленный из комнат. В каждой из комнат имеется не менее одной и не более четырех дверей, соединяющих между собой соседние комнаты. Одна из дверей называется входом в лабиринт, другая – выходом из лабиринта. За посещение каждой комнаты взимается пошлина в размере от одного до ста рублей. Найти путь от входа в лабиринт к его выходу, чтобы величина P была минимальна, где P - сумма пошлин за посещение комнат лабиринта. Если путей с одинаковыми значениями P несколько, то выбрать более короткий путь. Отобразить решение графически, выделив цветом найденный результат.
- 35. Программа поиска пути в лабиринте.** Описание: задан лабиринт, составленный из комнат. В каждой из комнат имеется не менее одной и не более четырех дверей, соединяющих между собой

соседние комнаты. Одна из дверей называется входом в лабиринт, другая – выходом из лабиринта. В каждой комнате хранятся сокровища: от одной до ста золотых монет. Найти путь от входа в лабиринт к его выходу, не проходящий одну и ту же комнату два раза и позволяющий унести как можно больше сокровищ. Отобразить решение графически, выделив цветом найденный результат.

Приложение Б

Минобрнауки России Юго-Западный государственный университет

Кафедра программной инженерии

КУРСОВАЯ РАБОТА (ПРОЕКТ)

по дисциплине «Языки программирования»
(наименование дисциплины)

на тему « _____ »

Направление подготовки (специальность) _____
(код, наименование)

Автор работы (проекта) _____
(инициалы, фамилия) (подпись, дата)

Группа _____

Руководитель работы (проекта) _____
(инициалы, фамилия) (подпись, дата)

Работа (проект) защищена _____
(дата)

Оценка _____

Члены комиссии _____
(подпись, дата) (инициалы, фамилия)

_____ (подпись, дата) (инициалы, фамилия)

Курск 20 __ г.

Приложение В**Минобрнауки России
Юго-Западный государственный университет
Кафедра программной инженерии****ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ (ПРОЕКТ)**Студент _____ шифр _____ группа _____
(фамилия, инициалы)

Тема « _____ »

Срок представления работы (проекта) к защите « _____ » 20 __ г.

Исходные данные (для проектирования, для научного исследования):

Содержание пояснительной записки курсовой работы (проекта):

Введение

1 Техническое задание

2 Технический проект

3 Рабочий проект

Заключение

Список использованных источников

Перечень графического материала:

_____Руководитель работы (проекта) _____
(инициалы, фамилия) (подпись, дата)

Задание принял к исполнению _____

(инициалы, фамилия)

(подпись, дата)

Приложение Г

Примерный график выполнения курсовой работы по неделям (с указанием контрольных мероприятий)

№ недели	Наименование и содержание этапа	Примечание
1	Изучение задания на курсовую работу и его согласование.	
2	Изучение метода, подлежащего реализации.	
3,4	Составления технического задания.	
5	Утверждение технического задания.	Контрольная точка
6	Разработка технического проекта. Функциональное проектирование.	
7	Разработка технического проекта. Состав и взаимосвязь функциональных модулей.	
8	Утверждение и оформление технического проекта.	Контрольная точка.
9	Разработка рабочего проекта. Определение состава программных модулей и их взаимосвязей.	
10	Разработка рабочего проекта. Составление схем алгоритмов верхнего уровня.	
11	Разработка тестов для алгоритмов верхнего уровня.	Контрольная точка
12	Программирование и отладка программ верхнего уровня.	
13	Составление схем алгоритмов нижнего уровня.	
14	Разработка тестов для алгоритмов нижнего уровня.	
15	Программирование и отладка алгоритмов нижнего уровня.	Контрольная точка
16	Тестирование и отладка всех модулей.	
17	Оформление пояснительной записки.	

18	Защита курсовой работы	
----	------------------------	--