

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

УТВЕРЖДАЮ
Заведующий кафедрой



Ломакин В.В.

____.____.2020

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по выполнению курсовых работ по дисциплине
«Программирование»

Направление подготовки

38.03.05 «Бизнес-информатика»

Профиль подготовки

Архитектура предприятия; цифровые технологии в бизнесе

Автор (ы):

старший преподаватель Резниченко О.С.

должность, ученая степень, ученое звание, инициалы и фамилия

Одобрено и рекомендовано к утверждению на заседании кафедры
прикладной информатики и информационных технологий

Протокол № 1 от 02.09.2020

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	2
ВВЕДЕНИЕ	3
1 Цели и задачи курсовой работы.....	3
2 Организация проведения курсовой работы	4
2.1 Тематика курсовых работ.....	4
2.2 Порядок выполнения курсовой работы	6
2.3 Сроки выполнения отдельных этапов.....	7
2.4 Порядок защиты курсовой работы.....	7
3 Формулировка задания	8
4 Основное содержание курсовой работы	9
5 Оформление курсовой работы	12
5.1 Оформление электронной версии разработанного приложения.....	12
5.2 Оформление пояснительной записки.....	12
6 Общие требования к курсовой работе.....	12
7 Правила оформления записки к курсовой работе	18
7.1 Общие положения	19
7.2 Содержание отчета о выполнении работы	19
7.3 Содержание.....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	22
ПРИЛОЖЕНИЕ А	24
ПРИЛОЖЕНИЕ Б	26

ВВЕДЕНИЕ

Курсовая работа является обязательным этапом при изучении дисциплины «Программирование», позволяющим систематизировать, расширить и закрепить теоретические знания и практические навыки студентом, а также определить уровень его подготовленности к выполнению функциональных обязанностей в соответствии с полученной квалификацией.

1 Цели и задачи курсовой работы

Основными целями курсовой работы являются:

- систематизация и закрепление полученных теоретических знаний, полученных студентами при изучении дисциплины «Программирование»;
- углубление практических навыков;
- формирование умений применять теоретические знания при решении поставленных практических вопросов;
- раскрытие содержательной характеристики выбранной темы;
- приобретение и закрепление навыков самостоятельной работы;
- проверка умения формулировать основные выводы по результатам анализа конкретной темы.

Курсовая работы представляет собой результат выполнения следующих взаимосвязанных этапов:

- 1) Выбор темы;
- 2) Разработка рабочего плана, проектирование структуры программного изделия;
- 3) Сбор, анализ и обобщение материалов исследования, формулирование основных теоретических положений;
- 4) Разработка алгоритмов и составление программ на выбранном языке (тестирование, отладка).

5) Документирование, созданного программного продукта и оформление курсовой работы;

6) Защита.

Работа выполняется самостоятельно в произвольное время и сдается в строго оговоренные сроки.

2 Организация проведения курсовой работы

2.1 Тематика курсовых работ

При выполнении работы студенты должны приобрести практические навыки проектирования программных продуктов среднего уровня сложности. Язык и среду программирования студент выбирает самостоятельно, исходя из выбранной технологии программирования, решаемой задачи, а также личных предпочтений.

Список примерных тем курсовых работ:

- 1) Программная реализация преобразователя диаграмм Вирта в формы Бэкуса-Науэра
- 2) Реализация игры «Бильярд»
- 3) Реализация метода получения исходного скобочного выражения и графа его разбора из прямой польской записи
- 4) Программная реализация построителя блок-схем по программе на языке Паскаль
- 5) Моделирование работы машины Тьюринга
- 6) Моделирование работы машины Поста
- 7) Программная реализация преобразователя форм Бэкуса-Науэра в диаграммы Вирта
- 8) Реализация игры в шашки с проверкой ходов и взятий
- 9) Реализация программы определения соответствия цепочки

символов диаграмме Вирта

- 10) Программная реализация игры «Домино»
- 11) Программная реализация преобразователя выражений в форму обратной польской записи, прямой польской записи
- 12) Программная реализация решения задачи о восьми ферзях
- 13) Разработка программы для форматирования текста программы на Паскале
- 14) Программная реализация электронных стрелочных часов-секундомера
- 15) Реализация игры «Сапёр»
- 16) Программная реализация электронной записной книжки
- 17) Программная реализация метода поиска минимального расстояния между пунктами А и В в системе двухсторонних дорог
- 18) Программная реализация игры «Пятнашки»
- 19) Реализация игры «Линии»
- 20) Программная реализация метода поиска минимального расстояния между двумя комнатами в системе прямоугольных комнат, имеющих входы-выходы
- 21) Реализация игры в шахматы с проверкой ходов и взятий
- 22) Программная реализация игры «Теннис»
- 23) Программная реализация метода поиска самого короткого по длине выхода в системе лабиринтов
- 24) Программная реализация преобразователя выражений из формы обратной польской записи в скобочную запись и форму прямой польской записи
- 25) Реализация программы для решения систем линейных уравнений методами Гаусса и Зейделя
- 26) Программная реализация головоломки «Ханойская башня»
- 27) Программная реализация настольного калькулятора для научных расчётов

28) Программная реализация метода определения минимального расстояния между двумя пунктами в системе односторонних дорог

29) Программная реализация настольного калькулятора для бухгалтерских расчетов

30) Программная реализация аппроксимации экспериментальных зависимостей методом МНК

Студенту в течении первых трех учебных недель необходимо определиться с темой курсовой работы, написать заявление на утверждение темы, подписать это заявление у научного руководителя и заведующего кафедрой (приложение Б).

2.2 Порядок выполнения курсовой работы

В процессе выполнения курсовой работы студенты должны:

1) разработать развернутое техническое задание на программный продукт;

2) выполнить анализ задания, выбрать технологию проектирования и разработать проект программного продукта;

3) выбрать структуры данных для реализации предметной области программного продукта;

4) разработать интерфейс пользователя;

5) выбрать стратегию тестирования и разработать тесты;

6) выбрать язык и среду программирования, наиболее удовлетворяющий проведенным разработкам;

7) разработать алгоритмы и реализовать их в выбранной среде разработки;

8) выполнить тестирование и отладку;

9) разработать необходимую документацию, указанную в техническом задании.

2.3 Сроки выполнения отдельных этапов

Выполнение курсовой работы осуществляется в несколько этапов. В конце каждого этапа студент демонстрирует преподавателю результаты выполнения работы.

Основные этапы курсовой работы и сроки их выполнения приведены в таблице 2.1.

2.4 Порядок защиты курсовой работы

На защиту студент предоставляет:

- 1) развернутое техническое задание;
- 2) программный продукт;
- 3) расчетно-пояснительную записку;
- 4) программную документацию, указанную в разделе «Требования к программной документации» технического задания.

программной документации» технического задания.

В процессе демонстрации программного продукта проверяется:

- соответствие программы техническому заданию;
- работоспособность в различных режимах.

На защите студент коротко (3-5 мин.) докладывает об основных проектных решениях, принятых в процессе разработки, и отвечает на вопросы членов комиссии.

Оценка за курсовая работы выставляется с учетом:

- качества выполненного программного продукта;
- правильности оформления записки;
- результатов защиты.

Таблица 2.1 - Этапы выполнения курсовой работы

Этап	Содержание этапа	Сроки	Представляемые результаты
0	Написание заявления на тему КП	1-3 неделя	Заявление
1	Разработка технического задания	4 неделя	Готовое техническое задание.
2	Выбор структур данных, разработка интерфейса пользователя, разработка иерархии объектов, подготовка данных для	8 неделя	Описание структур данных, структурная и функциональная схемы продукта, граф диалога интерфейса и описание основных экранных форм, стратегия тестирования и тестовые данные.
3	Разработка алгоритмов и реализация программы на выбранном языке программирования.	12 неделя	Схемы алгоритмов, тексты программ системы, сам продукт на дискете для проверки функционирования.
4	Подготовка расчетно-пояснительной записки	15 неделя	Полностью оформленная записка в рукописном или печатном варианте.
5	Защита курсовой работы	16 неделя	Программа, пояснительная записка, приложения.

3 Формулировка задания

Тематику курсовой работы предлагает преподаватель. Студент в подборе тематики курсовой работы может проявить инициативу и высказать свои пожелания преподавателю, ответственному за дисциплину.

Тема работы должна отвечать современным требованиям развития науки, техники, производства, экономики, культуры и образования.

В курсовой работе должна разрабатываться программа, программный комплекс или программная система. Наименование курсовой работы должно быть лаконичным и точно отражать суть работы.

4 Основное содержание курсовой работы

Примерная структура курсовой работы:

Титульный лист.

Содержание.

Введение.

1. Разработка эскизного и технического проекта программы

1.1. Назначение и область применения

1.2. Технические характеристики

1.3. Разработка алгоритма решения задачи

2. Разработка рабочего проекта

2.1 Разработка программы

2.2 Спецификация программы

2.3 Текст программы

2.4 Описание программы

3 Описание интерфейса пользователя и тестирование программы

3.1 Описание интерфейса пользователя

3.2 Анализ и тестирование программы

3.3 Разработка руководства пользователя

Заключение

Список использованных источников

Приложения

Содержание включает введение, наименования всех разделов, подразделов и пунктов, заключение, список литературы и наименования приложений с указанием номеров страниц, с которых начинаются эти элементы пояснительной записки.

Во введении обосновывается актуальность темы курсовой работы, формулируются основные задачи курсовой работы. В конце введения

можно раскрыть структуру работы, то есть дать перечень ее структурных элементов и обосновать последовательность их расположения.

Технологический раздел является основой курсовой работы, ему должны быть подчинены остальные разделы. Работа над ним требует от студента практического применения знаний технологии программирования.

Примерное содержание технологического раздела может быть следующим:

- Постановка задачи
- Требования к программе
- Требования к функциональным характеристикам
- Требования к надежности
- Требования к составу технических средств
- Требования к программной совместимости
- Описание применяемых математических методов
- Проектирование
- Выбор архитектуры программного обеспечения
- Выбор языка и среды программирования
- Выбор способа организации входных и выходных данных
- Представление в оперативной памяти
- Представление во внешней памяти
- Разработка структурной схемы программного продукта
- Алгоритм представления задачи
- Словесное описание алгоритма
- Блок-схема
- Проектирование пользовательского интерфейса
- Выбор типа пользовательского интерфейса
- Построение графа диалога
- Разработка форм ввода-вывода информации

- Тестирование
- Выбор стратегии тестирования
- Описание контрольного примера
- Результаты тестирования
- Руководство пользователя

В заключении приводятся выводы по отдельным разделам работы и результаты всей работы в целом. Иногда в заключении указывают пути продолжения исследуемой темы, методы ее дальнейшего изучения, а также конкретные задачи, которые придется решать в первую очередь. Раздел должен содержать описание функционального и эксплуатационного назначения программного продукта с указанием категорий пользователей.

Список источников должен содержать сведения обо всех источниках, на которые имеются ссылки в работе (книги, статьи, стандарты, публикации в Интернете и т.д.). Недопустимо включать в список те работы, которые фактически не были использованы.

В приложения в обязательном порядке помещают тексты программ и листы графической части курсовой работы, выполненные на листах формата А4. Также в приложения при необходимости могут помещаться вспомогательные и дополнительные материалы (таблицы вспомогательных цифровых данных, объемные расчеты, результаты тестирования программ, выходные документы: протоколы, отчеты и т.п.).

Содержание графической части определяется студентом совместно с руководителем курсовой работы. В курсовых работах, посвященных разработке программного обеспечения, в графическую часть могут выноситься:

- структурная схема системы;
- функциональная схема системы;
- структуры входных и выходных данных;
- схема пользовательского интерфейса;

- основные алгоритмы работы системы (блок-схемы);

5 Оформление курсовой работы

5.1 Оформление электронной версии разработанного приложения

Разрабатываемая программа должна включать справочную информацию об основных терминах, определяемых тематикой работы, и данные о работе программы.

Разрабатываемые программные модули должны быть самодокументированы, т.е. тексты программ должны содержать все необходимые комментарии. Программный код должен быть выполнен в соответствии с правилами хорошего стиля и максимально оптимизирован.

5.2 Оформление пояснительной записки

При оформлении текстовых и графических материалов, входящих в программную документацию, следует придерживаться действующих стандартов.

6 Общие требования к курсовой работе

Расчетно-пояснительная записка должна содержать обоснование основных проектных решений, принятых студентом на каждом этапе разработки. Решения должны приниматься исходя из особенностей проектируемого продукта и специфики области его применения. Не должно быть обоснований типа «удобнее», «целесообразнее» и т. п. Необходимо пояснить, чем удобнее, почему целесообразно. По возможности необходимо четко формулировать основания для принятия того или иного решения.

– **Описание задачи.** Оно дается на основе задания на курсовое проектирование с возможными уточнениями методов решения и формы представления результата. Данный пункт является уточнением темы с определением того, что фактически должно быть получено в результате выполнения курсовой работы, т.е. кратко определяются предполагаемые задачи проектирования, решение которых будет означать выполнение курсовой работы.

– **Выбор метода решения.** В этом разделе делается обзор существующих методов решения задач аналогичного класса, производится обоснование выбора конкретного метода и его описание (возможна *краткая историческая справка*)

– **Описание алгоритма решения.** Здесь приводится формализованное описание реализации задачи выбранным методом решения с приведением блок-схем алгоритма в целом, а также его наиболее существенных функциональных частей.

– **Описание структуры данных.** Здесь приводится описание способов хранения исходных данных проекта, промежуточных данных, использующихся при работе программы, а также структуры выходных данных.

– **Описание модульной структуры программы.** Здесь приводится описание полученной логической структуры программы. Осуществляется описание основных процедур и функций, их параметров, а также важных для понимания особенностей работы программы констант и переменных.

– **Примеры работы программы.** В этом разделе приводятся тестовые примеры начальных данных и результат их обработки программой, наиболее полно характеризующие возможное разнообразие вариантов обработки входных данных. Примеры, иллюстрирующие работу программы, можно представлять в виде копий экранов.

Структурная схема - схема, отражающая состав и взаимодействие частей разрабатываемого продукта. При объектной декомпозиции такими

частями являются объекты (рисунок 6.1), при структурной декомпозиции – подпрограммы (рисунок 6.2).

Для тем, связанных с нечисловой обработкой данных, этот раздел должен содержать информационную модель системы, которая может быть представлена функциональной схемой.

Функциональная схема - схема взаимодействия частей системы с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств.

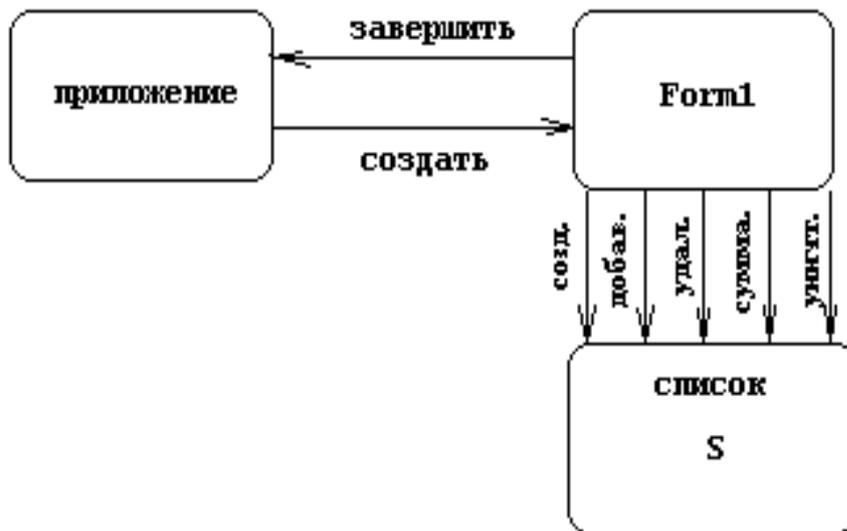


Рисунок 6.1 - Структурная схема программного продукта



Рисунок 6.2 - Структурная схема программного продукта

Большое значение при разработке структуры программного продукта имеет выбор алгоритмов предметной области и структур данных.

Для заданий, при выполнении которых используются сложные

математические методы, и заданий, при реализации которых студентом предлагался собственные оригинальные алгоритмы, обязательным является присутствие в записке обоснование выбора метода (по вычислительной сложности или другим соображениям). В этом же разделе приводятся и сами алгоритмы (в виде схем или псевдокодов), и пояснения к ним.

Выбор структур данных осуществляется на основе анализа основных процессов обработки информации (статические или динамические, массивы или другие структуры). При необходимости создаются новые структуры данных или модифицируются уже известные.

Обычно при выборе структур учитываются следующие параметры: объем и типы данных, а также основные операции над данными (хранение, поиск, сортировка) и частота обращения к ним в процессе выполнения программы. Если возможны варианты, то производится их оценка по объему требуемой памяти и вычислительной сложности выполнения основных операций.

1) Разработка пользовательского интерфейса. Этот раздел содержит обзор различных способов и форм взаимодействия пользователя с системой и обоснование выбора определенной формы диалога для общения с разрабатываемым программным продуктом. Проводится разработка структуры диалога и приводится граф диалога интерфейса, отражающий эту структуру. В случае табличной формы диалога дается описание всех оконных форм и меню (рисунки 6.3 и 6.4).



Рисунок 6.3 - Иерархии меню



Рисунок 6.4 - Экранная форма

В современных условиях наибольшее распространение получил графический интерфейс. Интерфейс командной строки используется все реже и, как правило, в приложениях системного характера. Под графическим пользовательским интерфейсом (ГПИ) понимается определенная среда, которая на основе многооконного графического представления данных организует интерфейс ПС и пользователя.

Существуют общие принципы, на которых базируется ГПИ:

- графический режим работы;
- многооконность;
- использование манипулятора типа мышь;
- представление ряда объектов иконками;
- идентичность изображаемого объекта объекту, представленному на экране, и наглядность изображения;
- стандартное представление всех основных действий и элементов графической среды;
- множество типовых элементов (строк меню, кнопок, активных областей и т.п.), используемых при разработке аналогичных программных средств и облегчающих процесс их кодирования.

2) Описание реализации программного продукта

Для программы, при разработке которой использовалась объектно-ориентированная технология, обязательно должна быть разработана диаграмма классов. Для каждого класса нужно указать необходимые

атрибуты и операции, соответственно обосновывая их назначение и функции.

В этом же разделе при необходимости можно привести алгоритмы некоторых методов.

Каждый алгоритм должен быть представлен:

- таблицей и (или) списком используемых в нем глобальных переменных;
- схемой алгоритма, использующей имена переменных, приведенных в таблице или списке;
- описанием процесса обработки данных в соответствии с приведенной схемой алгоритма.

Описание каждого алгоритма должно включать:

- функциональное назначение алгоритма;
- входные и выходные данные (результаты выполнения);
- список формальных параметров и их назначение;
- пример вызова модуля или подпрограммы;
- используемые технические средства;
- ссылку на таблицу переменных алгоритма;
- ссылку на рисунок со схемой алгоритма;
- описание процесса обработки данных в соответствии со схемой;
- если имеется приложение с полным текстом программы, то ссылку на соответствующую страницу приложения.

При описании процесса обработки данных в соответствии со схемой алгоритма необходимо пояснить все циклы, каждую альтернативу ветвления, принятое решение по результатам анализа альтернатив и последующие действия.

Тексты описания алгоритмов должны быть структурными, предложения короткими. Описание алгоритма должно отражать суть процесса обработки.

3) Выбор стратегии тестирования и отладка программного средства.

Данный раздел должен содержать обоснование выбора той или иной стратегии тестирования программного средства, тестовые наборы данных (тесты) по всем частям программного продукта как с использованием правильных входных данных, так и входных данных, не соответствующих принятым ограничениям, а также иллюстрироваться экранными распечатками и комментариями процесса отладки.

Отладка включает в себя поиск ошибки в тексте программного модуля (локализация ошибки) и исправление обнаруженной ошибки. Описывается проведенный анализ ошибок, выявленных в ходе написания, трансляции, тестирования и отладки программного средства. Приводятся распечатки экранных форм, отражающие полученные результаты решения поставленной задачи. Делается вывод о соответствии числовых значений результатов, их точности, форм выдачи и т.д. требованиям поставленной задачи. Можно привести данные статистической отчетности - количество допущенных ошибок (по видам), трудозатраты на разных этапах разработки и отладки модулей программного средства, расход вычислительных ресурсов на отдельных этапах выполнения задания. Описываются обнаруженные некорректные или нерациональные приемы программирования и программные конструкции, ошибки в программе, ошибки в алгоритме и постановке задачи.

4) Заключение. Раздел содержит выводы по разработанному продукту, рекомендации по его использованию и возможные направления дальнейшего усовершенствования.

Примечание. В зависимости от выбранной технологии и особенностей разрабатываемого продукта могут иметь место отклонения от рекомендуемой структуры записки. Разделы записки могут объединяться, опускаться, меняться местами. Возможно добавление новых разделов, если этого требует логика изложения.

7 Правила оформления записки к курсовой работе

7.1 Общие положения

Требования к оформлению пояснительной записки к курсовой работе в точности соответствуют «Правилам оформления выпускных квалификационных работ на кафедре прикладной информатики и информационных технологий». Оформлять пояснительную записку следует строго в соответствии с этими правилами [1].

Рекомендуемый объем записки к курсовой работе составляет, как правило, 22-40 страниц стандартного текста.

7.2 Содержание отчета о выполнении работы

Состав:

- титульный лист (*без указания номера страницы*) (шаблон и пример в приложении А);
- оглавление (*если текст работы делится на главы*) или содержание (*в том случае, если текст работы делится на разделы*);
- введение (*с указанием номера страницы*);
- текст глав или разделов (*основная часть курсовой работы*);
- заключение;
- библиографический список;
- приложения (*при необходимости, обязательно с нумерацией страниц*).

Содержание основной части курсовой работы зависит в первую очередь от её тематики и учитывает требования конкретного преподавателя, однако оно допускает творческую составляющую, вписывающуюся в тематику работы и отвечающую требованию логической завершенности каждого из разделов и подразделов.

7.3 Содержание

Пример - Содержание курсовой работы по дисциплине «**Программирование**» на тему «Программная реализация алгоритма архивации Хаффмана»:

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

	3
1	Теоретические основы сжатия даны	
	5
1.1	Обзор алгоритмов сжатия.....	8
1.2	Алгоритм Хаффмана	
	8
2	Описание алгоритма решения	
	10
3	Описание программы	
	13
3.1	Описание формата архива	
	13
3.2	Используемые константы, типы и переменные	
	14
3.3	Процедуры и функции общего назначения	
	15
3.4	Состав и описание блока архивации	
	17
3.5	Состав и принципы работы блока распаковки	
	18
4	Реализация и тестирование программы	
	19

4.1	Описание элементов формы.	Обработка событий	19
4.2	Ошибки периода исполнения	и их обработка	21
4.3	Недостатки и системные требования	программы	22
ЗАКЛЮЧЕНИЕ			23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ			24
ПРИЛОЖЕНИЯ			25

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Правила оформления письменных работ обучающихся на кафедре прикладной информатики и информационных технологий [Электронный ресурс], 2020. – 36 с. - URL: <http://iten.bsu.edu.ru/upload/iblock/303/Правила%20оформления%20работ%20ПИиИТ.pdf> (дата обращения 06.10.2020).

2 Приказ об утверждении Регламента использования системы "Антиплагиат. ВУЗ" по программам подготовки научно-педагогических кадров в аспирантуре НИУ "БелГУ (утв. 05.06.2020) [Электронный ресурс]. URL: <https://www.bsu.edu.ru/upload/iblock/279/522-ОД%20от%2005.06.2020%20Регламент%20исп.%20сист.%20Антиплагиат.Ву3.pdf> (дата обращения 06.10.2020)

3 Положение о порядке выполнения и защите курсовых работ (проектов) (утв. 29.04.2019). [Электронный ресурс]. URL: <https://www.bsu.edu.ru/upload/iblock/e2c/%D0%9F%D0%BE%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BE%20%D0%B7%D0%B0%D1%89%D0%B8%D1%82%D0%B5%20%D0%9A%D1%83%D1%80%D1%81%D0%BE%D0%B2%D1%8B%D1%85%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D1%85.doc> (дата обращения 06.10.2020)

4 Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 161 с. — (Высшее образование). — ISBN 978-5-534-10971-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454100> (дата обращения: 15.10.2020).

5 Нагаева, И. А. Программирование: Delphi : учебное пособие для среднего профессионального образования / И. А. Нагаева, И. А. Кузнецов ; под редакцией И. А. Нагаевой. — Москва : Издательство Юрайт, 2020. — 302 с. — (Профессиональное образование). — ISBN 978-5-534-09124-3. — Текст :

электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/455609> (дата обращения: 15.10.2020).

ПРИЛОЖЕНИЕ А

Бланк титульного листа

Зарегистрировано № _____
«___» _____ 2020 г.

подпись (расшифровка подписи)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ (ФАКУЛЬТЕТ) _____
Кафедра _____

[Т Е М А]
Курсовая работа

по дисциплине « _____ »
наименование дисциплины в соответствии с учебным планом
студента очной (очно-заочной, заочной) формы обучения
направления подготовки (специальности) _____
(код, наименование направления подготовки)
_____ курса группы _____
Фамилия Имя Отчество

Допущена к защите
«___» _____ 2020 г.

Подпись (расшифровка подписи)

Оценка
«___» _____ 2020 г.

Подпись (расшифровка подписи)

Научный руководитель:

(степень, должность)

(Ф.И.О. руководителя)

БЕЛГОРОД 2020

Пример титульного листа

Зарегистрировано № _____
«___» _____ 2020 г.

подпись (расшифровка подписи)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
Кафедра прикладной информатики и информационных технологий

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МАШИНЫ ПОСТА

Курсовая работы
по дисциплине «Программирование»

студента очной формы обучения
направления подготовки 38.03.05 «Бизнес-информатика»

1 курса группы 07001131
Клименко Надежды Анатольевны

Допущена к защите
«___» _____ 2020 г.

Подпись (расшифровка подписи)

Научный руководитель:
к.т.н., доцент

Асадуллаев Р.Г.

Оценка _____
«___» _____ 2020 г.

Подпись (расшифровка подписи)

БЕЛГОРОД 2020

ПРИЛОЖЕНИЕ Б

Бланк заявление на тему курсовой работы

Заведующему кафедрой
прикладной информатики и информационных
технологий

наименование факультета / института

НИУ «БелГУ»

В.В. Ломакину

инициалы и фамилия декана (директора)

студента(ки) _____ формы обучения
группы _____

инициалы и фамилия студента(ки)

ЗАЯВЛЕНИЕ

Прошу разрешить выполнение курсовой работы по дисциплине
«*Программирование*»

на тему: «»

Научный руководитель: *Резниченко Олег Сергеевич*

подпись

_____._____._____

Зав. кафедрой

подпись

В.В. Ломакин

расшифровка подписи

_____._____._____

Научный руководитель

старший преподаватель

должность

подпись

Резниченко О.С.

расшифровка подписи

_____._____._____