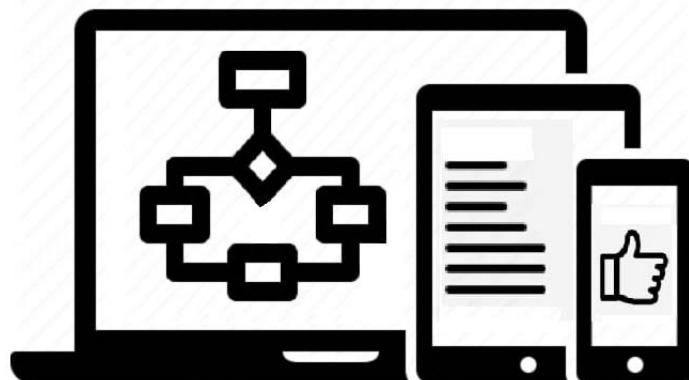


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ И ПРОГРАММИРОВАНИЕ

*Методические рекомендации к курсовому проектированию
для студентов специальности
1-40 05 01 «Информационные системы и технологии
(по направлениям)» дневной и заочной форм обучения*



Могилев 2023

УДК 621.01
ББК 36.4
О87

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «28» января 2023 г., протокол № 6

Составитель ст. преподаватель О. В. Сергиенко

Рецензент канд. техн. наук, доц. Д. М. Свирепа

Методические рекомендации к выполнению курсового проекта по дисциплине «Объектно-ориентированное проектирование и программирование» предназначены для студентов направления подготовки 1-40 05 01 «Информационные системы и технологии (по направлениям)» дневной и заочной форм обучения.

Учебное издание

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ И ПРОГРАММИРОВАНИЕ

Ответственный за выпуск	В. В. Кутузов
Корректор	А. А. Подошевка
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, Могилев.

© Белорусско-Российский
университет, 2023

Содержание

Введение.....	4
1 Общие требования к курсовому проекту.....	5
1.1 Тематика курсовых проектов.....	5
1.2 Общие требования к курсовому проекту.....	5
1.3 Функциональные требования к курсовому проекту.....	5
1.4 Требования к программной реализации курсового проекта.....	6
2 Содержание курсового проекта	6
2.1 Содержание пояснительной записки.....	6
2.2 Содержание графической части проекта.....	7
3 Требования к содержанию разделов пояснительной записки.....	7
3.1 Введение.....	7
3.2 Описание предметной области.....	7
3.3 Проектирование программного модуля.....	9
3.4 Реализация программного модуля.....	11
3.5 Тестирование программного модуля.....	12
3.6 Заключение.....	12
3.7 Список использованных источников.....	12
3.8 Приложения.....	12
4 Оформление пояснительной записки.....	13
4.1 Общие требования.....	13
4.2 Нумерация страниц.....	14
4.3 Построение текста.....	14
4.4 Рисунки.....	14
4.5 Таблицы.....	15
4.6 Ссылки.....	15
4.7 Приложения.....	15
4.8 Список использованных источников.....	16
Список литературы.....	17
Приложение А. Примерный перечень заданий на курсовое проектирование.....	18
Приложение Б. Образец оформления титульного листа.....	19

Введение

Целью курсового проекта является закрепление основ и углубление знаний приемов программирования на языке C++, получение практических навыков на всех этапах создания программного продукта: от постановки задачи до практической реализации, сопровождающейся документацией и инструкциями по его использованию. При выполнении курсового проекта студентам рекомендуется обратить внимание на качество создаваемых программ и оформления документации.

На проверку преподавателю предоставляются оформленная и скрепленная пояснительная записка и программа на любом электронном носителе (исходные и исполняемый файлы).

Также можно предоставить для проверки материалы курсового проекта в электронном виде. Проверка курсовых проектов в электронном виде прекращается за две недели до начала экзаменационной сессии.

1 Общие требования к курсовому проекту

1.1 Тематика курсовых проектов

Тематика подразделяется на типовую и выбираемую студентом самостоятельно.

В типовом проекте предлагается разработать программное средство по обработке информации о некоторой предметной области, созданию и обработке файлов, используемых для долговременного хранения информации. При этом студенту необходимо самостоятельно определить структуру и характеристики значений вводимых данных, обосновать используемый способ обмена информацией, создать файл во внешней памяти, предусмотреть ряд функций по обработке информации.

Каждое задание на курсовое проектирование уточняется с преподавателем. Примерный перечень заданий на курсовое проектирование представлен в приложении А.

1.2 Общие требования к курсовому проекту

Среда программирования – Microsoft Visual Studio 2017 и выше.

Язык программирования – C++.

Структура проекта – многофайловая.

Парадигма программирования – объектно-ориентированная.

Способ хранения данных – файлы.

Разрабатываемый интерфейс должен быть понятным и защищенным от случайных ошибок.

Ввод исходных данных должен осуществляться либо с клавиатуры, либо из файла (по выбору пользователя).

Результаты работы программы должны сохраняться в текстовый файл по запросу пользователя.

1.3 Функциональные требования к курсовому проекту

Программа реализуется в виде консольного или Windows-приложения. Основные возможности, которые должна выполнять программа, представлены ниже.

- 1 Работа с файлом данных:
 - а) создание файла;
 - б) открытие существующего файла;
 - в) удаление файла.
- 2 Работа с данными:
 - а) режим редактирования:
 - просмотр выбранных данных;
 - добавление новых данных;

- удаление данных;
- редактирование записи;
- б) режим обработки данных:
 - выполнение индивидуального задания;
 - поиск данных (как минимум по трем различным параметрам);
 - сортировка (как минимум по двум различным параметрам).

Предусмотреть:

- возможность навигации по пунктам меню;
- подтверждение выполнения необратимых действий, например, удаление файла должно подтверждаться сообщением вида «Вы действительно хотите удалить файл?»;
- обратную связь с пользователем, например, вывод сообщений об успешности создания файла / удаления записи и т. д.

1.4 Требования к программной реализации курсового проекта

1 Программный код должен быть оформлен в соответствии с правилами, приведенными в Code Convention.

2 Один метод решает только одну задачу (например, не допускается в одном методе считывать данные из файла и выводить их на консоль – это две разные функции). При этом внутри метода возможен вызов других методов.

3 Выполнение операций чтения/записи в файл должно быть сведено к минимуму (т. е. после однократной выгрузки данных из файла в массив/вектор дальнейшая работа ведется с этим массивом/вектором, а не происходит многократное считывание данных из файла в каждом методе).

4 Следует избегать длинных методов и глубокой вложенности: текст функции/метода должен уместиться на один экран, а вложенность блоков и операторов должна быть не более трёх.

5 Выносите код логически независимых модулей в отдельные .cpp-файлы и подключайте их с помощью заголовочных .h-файлов.

2 Содержание курсового проекта

2.1 Содержание пояснительной записки

В состав пояснительной записки (ПЗ) к курсовому проекту входит следующее.
Титульный лист.

Бланк задания курсового проекта.

Введение.

1 Описание предметной области.

1.1 Постановка задачи.

1.2 Объектно-ориентированный анализ предметной области.

2 Проектирование программного средства.

- 2.1 Проектирование структур данных.
- 2.2 Проектирование взаимодействия с пользователем.
- 2.3 Проектирование интерфейса.
- 3 Реализация программного средства.
- 3.1 Реализация классов и структур данных.
- 3.2 Реализация требований.
- 3.3 Реализация концепций ООП.
- 3.4 Описание структуры проекта.
- 3.5 Реализация интерфейса взаимодействия с пользователем.
- 4 Тестирование программного модуля.
- Заключение.
- Список использованных источников.
- Приложение А. Текст программы.
- Образец оформления титульного листа представлен в приложении Б.

2.2 Содержание графической части проекта

Графическая часть проекта выполняется на листе формата А1, где изображается следующее:

- снимки экрана;
- структура меню;
- диаграмма классов.

3 Требования к содержанию разделов пояснительной записки

3.1 Введение

Приводится цель, задачи и ожидаемый результат курсового проекта.

3.2 Описание предметной области

3.2.1 Постановка задачи. Выполняют подробное описание решаемой задачи: описывают назначение разрабатываемой программы, перечисляют входные и выходные данные программы, формулируют требования к программе. В требованиях должны быть отражены следующие пункты.

Требования к функциональным характеристикам программы – это перечень функций, которые должна реализовывать программа.

Пример 1

Разрабатываемая программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- ввода исходных данных;
- запуска процесса вычислений;

- просмотра результатов вычислений;
- записи результатов вычислений в файл.

Требования к входным и выходным данным программы. Входными данными для программы являются данные, вводимые пользователем через её интерфейс. Выходные данные – это информация, получаемая в результате выполнения программы. В записке обосновывают выбор способа ввода/выхода данных. Ввод данных может осуществляться с клавиатуры или из файла, а вывод – в файл, или на монитор компьютера в виде текста, таблицы, графика.

Пример 2

Требования к организации ввода входных данных:

- левая граница отрезка, содержащего корень уравнения. Способ ввода – с клавиатуры или из файла;
- правая граница отрезка, содержащего корень уравнения. Способ ввода – с клавиатуры или из файла;
- погрешность вычисления. Способ ввода – с клавиатуры;
- предусмотреть проверку на принадлежность входных данных допустимым диапазонам и повторение ввода при ошибочных данных.

Требования к организации вывода выходных данных:

- значение функции в точке, являющейся корнем уравнения. Способ вывода – на монитор компьютера в виде строки текста и в файл;
- количество итераций при нахождении корня уравнения. Способ вывода – на монитор компьютера в виде строки текста и в файл;
- график функции. Способ вывода – на экран в виде таблицы и в файл для построения данной зависимости с помощью программного продукта Microsoft Excel.

Требования к интерфейсу пользователя. Следует указать тип интерфейса пользователя.

Пример 3

Тип пользовательского интерфейса – консольное приложение.

Пример 4

Разрабатываемая программа должна иметь графический интерфейс пользователя в виде Windows-формы.

Требования к обработке ошибок. Перечисляются возможные ошибки пользователя при работе с программой, указываются способы диагностики и защиты от этих ошибок.

Пример 5

При выполнении программы необходимо предусмотреть обработку следующих ошибок:

- *неправильный формат исходных данных;*
- *невозможность выделения отрезка, содержащего корень уравнения;*
- *ошибка чтения/записи в файл.*

В результате выполнения должны быть выделены все требования к проектируемому программному средству.

3.2.2 Объектно-ориентированный анализ предметной области. Словесно описывается указанная в задании предметная область, выделяются сущность предметной области. Описываются характеристики объектов предметной области. Термины «класс», «поле» в данном пункте не употребляются.

Пример 6

Охотничий магазин продает оружие и амуницию. Все товары в магазине характеризуются ценой и артикулом. Оружие делится огнестрельное и холодное. Холодное оружие характеризуется...

3.3 Проектирование программного модуля

3.3.1 Проектирование структур данных. Приводится описание используемых структур данных. К рассматриваемым структурам относятся классы, массивы объектов классов, массивы с элементами базовых типов, списки и т. д. Для приведенных структур указывается назначение в программе.

В этом же пункте приводится диаграмма классов. Диаграмма классов должна отражать классы, свойства, методы, связи между классами. Для построения можно использовать возможности MS Visual Studio, пакеты моделирования или графические приложения.

Для каждого класса строится таблица с указанием полей, их типов, ограничений на значения (рисунок 3.1).

Таблица 2.1 – Описание полей класса Товар

<i>Имя поля</i>	<i>Тип данных</i>	<i>Ограничение</i>
<i>Наименование</i>	<i>string</i>	<i>Длина от 10 до 255 символов</i>
<i>Артикул</i>	<i>string</i>	<i>Длина 12 символов, может включать латинский алфавит и цифры</i>
<i>Цена</i>	<i>float</i>	<i>положительное</i>

Рисунок 3.1 – Пример оформления таблицы

При проектировании классов разработанные однотипные функции, например функции поиска, следует включить в отдельные специальные классы,

а обрабатываемые структуры, например, массивы объектов передавать в качестве параметров. Такие классы также описываются в данном пункте.

3.3.2 Проектирование взаимодействия с пользователем. Строится диаграмма взаимодействия, отражающая обращения пользователя, вводимые им данные и передачу этих данных классам для обработки. Для построения используется средство, выбранное в предыдущем пункте для диаграммы классов.

3.3.3 Проектирование интерфейса. Приводится описание форм и используемых элементов, изображение форм в режиме проектирования. Для консольного интерфейса приводится структура меню в виде иерархии, указываются вызываемые в пунктах функции. Пример фрагмента структуры меню представлен на рисунке 3.2.



Рисунок 3.2 – Пример структуры меню

3.4 Реализация программного модуля

3.4.1 Реализация классов и структур данных. Приводится программный код и словесные пояснения к рассмотренным в пункте 2.2 ПЗ классам и структурам данных.

3.4.2 Реализация требований. Для всех требований к функциям программного средства, описанных в пункте 1.1 ПЗ строится таблица (рисунок 3.3), в которой для каждого требования указывается его реализация (имя реализующей функции или класс и реализующий метод).

Таблица 2.2 – Соответствие требований реализации

Функциональное требование	Реализация	Примечание
Должна производиться сортировка товаров по артикулу в порядке возрастания	Метод <i>SortByArt</i> класса <i>Sort</i>	Параметр – массив объектов класса <i>product</i>
Должна производиться сортировка товаров по наименованию в порядке возрастания	Метод <i>SortByName</i> класса <i>Sort</i>	Параметр – массив объектов класса <i>product</i>

Рисунок 3.3 – Соответствие требований реализации

3.4.3 Реализация концепций ООП. Приводится определение концепций ООП, таких как наследование, полиморфизм, абстракция и инкапсуляция. Для каждой из них приводится краткое словесное описание реализации и фрагменты программного кода.

Пример 7

Виртуальные функции поддерживают концепцию полиморфизма. В данном проекте разработаны следующие виртуальные функции...

Пример 8

Для реализации инкапсуляции все поля классов имеют спецификатор доступа *private*.

3.4.4 Описание структуры проекта. Перечисляются созданные .сpp- и .h-файлы. Указывается, какие структурные единицы проекта в каких файлах реализованы. Указываются связи между файлами.

3.4.5 Описание интерфейса взаимодействия с пользователем. Приводятся снимки экрана с основными элементами интерфейса (формы в режиме работы или экран с различными пунктами меню), даётся кратное словесное пояснение.

3.5 Тестирование программного модуля

В разделе «Выполнение программы» должна быть указана последовательность действий пользователя, обеспечивающих загрузку, запуск, выполнение всех функций программы. Выполнение каждой функции сопровождается снимком экрана с соответствующим результатом работы программы.

В настоящем документе исключено прямое обращение к пользователю. Должны отсутствовать слова «откройте», «нажмите», «укажите» и пр. Следует применять штампы «следует открыть», «следует нажать» и им подобные.

3.6 Заключение

В разделе приводятся результаты проделанной работы и делаются выводы о том, достигнута ли цель проектирования и решены ли поставленные задачи.

3.7 Список использованных источников

В списке перечисляются книги, статьи, источники из интернета, которые были использованы при выполнении работы. Информацию о правилах оформления этого списка представлена в подразделе 4.8, а также в ГОСТ 7.1–2003 *Библиографическая запись. Библиографическое описание. Общие требования и правила.*

Список использованных источников должен содержать не менее пяти наименований не старше 10 лет.

3.8 Приложения

Приложения содержат материалы вспомогательного характера: алгоритмы, тексты программ, результаты тестирования, большие таблицы и т. д. В записке по данному курсовому проекту обязательным является наличие приложения с текстом программы. Наличие комментариев к программе является обязательным. Правила оформления представлены в подразделе 4.7.

4 Оформление пояснительной записки

4.1 Общие требования

Пояснительная записка к курсовой работе оформляется в соответствии с правилами оформления текстовых документов, изложенными в ГОСТ 2.105–95 *Общие требования к текстовым документам*, и с правилами оформления курсовых работ, изложенными в данных методических рекомендациях. Список использованных источников оформляется в соответствии с правилами, изложенными в ГОСТ 7.1–2003.

Записка должна быть напечатана на одной стороне листа белой бумаги формата А4. Рекомендуется: шрифт – Times New Roman, размер шрифта – 13, межстрочный интервал – 1,15.

Отступ от рамки – не менее 5 мм.

Цвет шрифта – черный, при необходимости вписывать также надо чернилами, пастой или тушью черного цвета. Опечатки не зачеркивать, а заклеивать, подчищать или закрашивать белой краской.

В тексте после знаков препинания обязательно ставится пробел. Нельзя сокращать слова (кроме сокращений, установленных правилами орфографии). Например, пишется целиком «то есть», «так как». Нельзя употреблять специальные знаки типа «=», «%», «+» и другие без цифр.

Записка оформляется на листах с рамкой.

Раздел «Содержание» печатается на листе с основной подписью для заглавного листа (пример заполнения представлен на рисунке 4.1), остальные листы курсовой работы выполняются на листах с основной подписью для последующих листов (пример заполнения представлен на рисунке 4.2).

					051.09.03.01.№зачетки.№журн.81-01						
Изм.	Лист	№ докум.	Подп.	Дата							
Разработал	Ваша фамилия						Лит.	Лист	Листов		
Проверил	Фамилия					Ваша тема		2	74		
Н.контр.							гр.ИСИТ - 191				
Утв.											

Рисунок 4.1 – Основная надпись для заглавного листа

					301.1 – 45.05.01.№по списку.№ зачетки					Лист
Изм.	Лист	№ докум.	Подп.	Дата						3

Рисунок 4.2 – Основная надпись для последующих листов

4.2 Нумерация страниц

Нумерация страниц – сквозная, по всем страницам записки, включая приложения. Первая страница – это титульный лист, на нем номер не ставится. Номер проставляется арабской цифрой без точки в нижней части листа справа.

4.3 Построение текста

Рассмотрим разбивку и нумерацию структурных единиц записки. Структурные единицы пояснительной записки приведены в подразделе 2.1. Каждая структурная единица записки начинается с нового листа.

Структурные единицы записки «Содержание», «Введение», «Заключение», «Список использованных источников» не нумеруются. Соответствующие заголовки записываются обычным шрифтом с прописной буквы с выравниванием по центру. Заголовки не подчеркиваются и точка в конце них не ставится.

Текст основной части делят на разделы (номер раздела состоит из одной цифры), разделы – на подразделы (номер подраздела состоит из двух цифр), подразделы – на пункты и т. д.

Разделы нумеруются арабскими цифрами без точки; подразделы – в пределах раздела. Номер состоит из номера раздела и подраздела, разделенных точкой. В конце номера подраздела точка не ставится. За номером раздела или подраздела следует его название, оно записывается обычным шрифтом с прописной буквы без точки в конце. Переносы слов в заголовках не допускаются. Если заголовков состоит из двух предложений, их разделяют точкой.

Каждый раздел начинается с нового листа, подразделы с нового листа не начинаются. Номер и название раздела или подраздела записываются с абзацного отступа. Расстояние между заголовком и текстом, заголовками раздела и подраздела – одна пустая строка.

4.4 Рисунки

Все иллюстрации (чертежи, схемы, графики, структурные схемы и др. (кроме таблиц)) называются рисунками. Они должны располагаться непосредственно после ссылки на них в тексте или на следующей странице. Рисунок располагается так, чтобы его удобно было смотреть без поворота листа или с поворотом по часовой стрелке.

Рисунок центрируется, подпись располагается с абзацного отступа.

Рисунки, за исключением рисунков приложений, следует нумеровать арабскими цифрами сквозной нумерацией. Если рисунок один, то он обозначается, например, «Рисунок 1». Слово «Рисунок» и его номер располагаются с абзацного отступа.

Рекомендуется нумеровать иллюстрации в пределах разделов. В этом случае номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой. Например, надпись «Рисунок 1.2» означает второй рисунок первого раздела.

Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами, добавляя перед цифрой обозначение приложения. Например, «Рисунок А.3», что обозначает третий рисунок в приложении А.

Иллюстрации должны иметь наименование и пояснительные данные (подрисуночный текст). Порядок расположения следующий: вначале – сам рисунок, затем – подрисуночный текст, далее – слово «Рисунок», номер рисунка и наименование в виде, например, «Рисунок 1 – Структурная схема».

4.5 Таблицы

В тексте пояснительной записки следует помещать итоговые и наиболее важные таблицы; таблицы вспомогательного и справочного характера – в приложениях.

Таблицы нумеруются и обозначаются по тем же правилам, что и иллюстрации, только вместо слова «Рисунок» пишется слово «Таблица» и располагается это слово и название таблицы без абзацного отступа над таблицей. Примеры обозначений: «Таблица 1», «Таблица В.1», «Таблица 3.1 – Список модулей».

Таблицы ограничиваются одинарными горизонтальными и вертикальными линиями. Использование диагональных линий не допускается.

Если таблица выходит за формат листа, то ее делят на части. Части таблицы можно помещать на одном листе одна над другой или рядом или переносить на другие листы. При переносе части таблицы на другой лист заголовок помещают только над первой частью, то есть слово «Таблица» с названием указывают только над первой частью, а над другими частями пишут слова «Продолжение таблицы» с указанием номера таблицы. Шапку таблицы при переносе части таблицы не повторяют, переносится строка с номерами столбцов.

4.6 Ссылки

Примеры ссылок в тексте пояснительной записки:

- 1) на иллюстрацию – «... в соответствии с рисунком 1.2 ...»;
- 2) на формулу – «... в формуле (2.1) ...»;
- 3) на приложение – «... (приложение Б) ...»;
- 4) на таблицу – «... в таблице 1.2 ...».

Повторные ссылки следует давать с сокращенным словом «смотри», например: (см. рисунок 1.2), (см. приложение Б).

На материалы, взятые из литературы или других источников, должны быть даны ссылки с указанием номера источника по списку использованных источников. Номер ссылки проставляется арабскими цифрами в квадратных скобках, например: [1], [1, 2, 5].

4.7 Приложения

Каждое приложение начинается с нового листа с указанием наверху посередине слова «Приложение» с первой прописной буквы и его обозначения.

Приложения обозначаются заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ъ. Например, «приложение А», «приложение Б» и т. д.

После строки, содержащей слово «Приложение», т. е. на второй строке, записывается симметрично по тексту с первой прописной буквы заголовков приложения. Например:

Приложение А
Текст программы

Затем через одну пустую строчку следует текст приложения. Текст каждого приложения может быть разделен на разделы, подразделы и т. д., которые нумеруются арабскими цифрами в пределах приложения по аналогии с разделами и подразделами основной части пояснительной записки. Например, «П.А.2» – это обозначение второго раздела приложения А.

Код программы допускается оформлять шрифтом меньшего размера до 10 пт.

4.8 Список использованных источников

Источники – это книги, учебники, диссертации, статьи из журналов, статьи из интернета и т. д., использованные при выполнении курсовой работы. Источники в списке располагаются в порядке ссылок в тексте записки или по алфавиту, нумеруются арабскими цифрами без точки и печатаются с абзацного отступа, при этом дается библиографическое описание каждого источника в соответствии с ГОСТ 7.1, ГОСТ 7.12.

Общий шаблон описания книги, у которой не более трех авторов: ФИО_автора, название книги, точка, тире, город, двоеточие, издательство, запятая, год издания, точка, тире, количество страниц, буква «с», точка.

Название города дается целиком, допустимы только сокращения «М.» (Москва) и «СПб.» (Санкт-Петербург); название издательства – без кавычек. Если у книги один, два или три автора, то вначале указывается фамилия, потом – инициалы.

Например,

Один, два или три автора

Шуп, Т. Решение инженерных задач на ЭВМ: практическое руководство: пер. с англ. / Т. Шуп. – Москва: Мир, 1982. – 238 с.

Четыре и более автора

Зубчатые передачи: справочник / Е. Г. Гинзбург [и др.]; под ред. Н. Т. Халебского. – Минск: Машиностроение, 1980. – 416 с.

Книги под общей редакцией

Шейнин, А. М. Эксплуатация дорожных машин / А. М. Шейнин; под общ. ред. А. М. Шейнина. – Москва: Транспорт, 1992. – 328 с.

Стандарты

ГОСТ 19.701–90. Схемы алгоритмов, программ данных и систем. – Москва: Изд-во стандартов, 2004. – 26 с.

Список литературы

- 1 **Страуструп, Б.** Программирование: принципы и практика с использованием С++ / Б. Страуструп. – Москва: Вильямс, 2016. – 1328 с.
- 2 **Боровский, А. Н.** Qt4.7+. Практическое программирование на С++ / А. Н. Боровский. – Санкт-Петербург: ВHV, 2012. – 496 с.
- 3 **Васильев, А. Н.** Объектно-ориентированное программирование на С++ / А. Н. Васильев. – Санкт-Петербург: Наука и техника, 2016. – 544 с.
- 4 **Васильев, А. Н.** Программирование на С++ в примерах и задачах / А. Н. Васильев. – Москва: ЭКСМО, 2017. – 416 с.
- 5 **Кенег, Э.** Эффективное программирование на С++. Практическое программирование на примерах. Серия «С++ In-Depth» / Э. Кенег. – Москва: Диалектика, 2019. – 368 с.
- 6 **Кениг, Э.** Эффективное программирование на С++. Практическое программирование на примерах: в 2 т. / Э. Кениг. – Москва: Вильямс, 2016. – Т. 2. – 368 с.
- 7 **Лафоре, Р.** Объектно-ориентированное программирование в С++. Классика Computer Science / Р. Лафоре. – Санкт-Петербург: Питер, 2013. – 928 с.
- 8 **МакГрат, М.** Программирование на С для начинающих / М. МакГрат. – Москва: Эксмо, 2015. – 192 с.
- 9 **Мартин, Р.** Чистый код. Создание, анализ и рефакторинг. Библиотека программиста / Р. Мартин. – Санкт-Петербург: Питер, 2018. – 464 с.
- 10 **Мейерс, С.** Эффективное использование С++. 35 новых рекомендаций по улучшению ваших программ и проектов / С. Майерс. – Москва: ДМК Пресс, 2014. – 294 с.
- 11 **Павловская, Т. А.** С/С++. Программирование на языке высокого уровня : учебник / Т. А. Павловская. – Санкт-Петербург: Питер, 2016. – 461 с.
- 12 **Полубенцева, М.** С/С++. Процедурное программирование / М. С. Полубенцева. – Санкт-Петербург: ВHV, 2017. – 432 с.
- 13 **Перри, Г.** Программирование на С для начинающих / Г. Перри, Д. Миллер. – Москва: ЭКСМО, 2016. – 192 с.
- 14 **Хенкеманс, Д.** Программирование на С++ / Д. Хенкеманс, М. Ли. – Санкт-Петербург: Символ-плюс, 2015. – 416 с.
- 15 **Шилдт, Г.** С++. Базовый курс / Г. Шилдт. – Москва: Вильямс, 2015. – 624 с.

Приложение А (обязательное)

Примерный перечень заданий на курсовое проектирование

1 Создать шаблон класса «стек». Написать программу, использующую этот шаблон класса для моделирования Т-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа).

2 Создать шаблон класса «стек». Написать программу, использующую этот шаблон для отыскания прохода по лабиринту с использованием данного шаблона класса.

3 Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат определяется его координатами в матрице. После отыскания прохода программа печатает найденный путь в виде координат квадратов.

4 Создать шаблон класса «бинарное дерево». Использовать его для сортировки целых чисел и строк, задаваемых с клавиатуры или из файла.

5 Создать шаблон класса «очередь». Написать программу, демонстрирующую работу с этим шаблоном для различных типов параметров шаблона. Программа должна содержать меню, позволяющее осуществить проверку всех методов шаблона.

6 Создать шаблон класса «очередь с приоритетами». При добавлении элемента в такую очередь его номер определяется его приоритетом. Написать программу, демонстрирующую работу с этим шаблоном для различных типов параметров шаблона. Программа должна содержать меню, позволяющее осуществить проверку всех методов шаблона.

7 Создать шаблон класса «бинарное дерево», обладающее возможностью добавления новых элементов, удаления существующих, поиска элемента по ключу, а также последовательного доступа по всем элементам.

8 Написать программу, использующую этот шаблон для представления англо-русского словаря. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Предусмотреть возможность формирования словаря из файла и с клавиатуры.

9 Разработать программное средство для учета данных о товарах в аптеке. Программа должна содержать меню, позволяющее осуществить редактирование ассортимента, поиск и сортировку по различным критериям.

10 Разработать программное средство для учета данных о товарах в аптеке. Программа должна содержать меню, позволяющее осуществить редактирование ассортимента, поиск и сортировку по различным критериям.

**Приложение Б
(обязательное)**

Образец оформления титульного листа

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ТЕМА КУРСОВОГО ПРОЕКТА

Курсовой проект
по дисциплине «Объектно-ориентированное проектирование
и программирование»

301.1 – 45.05.01. № по списку. № зачетки

Исполнитель	_____	ФИО	студента,	группа
	(подпись)			
Руководитель	_____	ФИО		руководителя
	(подпись)			
Дата допуска к защите	_____			
Дата защиты	_____			
Оценка	_____			