**Методические указания к лабораторной работе № 10 по курсу**

**Основы программирования**

#### " Комплексная Лабораторная работа/Домашнее задание.

#### Документация. Разработка Технического задания проекта"

Москва, МГТУ - 2017

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

СОДЕРЖАНИЕ

[1. Цель работы 3](#_Toc405801483)

[2. Задачи, решаемые в лабораторной работе 3](#_Toc405801484)

[3. Основные понятия и примеры 4](#_Toc405801485)

[3.1. Программная документация 4](#_Toc405801486)

[3.2. Программная документация, разрабатываемая в домашнем задании 4](#_Toc405801487)

[3.3. Особенности и принципы разработки программной документации (ПД) 4](#_Toc405801488)

[3.4. Требования к оформлению ПД в ДЗ 5](#_Toc405801489)

[3.5. Документ техническое задание (ТЗ) и его назначение 6](#_Toc405801490)

[3.6. Стиль изложения в ТЗ 6](#_Toc405801491)

[3.7. Требования к ТЗ 6](#_Toc405801492)

[3.8. Разработка ТЗ 6](#_Toc405801493)

[3.9. Содержание ТЗ 7](#_Toc405801494)

[3.10. Раздел ТЗ – 5. Технические требования (общее) 8](#_Toc405801495)

[3.11. Раздел ТЗ – 5.1 Требования к функциональным характеристикам 8](#_Toc405801496)

[3.12. Разделы ТЗ – Другие технические требования 8](#_Toc405801497)

[3.13. Раздел ТЗ – требования к документации 8](#_Toc405801498)

[3.14. Раздел ТЗ – этапы сроки выполнения 8](#_Toc405801499)

[3.15. Раздел ТЗ – порядок приема и контроля 9](#_Toc405801500)

[3.16. Порядок изменения и корректировки ТЗ 9](#_Toc405801501)

[4. Порядок работы и методические указания (основные требования) 9](#_Toc405801502)

[4.1. Прочитать и усвоить основные понятия данной лабораторной работы. 9](#_Toc405801503)

[4.2. Прочитать документ требования к ДЗ/КЛР. 9](#_Toc405801504)

[4.3. Определить номер своего варианта и получить тему ДЗ. 10](#_Toc405801505)

[4.4. Познакомиться с требованиями к оформлению документа ТЗ. 10](#_Toc405801506)

[4.5. Проработать предметную область ДЗ/КЛР. 10](#_Toc405801507)

[4.6. Познакомиться с образцом документа ТЗ. 10](#_Toc405801508)

[4.7. Создать в VS 2005 консольный проект (DZ\_XXXXX\_ХDD). 10](#_Toc405801509)

[4.8. Обеспечить русификацию консольного ввода и вывода. 11](#_Toc405801510)

[4.9. Провести макетирование системы классов. 11](#_Toc405801511)

[4.10. Использование шаблона документа ТЗ 13](#_Toc405801512)

[4.11. Разработать разделы документа: Наименование, назначение и исполнитель. 13](#_Toc405801513)

[4.12. Разработать раздел документа: Функциональные требования. 13](#_Toc405801514)

[4.13. Прочитать раздел документа образец: Требования к ПД. 13](#_Toc405801515)

[4.14. Прочитать раздел документа образец: Стадии и этапы разработки. 13](#_Toc405801516)

[4.15. Прочитать раздел документа образец: Порядок приема и контроля. 14](#_Toc405801517)

[4.16. Построить предварительную диаграмму классов для ДЗ. 14](#_Toc405801518)

[4.17. Дополнительные требования для сильных студентов 14](#_Toc405801519)

[5. Варианты по группам и студентам 14](#_Toc405801520)

[6. Предварительные диаграммы классов домашнего задания 17](#_Toc405801521)

[7. Контрольные вопросы к ЛР 18](#_Toc405801522)

[8. Сроки и порядок защиты документа ТЗ 18](#_Toc405801523)

[9. Литература 18](#_Toc405801524)

[10. Приложение: Шаблон документа ТЗ 19](#_Toc405801525)

[11. Приложение: Образец документа ТЗ (\*.doc) 19](#_Toc405801526)

[12. Приложение: Образец документа ТЗ (\*.pdf) 19](#_Toc405801527)

[13. Справочные материалы 19](#_Toc405801528)

Цель работы

Целью лабораторной работы получение навыков разработки документа на программное обеспечение - технического задания (ТЗ). Работа выполняется в рамках ДЗ/КЛР. На основе образца и шаблона данного документа, для своего варианта задания, студенты разрабатывают и защищают собственное техническое задание для своего варианта ДЗ. Данная лабораторная работа входит в цикл работ, выполняемых студентами кафедры в рамках практикума по программированию (во 2-м семестре кафедры ИУ5). Кроме этой работы в цикл входят ЛР № 13,14, и 15. Они объединены общей задачей. Порядок работ и требования к ним изложены в документе “ Методические указания для выполнения ДЗ/КЛР по дисциплине Программирование на основе классов и шаблонов кафедры ИУ5”[6], он размещен на сайте дисциплины. Документ оформляется в MS WORD в виде текстового документа.

Задачи, решаемые в лабораторной работе

В процессе выполнения ЛР студенты индивидуально должны выполнить следующие задачи. Перечислим основные требования к лабораторной работе:

* Прочитать и усвоить основные понятия данной лабораторной работы.
* Прочитать документ требования к ДЗ/КЛР.
* Определить номер своего варианта и получить тему ДЗ.
* Познакомиться с требованиями к оформлению документа ТЗ.
* Познакомиться с образцом документа ТЗ.
* Провести макетирование системы классов.
* Загрузить шаблон документа и внести необходимые данные для идентификации студента, группы и индивидуального задания студента по вариантам.
* Разработать разделы документа: Наименование, назначение и исполнитель.
* Разработать раздел документа: Технические требования.
* Построить предварительную диаграмму наследования классов.
* Прочитать и изучить раздел документа образец: Требования к программной документации.
* Прочитать и изучить раздел документа образец: Стадии и этапы разработки.
* Прочитать и изучить раздел документа образец: Порядок приема и контроля.
* После выполнения перечисленных действий студенты: демонстрируют преподавателю документ, разработанный во время ЛР. Документ оформляется в MS WORD на основе шаблона документа с использованием образца документа.

**Примечания**. 1. Для удобства восприятия текста используется цвет и тип шрифта. Фрагменты исходного текста, включаемые в программные модули, я буду выделять синим цветом. Например, вставка заголовочного файла в программу будет выглядеть так:

#include <iostream>

2. Вывод результата в консольное окно, который формируется программой, будем помечать коричневым цветом и устанавливать непропорциональный шрифт **Courier New**. Например:

**Введите iVal: 10**

3. Формализованные описания языка и синтаксические правила будем записывать зеленым цветом:

**<левая часть выражения присваивания> = <правая часть выражения присваивания>;**

4. Если в тексте встречается переменная, которая подчеркнута, то это означает, что дается определение важного понятия и это понятие встречается в данном тексте первый раз. Например:

Программа – это упорядоченная совокупность операторов …

Основные понятия и примеры

В рамках освоения дисциплины “Программирование на основе классов и шаблонов” студенты выполняют домашнее задание (комплексную лабораторную работу - КЛР) по разработке собственной системы классов. Назначение системы классов определяется вариантом задания. Работы выполняются студентами индивидуально. Работа выполняется частично в часы проведения лабораторных работ – КЛР (комплексная лабораторная работа) по отдельным частям и в часы самостоятельной проработки курса. Результаты работы учитываются во втором модуле курса.

## Программная документация

Любой программный проект должен сопровождаться разработкой комплекса программной документации. Время, которое затрачивается на разработку такой документации соизмеримо со временем самой программной разработки (не менее 50% времени всего проектирования). Иногда это время превышает время программирования и отладки проекта. Некоторые документы разрабатываются до начала разработки. Например, техническое задание (ТЗ) на программную разработку. Без создания грамотного ТЗ проект вообще нереализуем или, скорее всего, не будет успешным. Кроме того, без хорошо разработанного технического задания, невозможно успешно предъявить и сдать работу заказчику, а также, успешно закрыть работу по проекту (включая и получения оплаты за выполненную работу). Программисты (или специальные разработчики документации) разрабатывая документы, по сути, являются “техническими писателями”, а освоение таких умений является неотъемлемой частью получаемых навыков в рамках подготовки по нашей специальности.

## Программная документация, разрабатываемая в домашнем задании

Для получения основных навыков в разработке технической документации на программное обеспечение выбраны главные документы из значительного множества документов, требуемых в ГОСТ [8]. Этот комплект документов содержит следующие документы:

* Техническое задание (ТЗ);
* Описание применения программного продукта (ОП);
* Техническое описание программного продукта (ТО);
* Руководство пользователя (РП);
* Руководство системного программиста (РСП);
* Исходные тексты программ системы классов и тестового примера (ИТ);
* Программа и методика испытаний на основе тестового примера (ПМИ);
* Описание тестового примера (ОПТ);

Последний документ (“Описание тестового примера”) может не выделяться отдельно и входить составной частью других документов: “ Руководство пользователя ”, ” Техническое описание программного продукта ”, “ Программа и методика испытаний ”. В рамках домашнего задания и КЛР студенты должны разработать весь комплект документации на программный продукт, разрабатываемый по их варианту. По каждому документу у вас предусмотрена отдельная лабораторная работа, которую вы можете выполнить и заранее в рамках часов, отводимых под самостоятельных работу студентов (34 академических часа)

## Особенности и принципы разработки программной документации (ПД)

При разработке ПД нужно учесть следующие обстоятельства:

* Кто из участников процесса разрабатывает данный документ ПД.
* Для кого разрабатывается данный документ ПД
* Каков стиль изложения данного документа ПД
* Каково содержание данного документа ПД
* На каком этапе проектирования разрабатывается данный документ ПД
* Каковы требования к данному документу.

Четкие ответы на перечисленные вопросы должны быть у каждого разработчика данного документа. В процессе разработки документации на программный продукт могут участвовать следующие группы и разновидности специалистов:

* Руководители проекта.
* Разработчики программисты (или исполнители проекта).
* Заказчики проекта.
* Пользователи разрабатываемого программного обеспечения (ПО).
* Системные программисты, устанавливающие ПО для пользователя.
* Технические писатели (разработчики технической документации).

Эти специалисты могут выполнять следующие роли по отношению к документам комплекта ПД: писать или разрабатывать документы, контролировать содержание документов, согласовывать (подписывать) документы и утверждать документы (подписи). В нашем случае, в игровом обучающем режиме студенты могут выполнять разные роли: руководителей, писателей и разработчиков, а преподаватели – роль заказчиков проекта.

В каждой ЛР из цикла КЛР мы рассмотрим все эти особенности, применительно к каждому типу рассматриваемого документа. Здесь выделим общие особенности стиля изложения и разработки ПД:

* Документы должны содержать информацию, имеющую отношение только к данному проекту и данному документу.
* Документы должны быть написаны на техническом языке и не должны содержать неопределенности и неоднозначности.
* Документы должны быть написаны короткими предложениями, и они должны включать минимум распространенных предложений (с придаточными предложениями).
* Стиль документа должен соответствовать стилю документа данного типа.
* Документы не должны содержать “воды” и отступлений от содержания. Никакой “лирики”.
* Все пункты должны быть пронумерованы (разделы, позиции и т.д.) и допускать ссылки из других документов. Не допускается перечисления в виде маркированных списков (**bullets**).
* При ссылках на другие документы должны указываться прямые ссылки, содержащие: полное название документа и позицию (пункт), на который производиться ссылка.
* В пределах проекта устанавливается требования к оформлению документов: редактор текста, шрифт, кегль шрифта, интервал, способы форматирования текста, размеры страниц, содержание колонтитулов, способы рисования иллюстраций документа (рисунков), обязательная нумерация страниц в документе.

## Требования к оформлению ПД в ДЗ

В нашем случае требования к оформлению следующие:

* Текстовый редактор - **MS WORD** (не Open Office!),
* Шрифт - **Times New Roman**.
* Кегль шрифта - **12**,
* Интервал между строками - **одинарный**,
* Способы форматирования текста – **по ширине**,
* Размеры страниц – **А4 (верх - 2 , низ - 2 , слева - 3 , справа - 2)**,
* Содержание колонтитулов (**вариант, группа, ФИО студента)**,
* Способы рисования иллюстраций документа (Предпочтительнее **MS Visio** можно **MS WORD**),
* Нумерация страниц – **центр – верх**.

Данные требования необходимо соблюдать при разработке всех документов домашнего задания.

## Документ техническое задание (ТЗ) и его назначение

Документ техническое задание (ТЗ) разрабатывается на ранних этапах проектирования программного продукта и содержит требования к разработке: функции, сроки, требования к продукту в различных аспектах и порядок их реализации. Этот документ является важнейшим и иногда разрабатывается до заключения финансового договора с заказчиком. Считается, что после утверждения ТЗ (его подписания сторонами), оно может быть изменено только по обоюдному согласию заказчика и разработчика и только в определенном порядке.

Документ ТЗ разрабатывается совместно заказчиком и разработчиками (руководителем проекта и исполнителями) и утверждается (подписывается). После этого, не выполнение отдельных пунктов ТЗ может являться основанием для расторжения договора (по закону). Более того, утверждается, что невыполнение ТЗ преследуется по закону.

**Примечание**. Студенты часто стараются разработать ТЗ после завершения разработки. Это не правильно, так как заказчик (преподаватель у нас) может ТЗ не утвердить это очень рискованно.

ТЗ является, кроме того, основой для самой разработки и основой для разработки других документов.

## Стиль изложения в ТЗ

Стиль изложения документа ТЗ – декларативный (предписывающий): все предложения должны соответствовать предписывающему стилю ("программа должна обеспечивать …" или "в процедуре необходимо обеспечить" или "система должна выполнять" и т.д.). Кроме этого в полной мере при разработке ТЗ нужно учесть общие требования к стилю документов.

## Требования к ТЗ

Документ ТЗ является важнейшим документом для проектирования и реализации проектов. В частности, проведение приемно-сдаточных испытаний программного продукта основывается (см. пояснения к ЛР по “ Программе и методике испытаний ”). Поэтому все пункты требования должны быть четкими и не допускать разных трактовок. Они должны быть такими, чтобы их можно было бы проверить. В противном случае их из ТЗ нужно исключить, обоснованно доказав это заказчику.

При разработке ТЗ реально сталкиваются разнонаправленные интересы заказчиков и разработчиков (фактически возникают противоречия, которые нужно разрешить). Заказчики хотят с минимальными затратами сделать как можно больше, а разработчики хотят сделать минимум также со своими минимальными затратами трудоемкости.

## Разработка ТЗ

Разработка ТЗ должна производиться на ранних стадиях проектирования программных продуктов. Иногда, перед разработкой ТЗ выполняют работы по пробному макетированию отдельных частей программы или ее в целом. Макетирование технических решений позволяет проверить правильность формулировок отдельных пунктов ТЗ, определить трудоемкость и сроки выполнения работ. Результаты макетирования можно предъявить заказчику для подтверждения реальности сроков выполнения работ и пояснения согласованности и наглядности функций, возложенных на программный продукт. В макете может быть продемонстрирован интерфейс пользователя с будущей системой.

Для сложных программных разработок (программных систем) предварительно могут проводиться дополнительные научно - исследовательские работы (НИР) или опытно-конструкторские разработки (ОКР). Результатами таких работ должны быть предложения к разработке ТЗ на систему или сам документ ТЗ.

В любом случае, в тексте ТЗ должны содержаться такие формулировки требований, которые не ограничивают разработчика и предельно понятны заказчику. Так, например, в нашем случае не следует в разделе функциональных требований указывать имена классов, методов и т.д. В процессе разработки они могут измениться, но это фактически может привести к формальному невыполнению пунктов ТЗ, на реализации которых заказчик, возможно, будет настаивать. С другой стороны, может быть внесена такая формулировка, которая понимается заказчиком и разработчиком по-разному. Это не правильно. Таких текстов в документе следует избегать.

## Содержание ТЗ

Содержание документа ТЗ по пунктам приведено ниже. В образце документа ТЗ приведено ТЗ для варианта улиц и домов, описанного в общем пособии по курсу [3]. В шаблоне документа ТЗ (см. в конце данных методических указаний) даны методические указания к написанию и приспособлению документа применительно к конкретному варианту студента. Содержание документа ТЗ:

1. НАИМЕНОВАНИЕ

2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

4. ИСПОЛНИТЕЛЬ

5. ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

5.1. Требования к функциональным характеристикам

5.2. Требования к программному обеспечению

5.3. Требования к условиям эксплуатации

5.4. Требования к информационному обеспечению

5.5. Требования к надежности

5.6. Требования к составу и характеристикам технических средств

5.7. Требования к программной совместимости

6. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

6.1. Разрабатываемые технические и эксплуатационные документы

7. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

1. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

8.1. Сроки выполнения отдельных этапов работ

1. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ ЗАДАНИЯ

9.1. Требования к сдаче и условия приемки

10. ДОПОЛНИТЕЛЬНЫЕ ТРЕБОВАНИЯ

Дополнительно к методическим указаниям шаблона выделим главные требования к основным разделам ТЗ (на выполнение этих требований будет обращаться повышенное внимание при предъявлении программ и защите документов):

В пункте 3 (НАЗНАЧЕНИЕ РАЗРАБОТКИ) очень кратко (2-3 предложения) формулируется назначение разработки: какие преимущества возникают при его применении, какие новые возможности появляются у пользователя, какие операции автоматизируются при использовании программного продукта, для решения каких задач может использоваться данная система классов.

В раздел 5.1 на должны быть включены основные функции системы классов, включающие способы создания всех типов объектов, использования методов классов. Требования к функциям должны формулироваться на содержательном уровне (а не программистском). Эти основные функции должны быть размещены в начале раздела 5.1. Кроме того, должны быть отражены и дополнительные функции.

Все позиции в ТЗ (для ссылок на них) должны быть пронумерованы с помощью многоуровневой нумерации (5.1.1, 5.1.2 и т.д.). В разделах п.7 и п.10 информация может отсутствовать при выполнении ДЗ/КЛР. В раздел 5 могут быть добавлены новые позиции по соглашению с заказчиком (Например, функции защиты информации, безопасность использования программного и технического обеспечения и т.д.).

## Раздел ТЗ – 5. Технические требования (общее)

В этом разделе ТЗ перечисляются технические требования к программному изделию. Эти требования должны быть выполнены при реализации программного проекта. Формулировки должны быть однозначными. Требования должны быть выполнимы разработчиком и проверяемы заказчиком при проведении приемно – сдаточных испытаниях. В главный раздел могут выноситься требования общего характера, которые должны быть выполнены (Смотрите шаблон и образец ТЗ).

## Раздел ТЗ – 5.1 Требования к функциональным характеристикам

Данный раздел – один из основных документа ТЗ. Он разрабатывается на основе изучения поставленной задачи. Набор функциональных требований (или функциональных характеристик) должен быть полным. Требования не должны противоречить друг другу и ясно восприниматься заказчиком – пользователем. (Смотрите шаблон и образец ТЗ).

## Разделы ТЗ – Другие технические требования

Разделы ТЗ, связанные с другими техническими требованиями, задают требования к реализации и эксплуатации программного продукта. Их правильная разработка очень важна. Здесь определяются средства реализации проекта, условия его эксплуатации, форматы данных и файлов, требования к надежности ПО. В нашем случае (ДЗ/КЛР) эти требования, за исключением некоторых дополнительных вариантов, могут быть одинаковыми для всех заданий. Однако их внимательно необходимо прочитать, проверить и осмыслить. (Смотрите шаблон и образец ТЗ).

## Раздел ТЗ – требования к документации

В данном разделе определяется перечень документации, разрабатываемый в программном проекте. В нашем случае данный перечень уже задан, если необходимо для реализации проекта в него можно внести дополнения (см. дополнительные варианты). В курсовых работах и проектах в данный перечень добавляются листы: диаграммы классов и объектов, блок-схемы алгоритмов, структуры данных и т.д. В нашем случае диаграммы и блок-схемы можно оформить в виде рисунков, вставляемых в соответствующие документы. (Смотрите шаблон и образец ТЗ).

## Раздел ТЗ – этапы сроки выполнения

Этапы и сроки выполнения ДЗ определяются учебным планом и длительностью семестра. В нашем случае приведенный список этапов и сроков можно сохранить в неизменном виде. Нужно только учесть, что формально ЛР № 12 выполняется в конце семестра, а работу над заданием необходимо начать как можно раньше. Поэтому желательно познакомиться с темой задания и разработать ТЗ в сроки, определенные в образце и шаблоне документа. (Смотрите шаблон и образец ТЗ).

## Раздел ТЗ – порядок приема и контроля

В данном разделе ТЗ формулируются требования к приемке программного продукта. Определяются принципы проверки и его этапы. Проверка и сдача программного проекта может выполняться на основе различных принципов, и даже в несколько этапов:

* Проверка принципов построения продукта на макете и ТЗ (ранние стадии).
* Проверка работоспособности продукта – целенаправленно проверяется наличие ошибок в ПО (завершение разработки).
* Проверка документации на соответствие требованиям (завершение разработки).
* Выборочная проверка работоспособности на основе ТЗ и его выполнения, на основе программы и методики испытаний (ПМИ). По завершению разработки.
* Полная комплексная проверка работоспособности и документации на программный продукт. Такая проверка выполняется на основе специального согласованного документа ПМИ.
* Выборочная проверка работоспособности и документации на программный продукт. Такая проверка выполняется на основе специального согласованного документа ПМИ.

В нашем случае (сдача ДЗ студентами) мы выполняем выборочную проверку программ и документации на основе специального документа ПМИ. Для проведения проверки разрабатывается специальный тестовый пример, который позволяет выборочно проверить каждый пункт ТЗ из раздела 5.1 – требования к функциональным характеристикам или, по-другому, функциональным требованиям. Разработка этого примера и документа ПМИ будет рассмотрена в специальной лабораторной работе по курсу. (Смотрите шаблон и образец ТЗ).

## Порядок изменения и корректировки ТЗ

Документ техническое задание может быть, в порядке исключения, изменен или откорректирован даже после его согласования и утверждения (подписи заказчиком и разработчиком). Для этого разрабатывается отдельный документ с названием: “Корректировка ТЗ”. Далее, после всех согласований документ может быть заново переоформлен либо документ корректировки просто прикладывается к основному ТЗ. Эти действия (переоформление или прикладывание) выполняются по обоюдному согласию разработчика и заказчика.

Порядок работы и методические указания (основные требования)

## Прочитать и усвоить основные понятия данной лабораторной работы.

Предварительно перед началом лабораторной работы, а желательно и до ее начала в дисплейном классе, необходимо прочитать раздел этого документа “Основные понятия”. Необходимо усвоить понятия: программная документация; комплект ПД; техническое задание; “технический писатель”; требования к сути документа и его оформлению; группы специалистов, имеющих отношение к ПД; стиль изложения документа; назначение ТЗ; содержание ТЗ; стиль изложения ТЗ; требования к ТЗ; технические требования; приемно–сдаточные испытания и их разновидности; функциональные требования; макетирование и его назначение.

## Прочитать документ требования к ДЗ/КЛР.

Нужно познакомиться с документом “Методические указания к выполнению комплексной лабораторной работы (ДЗ/КЛР)”, размещенному на сайте и/или с документом “Методические указания по ЛР ПКШ”[3]. Их найдете на закрытой страничке со ссылкой “Домашнее задание”. Необходимо составить представление о том, какой объем работ предстоит, и какую работу необходимо сделать в рамках работы над домашним заданием.

## Определить номер своего варианта и получить тему ДЗ.

У старосты, по журналу группы определить свой номер, применительно к данному семестру. Познакомиться и переписать свою тему ДЗ к себе в тетрадь. Варианты тем представлены в данных МУ (ЛР№12) или в отдельном документе вариантов на сайте.

## Познакомиться с требованиями к оформлению документа ТЗ.

Далее необходимо познакомиться с документом “ Методические указания для выполнения ДЗ/КЛР по дисциплине Программирование на основе классов и шаблонов кафедры ИУ5”, он размещен на сайте дисциплины. Особое внимание обратите на перечень документов и на описание документа – Техническое задание. Все полученные документы сохраните для последующей работы.

## Проработать предметную область ДЗ/КЛР.

Выявить содержательно основные объекты, которые будут использоваться для ДЗ [6].

## Выполнить предварительную модульную декомпозицию ПО проекта.

Присвоить названия основным модулям проекта и определить их предварительное назначение [6].

## Познакомиться с образцом документа ТЗ.

Далее необходимо познакомится с образцом документа техническое задание, ссылка на который дана в конце настоящих методических указаний к ЛР №12. Представленный образец ТЗ соответствует теме разобранной в Общих МУ[7] и в МУ по ДЗ/КЛР[6].

(Для тех, кто ранее освоил создание проектов VS и русификацию консольного ввода и вывода хорошо следующие пункты можно пропустить).

## Создать в VS макет консольного проекта ДЗ (DZ\_XXXXX\_ХDD).

Для создания макета консольного проекта необходимо, на основе декомпозиции ПО:

* Запустить систему программирования VS 2005/08/10/12;
* В меню “**File**” выбрать пункт “**New**” и в подменю выбрать позицию “**Project**…”;
* В списке “**Project types**” выбрать “**Visual C++/Win32**”, а в списке “**Templates**” выбрать “**Win32 Console Application**”;
* В поле “Name” ввести: DZ\_XXXXX\_ХDD (где: XXXXX - часть шифра разработки, X – номер группы, а DD – номер варианта по журналу группы текущего семестра. Например, для студента группы ИУ5-22 с вариантом 15 и темой Street – введем – DZ\_Street\_215). Далее нажать “**OK**”;
* В новом окне мастера проектов нажать “**Next**”. Проверить настройки проекта: “**Application Type**” должно быть – “**Console Application**”, “**Additional option**” -> “**Empty Project**” должен быть выключен, “**Add common header files**”-> “**MFC**” и “**ATL**” должны быть включены.
* Далее необходимо нажать кнопку “**Finish**”. Новый проект будет создан.
* Необходимо убрать из главных моделей проекта (DZ\_Street\_ХDD.CPP и DZ\_ Street\_ХDD.H) все лишнее. Этого: в файле DZ\_XXXXX\_ХDD.H (у нас в примере DZ\_Street\_215.H) уберем все, а в файле DZ\_XXXXX\_ХDD.CPP (DZ\_Street\_215.CPP) оставим следующий текст:

#include "stdafx.h"

#include "DZ\_Street\_215.h"

#include <iostream>

using namespace std;

void main(void)

{ … }

* Для контроля правильности создания пустого проекта, нажмем клавишу “**F7**” для проверки создания программы (build) и “**F5**” для проверки ее выполнения (run/debug). Все перечисленные действия должны быть выполнены безошибочно.
* Перечень модулей для создания проекта ДЗ можно уточнить в документе “ Методические указания для выполнения ДЗ/КЛР по дисциплине Программирование на основе классов и шаблонов кафедры ИУ5”, он размещен на сайте дисциплины. Нужно проработать разделы 5, 6 и 7.

## Обеспечить русификацию консольного ввода и вывода.

Для корректного отображения текстов на русском языке и его ввода в окне командной строки (после первого запуска программы) нужно сделать настройки шрифта этого окна. Переключаем шрифт в тип - Lucida Console. Выбираем настройки (после вывода консольного окна на экран, правой кнопкой вызываем системное меню): СВОЙСВА->ШРИФТ -> Lucida Console). После переключения шрифта, на запрос в отдельном окошке нужно выбрать режим – “**Для всех окон с данным именем!**”. Для правильной русификации окна консоли, кроме этого, в самом начале главной программы нужно переключить кодовую страницу для вывода:

system(" chcp 1251 > nul");

Для приостановки завершения программы в консольном окне в конце ее работы можно вызвать паузу следующим образом (например, в конце текста программы):

system(" PAUSE");

На экране появиться следующая строка (смотри ниже) и программа будет ожидать нажатия клавиши:

**Для продолжения нажмите любую клавишу . . .**

Обратите внимание на то, что при другом способе локализации (setlocale(0,"rus");) не все работает правильно. Вывод на консоль и ввод с консоли выполняется правильно, но после этого введенные в консольном окне данные (например, строка) имеют другую кодировку и выводятся неверно! Можете сами это проверить. Поэтому предпочтительно использовать предложенный выше способ с переключением кодовой страницы.

**Примечание**. Если вы затрудняетесь выполнить заданный пункт ЛР, обратитесь к разделу “Основные понятия”, где приведены примеры для иллюстрации данного пункта.

**Примечание**. Все описания классов и шаблонов выполнять в заголовочном файле (у нас в примере DZ\_Street\_215.H). Описания объектов и вызов методов в основном файле: DZ\_XXXXX\_ХDD.CPP (DZ\_Street\_215.CPP).

## Провести макетирование системы классов.

Первоначально нужно описать в DZ\_XXXXX\_ХDD.H базовый абстрактный класс (см. ниже). Напомним, что здесь создается только макет. Окончательное содержание классов и их структура может отличаться от первоначальной структуры.

// Файл - DZ\_XXXXX\_ХDD.H

include "stdafx.h"

#include <iostream>

using namespace std;

// Абстрактный класс для дома

class AbstrHome : public CObject {

public:

virtual int classType() = 0;

virtual char \*className() = 0;

virtual void printOn(ostream &) = 0;

AbstrHome(){ name = (char \*)NULL;

no = NULL;};

~AbstrHome(){

if( name != (char \*)NULL)

delete [] name;

};

char \*name; // ? Резерв Поисковое имя

int no; // ? РезервНомер в списке

};

Далее в скелетную структуру классов (структура с наследованием, названиями классов, но без наполнения) добавим описание классов **Street** и **Home**:

//////////////////////

//// Класс домов

//////////////////////

class Home: public AbstrHome {

// Конструкторы

public:

Home();

Home( Home & H) ;

Home( Home \* pH);

Home(const char \*HomName, const char \*Number);

Home(const char \*HomName, const char \*Number, int Numb) ;

Home(const char \*HomName, const char \*Number, int Numb,

int Etag, int Men=0,HomeType Type = fast,int Apart=0);

// Деструктор

~Home();

//Оператор присваивания для поддержки перегрузки "+"

Home operator =(Home & H );

// Виртуальные методы

public:

virtual int classType() { return HomeClass; }

virtual char \*className() { return "Home"; }

virtual void printOn(ostream & out);

// Методы класса Home

void setName(const char \*HomName , const char \*Number=NULL);

const char \*getName() { return (const char \*)name; };

const char \*getNumb() { return (const char \*)Home\_Number; };

int getNo() { return no; };

void getParam(int & iH, int & Etag ,int & Men ,HomeType & Type, int & Apart )

{iH = iHome; Etag = EtagCount; Men = MenCount;

Type = TypeHome; Apart = NumbApartament ; };

void setParam(int iH, int Etag ,int Men ,HomeType Type, int Apart )

{ iHome= iH; EtagCount= Etag; MenCount= Men;

TypeHome = Type; NumbApartament = Apart; };

void setAllParam(const char \*HomName , const char \*Number, int iH, int Etag ,

int Men ,HomeType Type, int Apart , BOOL rem = false);

// Дружественная функция для перегрузки

friend Home & operator +(Home & H1 , Home & H2);

// Свойства класса Home

public:

char \*Home\_Number; // Символьный номер дома

int iHome; // Номер дома числовой

int EtagCount; // Число этажей

int MenCount ; // Число жителей в доме

HomeType TypeHome ; // Тип дома

BOOL HomeRemont ; // Требуется ли ремонт дома

int NumbApartament; // Число квартир// …

};

//////////////////////

Описание класса Street, можно посмотреть в [6]. На первом этапе создания ТЗ не просто создать полный перечень возможных свойств и методов классов. Первоначально можно их задать в виде комментариев с последующей детализацией. В нашем проекте, в главном модуле проверим описание объектов новых классов: Home и Street. Например, в главной программе (DZ\_XXXXX\_ХDD.CPP) запишем:

system(" chcp 1251 > nul");

// Описание дома

Home H1;

// Описание улицы

Street S1;

system(" PAUSE");

Теперь можно первоначально компилировать программы проекта и проверить сборку.

## Использование шаблона документа ТЗ

В конце данных МУ к ЛР№12 приведен шаблон документа ТЗ. Нужно загрузить шаблон документа и внести необходимые изменения для идентификации студента, группы и индивидуального задания студента по вариантам. Места для изменений помечены красным цветом, а методические указания к его оформлению – синим цветом.

## Разработать разделы документа: Наименование, назначение и исполнитель.

Внимательно проработать разделы документа: наименование, назначение и исполнитель. Методические указания по оформлению этих разделов даны в шаблоне. Шифр каждого проекта у студентов должен быть индивидуальным. Особое внимание обратите на раздел “Назначение”. В этом разделе необходимо проявить все свои знания и “пофантазировать” (спроектировать), придумать назначение и области применения вашей системы классов. Не пугайтесь ничего, этот текст потом изменить можно с полученной новой информацией, при окончательной сдаче ДЗ.

## Разработать раздел документа: Функциональные требования.

Раздел ТЗ функциональные требования (или требования к функциональным характеристикам) при создании ТЗ является самым трудоемким. Необходимо представить какими свойствами и методами должны обладать проектируемые вами классы. Для удобства не закрывайте ваш проект в VS. После осмысления вашего задания изучения материалов примеров из методических указаний и примеров текстов программ (найдете на сайте в разделе ДЗ), вы можете в виде комментариев добавить в проект содержательные названия свойств и методов ваших новых классов. Далее сформулируйте эти названия свойств и методов в виде технических функциональных требований этого раздела ТЗ. Повторю, здесь не нужно приводить латинские идентификаторы (Street и Home), а нужно использовать названия на содержательном уровне (улицы и дома). Например, сформулированные требования могут быть такими (более детально см. образец ТЗ):

### 5.1.5 Обеспечить операцию сложения двух улиц.

### …

### 5.1.9 Учесть свойство названия улицы и возможность ее переименования

### …

Разработанный перечень функциональных требований для вашего варианта необходимо внести в данный раздел ТЗ. Не забудьте, что все пункты ТЗ, а особенно функциональные требования должны быть пронумерованы, иное не допускается.

## Прочитать и разработать раздел документа образец: Требования к ПД.

Необходимо прочитать в образце и шаблоне ТЗ раздел требования к ПД. Проверить в своем документе ТЗ правильность его оформления.

## Прочитать и разработать раздел документа образец: Стадии и этапы разработки.

Необходимо прочитать в образце и шаблоне ТЗ раздел Стадии и этапы разработки. Проверить в своем документе ТЗ правильность его оформления.

## Прочитать и разработать раздел документа образец: Порядок приема и контроля.

Необходимо прочитать в образце и шаблоне ТЗ раздел: “Порядок приема и контроля”. Проверить в своем документе ТЗ правильность его оформления.

## Построить предварительную диаграмму классов для ДЗ.

Необходимо построить предварительную диаграмму классов ДЗ. Пример такой диаграммы смотрите ниже.

## Дополнительные требования для сильных студентов

Для сильных студентов предлагаются дополнительные требования при выполнении ЛР №12 и ДЗ/КЛР. Эти требования могут быть выполнены в любой последовательности и в любом объеме. На титульном листе отчета по ЛР необходимо указать, что дополнительные требования выполнены. Нужно выбрать один из вариантов с дополнительными требованиями (31-36), согласовав свой выбор с преподавателем. Нужно учесть специальные дополнительные требования (по группам), приведенные в разделе вариантов для сильных студентов.

В документе ТЗ необходимо уже на этой стадии разработки учесть дополнительные требования.

Варианты по группам и студентам

Ниже приведены примерные варианты тем заданий для выполнения комплексной лабораторной работы (КЛР) или домашнего задания (ДЗ) по дисциплине “Программирование на основе классов и шаблонов”. Номера тем заданий соответствуют номерам студента по журналу группы на текущий семестр. Номера вариантов заданий меняются ежегодно, поэтому уточнить тему заданий можно в специальном документе, доступном оперативно студентам на сайте дисциплины: [www.sergebolshakov.ru](http://www.sergebolshakov.ru).

| **Номер варианта** | **Тема задания** |
| --- | --- |
|  | Класс публикаций в разных журналах и класс их каталогов этих публикаций. Предусмотреть операции объединения каталогов и сортировки по авторам, журналам, году издания и названиям статей и т.д. |
|  | Класс факультетов и кафедр факультета, предусмотреть операции слияния/разделения кафедр и факультетов, перевода кафедр с одного факультета в другой. |
|  | Класс слов и класс предложений. Предусмотреть операции объединения предложений, добавления, удаления и замены слов на заданных позициях и т.д. |
|  | Класс строк переменной длинны и класс массивов этих строк, предусмотреть операции над строками и массивами строк (объединение и разбиения массивов) и т.д. |
|  | Класс файлов и класс каталогов файлов, предусмотреть поиск по имени файла. Предусмотреть операции перемещения файлов, их добавления и удаления, поиска, переименования, сравнения и объединения каталогов и т.д. |
|  | Класс словарей (русско-английских) и класс гнезд словарей (ячеек словарей), предусмотреть поиск и сортировку. Объединение словарей и разделение по алфавитному принципу и т.д. |
|  | Класс множеств объектов и класс их элементов, предусмотреть весь известный из теории набор операций над множествами: объединение, вычитание, пересечение и др. Операции из дискретной математики. |
|  | Класс групп студентов и класс студентов, предусмотреть операции слияния/разделения групп и редактирование всех видов объектов |
|  | Класс перечней программных продуктов инсталлированных на разных ЭВМ и класс программных продуктов с атрибутами. Предусмотреть возможность объединения списков и замены версий программных продуктов в списке и т.д. |
|  | Класс строительных бригады и класс работников бригад. Предусмотреть объединение и разделение бригад, добавление и удаление работников. |
|  | Класс аннотированных ссылок на ИНТЕРНЕТ ресурсы и класс страниц (перечней), на которых они расположены, с возможностью поиска, предусмотреть операции объединения и очистки страниц и т.д. |
|  | Класс программных продуктов и класс компакт дисков, где они записаны. Предусмотреть возможность слияния компакт дисков, их корректной очистки и разбиения на каталоги (каталог – возможно, это новый класс и объекты) и т.д. |
|  | Класс стеллажей книг в библиотеке и класс книг. Предусмотреть объединение разделение стеллажей, добавление и удаление книг. |
|  | Класс таблиц баз данных и класс записей в таблице (реляционная таблица). Предусмотреть класс связей между таблицами. |
|  | Класс наборов компьютеров в локальной сети, класс компьютеров и класс их связей. Предусмотреть операции объединения и разделения наборов компьютеров сети, добавления, удаления и замены компьютеров в списках, изменения соединений, структуры сети и т.д. |
|  | Класс изображений и класс слайдов презентаций (последовательностей слайдов). Изображения не выводятся на экран, а распечатываются в поток с названиями и характеристиками. Предусмотреть возможность включения вложенных презентаций. |
|  | Класс отделов сотрудников и класс сотрудников, предусмотреть операции приема на работу, увольнения, изменение окладов и должностей. Операции объединения и разделения отделов, подсчета фонда зарплаты и средних характеристик отдела: стаж, возраст зарплата и т.д.. |
|  | Класс перечня (списка) блоков занятой и свободной оперативной памяти и класс отдельных блоков, предусмотреть сборку мусора объединения списков памяти, объединения и разделения блоков памяти и т.д. |
|  | Класс комплектующих компьютера и класс их наборов (перечней комплектующих). Предусмотреть операции замен комплектующих по типу и множественность некоторых элементов компьютеров (например, диски). Предусмотреть очистку набора и изменение типа комплектующих элементов. |
|  | Класс документов и класс папок с множеством документов, предусмотреть операции слияния папок, добавления и удаления из них документов. |
|  | Классы заголовков текста в документах и классы документов. Предусмотреть возможности объединения документов. Документа на экран не выводятся. В заголовках должна быть указана позиция расположения в документах, они могут иметь многоуровневый вид (не менее 3-х уровней заголовков). |
|  | Класс карточек учета товаров и класс картотек с поиском по названию товара, фирмы их продающих. Учесть даты их поступления. Предусмотреть операции добавления карточек, слияния фирм и ох очистки. |
|  | Класс информации о различных событиях и класс их наборов (перечней событий). Обеспечить поиск по контексту названия, интервала дат и типу событий, которых должно быть не менее пяти. Объединение списков событий и т.д. |
|  | Класс звуковых сигналов (нот) и класс мелодий из них, предусмотреть операции слияния мелодий и включения новых звуков в них (необязательно проигрывать мелодии и звуки, достаточно давать их буквенное обозначение) |
|  | Класс очередей задач к ресурсам операционной системы и класс задач. Задача рассматривается как элемент с различными атрибутами (требуемая ОП, время счета и т.д.). Очередь организуется в режиме FIFO. Предусмотреть операции объединения очередей и изменения последовательности расположения задач в зависимости от характеристик и т.д. |
|  | Класс окон интерфейса и класс управляющих элементов в этих окнах (кнопки, поля и т.д.). Сами окна и элементы интерфейса не выводятся на экран. Имитируется распечатка списка элементов и окон. Предусмотреть операции объединения окон и проверки корректности расположения элементов (отсутствие наложения друг на друга, нахождения в рамке окна и т.д.). |
|  | Класс векторов и класс их списков (двумерные вектора), предусмотреть операции над векторами и списками векторов (сложение и т.д. из математики) |
|  | Класс магазинов и класс товаров, которые продаются в данном магазине. Предусмотреть завоз новых товаров, продажу товаров. Объединение магазинов и выделение филиалов с товарами. |
|  | Класс списка литературы и класс элементов списка литературы. Предусмотреть возможность объединения списков, их сортировки по разным критериям и распечатки и т.д. |
|  | Класс мультимедиа объектов (звуки, рисунки, текст и т.д.) и класс их хранилищ ("холста" для рисунков). Объекты не воспроизводятся, нужно предусмотреть стандартный вывод названий и свойств. Предусмотреть операции объединения и расслоения холстов по введенным признакам, например номер слоя. |
| 1. (Д.т.) | Класс двунаправленных списков и их произвольных элементов: строк, дат и чисел (нужно использовать указатели). Операции со списками их сортировки и их объединения. Создать собственные классы, а не копировать из VS и т.д. Необходимо придумать содержательный контейнерный класс для наследования от двунаправленных списков и элементный класс, объекты которого включаются в него. |
| 1. (Д.т.) | Классы таблиц (строки и столбцы – произвольной размера) и их содержимого в виде текстовых ячеек. Операции объединения текста в ячейках, добавление текста, обмена ячейками и т.д. |
| 1. (Д.т.) | Классы векторов и классы массивов векторов (трехмерные вектора) предусмотреть операции над векторами и массивами векторов (сложение и т.д. из математики) |
| 1. (Д.т.) | Классы массив целых чисел и классы их массивов (массивов указателей для массивов), предусмотреть операции над массивами и числами: сложения, вычитания и объединения. |
| 1. (Д.т.) | Класс баз данных и класс таблиц баз данных (реляционная таблица). Предусмотреть класс записей в таблицах. |
| 1. (Д.т.) | Класс баз данных и класс таблиц баз данных (реляционная таблица). Предусмотреть класс связей между таблицами. |

**Примечание:** Для того чтобы задания каждого студента были индивидуальными введены специальные требования для каждой из групп ( 2014/2015 учебный год):

**гр. ИУ5-21** - обязательным является перегрузка операции вычитания ("-") двух контейнерных классов для получения нового объекта контейнерного класса, содержащего только те элементы, которые отсутствуют во втором контейнером класса. (Пример. создание новой улицы). Для описания контейнерного класса необходимо использовать базовый класс **CObArray**.

**гр. ИУ5-22** - обязательным является перегрузка операции ввода (**>>**) из стандартного потока **cin** для элементного класса. (Пример. Класс домов). Для описания контейнерного класса необходимо использовать базовый шаблонный класс **CArray**.

**гр. ИУ5-23** – обязательным требованием является использование дополнительной, перегруженной операций. Операция выполняет удаление объектов для контейнерного класса (использовать знак "-" - минус). (Пример. Удаление дома с улицы). Для описания контейнерного класса необходимо использовать базовый шаблонный класс **CList**.

**гр. ИУ5-24** - обязательным является перегрузка операции вывода (**<<**) в стандартный поток **cout** для элементного класса. (Пример. Класс домов. Смотрите документ требований к КЛР). Для описания контейнерного класса необходимо использовать базовый класс **CObList**.

**Дополнительные требования для сильных студентов**

Для сильных студентов предлагается более сложный вариант задания. Во-первых, можно выбрать в таблице вариантов более сложный вариант задания (д.т.), согласовав выбранную тему с преподавателем.

Во-вторых, в стандартном задании можно выполнить следующие дополнительные требования:

1. Создать собственные классы для двунаправленных списков (или специальных динамических массивов) и использовать их для наследования в ваших тематических контейнерных классах.
2. Выполнить оригинальную перегрузку операций ввода/вывода в поток(**<< , >>**). Это нужно сделать и для элементного и контейнерного классов задания.
3. Использовать бинарные файлы для хранения информации из контейнерных классов. Предусматривается загрузка и выгрузка из файлов из/в ОП. Возможна выборка объектов по номеру в контейнере и/или по условиям, наложенным на атрибуты элементных объектов.
4. Использовать динамически создаваемые объекты, включаемые в контейнер, с контролем создания и удаления объектов и их составляющих (например, для строковых полей класса). Предусмотреть возможность удаления и сохранения объектов после выборки из контейнера.

Студент может выбрать одно или несколько дополнительных требований и указать их содержание на титульных листах документов ДЗ. Выполнение дополнительных требований может быть полезно в учебном плане для тренировки программиста, а также учитывается при подведении итогов семестра по данной дисциплине (автоматы или значение отметки).

**Примечание**: Для утверждения дополнительных требований и дополнительных тем нужно(31-36) их нужно выбрать и согласовать с преподавателем в начале семестра. На титульных листах документов должен быть отмечен факт выполнения дополнительных требований и их содержания.

Предварительные диаграммы классов домашнего задания

Для классов домашнего задания необходимо построить предварительную диаграмму классов. Для классов нашего примера получим:

Контрольные вопросы к ЛР



1. Для чего нужна программная документация?
2. Что включается в комплект ПД для ДЗ/КЛР?
3. Для чего нужно ТЗ и его назначение?
4. Каково содержание документа ТЗ?
5. Какие группы специалистов участвуют в разработке ТЗ, другой документации?
6. Какой стиль должен быть у документа ТЗ?
7. Каковы основные требования к тексту документа ТЗ?
8. Что включается в раздел ТЗ технические требования?
9. Какие разновидности приемно-сдаточных испытаний вы знаете?
10. Что включается в раздел ТЗ функциональные требования?
11. Как разрабатывается раздел ТЗ функциональные требования, в чем особенность?
12. Для чего нужно макетирование программных продуктов?
13. В чем состоит противоречие между заказчиком и разработчиком при разработке ТЗ?
14. Кто разрабатывает ТЗ?
15. Какие принципы разработки документов ПД вы знаете?
16. Какие требования предъявляются к разработке документов (общие)?
17. Какие требования к оформлению документа ТЗ предложены для ДЗ/КЛР?
18. Как можно изменить ТЗ после его утверждения?
19. Почему необходима нумерация пунктов ТЗ?
20. Какой из разделов ТЗ является самым сложным для разработки и почему?
21. Какой способ приемки программного продукта предложен в ДЗ?

Сроки и порядок защиты документа ТЗ

Сроки выполнения ДЗ устанавливаются в техническом задании (14 неделя). ДЗ считается выполненным и защищенным при предоставлении работоспособной программы, полного комплекта ПД и успешного проведения приемно-сдаточных испытаний по ПМИ. Студенты, не выполнившие работу самостоятельно, не могут рассчитывать на экзамене на оценку выше тройки. При защите ДЗ студенты должны иметь все необходимое для выполнения сборки исполнимого модуля и самостоятельного внесения изменений в проект и его модули по требованию преподавателя.

Студенты, выполнившие ЛР и ДЗ самостоятельно и с дополнительными требованиями, имеют возможность получить автоматическую оценку на экзамене (разумеется, ОТЛИЧНО).

Сдача ДЗ/КЛР на экзамене не допускается. Результаты выполнения ДЗ значительно влияют на итоговую оценку по дисциплине в семестре. Студенты, не выполнившие ДЗ/КЛР, к экзамену не допускаются.

Литература

1. Г. Шилдт “С++ Базовый курс”: Пер. с англ.- М., Издательский дом “Вильямс”, 2011 г. – 672с
2. Г. Шилдт “С++ Руководство для начинающих” : Пер. с англ. - М., Издательский дом “Вильямс”, 2005 г. – 672с
3. Г. Шилдт “Полный справочник по С++”: Пер. с англ.- М., Издательский дом “Вильямс”, 2006 г. – 800с
4. Бьерн Страуструп "Язык программирования С++"- М., Бином, 2010 г.
5. MSDN Library for Visual Studio 2005 (Microsoft Document Explorer – входит в состав дистрибутива VS. Нужно обязательно развернуть при установке!)
6. Большаков С.А. - “Методические указания для выполнения ДЗ/КЛР по дисциплине Программирование на основе классов и шаблонов кафедры ИУ5”.(см. на сайте 1-й курс [www.sergebolshakov.ru](http://www.sergebolshakov.ru))
7. Большаков С.А. “Общее методическое пособие по курсу для выполнения ЛР “(см. на сайте 1-й курс [www.sergebolshakov.ru](http://www.sergebolshakov.ru)) – см. кнопку в конце каждого раздела сайта!!!
8. Г.С.Иванова, Т.Н. Ничушкина, Е.К.Пугачев "Объектно-ориентированное программирование". – М., МГТУ, 2001 г.
9. Другие методические материалы по дисциплине с сайта [www.sergebolshakov.ru](http://www.sergebolshakov.ru).
10. Конспекты лекций по дисциплине “Программирование на основе классов и шаблонов”.
11. Страуструп Б. "Дизайн и эволюция С++. Классика CS" – СПб,: Питер , 2007. – 445с.

Приложение: Шаблон документа ТЗ

Данный документ представлен для разработки собственного технического задания по предложенному варианту (список группы и уточнения для группы). Красным цветом в документе помечены фрагменты и разделы текста, которые необходимо переработать полностью или изменить. После изменения необходимо установить стандартный для документа цвет (только черный). Синим цветом помечены методические указания для выполнения пункта задания. После внимательного прочтения и использования этот текст из итогового документа должен быть удален.



Приложение: Образец документа ТЗ (\*.doc)

Данный документ является примером оформления ТЗ для варианта рассмотренного в методических указаниях по курсу [6]. Его необходимо прочитать предварительно, перед разработкой собственного документа. Не нужно его копировать и представлять при сдаче собственной работы!



Приложение: Образец документа ТЗ (\*.pdf)

Данный документ является примером оформления ТЗ для варианта рассмотренного в методических указаниях по курсу [6]. Его необходимо прочитать предварительно, перед разработкой собственного документа. Не нужно его копировать и представлять при сдаче собственной работы!



Справочные материалы

Пока в этом разделе не размещена информация.