# 1. Техническое задание

## 1.1. 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 (у нас -преподаватель) может ТЗ не утвердить. Это очень рискованно. ТЗ является, кроме того, основой для самой разработки и основой для разработки других документов.

## 1.2. 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.

## 1.3. Требования к ТЗ

Документ ТЗ является важнейшим документом для проектирования и реализации проектов. В частности, проведение приемно-сдаточных испытаний программного продукта основывается на ТЗ (см. пояснения к ЛР по “ Программе и методике испытаний”). Поэтому все пункты требований к программному продукту 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 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.

## 1.4. 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

8. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

8.1. Сроки выполнения отдельных этапов работ

9. 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. ), чтобы обеспечивалась возможность ссылки на них из других документов.

В разделах п.7 и п.10 информация может отсутствовать при выполнении заданий ЛВП.

 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.).

## 1.5. 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 приемо-сдаточных испытаний. В главный раздел могут выноситься требования общего характера, которые должны быть выполнены (Смотрите шаблон и образец ТЗ).

#### 1.5.1. 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).

####  1.5.2. 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 (ЛВП) 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).

##  1.6. 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).

## 1.7. 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 работу над заданием необходимо начать как можно раньше, чтобы разработать ТЗ в сроки, определенные в программе ЛВП.

##  1.8. 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 (демонстрация задания 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.
(Смотрите шаблон и образец ТЗ).

## 1.9. 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.

#  2. Программа и методика испытаний

 Документ “Программа и методика испытаний” (ПМИ) разрабатывается специально для проведения приемно-сдаточных испытаний для сдачи ПО. В нашем случае этот документ разрабатывается для проведения испытаний на основе пунктов ТЗ. Данный документ ориентирован на заказчика, который будет выполнять приемку программного продукта. Возможны различные варианты построения документа (ПМИ), который устанавливается по соглашению с заказчиком:
- приемка на основе проверки выполнения пунктов ТЗ (п. 5.1 – требования к функциональным характеристикам в первую очередь);
- приемка, ориентированная на проверку работоспособности системы;
- комбинированная приемка по ТЗ и работоспособности и т.д.

 Для нашего случая используется вариант приемки на основе пунктов ТЗ и проверки работоспособности системы, поэтому, отметим еще раз, что в ТЗ все позиции должны быть пронумерованы, для того, чтобы на них можно было ссылаться в ПМИ. Документ ПМИ должен быть фактически пошаговой инструкцией для проведения испытаний. Специалист, принимающий испытания (представитель заказчика), должен без подсказок разработчика самостоятельно выполнить действия по проверке и удостоверится в работоспособности программного продукта и выполнения всех пунктов ТЗ.

## 2.1. Стиль изложения документа ПМИ

Стиль изложения должен быть предписывающим, и основан на техническом языке, принятом программистами и специалистами по разработке ПП (программистский жаргон здесь недопустим). Недопустимо, также, использование бытовых терминов. Все тексты должны трактоваться однозначно. Например, "нажмите … клавишу", "получите на экране …", "испытания должны проводиться в режиме командной строки для CMD.EXE" и т.д. Неоднозначностей и ошибок в тексте документа ПМИ не должно быть. Данный документ должен быть лаконичным и четким. Не нужно строить сложные предложения.

## 2.2. Разработка документа ПМИ

Разработка документа ПМИ выполняется в конце всей разработки программного продукта специалистами по программированию и заказчиком, но ее план задумывается уже при составлении ТЗ на программное обеспечение. Перед разработкой ПМИ должен быть подготовлен специальный тестовый пример, ориентированный для проверки пунктов ТЗ. Нужно иметь в виду, что при разработке ТЗ вы уже должны понимать, как могут быть проверены эти пункты в документе ПМИ. Для проверки правильности документа необходимо провести испытания самостоятельно, использую в роли заказчика представителя не связанного с вашим проектом.

## 2.3. Содержание документа ПМИ

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

Содержание документа ПМИ:

1. ОБЪЕКТ ИСПЫТАНИЙ

2. ЦЕЛЬ ИСПЫТАНИЙ

3. СОСТАВ ПРЕДЪЯВЛЯЕМОЙ ДОКУМЕНТАЦИИ

 4. ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ К ИСПЫТАНИЯМ И УСЛОВИЯ ПРОВЕДЕНИЯ ИСПЫТАНИЯ
 4.1. Требования к условиям проведения испытаний
 4.2. Требования к техническим средствам

5. ПОРЯДОК ПРОВЕДЕНИЯ ИСПЫТАНИЙ
 5.1. Состав и структура технических и программных средств для проведения испытаний программного продукта.
 5.2. Последовательность испытаний (в виде таблицы)

6. МЕТОДЫ ИСПЫТАНИЯ

7. РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

## 2.4. Требования к документу ПМИ

 Главные требования к основным разделам документа “Программы и методики испытания” (на выполнение этих требований будет обращаться повышенное внимание при сдаче ПО) следующие:
 В п.1 (“ОБЪЕКТ ИСПЫТАНИЙ”) описывается, что будет испытываться, включая и наименование программного изделия. В учебных целях здесь, не допускаются ссылки на другие документы проекта, особенно при задании условий проведения испытаний.

 В п.2 (“ЦЕЛЬ ИСПЫТАНИЙ”) устанавливается цель испытания, то есть фактически устанавливается вариант проведения приемно-сдаточных испытаний программного изделия.

 В п.3 (“СОСТАВ ПРЕДЪЯВЛЯЕМОЙ ДОКУМЕНТАЦИИ”) приводится полный перечень представляемой документации для проведения испытаний.

 В п.4 (“ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ К ИСПЫТАНИЯМ”) определяются условия проведения испытаний, включая и технические (OC, требования к компьютеру и т.д.). При формировании условий проведения испытаний необходимо учитывать возможности заказчика (преподавателя), которые он может предоставить для проведения приемки. Сдача программного продукта на технических и программных средствах разработчика недопустима.

 п.5 (“ПОРЯДОК ПРОВЕДЕНИЯ ИСПЫТАНИЙ”)
В пункте 5.1 определены конкретные условия проведения испытания для проверки.
В пункте 5.2 в виде таблицы, формат которой приведен ниже, дается перечень действий и результатов, необходимых для проведения испытаний:

№ п.п. | № пункта ТЗ | Выполняемые действия | Ожидаемый результат

Содержание строки таблицы для проверки функционального требования ТЗ «Должно быть обеспечено создание объекта типа дом»:
 Выполните пункт меню № 1.
 Введите “1” и нажмите ENTER.
 ( Выполнится фрагмент тестового примера, где создается объект. См. п 2.1 "Описание тестового примера" – Home H1("Жилой", "д.10", 7,2,10, multiple , 8, false);)
 На экране будет распечатано содержимое объекта:

 Имя – Жилой, д.10 Номер - 7 Этажей - 2 Жителей - 10 Ремонт не нужен!
Тип дома - много строений Число квартир – 8

 (Примечания для студентов: в ячейке таблицы отображаются все результаты, которые проверяющий должен проверить – детальный вывод на экран и другие результаты).

 … … … … …

…

Завершение программ

… …

Примечания: 1. В колонке 2 можно не раскрывать полностью текстовое содержание пункта ТЗ, но желательно это сделать. Ссылка на пункт ТЗ обязательна. Номера по порядку в таблице должны соответствовать логически выделенным шагам проведения испытаний. Они не должны объединять одновременно несколько пунктов ТЗ.

2. Выполняемые действия должны быть точными, расписанными в точности до нажатия отдельной клавиши.

3. Если результат трудно поместить в ячейке данной таблицы, то его нужно разместить в конце данного документа (в разделе результаты испытаний), а в таблице дать ссылку на раздел и страницу, где размещены эти результаты.

4. Каждая строка таблицы проверки пунктов ТЗ должна содержать описание действий для выборочной проверки этого пункта. Поэтому нужно продумать все необходимое и возможно продублировать действия других пунктов или выполнить ссылку на них (по пункту в колонке №1).

 В п.6 (“МЕТОДЫ ИСПЫТАНИЯ”) в данном разделе описываются специальные методы, которые используются при проведении испытаний (Например, способы измерения или измерительные инструменты). Рассматриваются методики проведения испытаний, которые трудно отобразить в таблице п.5 (например, выводимые результаты на экран, сложные эксперименты для проверки работоспособности и т.д.).

В этом случае пункты должны быть пронумерованы, а в таблице ПМИ будут сделаны ссылки на эти пункты.

 В п.7 (“РЕЗУЛЬТАТЫ ИСПЫТАНИЙ”) размещаются результаты испытаний с возможностью ссылки на них из таблицы испытаний. Они даются с стиле рисунков или текста результатов, полученных при проведении испытаний.

# 3. Вопросы для самопроверки

1. Дайте краткое определение понятия класса.

2. Дайте развернутое определение понятия класса.

3. Что такое конструктор и для чего они нужно?

4. Что такое деструктор и для чего они нужны?

5. Что такое инкапсуляция?

 6. Дайте определение понятия наследования.

7. Что такое базовый и производный классы?

8. Что такое множественное наследование?

9. Какие изменения в порожденном классе можно сделать при наследовании?

10. Чем различаются классы и структуры данных?

11. Что такое перегруженные операции в классе?

12. Что такое статическое связывание?

13. Что такое динамическое связывание?

14. Какие виды перегрузки операций Вы знаете?

15. Каково назначение перегрузки операций с точки зрения программиста?

16. Что такое контейнерный объект?

17. Какие разновидности контейнеров вы знаете?

18. Какие операции выполняются с контейнерами?

19. Какие операции с массивами вы знаете?

20. Перечислите основные методы класса vector.

21. В каком отношении элементный класс находиться с контейнерным классом?

22. Какие признаки классификации контейнеров вы знаете?

23. Что такое позиция? Как выполняется навигация по спискам.

24. Что такое итератор и для чего он используется?

25. Какие операции со списками вы знаете?

26. Перечислите основные методы класса stack.

27. Для чего нужна программная документация?

28. Что включается в комплект ПД для ЛВП?

29. Кто разрабатывает ТЗ?

30. Какой из разделов ТЗ является самым сложным для разработки и почему?

31. Для чего нужен документ «Описание применения» (ОП )?

32. Каково содержание документа ОП?

33. Каково содержание документа «Техническое описание» (ТО)?

34. Почему необходима нумерация пунктов ТО?

35. Для чего нужен документ «Руководство пользователя» (РП) и его содержание?

36. Для чего нужен документ «Руководство системного программиста» (РСП) и его содержание?

37. Для чего нужен документ ПМИ и его содержание?

38. Каковы основные требования к тексту документа ПМИ?

39. Какой из разделов ПМИ является самым сложным для разработки и почему?

40. Что такое предметная область?

41. Как формируются функциональные требования к решаемой задаче.

4. Литература по курсу ПКШ

 1. Г. Шилдт “С++ Базовый курс”: Пер. с англ.- М., Издательский дом “Вильямс”, 2011 г. – 672с

2. MSDN Library for Visual Studio 2012

3. Общее методическое пособие по курсу для выполнения ЛР и КЛР/ДЗ (см. на сайте 1-й курс www.sergebolshakov.ru) – см. кнопку в конце каждого раздела сайта!!!

 4. Г.С.Иванова, Т.Н. Ничушкина, Е.К.Пугачев "Объектно-ориентированное программирование". – М., МГТУ, 2001 г.

8. Другие методические материалы по дисциплине с сайта www.sergebolshakov.ru. 9. Конспекты лекций по дисциплине “Программирование на основе классов и шаблонов”.

 6. Глаголев В.А. Разработка технический документации. Руководство для технических писателей и локализаторов ПО – СПб.: Питер, 2008. – 192с.:ил. ISBN 978-5-388-00101-6

7. Эккель Б. Философия С++. Введение в стандартный С++. 2-е изд.- СПб.: Питер, 2004.- 572с.: ил.

8. Страуструп, Бьярн. Программирование: принципы и практика с использованием С++, 2-е изд. : Пер. с англ. - М . : ООО "И . Д. Вильяме", 2016. - 1328с. : ил .