

Методические рекомендации по выполнению курсовой работы по дисциплине «Технологии индустриального программирования»

О курсовой работе

Курсовая работа представляет собой одну из форм семестровой работы студентов в рамках прохождения дисциплины. В данном случае дисциплины «Технологии индустриального программирования».

По итогам выполнения курсовой работы **ОБЯЗАТЕЛЬНО** необходимо подготовить отчёт, содержащий основные пункты работы (о них позднее), которые будут оформлены в соответствии с правилами оформления работы (приведены в конце рекомендаций).

В качестве темы курсовой работы можно взять разработку любого программного продукта на усмотрение студента. Каждая тема должна быть утверждена в соответствующем порядке защиты тем. Возможно выполнение совместного проекта, но в таком случае каждый студент внутри группы выполняет отдельную полноценную разработку части системы продукта.

В пределах методических рекомендаций предусмотрен выбор категорий тем работ:

- базовый уровень: моделирование и решение квадратных уравнений с заданными коэффициентами и сравнение различных методов решения квадратных уравнений;
- продвинутый уровень: моделирование работы конвейерной системы и промышленного логического контроллера технологического процесса;
- тема по выбору: студент самостоятельно определяет тему работы.

Важно, что при создании продукта значительная основная часть разработки должна быть на языке C++ с использованием принципов объектно-ориентированного программирования.

Части подготовки курсовой работы в разрезе практических занятий:

Практика	Основное содержание практики/ Часть подготовки к курсовой работе
1	Вводное занятие. Система требований. Знакомство с методическими материалами. Объёмы и сроки поставленных задач.
2	Выбор тем курсовых работ, формирование рабочих групп, постановка задач курсовой работы и технического задания на продукт.
3	Теоретическая часть курсовой работы. Обзор существующих продуктов-аналогов или продуктов смежного функционала.

4	Формирование решения, описание решения, алгоритм работы продукта (диаграмма процессов). Презентация темы курсовой работы.
5	Определение функциональных частей системы, схема объектов продукта (диаграмма классов). Презентация темы курсовой работы.
6	/Самостоятельная разработка продукта, ответы на возникающие вопросы/
7	/Самостоятельная разработка продукта, ответы на возникающие вопросы/
8	/Самостоятельная разработка продукта, ответы на возникающие вопросы/
9	/Самостоятельная разработка продукта, ответы на возникающие вопросы/
10	/Самостоятельная разработка продукта, ответы на возникающие вопросы/
11	/Самостоятельная разработка продукта, ответы на возникающие вопросы/
12	/Самостоятельная разработка продукта, ответы на возникающие вопросы/
13	/Самостоятельная разработка продукта, ответы на возникающие вопросы/
14	Предпоказ продукта
15	Предпоказ продукта
16	Предпоказ продукта

Балльная система

Во время подготовки в семестре и сдачи курсовой работы будет применена 100-балльная система оценивания. В зависимости от полученных баллов и формата защиты будет выставлена соответствующая оценка по итогам защиты курсового проекта.

Пункты будут оцениваться в своё количество баллов, включая объём выполнения данного пункта по системе 125% - 100% - 75% - 50% - 25% - 0%. Возможность получения 125% возможна при своевременной сдаче каждого из пунктов подготовки курсовой работы.

Важным моментом является то, что получение положительной оценки (выше неудовлетворительно) возможна только в случае разработки РАБОТАЮЩЕГО программного продукта. В остальном, оценка зависит только от раскрытия каждого из пунктов.

Обратите внимание, что полностью готовый проект можно показать впервые и в день сдачи курсовой работы, но в случае неудовлетворительного результата возможность исправить и представить готовый вариант будет уже в момент пересдачи.

Закрытые и проверенные пункты на сдаче проверяться не будут, поэтому одной из выгодных стратегий является сдача пунктов курсовой в рекомендуемые сроки.

Таблица пунктов оценивания и баллов:

№	Описание	Баллы	Сроки для повышенных баллов	Если не выполнить нельзя получить оценку выше
1	Тема работы	1	Практика 5	неудовл.
2	Актуальность работы	2	Практика 5	удовл.
3	Цель работы	1	Практика 5	неудовл.
4	Задачи работы	3	Практика 5	удовл.
5	Критерии достижения успеха	10	Практика 5	отлично
6	Обзор области	10	Практика 5	удовл.
7	Обзор 3 решений	10	Практика 5	отлично
8	Схема работы	15	Практика 7	хорошо
9	Архитектура классов	15	Практика 8	хорошо
10	Разработка продукта	15	Практика 16	неудовл.
11	Тестирование	15	Практика 16	неудовл.
12	Заключение	2	Практика 16	неудовл.
13	Список литературы	1	Практика 16	удовл.

Для повышения баллов, заработанных в семестре необходимо отвечать на защите в день представления курсовой работы. При этом **НАБРАННЫЕ БАЛЛЫ НЕ ЯВЛЯЮТСЯ ГАРАНТИЕЙ ПОЛУЧЕНИЯ ОЦЕНКИ**, в случае, если не удастся ответить на вопросы по работе, что может говорить о возможной попытке сдать не свою работу, либо прочих махинациях, противоречащих Уставу Университета.

Диапазон	Предварительная оценка
от 90 и выше	отлично
от 75 до 90	хорошо
от 60 и выше	удовлетворительно
до 60	неудовлетворительно

Необходимые пункты

Курсовая работа должна содержать следующие пункты и их разновидности:

1. Введение:
 - 1.1. Тема работы.
 - 1.2. Актуальность работы.
 - 1.2. Цель работы.
 - 1.3. Задачи работы.
 - 1.4. Критерии достижения успеха.
2. Теоретическая часть:
 - 2.1. Обзор области разработки.
 - 2.2. Обзор существующих продуктов.
3. Технологическая часть:
 - 3.1. Схема работы системы.
 - 3.2. Архитектура классов продукта.
4. Практическая часть:
 - 4.1. Разработка продукта.
 - 4.2. Тестирование.
5. Заключение.
6. Список литературы.
7. Приложения.

Каждый из пунктов будет рассмотрен далее более подробно в разрезе подготовки конкретной темы работы.

Введение

В введение следует раскрыть предпосылки к разработке продукта, то есть **основную идею**, раскрыть **актуальность работы**, поставить **цель работы**, определить **задачи** и **критерии достижения успеха**.

Задачи для каждой курсовой работы должны соответствовать конечной цели, которую можно сформулировать по шаблону:

Цель работы - разработка программного продукта «Название продукта» на языке C++.

Задачи:

1. Провести библиографический обзор источников по теме работы.
2. Выделить от 3 существующих решений, соответствующих поставленной задаче, либо близких по смыслу. Выделить плюсы и минусы каждого из них.
3. Составить алгоритм работы программного продукта в виде ... (например, диаграммы процессов).
4. Составить иерархию объектов системы в виде ... (например, диаграммы классов).
5. Разработать программный продукт на языке C++.
6. Протестировать работу продукта.

7. Составить отчёт по выполненной работе в соответствии с требованиями и защитить его.

Важно отметить, что важным фактором при оценке курсовой работы будет являться макет технического задания на продукт, который должен быть подготовлен и представлен в упрощённом виде с помощью соответствующей таблицы критериев, состоящей из **10 основных пунктов**.

Например:

№	Критерий	Значение
1	Вариативность входных параметров	от -2143967 до +2143967
...
10	Время работы	Не более 0,5 с

Такая таблица позволит не только объективно оценить выполненный проект, но и послужит в качестве основы для проведения тестирования продукта и самоконтроля выполнения его пунктов.

Данный пункт можно сдать предварительно не позднее 5 практического занятия для получения дополнительных повышенных баллов, либо сдать до сдачи курсовой для предварительной проверки.

Оценка пунктов	Срок
до 125%	не позднее 5 практического занятия
до 100%	не позднее дня сдачи курсовой работы

Презентацию тем курсовых работ необходимо предоставить к пятому практическому занятию (либо в соответствии с таблицей задержек), к которым необходимо в виде презентации представить следующие пункты:

- цель курсовой работы;
- задачи курсовой работы;
- техническое задание в виде таблицы из 10 пунктов;
- краткий теоретический обзор темы работы.

Типовые темы курсовых работ

Как было указано ранее, в качестве темы курсовой работы можно взять базовый уровень работы: моделирование и решение квадратных уравнений с заданными коэффициентами и сравнение различных методов решения квадратных уравнений. Таблица с типовыми вариантами в конце методических рекомендаций.

Продвинутый вариант работы: моделирование работы конвейерной системы и промышленного логического контроллера технологического процесса. Таблица с типовыми вариантами в конце методических рекомендаций.

Теоретическая часть

Теоретическая база для курсовой работы представляет собой первую главу курсовой, в которой рассматривается, как область внедрения продукта, так и существующие продукты аналоги.

Для библиографического обзора необходимо рассмотреть область внедрения продукта и вкратце рассказать о ней (не менее 2 страниц). Необходимо представить краткий обзор 3 существующих продуктов, решающих такую же задачу, либо похожую. Выделить плюсы и минусы каждого решения и сравнить их между собой.

Технологическая часть. Схема работы

Технологической части работы будут посвящены 3 и 4 практическое занятие. Можно воспользоваться как примерами далее, так и самостоятельно описать систему и алгоритм её работы.

Для обеспечения качественной, эффективной и, самое главное, наглядной передачи основных процессов, за счёт которых работает тот или иной программный продукт, прибегают к различного рода интерпретациям и описаниям происходящих процессов. Одни могут быть более формализованы, чем другие, но каждый способ отображения обладает своими ключевыми особенностями, о которых не следует забывать при проектировании и разработке программных продуктов.

Основные способы (подходы) к отображению процессов в системе:

- текстовый (словесный), когда с помощью языка описывается работа системы;
- графический, когда схема работы и процессы описываются с помощью графической интерпретации данных процессов (от формализованных по типу BPMN, IDF и прочих до абстрактных систем на усмотрение автора);
- знаковый, как ответвление от графического, где каждый процесс и/или его особенность отображаются в строгом соответствии с некоторой системой отображения.

Зачастую комбинируют несколько способов одновременно для того, чтобы сохранить возможность отображения отдельных аспектов продукта. Например, словесное описание, предусмотренное ГОСТом на техническое задание охватывает огромное количество аспектов разработки и особенностей программного продукта, которые зачастую нельзя отобразить с помощью одного изображения.

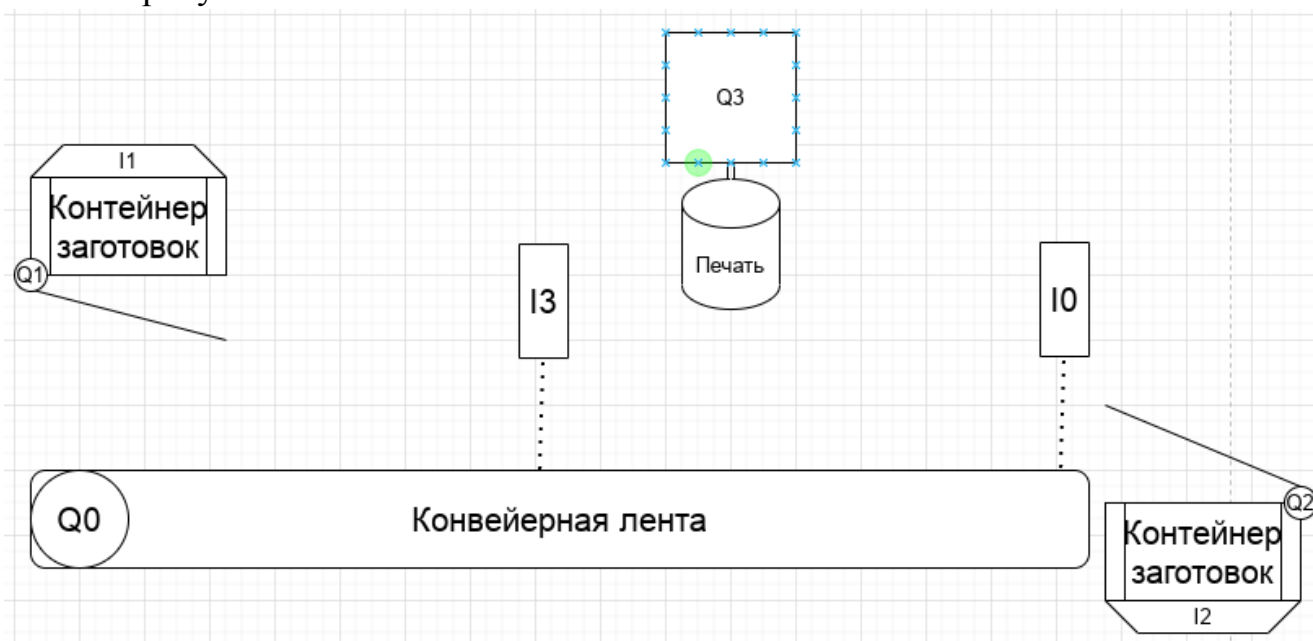
В рамках работы в данном семестре мы с вами будем пользоваться комбинированным способом, сочетая словесное описание, графическое (для отображения фактического расположения частей системы) и диаграмму состояний (формата, близкого к конечным автоматам).

Практические задания данного занятия можно выполнять по одной из трёх основных траекторий:

1. Задание общего варианта, которое соответствует базовому уровню выполнения задания на курсовую работу.
2. Задание общего варианта, которое соответствует продвинутому уровню выполнения задания на курсовую работу.
3. Задание по собственному варианту.

В качестве основного автоматизируемого процесса будет выступать технологический процесс автоматической маркировки товара, которая обеспечивается работой промышленного логического контроллера.

Итак, в одном из отделов технологического производства располагается автоматическая установка, которая представляет из себя конвейерную ленту с электроприводом (Q0), в начале ленты располагается контейнер с заготовками, заслонка которого открывается при помощи привода (Q1), о наличии заготовок в контейнере сообщает датчик (I1). О том, что заготовка дошла до края ленты сообщает датчик (I0). В конце ленты располагается контейнер для маркированных заготовок, с крышкой на электроприводе (Q2) и датчиком наполнения (I2). В середине ленты располагается автоматический станок печати (опускается оттиск и печать остаётся на заготовке) с приводом (Q3), перед местом печати на ленте располагается датчик приближения заготовки (I3) на достаточном расстоянии, чтобы успел опуститься оттиск за время движения товара, а скорости достаточно, чтобы ленту не нужно было останавливать. Схематическое изображение установки показано на рисунке:



Дополнительно есть пульт оператора, который содержит кнопку старта (I4), экстренной остановки (I5), лампочка, сигнализирующая о том, что заготовки закончились (Q6) и что контейнер для пропечатанных заготовок полон (Q7), а также об аварийном состоянии (Q8).

Система работает следующим образом:

1. Оператор нажимает кнопку старта. Запускается лента.
2. а) Контейнер с заготовками пустой. Вызов сигнала и остановка ленты.

- б) Контейнер с заготовками не пустой. Открывается контейнер.
3. Заготовка едет по ленте, но ещё не достигла датчика печати.
 4. Заготовка достигла датчика печати. Печать опускается.
 5. Деталь продолжает движение по ленте.
 6. Деталь достигла конца ленты. Открывается крышка контейнера для готовых заготовок.
 7. а) Контейнер стал заполнен. Останавливаем процесс.
б) Контейнер не заполнен. Продолжаем процесс в шаг 2.
 8. В любой момент, когда нажата кнопка остановки, процесс должен быть остановлен.
 9. При переходе в неизвестные состояния процесс должен быть остановлен.

Данное описание системы было комбинацией графического способа и словесного описания процессов. Далее рассмотрим диаграмму состояний для него же.

Одним из способов представления процессов программного продукта, помимо привычных блок-схем, могут выступать диаграммы процессов различного вида, в том числе диаграммы состояний объектов, которыми мы и будем пользоваться в нашей разработке. В конечном счёте, нам дается комбинация входных сигналов, по которой необходимо сформировать комбинацию выходных (управляющих) сигналов.

Рассмотрим таблицу состояний в соответствии с шагами:

Шаг	I0	I1	I2	I3	I4	I5	Q0	Q1	Q2	Q3	Q6	Q7	Q8
1	0	X	0	0	1	0	1	0	0	0	0	0	0
2a	0	0	0	0	0	0	0	0	0	0	1	0	1
2б	0	1	0	0	0	0	1	1	0	0	0	0	0
3	0	X	0	0	0	0	1	0	0	0	0	0	0
4	0	X	0	1	0	0	1	0	0	1	0	0	0
5	0	X	0	0	0	0	1	0	0	0	0	0	0
6	1	X	0	0	0	0	1	0	1	0	0	0	0
7a	0	X	1	0	0	0	0	0	0	0	0	1	1
7б	0	X	0	0	0	0	1	0	0	0	0	0	0
8	X	X	X	X	X	1	0	0	0	0	0	0	0
9	?	?	?	?	?	?	0	0	0	0	0	0	1

В данной таблице X означает, что неважно, в каком состоянии находится конкретный элемент системы (датчик).

Таким образом, имея диаграмму состояний можно планировать комбинации элементов так, чтобы не возникали сопутствующие ошибки процессов и контролировать отдельные их параметры.

Для базового уровня:

Сравним между собой вычисление корней квадратного уравнения двумя разными способами, рассмотрев при этом отдельные случаи для различных коэффициентов квадратных уравнений. В общем случае уравнение второй степени можно записать в виде $Ax^2 + Bx + C = 0$. В зависимости от значений коэффициентов можно выделить различные модификации данного уравнения:

	B = 0		B != 0	
A = 0	$0 = 0$ (1)	$C = 0$ (3)	$Bx + C = 0$ (5)	$Bx = 0$ (7)
A != 0	$Ax^2 = 0$ (2)	$Ax^2 + C = 0$ (4)	$Ax^2 + Bx + C = 0$ (6)	$Ax^2 + Bx = 0$ (8)
	C = 0	C != 0		C = 0

- (1) - уравнение, верно при любом x ;
- (2) - уравнение, имеет единственный корень $x = 0$;
- (3) - уравнение, которое не имеет корней и является неверным;
- (4) - уравнение, которое имеет корни $\sqrt{-C/A}$ и $-\sqrt{-C/A}$, если они существуют, то есть если $-C/A \geq 0$;
- (5) - уравнение с единственным корнем $-C/B$;
- (6) - классическое квадратное уравнение;
- (7) - уравнение, имеет единственный корень $x = 0$;
- (8) - уравнение, у которого левая часть раскладывается в вид $x \cdot (Ax + B) = 0$, то есть всего есть 2 корня: $x = 0$ и $x = -B/A$.

В соответствии с вариантом создаются 8 уравнений, каждое из которых решается соответствующим заданному уравнению способом и методом подбора. Результаты решения сравниваются между собой и производятся соответствующие выводы.

Графической интерпретации для заданного примера нет, так как оно представляется в виде соответствующей матрицы возможных коэффициентов, а последовательность действий достаточно описывается словесным алгоритмом.

Технологическая часть. Архитектура классов

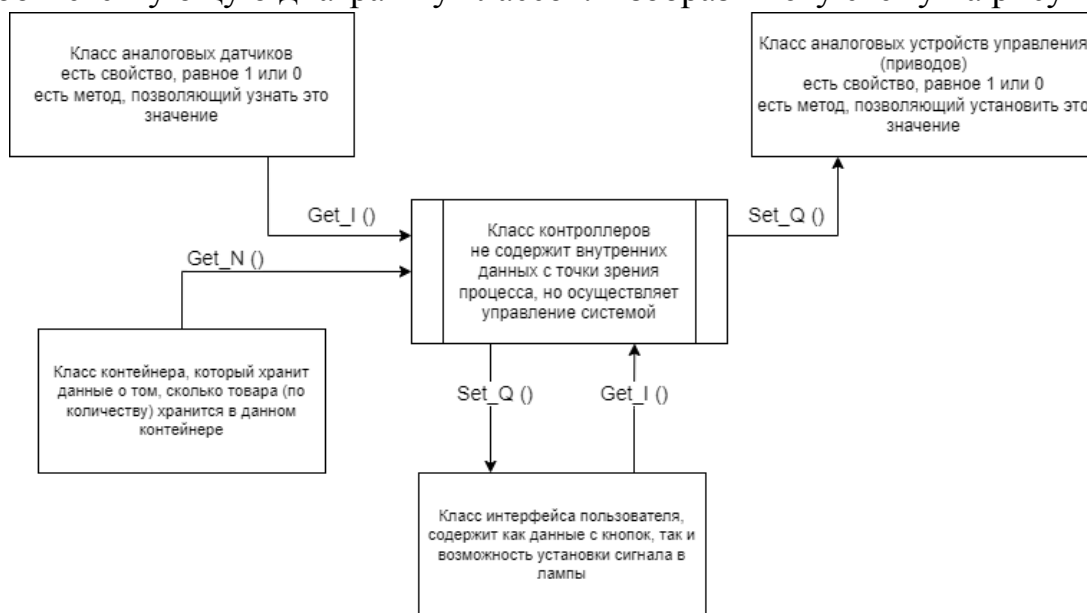
Так или иначе, вне зависимости от конечной реализации, диаграмма классов представляет собой графически, либо словесно описанную схему системы взаимодействия классов и способов такого взаимодействия. Так как, фактически, классы представляют собой отдельные абстрактные структуры, которые позволяют раскрыть и/или описать реальное взаимодействие подобных объектов. Рассмотрим это далее на примерах.

Важно, что для каждого класса необходимо написать таблицу классов с описанием их свойств и методов.

Возьмём в качестве примера ту же установку по маркированию заготовок. Фактически, мы можем наблюдать несколько проявлений различного рода объектов:

- аналоговые датчики;
- аналоговые элементы управления;
- контроллер;
- интерфейс оператора;
- контейнеры с продуктами.

Если каждый из данных родов объектов мы будем рассматривать, как отдельные классы, то их взаимодействие, фактически, и вид такого взаимодействия и позволит описать соответствующую диаграмму классов. Изобразим эту схему на рисунке:



Для примера базового уровня характерно наличие 8 классов под каждый из возможных видов уравнения. При этом есть интерфейс инициализации значений каждого из уравнений, а также получения значения корней такого уравнения, если они существуют.



Практическая часть

В практической части достаточно привести несколько примеров реализации отдельных пунктов программного продукта с описанием технологических его технологических частей.

Для тестирования необходимо подготовить **журнал испытаний программы** (таблица из 4 столбцов и 10 строк) на 10 испытаний, из которых: 6 испытаний стандартной работы, 2 испытания краевых значений (проверка пределов работы программы) и 2 испытания внеформатных значений (ошибок).

Заключение

В заключении необходимо сделать вывод о работе над продуктом, его результатах, а также привести систему критериев по их достижимости.

Список литературы

В конце работы необходимо указать список использованных источников, на который необходимо ссылаться в теоретической части и по ходу работы. Если какие-то фрагменты взяты из литературы, то необходимо указывать в конце скобки [1] с номером источника в списке. НЕ ДОПУСКАЕТСЯ ПРЯМОГО ЦИТИРОВАНИЯ, но можно перефразировать, донося основную мысль источника.

Приложение

В приложении следует привести листинг программы (скриншот полного кода программы по отдельным файлам).

Требования к оформлению

Требования к оформлению располагаются в [файле](#)