

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Б1.О.19
(индекс дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Основы моделирования и проектирования программного обеспечения

(наименование дисциплины)

по направлению подготовки
09.03.04 Программная инженерия

направленность (профиль)
Программная инженерия с применением ИИ-технологий

Форма обучения: очная

Год набора: 2026

Общая трудоемкость: 5 ЗЕ

Распределение часов дисциплины по семестрам

Семестр	5	Итого
Вид занятий	Форма контроля	
Лекции	КР, экзамен	32
Лабораторные		
Практические		32
Руководство: курсовые работы		1
Промежуточная аттестация		0,35
Контактная работа		65,35
Самостоятельная работа		79
Контроль		35,65
Итого		180

Рабочую программу составил(и):

доцент института цифровых технологий, канд. техн. наук, Хрипунов Н.В.

(должность, ученое звание, степень, Фамилия И.О.)

(должность, ученое звание, степень, Фамилия И.О.)

Рецензирование рабочей программы дисциплины:



Отсутствует



Рецензент

(должность, ученое звание, степень, Фамилия И.О.)

Рабочая программа дисциплины составлена на основании ФГОС ВО и учебного плана
направления подготовки

09.03.04 Программная инженерия

Срок действия рабочей программы дисциплины до «31» августа 2030 г.

УТВЕРЖДЕНО

На заседании института цифровых технологий

(протокол заседания № 1 от «05» сентября 2025 г.).

1. Цель освоения дисциплины

Цель освоения дисциплины – сформировать у обучающихся теоретические представления о моделировании и проектировании программного обеспечения, закрепить основные навыки разработки программного обеспечения.

2. Место дисциплины в структуре ОПОП ВО

Дисциплины и практики, на освоении которых базируется данная дисциплина: «Объектно-ориентированное программирование».

Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее: «Архитектура информационных систем и методы интеграции», «Выполнение и защита выпускной квалификационной работы».

3. Планируемые результаты обучения

Формируемые и контролируемые компетенции (код и наименование)	Индикаторы достижения компетенций (код и наименование)	Планируемые результаты обучения
ПК - 1. Способен формализовать и алгоритмизировать задачи для разработки программного обеспечения	ПК-1.1. Знает понятие алгоритма, методы формализации и способы описания алгоритмов	Знать: понятие алгоритма Уметь: описать алгоритм задачи для разработки программного обеспечения Владеть: навыками разработки программного обеспечения
	ПК-1.2. Умеет описать и построить алгоритм задачи для разработки программного обеспечения	Знать: методы формализации алгоритмов Уметь: построить алгоритм задачи для разработки программного обеспечения Владеть: навыками проектирования программного обеспечения
	ПК-1.3. Владеет навыками алгоритмизации, проектирования и разработки программного обеспечения	Знать: способы описания алгоритмов Уметь: оценить качество алгоритма задачи для разработки программного обеспечения Владеть: навыками алгоритмизации программного обеспечения
ПК-4 Способен разрабатывать технические спецификации компонентов программных систем	ПК-4.1. Понимает технические спецификации компонентов программных систем	Знать: технические спецификации компонентов программных систем Уметь: оценить качество алгоритма задачи для разработки программного обеспечения Владеть: навыками оценки требований к техническим спецификациям компонентов программных систем
	ПК-4.2. Умеет описывать технические спецификации компонентов программных систем	Знать: способы описания технических спецификаций компонентов программных систем Уметь: оценить качество алгоритма задачи для разработки программного обеспечения Владеть: навыками проектирования

Формируемые и контролируемые компетенции (код и наименование)	Индикаторы достижения компетенций (код и наименование)	Планируемые результаты обучения
		технических спецификаций компонентов программных систем
	ПК-4.3. Владеет навыками разработки технических спецификаций компонентов программных систем	Знать: методы разработки технических спецификаций компонентов программных систем Уметь: оценить качество алгоритма задачи для разработки программного обеспечения Владеть: навыками разработки технических спецификаций компонентов программных систем

4. Структура и содержание дисциплины

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
Модуль 1. Содержание проектирования ПО и смежные области программной инженерии	Лек 1	Проектирования ПО в составе программной инженерии	5	2		-	
	Лек 2	Основные вопросы проектирования ПО	5	2		-	
	Ср 1	Работа с лекционным материалом и учебной литературой, подготовка к практическим работам	5	12		-	
	Пр 1	Пр 1 Разработка организационной структуры и анализ функциональных областей предприятия (часть 1).	5	2	10	-	Отчет по практической работе 1
	Пр 2	Пр 1 Разработка организационной структуры и анализ функциональных областей предприятия (часть 2).	5	2		-	
Модуль 2. Разработка требований к проектированию ПО	Лек 3	Функциональные спецификации	5	2		-	
	Лек 4	Техническое задание по ГОСТ 34.602	5	2		-	
	Ср 2	Работа с лекционным материалом и учебной литературой, подготовка к практическим работам	5	12		-	
	Пр 3	Пр 2. Разработка технического задания (часть 1).	5	2	10	-	Отчет по практической работе 2
	Пр 4	Пр 2. Разработка технического задания (часть 2).	5	2		-	
Модуль 3 Архитектурное проектирование программного обеспечения и моделирование	Лек 5	Порядок и принципы архитектурного проектирования ПО	5	2		-	
	Лек 6	Архитектурные принципы и шаблоны проектирования программного обеспечения	5	2		-	
	Лек 7	Процессы архитектурного	5	2		-	

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
процессов		проектирования и архетипы приложений					
	Ср 3	Работа с лекционным материалом и учебной литературой, подготовка к практическим работам	5	12		-	
	Пр 5	Пр 3. Функционально-ориентированное моделирование программного обеспечения в методологии SADT (часть 1).	5	2	10	-	Отчет по практической работе 3
	Пр 6	Пр 3. Функционально-ориентированное моделирование программного обеспечения в методологии SADT (часть 2).	5	2		-	
Модуль 4. Функционально-ориентированное проектирование ПО	Лек 8	Процессный подход	5	2		-	
	Лек 9	Функциональное моделирование программного обеспечения	5	2		-	
	Ср 4	Работа с лекционным материалом и учебной литературой, подготовка к практическим работам	5	12		-	
	Пр 7	Пр 4. Моделирование бизнес-процессов в методологии BPMN (часть 1).	5	2	15	-	Отчет по практической работе 4
	Пр 8	Пр 4. Моделирование бизнес-процессов в методологии BPMN (часть 2).	5	2		-	
	Пр 9	Пр 4. Моделирование бизнес-процессов в методологии BPMN(часть 3).	5	2		-	
Модуль 5. Объектно-	Лек 10	Методы проектирования архитектуры ПО	5	2		-	

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
ориентированное проектирование ПО	Лек 11	Основы и правила унифицированного языка моделирования	5	2		-	
	Лек 12	Диаграммы объектной модели	5	2		-	
	Ср 5	Работа с лекционным материалом и учебной литературой, подготовка к практическим работам	5	19		-	
	Пр 10	Пр 5. Модель вариантов использования (часть 1).	5	2	15	-	Отчет по практической работе 5
	Пр 11	Пр 5. Модель вариантов использования (часть 2).	5	2		-	
	Пр 12	Пр 6. Модели детализации прецедентов (часть 1).	5	2	15	-	Отчет по практической работе 6
	Пр 13	Пр 6. Модели детализации прецедентов (часть 2).	5	2		-	
	Лек 13	Моделирование данных в UML	5	2		-	
	Пр 14	Пр 7. Логическое моделирование данных с использованием диаграммы классов (часть 1).	5	2	15	-	Отчет по практической работе 7
	Пр 15	Пр 7. Логическое моделирование данных с использованием диаграммы классов (часть 2).	5	2		-	
	Пр 16	Пр 7. Логическое моделирование данных с использованием диаграммы классов (часть 3).	5	2		-	
Модуль 6. Технологии проектирования ПО	Лек 14	Формальные методы проектирования	5	2		-	
	Лек 15	Прототипное проектирование и разработка ПО	5	2		-	
	Лек 16	Гибкие методы проектирования и разработки ПО	5	2		-	

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
	Ср 6	Работа с лекционным материалом и учебной литературой	5	12		-	
	Курсовая работа		5	1		-	Защита курсовой работы
	ПА	Промежуточная аттестация	5	0,35		-	
	Псщ.		5		10		
	Контроль	Экзамен	5	35.65	100		Итоговый тест
Итого:				180			

Схема расчета итогового балла: по накопительному рейтингу

Текущий рейтинг + Результат итогового теста и все делится на 2

5. Образовательные технологии

При изучении дисциплины используются следующие образовательные технологии:

- технологии традиционного обучения в форме лекций, практических работ и самостоятельной работы обучающихся;
- технология проектного обучения: реализация и защита отчетов по практическим работам.

Технологии традиционного обучения - организация учебного процесса в вузе, основанная на лекционных и практических формах обучения: объяснительно-иллюстративное обучение. Данная технология применяется во всех модулях курса.

Технология интерактивного обучения - организация учебного процесса, которая предполагает максимальную активность обучающихся в процессе формирования ключевых компетенций. На практическом занятии обучающиеся представляют результат выполнения заданной работы.

6. Методические указания по освоению дисциплины

6.1. Рекомендации по подготовке к практическим занятиям

Обучающимся следует:

- при подготовке к практическим занятиям следует обязательно использовать не только лекции, учебную литературу, но и другие источники;
- в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения;
- на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Для того чтобы практические занятия приносили максимальную пользу, необходимо помнить, что решение задач проводится по рассмотренному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться обучающимся на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях обучающийся не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если обучающийся видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

6.2. Рекомендации по подготовке к экзамену

Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, обучающийся ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене обучающийся демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

На консультации перед экзаменом обучающиеся должны быть ознакомлены с основными требованиями и получить ответы на возникающие в процессе подготовки вопросы.

Необходимо ориентировать обучающихся на систематическую подготовку к занятиям в течение семестра, что позволит использовать время экзаменационной сессии для систематизации знаний.

7. Оценочные средства

7.1. Паспорт оценочных средств

Семестр	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
5	ПК-1; ПК-4	Тестовые задания 1-621 Вопросы к экзамену Отчеты по практическим работам 1-7 Защита курсовой работы

7.2. Типовые задания или иные материалы, необходимые для текущего контроля

7.2.1. Отчеты по практическим занятиям (примеры)

Практическая работа 1. Разработка организационной структуры и анализ функциональных областей предприятия

Цель работы: получить практические навыки разработки организационной структуры и анализ функциональных областей предприятия.

Задание

1. Выбрать и согласовать с преподавателем предприятие (реальное или гипотетическое).

2. Выполнить словесное описание предприятия, достаточное для построения организационной структуры и таблицы распределения функциональных областей

3. Построить организационную структуру и таблицу распределения функциональных областей

Указания к выполнению:

Выбор предприятия начинается с предметной области

1. Перевозки
2. Коммуникации
3. Недвижимость
4. Кинопрокат
5. Торговля
6. Медицина
7. Услуги
8. Обучение

9. Ремонт
10. Строительство
11. Производство
12. Информационные технологии

Затем из предметной области выбрать (придумать) предприятие/организация/учреждение, например для области 1 – Судоходная компания

Затем используя Интернет или личный опыт изучить деятельность компаний выбранного типа и дать краткое описание деятельности. Например, компания занимается перевозками грузов между континентами. В ее собственности несколько десятков судов различного класса и грузоподъемности. К услугам этой компании обращаются тысячи клиентов из различных стран мира. На судне может находиться несколько партий грузов для различных грузополучателей из различных стран и городов. Одна партия груза может состоять из нескольких разновидностей грузов. У одной партии груза может быть только один отправитель и только один получатель. Судно следует по маршруту. Маршрут разрабатывается главным менеджером компании и проходит через несколько портов. В очередном порту назначения производится лишь частичная погрузка и выгрузка грузов, и судно следует дальше.

Для больших предприятий выбрать область автоматизации, например, управление грузооборотом

2. Описать словами организационную структуру и должностные обязанности структурных подразделений и работников предприятия. Для больших предприятий достаточно описать структуру и функции области автоматизации

3. Построить организационную структуру используя различные CASE средства (например, ARISexpress, bpmn.studio)

4. Построить таблицу распределения функциональных областей исходя из записанных в п.2 функций

Пример таблицы распределения функциональных обязанностей агентства по недвижимости показан в таблице.

Таблица 1.1

Распределение функциональных обязанностей

	Менеджмент	Финансы	Маркетинг	Клиентский сервис	Документооборот
Директор, заместитель директора	х	х			х
Бухгалтер		х			
Риелторы и агенты			х	х	
Оценщики		х		х	
Юристы				х	х

Практическая работа 2. Разработка технического задания

Цель работы: приобретение практических навыков по разработке технического задания (ТЗ) на разработку программного обеспечения.

Задание: Разработать техническое задание на разработку программного обеспечения (ПО) для выбранной предметной области.

Указания к выполнению задания:

Изучите ГОСТ 34.602-89. Перед началом работы внимательно изучите ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы», определяющий структуру и содержание технического задания.

Изложите техническое задание в соответствии со структурой по ГОСТ 34.602-89:

1. Общие положения:

1.1 Полное наименование системы и ее условное обозначение.

1.2 Наименование предприятия (организации) – заказчика (пользователя) системы и, при необходимости, наименование организации – разработчика системы.

1.3 Перечень документов, на основании которых создается система. Указываются документы, на основании которых принято решение о разработке системы (например, приказ руководителя, результаты анализа).

1.4 Плановые сроки начала и окончания работ по созданию системы.

1.5 Сведения об источниках финансирования работ по созданию системы.

1.6 Порядок оформления и предъявления заказчику результатов работ по созданию системы (этапы, сроки, форма представления).

2. Назначение и цели создания системы:

2.1 Назначение системы. Описывается, для чего предназначена разрабатываемая система (какие задачи она должна решать).

2.2 Цели создания системы. Описываются цели, которые должны быть достигнуты в результате создания системы (например, повышение эффективности работы, снижение затрат, улучшение качества обслуживания). Необходимо указать конкретные измеримые показатели.

3. Характеристики объекта автоматизации:

3.1 Краткие сведения об объекте автоматизации. Описывается предметная область, для которой разрабатывается система (например, библиотека, пиццерия, склад).

3.2 Перечень функций, выполняемых объектом автоматизации. Перечисляются основные функции, которые выполняет объект автоматизации (например, выдача книг, прием заказов, учет товаров).

3.3 Характеристики потоков информации, циркулирующих в объекте автоматизации. Описываются основные информационные потоки (например, запросы на выдачу книг, заказы, накладные).

4. Требования к системе в целом:

4.1 Требования к структуре и функционированию системы. Описываются основные компоненты системы и их взаимодействие.

4.2 Требования к характеристикам технических средств. Описываются требования к аппаратному и программному обеспечению, необходимому для работы системы (например, тип сервера, операционная система, СУБД).

4.3 Требования к информационной безопасности. Описываются требования к защите информации от несанкционированного доступа, модификации или уничтожения.

4.4 Требования к надежности. Описываются требования к надежности работы системы (например, время наработки на отказ, время восстановления после сбоя).

4.5 Требования к эргономике и технической эстетике. Описываются требования к удобству и привлекательности интерфейса пользователя.

4.6 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению системы.

5. Требования к функциям (задачам), выполняемым системой:

Описываются требования к каждой функции (задаче), которую должна выполнять система. Для каждой функции необходимо указать:

Наименование функции.

Описание функции.

Входные данные.

Выходные данные.

Алгоритм выполнения (если необходимо).

6. Требования к видам обеспечения:

6.1 Требования к математическому обеспечению. Описываются требования к математическим методам и алгоритмам, используемым в системе.

6.2 Требования к программному обеспечению. Описываются требования к программным средствам, необходимым для работы системы (например, языки программирования, библиотеки, фреймворки).

6.3 Требования к информационному обеспечению. Описываются требования к структуре и организации данных, используемых в системе (например, структура базы данных, форматы файлов).

6.4 Требования к техническому обеспечению. Описываются требования к аппаратному обеспечению (см. 4.2).

6.5 Требования к организационному обеспечению. Описываются требования к организации работы пользователей с системой (например, инструкции, регламенты).

6.6 Требования к методическому обеспечению. Описываются требования к методическим материалам, необходимым для использования системы (например, учебные пособия, руководства).

6.7 Требования к лингвистическому обеспечению. Описываются требования к языку интерфейса пользователя (например, русский, английский).

6.8 Требования к правовому обеспечению. Описываются требования к соблюдению авторских прав и других юридических норм при использовании системы.

7. Требования к составу и содержанию работ по созданию системы:

Перечисляются основные этапы работ по созданию системы (например, анализ требований, проектирование, разработка, тестирование, внедрение). Для каждого этапа необходимо указать:

Наименование этапа.

Содержание работ.

Сроки выполнения.

Результаты.

8. Порядок контроля и приемки системы:

Описывается порядок проведения испытаний и приемки системы. Указываются виды испытаний (например, функциональное тестирование, нагрузочное тестирование), критерии приемки и порядок оформления результатов.

9. Требования к документированию:

Описывается состав документации, которая должна быть разработана вместе с системой (например, руководство пользователя, руководство администратора, техническое описание).

10. Источники разработки:

Перечисляются источники информации, использованные при разработке ТЗ (например, стандарты, техническая литература, документация на существующие системы).

Практическая работа 3. Функционально-ориентированное моделирование программного обеспечения в методологии SADT

Цель работы - получить практические навыки моделирования бизнес-процессов в методологии SADT в нотациях IDEF0 и DFD

Задание:

1. Разработать модель процессов в нотации IDEF0.
2. Дополнить модель диаграммами DFD.

Указания к выполнению задания.

В работе следует использовать свободное ПО Ramus2.0 <https://github.com/Vitaliy-Yakovchuk/ramus/releases> и предприятие предметной области, описанное в предыдущих работах.

1. Разработать модель процессов в нотации IDEF0:

При разработке модели процессов заданной предметной области необходимо:

Определить контекстную диаграмму (A-0). Начните с построения контекстной диаграммы (A-0), которая представляет всю систему как один процесс. Необходимо описать Цель создания модели.

Точку зрения (например, с точки зрения клиента, менеджера, сотрудника).

Входы (Inputs): Какие данные или материалы необходимы для работы системы?

Выходы (Outputs): Какие результаты производит система?

Управление (Controls): Какие правила, стандарты или политики регулируют работу системы?

Механизмы (Mechanisms): Какие ресурсы (люди, оборудование, программное обеспечение) используются для выполнения процессов?

Выполнить декомпозицию (разбиение) диаграммы A-0 на дочерние диаграммы (A0). Разбейте контекстную диаграмму на несколько дочерних процессов, каждый из которых представляет собой более детальное описание части системы.

Убедитесь, что входы, выходы, управление и механизмы родительской диаграммы согласуются с входами, выходами, управлением и механизмами дочерних диаграмм. Нельзя добавить что-то, чего не было на родительской диаграмме.

Старайтесь, чтобы количество процессов на одной диаграмме составляло 3-6.

Каждый процесс должен иметь уникальное название, отражающее его суть. Используйте глаголы или отглагольные существительные (например, «Принять заказ», «Доставка товара»).

2. Дополнить модель диаграммами DFD.

Для дополнения модели IDEF0 диаграммой DFD необходимо:

Выбрать процесс из модели IDEF0 для детализации. DFD детализирует отдельные процессы, представленные в IDEF0. Выберите наиболее важный или сложный процесс из модели IDEF0, который требует более подробного описания потоков данных.

Определить внешние сущности (источники и приемники данных). Определите, кто или что поставляет данные в выбранный процесс и кто или что получает данные из этого процесса. Эти сущности представляются прямоугольниками на диаграмме DFD.

Определить основные потоки данных. Выявите, какие данные передаются между внешними сущностями и процессами, а также между процессами внутри выбранного процесса IDEF0. Эти потоки данных представляются стрелками на диаграмме DFD.

Определить хранилища данных (data stores). Выявите, какие данные хранятся внутри процесса. Хранилища данных представляются параллельными линиями на диаграмме DFD.

Детализировать выбранный процесс IDEF0. Разбейте выбранный процесс из IDEF0 на более мелкие процессы, которые выполняют отдельные шаги обработки данных. Каждый процесс должен иметь четко определенные входы и выходы.

Построить диаграмму DFD, отражающую потоки данных. Используя определенные внешние сущности, потоки данных, хранилища данных и процессы, постройте диаграмму DFD, которая показывает, как данные перемещаются и преобразуются внутри выбранного процесса IDEF0.

Убедиться в согласованности DFD и IDEF0. Проверьте, что данные, входящие и выходящие из процесса IDEF0, соответствуют данным, входящим и выходящим из соответствующей диаграммы DFD. DFD должна детализировать то, что уже представлено в IDEF0, а не вводить новую информацию.

Практическая работа 4. Моделирование бизнес-процессов в методологии BPMN

Цель работы - получить практические навыки моделирования бизнес-процессов в методологии BPMN

Задание:

1. Описать алгоритмический процесс.
2. Построить модель процесса в нотации BPMN .

Указания к выполнению задания.

В работе следует использовать свободное онлайн средство моделирования <https://bpmn.studio/ru> и предприятие предметной области, описанное в предыдущих работах.

1. Описать алгоритмический процесс

Описать алгоритмический процесс необходимо следующим образом:

Выбрать процесс для моделирования. Используйте предметную область из работ 1 и 2 (SADT, DFD). Выберите конкретный бизнес-процесс, который будет смоделирован в BPMN. Это может быть процесс целиком или подпроцесс из ранее созданной модели. Например для процесса «Обработка заказа в интернет-магазине» можно выбрать подпроцесс «Проверка наличия товара на складе».

Определить цель процесса. Четко сформулируйте цель моделируемого процесса. Что должно быть достигнуто в результате его выполнения?

Определить участников процесса (роли). Какие сотрудники или системы участвуют в выполнении процесса? Определите роли, ответственные за выполнение различных задач.

Составить подробное описание шагов процесса. Разбейте процесс на последовательность отдельных шагов (задач). Для каждого шага опишите:

Входные данные: Какие данные необходимы для выполнения шага.

Действие: Что необходимо сделать на этом шаге.

Выходные данные: Результат выполнения шага.

Участник (роль): Кто отвечает за выполнение шага.

Критерии успешного завершения: Когда шаг считается выполненным.

Определить условия и развилки (шлюзы). Выявите точки принятия решений в процессе. Какие условия определяют, по какому пути пойдет процесс? Опишите логику работы каждого шлюза.

Определить события, инициирующие и завершающие процесс. Какие события запускают процесс и какие события указывают на его завершение?

2. Построить модель процесса в нотации BPMN.

Создать новую диаграмму BPMN. Начните новый проект и выберите шаблон "Process Diagram".

Использовать палитру элементов BPMN. BPMN.studio предоставляет палитру с различными элементами BPMN (события, задачи, шлюзы, потоки управления и сообщений, артефакты). Изучите палитру и научитесь добавлять элементы на диаграмму.

Изобразить начало и конец процесса. Начните с добавления стартового события (Start Event) и конечного события (End Event).

Добавить задачи (Tasks). Добавьте задачи, представляющие отдельные шаги процесса, описанные вами в пункте 1. Выберите подходящий тип задачи (например, User Task, Service Task, Script Task) в зависимости от характера задачи.

Соединить элементы потоками управления (Sequence Flows). Соедините стартовое событие, задачи и конечное событие потоками управления, чтобы показать последовательность выполнения шагов процесса.

Добавить шлюзы (Gateways). Добавьте шлюзы, представляющие точки принятия решений. Используйте различные типы шлюзов (Exclusive Gateway, Parallel Gateway, Inclusive Gateway) в зависимости от логики процесса.

Добавить потоки сообщений (Message Flows). Если в процессе участвуют разные участники или системы, используйте потоки сообщений для отображения обмена сообщениями между ними.

Добавить пулы и дорожки (Pools and Lanes). Используйте пулы и дорожки для разделения процесса на отдельные участки, представляющие разных участников или системы. Пул представляет собой независимый процесс, а дорожки - разделение пула на более мелкие участки, представляющие роли или отделы.

Добавить аннотации (Text Annotations). Добавьте текстовые аннотации для пояснения элементов модели.

Настроить свойства элементов. Кликните на каждый элемент, чтобы настроить его свойства (например, имя, тип, описание, исполнитель).

Проверить правильность модели. Убедитесь, что модель BPMN соответствует описанию процесса и логически непротиворечива.

Практическая работа 5. Модель вариантов использования

Цель работы - получить практические навыки моделирования вариантов использования программного обеспечения

Задание:

1. Провести описание своей предметной области в ключе объектно-ориентированного подхода.
2. Выполнить построение диаграммы прецедентов в StarUML.
3. Выполнить документирование элементов модели в StarUML.
4. Добавить описание потока событий одного прецедента.

Указания к выполнению задания.

В работе следует использовать свободное ПО StarUML <http://staruml.io>.

1. Провести описание своей предметной области в ключе объектно-ориентированного подхода:

Определите основные объекты (классы) в вашей предметной области. Подумайте, какие существительные (вещи, концепции, объекты) играют важную роль в работе системы. Например, для системы управления библиотекой это могут быть: Книга, Читатель, Автор, Заказ, Абонемент.

Опишите характеристики (атрибуты) каждого объекта. Какие данные необходимо хранить о каждом объекте? Например, для объекта «Книга» это могут быть: Название, Автор, ISBN, Год издания, Количество экземпляров.

Определите взаимосвязи между объектами. Как объекты связаны между собой? Например: «Читатель может взять несколько Книг», «Книга написана одним или несколькими Авторами».

Определите основные операции (методы) каждого объекта. Что можно делать с каждым объектом? Например, для объекта «Книга» это могут быть: «Взять книгу», «Вернуть книгу», «Зарезервировать книгу».

Сформулируйте описание предметной области, используя объектно-ориентированную терминологию. Например: «Система управления библиотекой позволяет Читателям брать Книги. Каждая Книга имеет определенные атрибуты, такие как Название, Автор, ISBN. Читатель может выполнить операции Взять книгу и Вернуть книгу».

2. Выполнить построение диаграммы прецедентов в StarUML:

Запустите StarUML.

Создайте новый проект или откройте существующий.

Создайте новую диаграмму прецедентов (Use Case Diagram). Выберите «Model» -> «Add Diagram» -> «Use Case Diagram».

Определите акторов (Actors). Акторы представляют собой роли, которые взаимодействуют с системой. Определите, кто или что будет использовать вашу систему. Примеры акторов: Читатель, Администратор библиотеки, Кладовщик (в зависимости от предметной области). Добавьте акторов на диаграмму, перетаскивая элемент «Actor» из панели инструментов. Дайте каждому актору осмысленное имя.

Определите прецеденты (Use Cases). Прецеденты представляют собой цели, которые пользователи хотят достичь, используя систему. Определите, что пользователи смогут делать с помощью вашей системы. Примеры прецедентов: «Взять книгу», «Вернуть книгу», «Зарегистрировать читателя», «Поиск книги». Добавьте элемент «Use Case» из панели инструментов. Дайте каждому прецеденту осмысленное имя (используйте глаголы в инфинитиве).

Установите связи между акторами и прецедентами. Используйте ассоциации (Association) для соединения акторов с прецедентами, которые они используют. Проведите линии от акторов к соответствующим прецедентам.

3. Выполнить документирование элементов модели в StarUML:

Выберите актора или прецедент на диаграмме.

Откройте окно свойств (Property Window). Обычно оно находится в нижней части экрана. Если его нет, выберите «View» -> «Property».

Заполните поле «Documentation». В этом поле необходимо описать назначение и функциональность выбранного элемента. Для акторов укажите их роль и обязанности. Для прецедентов опишите, что он делает, какие входные данные ему нужны и какие результаты он выдает. Будьте конкретны и информативны. Чем подробнее будет описание, тем лучше будет понимание вашей модели.

При необходимости, используйте другие поля свойств. Например, можно указать приоритет прецедента, сложность его реализации и т.д.

4. Добавить описание потока событий одного прецедента:

Выберите один из прецедентов на вашей диаграмме. Выберите прецедент, который кажется вам наиболее интересным или важным.

Определите основной поток событий (Main Success Scenario). Опишите шаги, которые выполняются при нормальном, успешном выполнении прецедента. Опишите, что делает актор и что делает система на каждом шаге.

Определите альтернативные потоки событий (Alternative Flows). Опишите, что происходит, если что-то идет не так. Какие альтернативные варианты развития событий возможны? Например, если пользователь вводит неверные данные, что должна сделать система?

ЗадOCUMENTИРУЙТЕ потоки событий в StarUML. Для этого необходимо использовать поле «Documentation» для выбранного прецедента.

Пример описания потока событий (для прецедента «Взять книгу»):

Основной поток событий:

1. Читатель предоставляет библиотекарю свою читательскую карточку.
2. Библиотекарь сканирует читательскую карточку.
3. Система проверяет, не заблокирован ли читатель (например, из-за просроченных книг).

4. Читатель предоставляет книгу для выдачи.
5. Библиотекарь сканирует ISBN книги.
6. Система проверяет, доступна ли книга (не находится ли она в резерве или на ремонте).

7. Система регистрирует выдачу книги читателю.

8. Система печатает квитанцию с указанием срока возврата книги.

9. Библиотекарь отдает книгу и квитанцию читателю.

Альтернативные потоки событий:

3а. Читатель заблокирован: Система отображает сообщение о блокировке и прерывает процесс выдачи.

6а. Книга недоступна: Система отображает сообщение о недоступности книги и предлагает читателю выбрать другую книгу.

Практическая работа 6. Модели детализации прецедентов

Цель работы - получить практические навыки детализации диаграммы прецедентов средствами языка UML

Задание:

1. Провести описание своей предметной области в ключе объектно-ориентированного подхода.
2. Выполнить построение диаграммы прецедентов в StarUML.
3. Выполнить документирование элементов модели в StarUML.
4. Добавить описание потока событий одного прецедента.

Указания к выполнению задания.

В работе следует использовать свободное ПО StarUML <http://staruml.io>.

1. Провести описание своей предметной области в ключе объектно-ориентированного подхода:

Определите основные объекты (классы) в вашей предметной области. Подумайте, какие существительные (вещи, концепции, объекты) играют важную роль в работе системы. Например, для системы управления библиотекой это могут быть: Книга, Читатель, Автор, Заказ, Абонемент.

Опишите характеристики (атрибуты) каждого объекта. Какие данные необходимо хранить о каждом объекте? Например, для объекта «Книга» это могут быть: Название, Автор, ISBN, Год издания, Количество экземпляров.

Определите взаимосвязи между объектами. Как объекты связаны между собой? Например: «Читатель может взять несколько Книг», «Книга написана одним или несколькими Авторами».

Определите основные операции (методы) каждого объекта. Что можно делать с каждым объектом? Например, для объекта «Книга» это могут быть: «Взять книгу», «Вернуть книгу», «Зарезервировать книгу».

Сформулируйте описание предметной области, используя объектно-ориентированную терминологию. Например: «Система управления библиотекой позволяет Читателям брать Книги. Каждая Книга имеет определенные атрибуты, такие как Название, Автор, ISBN. Читатель может выполнить операции Взять книгу и Вернуть книгу».

2. Выполнить построение диаграммы прецедентов в StarUML:

Запустите StarUML.

Создайте новый проект или откройте существующий.

Создайте новую диаграмму прецедентов (Use Case Diagram). Выберите «Model» -> «Add Diagram» -> «Use Case Diagram».

Определите акторов (Actors). Актеры представляют собой роли, которые взаимодействуют с системой. Определите, кто или что будет использовать вашу систему. Примеры акторов: Читатель, Администратор библиотеки, Кладовщик (в зависимости от предметной области). Добавьте акторов на диаграмму, перетаскивая элемент «Actor» из панели инструментов. Дайте каждому актору осмысленное имя.

Определите прецеденты (Use Cases). Прецеденты представляют собой цели, которые пользователи хотят достичь, используя систему. Определите, что пользователи смогут делать с помощью вашей системы. Примеры прецедентов: «Взять книгу», «Вернуть книгу»,

«Зарегистрировать читателя», «Поиск книги». Добавьте элемент «Use Case» из панели инструментов. Дайте каждому прецеденту осмысленное имя (используйте глаголы в инфинитиве).

Установите связи между акторами и прецедентами. Используйте ассоциации (Association) для соединения акторов с прецедентами, которые они используют. Проведите линии от акторов к соответствующим прецедентам.

3. Выполнить документирование элементов модели в StarUML:

Выберите актора или прецедент на диаграмме.

Откройте окно свойств (Property Window). Обычно оно находится в нижней части экрана. Если его нет, выберите «View» -> «Property».

Заполните поле «Documentation». В этом поле необходимо описать назначение и функциональность выбранного элемента. Для акторов укажите их роль и обязанности. Для прецедентов опишите, что он делает, какие входные данные ему нужны и какие результаты он выдает. Будьте конкретны и информативны. Чем подробнее будет описание, тем лучше будет понимание вашей модели.

При необходимости, используйте другие поля свойств. Например, можно указать приоритет прецедента, сложность его реализации и т.д.

4. Добавить описание потока событий одного прецедента:

Выберите один из прецедентов на вашей диаграмме. Выберите прецедент, который кажется вам наиболее интересным или важным.

Определите основной поток событий (Main Success Scenario). Опишите шаги, которые выполняются при нормальном, успешном выполнении прецедента. Опишите, что делает актор и что делает система на каждом шаге.

Определите альтернативные потоки событий (Alternative Flows). Опишите, что происходит, если что-то идет не так. Какие альтернативные варианты развития событий возможны? Например, если пользователь вводит неверные данные, что должна сделать система?

ЗадOCUMENTИРУЙТЕ потоки событий в StarUML. Для этого необходимо использовать поле «Documentation» для выбранного прецедента.

Пример описания потока событий (для прецедента «Взять книгу»):

Основной поток событий:

1. Читатель предоставляет библиотекарю свою читательскую карточку.
2. Библиотекарь сканирует читательскую карточку.
3. Система проверяет, не заблокирован ли читатель (например, из-за просроченных книг).
4. Читатель предоставляет книгу для выдачи.
5. Библиотекарь сканирует ISBN книги.
6. Система проверяет, доступна ли книга (не находится ли она в резерве или на ремонте).
7. Система регистрирует выдачу книги читателю.
8. Система печатает квитанцию с указанием срока возврата книги.
9. Библиотекарь отдает книгу и квитанцию читателю.

Альтернативные потоки событий:

3а. Читатель заблокирован: Система отображает сообщение о блокировке и прерывает процесс выдачи.

6а. Книга недоступна: Система отображает сообщение о недоступности книги и предлагает читателю выбрать другую книгу.

Практическая работа 7. Логическое моделирование данных с использованием диаграммы классов

Цель работы - получить практические навыки моделирования данных с использованием диаграммы классов UML

Задание:

1. Выполнить логическое проектирование базы данных для предметной области на уровне описания таблиц, полей, ключей и связей
2. Визуализировать логическую модель с использованием диаграммы классов UML

Указания к выполнению задания.

В работе следует использовать свободное ПО StarUML <http://staruml.io>.

1. Выполнить логическое проектирование базы данных для предметной области на уровне описания таблиц, полей, ключей и связей.

На основе описания предметной области (из предыдущих работ) определите основные сущности, информацию о которых необходимо хранить в базе данных. Каждая сущность будет соответствовать таблице в базе данных.

Для каждой сущности определите атрибуты, которые описывают ее характеристики. Каждый атрибут будет соответствовать полю в таблице базы данных. Определите тип данных для каждого атрибута (текст, число, дата).

Определите первичные ключи (primary keys) для каждой таблицы.

Определите внешние ключи (foreign keys) для установления связей между таблицами

Определите типы связей между таблицами (один-к-одному, один-ко-многим, многие-ко-многим). Тип связи определяет, сколько экземпляров одной сущности могут быть связаны с экземплярами другой сущности.

Для реализации связи многие-ко-многим обычно создается промежуточная таблица (связующая таблица), которая содержит внешние ключи, ссылающиеся на первичные ключи обеих таблиц.

Оформите результаты логического проектирования в виде таблицы или списка, указывая название таблицы, поля, типы данных, первичные и внешние ключи, а также тип связи между таблицами

2. Визуализировать логическую модель с использованием диаграммы классов UML:

Создайте новую диаграмму классов (Class Diagram) в StarUML. В проекте StarUML выберите «Model» -> «Add Diagram» -> «Class Diagram».

Добавьте классы (Classes) для каждой сущности (таблицы). Добавьте классы на диаграмму, перетаскивая элемент «Class» из панели инструментов. Дайте каждому классу имя, соответствующее названию таблицы.

Добавьте атрибуты (Attributes) для каждого класса (поля таблицы). Для каждого класса добавьте атрибуты, соответствующие полям таблицы. Укажите тип данных для каждого атрибута. Для этого щелкните правой кнопкой мыши по классу, выберите «Add» -> «Attribute». В поле «Type» укажите тип данных атрибута.

Установите связи (Associations) между классами. Используйте ассоциации для отображения связей между таблицами. Для установления ассоциации выберите инструмент «Association» на панели инструментов и проведите линию от одного класса к другому.

Укажите кратность (Multiplicity) на концах ассоциаций. Кратность определяет, сколько экземпляров одного класса могут быть связаны с экземплярами другого класса. Укажите кратность на концах ассоциаций в соответствии с типом связи между таблицами.

Разместите элементы на диаграмме аккуратно и понятно. Убедитесь, что диаграмма легко читается и все элементы четко видны.

Требования к оформлению

Отчет должен содержать подробное описание (включая иллюстрации). Отчёт по практическому занятию выполняется на страницах формата А4 в электронном виде.

При оформлении отчёта используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страницы ставятся по центру сверху.

При оформлении отчёта соблюдать следующие требования:

- Для заголовков: полужирный шрифт, 14 пт, центрированный.
- Для основного текста: нежирный шрифт, 14 пт, выравнивание по ширине.
- Во всех случаях тип шрифта – Times New Roman, отступ абзаца 1.25 см, полуторный междустрочный интервал.
- Поля: левое – 2 см, правое, верхнее и нижнее – 1 см.

Процедура оценивания

Оценка выполненной практической работы проводится по следующим критериям:

1. Наличие всей существенной информации по работе
2. Точность и полнота предоставляемых сведений
3. Непротиворечивость приводимой информации
4. Правильность интерпретаций и выводов, которые сделаны по результатам работы
5. Степень достижения обучающимся поставленной цели
6. Обоснованность применяемого решения
7. Грамотность (содержательная) используемых формулировок

Критерии оценки:

Формы текущего контроля	Критерии и нормы оценки
Отчет по практической работе 1-3	10 баллов – задание выполнено в полном объёме без замечаний 8 баллов – задание выполнено в полном объёме, присутствуют замечания по выполнению задания 6 баллов – задание выполнено не в полном объёме, присутствуют несущественные замечания 3 балла – задание выполнено не в полном объёме, присутствуют замечания по выполнению задания 0 баллов – задание не выполнено
Отчет по практической работе 4-7	15 баллов – задание выполнено в полном объёме без замечаний 10 баллов – задание выполнено в полном объёме, присутствуют замечания по выполнению задания 6 баллов – задание выполнено не в полном объёме, присутствуют несущественные замечания 3 балла – задание выполнено не в полном объёме, присутствуют замечания по выполнению задания 0 баллов – задание не выполнено

7.2.2. Типовые тестовые материалы

(наименование оценочного средства)

1. Методология SADT содержит нотацию

- IDEF0
- UML
- BPMN
- CORBA

2. Документ, представляющий графическое изображение форм документов, частей, элементов системы и связей между ними в виде условных обозначений по ГОСТ 34.201-89 называется

- Ведомость
- Схема
- Инструкция
- Обоснование

3. В MSF фаза «Стабилизация» завершается вехой

- составление документа «Образ и границы проекта»
- веха «Функциональные спецификации»
- веха «Завершение разработки»
- веха «Выпуск версии (Релиз)»

4. В множество структурных сущностей UML входит

- прецедент
- сущность
- связь
- атрибут

5. Укажите диаграмму поведения UML

- диаграмма прецедентов
- диаграмма последовательности.
- диаграмма обзора взаимодействия
- диаграмма синхронизации

6. Диаграмма деятельности содержит

- состояния действия
- процессы
- сущности
- артефакты

7. Отношения между классами в UML

- зависимость
- агрегация
- композиция
- навигация

Комплект заданий для итогового теста

ОМ закрытого типа

Задание 1

Выберите один правильный вариант ответа.

UML – это

- а) унифицированный язык моделирования
- б) универсальный язык моделирования
- в) унифицированный метод моделирования
- г) универсальный метод моделирования

Правильный ответ: а

Задание 2

Выберите несколько правильных вариантов ответа.

В методологию SADT входит

- а) IDEF0
- б) UML
- в) DFD
- г) BPMN

Правильный ответ: а, в

Задание 3

Выберите один правильный вариант ответа.

Укажите обязательные элементы диаграммы прецедентов UML

- а) прецеденты
- б) примечания
- в) ограничения
- г) пакеты

Правильный ответ: а

ОМ открытого типа

Задание 4

Дайте развернутый ответ.

В чем заключается разница между моделью и метамodelью в контексте разработки программного обеспечения?

Правильный ответ:

Модель представляет собой упрощенное представление реального объекта или системы, созданное для анализа, понимания и прогнозирования ее поведения. Метамодель, в свою очередь, описывает структуру и правила построения моделей.

Задание 5

Дайте развернутый ответ.

Какие преимущества и недостатки имеет водопадная модель разработки программного обеспечения (Waterfall model)? В каких ситуациях целесообразно ее использовать?

Правильный ответ:

Преимущества:

- Простая и понятная методология.
- Четкая структура и этапы работ.
- Строгая документация на каждом этапе.
- Подходит для проектов с четко определенными и стабильными требованиями.

Недостатки:

- Негибкость и сложность внесения изменений после завершения этапа.
- Позднее обнаружение ошибок, так как тестирование проводится только на последних этапах.
- Не подходит для проектов с изменяющимися требованиями.

Целесообразно использовать в проектах с четко определенными, стабильными требованиями и небольшим риском изменений.

Задание 6

Решите кейс

Команда разработчиков столкнулась с проблемой: сложно поддерживать существующий модуль в программном обеспечении из-за его запутанной структуры и

большого количества дублирующегося кода. Какие действия можно предпринять для улучшения ситуации, используя принципы объектно-ориентированного проектирования и рефакторинга?.

Правильный ответ:

1. Провести тщательный анализ кода модуля для выявления участков с высокой сложностью и дублирующимся кодом.
2. Разработать набор автоматических тестов, покрывающих функциональность модуля.
3. Применить методы рефакторинга для улучшения структуры кода - выделить повторяющиеся фрагменты кода в отдельные методы, переместить методы в более подходящие классы.
4. После каждого шага рефакторинга запускать тесты для проверки, что поведение модуля не изменилось.
5. Документировать внесенные изменения для облегчения дальнейшей поддержки.

Задание 7

Дайте развернутый ответ.

Опишите основные этапы разработки программного обеспечения в рамках гибкой методологии (Agile). Чем они отличаются от этапов водопадной модели?

Правильный ответ:

В гибкой методологии (Agile) разработка программного обеспечения происходит итеративно, с акцентом на быструю обратную связь и адаптацию к изменяющимся требованиям.

Основные этапы:

Планирование спринта (Sprint Planning): Определение целей спринта и выбор задач из бэклога продукта.

Разработка (Development): Реализация выбранных задач.

Ежедневные собрания (Daily Scrum): Короткие встречи команды для обсуждения прогресса и выявления проблем.

Обзор спринта (Sprint Review): Демонстрация результатов спринта заказчику и получение обратной связи.

Ретроспектива спринта (Sprint Retrospective): Анализ процесса разработки в спринте и определение улучшений.

Отличия от водопадной модели:

Гибкие методологии ориентированы на итеративную разработку, а водопадная модель - на последовательную.

В Agile требования могут меняться в процессе разработки, а в Waterfall требования должны быть зафиксированы в начале проекта.

В Agile заказчик активно участвует в процессе разработки, а в Waterfall - в основном на этапах сбора требований и приемки результатов.

Задание 8

Решите кейс

Вы разрабатываете систему управления проектами. Определите основных акторов и прецеденты для этой системы. Опишите кратко сценарий для одного из прецедентов.

Правильный ответ:

Основные акторы:

Менеджер проекта

Разработчик

Заказчик

Основные прецеденты:

Создать проект (Менеджер проекта)

Назначить задачу (Менеджер проекта)

Выполнить задачу (Разработчик)

Отслеживать прогресс (Менеджер проекта, Заказчик)
Сообщить о проблеме (Разработчик)
Утвердить результат (Заказчик).

7.2.2. Выполнение курсовой работы

Темы курсовых работ

1. Моделирование и проектирование программного обеспечения информационной системы управления (ИСУ) деканата вуза.
2. Моделирование и проектирование программного обеспечения ИСУ турагентством.
3. Моделирование и проектирование программного обеспечения ИСУ страховой компании.
4. Моделирование и проектирование программного обеспечения информационной системы управления (ИСУ) кадровым агентством.
5. Моделирование и проектирование программного обеспечения ИСУ спортивного центра.
6. Моделирование и проектирование программного обеспечения информационной системы управления (ИСУ) центром занятости.
7. Моделирование и проектирование программного обеспечения информационной системы управления (ИСУ) регистратурой поликлиники.
8. Моделирование и проектирование программного обеспечения информационной системы управления (ИСУ) продажами оргтехники.
9. Моделирование и проектирование программного обеспечения информационной системы управления (ИСУ) складом.
10. Моделирование и проектирование программного обеспечения ИСУ СТО.
11. Моделирование и проектирование программного обеспечения ИСУ школы.
12. Моделирование и проектирование программного обеспечения ИСУ ресторана.
13. Моделирование и проектирование программного обеспечения ИСУ системой онлайн-образования (платформы онлайн-курсов).
14. Моделирование и проектирование программного обеспечения ИСУ для управления автопарком предприятия.
15. Моделирование и проектирование программного обеспечения ИСУ системой бронирования номеров в отеле.
16. Моделирование и проектирование программного обеспечения ИСУ системой управления взаимоотношениями с клиентами (CRM) для небольшого бизнеса.
17. Моделирование и проектирование программного обеспечения ИСУ системой управления контентом (CMS) для новостного портала.
18. Моделирование и проектирование программного обеспечения ИСУ системой учета коммунальных платежей.
19. Моделирование и проектирование программного обеспечения ИСУ системой управления проектами для IT-компаний.
20. Моделирование и проектирование программного обеспечения ИСУ системой управления библиотечным фондом научной библиотеки.
21. Моделирование и проектирование программного обеспечения ИСУ системой автоматизированного тестирования программного обеспечения.
22. Моделирование и проектирование программного обеспечения ИСУ системой поддержки принятия решений для финансовых аналитиков.
23. Моделирование и проектирование программного обеспечения ИСУ системой управления инцидентами в службе технической поддержки.
24. Моделирование и проектирование программного обеспечения ИСУ логистикой для интернет-магазина.

25. Моделирование и проектирование программного обеспечения ИСУ мониторингом состояния здоровья пациентов с использованием носимых устройств.
26. Моделирование и проектирование программного обеспечения ИСУ системой персонализированных рекомендаций для онлайн-кинотеатра.
27. Моделирование и проектирование программного обеспечения ИСУ производством на небольшом предприятии.
28. Моделирование и проектирование программного обеспечения ИСУ отдела снабжения
29. Моделирование и проектирование программного обеспечения ИСУ отдела продаж
30. Моделирование и проектирование программного обеспечения ИСУ отдела кадров

Методические рекомендации по выполнению курсовой работы

Структура курсовой работы:

Титульный лист

Заявление на утверждение темы

Введение

Глава 1 Концептуальное проектирование

1.1 Идентификация предметной области

1.2 Выбор методологии и технологии концептуального моделирования

1.3 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»

1.4 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»

1.5 Разработка требований к ПО

1.6 Обзор и анализ аналогов ПО

1.7 Постановка задачи на разработку

Выводы по главе

Глава 2 Логическое моделирование программного обеспечения

2.1 Выбор методологии и технологии логического моделирования

2.2 Разработка диаграмм логической модели

2.3 Разработка логической модели данных

Выводы по главе

Заключение

Список использованной литературы

Приложение

Содержание элементов курсовой работы.

Содержание введения

1. Краткое описание предметной области и определение актуальности выбранной темы:

2) цель и задачи работы;

3) описание объекта, предмета и методов исследования;

4) краткий обзор работы по главам.

Объект исследования – сущность, на которую направлен взгляд исследователя. Объектами исследования прикладной информатики чаще всего являются различные системы.

Предмет исследования – свойства объекта, интересующие исследователя. В предмет включаются только те элементы, связи и отношения объекта, которые подлежат исследованию в данной работе. Поэтому определение предмета исследования означает также установление границ поиска и предположение о наиболее существенных в плане поставленной проблемы связях.

Метод исследования – характеристика процесса получения новых знаний о предмете. Основные подходы, методологии, средства и методы, которые могут выступать в качестве теоретико-методологической основы курсовой работы:

- гибкие технологии проектирования программного обеспечения;
- методологии структурного анализа и проектирования программного обеспечения;
- методологии объектно-ориентированного анализа и проектирования программного обеспечения;
- комплексные методы анализа и проектирования программного обеспечения;
- CASE-технологии анализа и проектирования программного обеспечения

Содержание главы 1

Целью данной части курсовой работы является изложение результатов анализа предметной области автоматизации и разработка концептуальной модели.

Концептуальная модель основывается на результатах анализа предметной области и представляет собой ее описание, выполненное с использованием естественного языка, математических выражений, таблиц, графов и других средств.

Раздел 1.1 Идентификация предметной области

В этом разделе описываются основные сведения о предприятии, в состав которого входит подразделение, связанное с объектом исследования:

- направления предпринимательской деятельности предприятия;
- организационно-правовая форма предприятия;
- номенклатура выпускаемой продукции и виды услуг;
- преобладающий тип производства;
- перспективы развития предприятия.

В данной части курсовой работы требуется привести описание организационной структуры предприятия и определить место подразделения (объекта исследования) в производственно-хозяйственной деятельности предприятия.

Целесообразно разработать схему организационной структуры предприятия и выделить в ней исследуемое подразделение.

Описываются также состав подразделения и его задачи.

Раздел 1.2 Выбор методологии и технологии концептуального моделирования ИС

В этом разделе дается краткая характеристика современных методологий и технологий концептуального моделирования информационных систем, проводится их сравнительный анализ.

Для разработки концептуальной модели выполняются моделирование и анализ (обследование) бизнес-процессов на основе реинжиниринга.

К числу наиболее распространенных методологий моделирования бизнес-процессов относятся:

- методологии структурного анализа и проектирования: IDEF0, DFD, IDEF3;
- методологии объектно-ориентированного анализа и проектирования, основанные на стандартах языка UML;
- методологии, основанные на стандартах нотации BPMN, и др.

Выбор той или иной методологии необходимо обосновать. Целесообразно представить результаты сравнительного анализа различных методологий в таблице.

Приветствуются решения, основанные на интеграции различных методологий моделирования бизнес-процессов.

Следует также выбрать CASE-средство, поддерживающее выбранную методологию моделирования бизнес-процессов.

Раздел 1.3 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»

Данный раздел предназначен для разработки с помощью выбранной методологии графического представления модели бизнес-процесса «КАК ЕСТЬ» (AS-IS).

Модель «КАК ЕСТЬ» отражает существующее состояние исследуемого бизнес-процесса.

Для обеспечения наглядности и простоты понимания модели следует выполнить ее декомпозицию.

Созданная модель «КАК ЕСТЬ» должна быть проанализирована для определения узких мест существующего бизнес-процесса.

Перечисляются недостатки существующего бизнес-процесса. Для их устранения предлагается усовершенствовать бизнес-процесс путем его автоматизации – внедрения новой ИС.

Раздел 1.4 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»

В данном разделе с помощью выбранной методологии строится развернутая модель бизнес-процесса «КАК ДОЛЖНО БЫТЬ» (TO BE).

Модель «КАК ДОЛЖНО БЫТЬ» отражает целевое состояние бизнес-процесса, которое в дальнейшем предполагается претворить в жизнь. В контексте курсовой работы модель «КАК ДОЛЖНО БЫТЬ» является моделью автоматизированного бизнес-процесса.

Для обеспечения наглядности и простоты понимания модели следует выполнить ее декомпозицию. Целесообразно выделить на рисунках новые элементы.

Раздел 1.5 Разработка требований

Данный раздел курсовой работы предназначен для формирования требований, обеспечивающих повышение эффективности исследуемого бизнес-процесса.

Для разработки требований рекомендуется использовать технологию FURPS+.

Следует напомнить, что название технологии FURPS+ происходит от аббревиатуры, представляющей собой усовершенствованную модель для классификации атрибутов качества программного обеспечения (функциональных и нефункциональных требований). Данная технология широко применяется в программной индустрии в настоящее время.

Пример формулировки требований к ИС по технологии FURPS+:

- 1) functionality, функциональность:
 - регистрация пользователей;
 - формирование операционной отчетности;
 - запись данных в базу данных (БД);
- 2) usability, удобство использования: наличие справочной информации;
- 3) reliability, надежность: обеспечение резервного копирования;
- 4) performance, производительность: допустимое количество одновременно работающих пользователей – 20;
- 5) supportability, поддерживаемость:
 - возможность масштабирования;
 - простота инсталляции;
- 6) проектные ограничения: реализация на платформе «1С-Битрикс».

Раздел 1.6 Обзор и анализ аналогов

Рекомендуется выбрать в качестве аналогов готовые ИТ-решения (два три), используемые для автоматизации исследуемой предметной области.

В форме таблицы приводится сравнительный анализ аналогов на предмет соответствия сформулированным в предыдущем разделе требованиям.

Приводится обоснование решения о разработке новой ИС (например, по причине сложности адаптации известных ИТ-решений к специфике предметной области).

Раздел 1.7 Постановка задачи на разработку новой ИС

В данном разделе описывается техническое задание (ТЗ) на разработку ИС.

В качестве основы для разработки ТЗ рекомендуется использовать ГОСТ 34.602–89 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы».

Содержание главы 2

Целью данной части курсовой работы является разработка логической модели.

Логическое моделирование – это стадия проектирования ПО, необходимая для уточнения основных выводов из ее концептуальной модели и постановки задачи на разработку программного обеспечения и модели данных.

Логическая модель – совокупность описания объектов и связей между ними.

Раздел 2.1 Выбор методологии и технологии логического моделирования

В данной части работы дается краткая характеристика современных методологий и технологий логического моделирования информационных систем, проводится их сравнительный анализ.

Рекомендуется использование методологий и технологий объектно-ориентированного анализа и проектирования, основанных на стандартных нотациях языка UML.

В форме таблицы приводится обоснование выбора конкретной методологии и средств моделирования.

Раздел 2.2 Разработка диаграмм логической модели

В данном разделе разрабатываются UML-диаграммы логической модели.

Рекомендуется представить рисунки диаграмм ядра языка UML, отражающие ключевые аспекты логической модели ИС:

- диаграммы вариантов использования;
- диаграммы классов;
- диаграммы последовательности или кооперации.

Приветствуется также разработка расширенного набора диаграмм языка UML, включающего диаграмму состояния, диаграмму деятельности и др.

Раздел 2.3 Разработка логической модели данных

В данном разделе разрабатывается логическая модель данных. Логическая модель данных отображает связи между элементами данных. Она не ориентирована на конкретную СУБД. Примером такой модели является диаграмма «сущность – связь».

Для разработки логической модели данных рекомендуется использовать методологию IDEF1X.

Возможно также применение встроенных опций моделирования данных CASE-средств, поддерживающих методологии объектно-ориентированного анализа и проектирования.

Содержание заключения

В заключении рекомендуется указать, какие задачи были решены, определить пути их внедрения и направления дальнейшего совершенствования.

В данном разделе указываются основные результаты выполнения проекта и сделанные по ним выводы. Обязательным является описание подходов, методик и способов:

- 1) получения исходных данных;
- 2) проведения экспериментов;
- 3) проверки результатов;
- 4) анализа результатов с целью обобщения и формулировки выводов;
- 5) сравнения с другими работами в данной области.

Заключение должно соответствовать (но не дублировать) введению работы. В нем необходимо отразить доказанную актуальность работы, цель и поставленные задачи. По каждой задаче необходимо представить полученные результаты.

Выводы по разделам представляют собой описание конкретных результатов, полученных в ходе работы над конкретным материалом исследования. Этот же принцип реализуется и в рамках написания каждого параграфа главы.

Оформление списка использованной литературы

Указываются литературные источники, использованные при написании проекта в принятой для библиографического описания форме. На все источники должны быть сделаны ссылки в тексте курсовой работы.

Обучающемуся необходимо выполнить курсовую работу по своей теме в соответствии с методическими указаниями по выполнению курсовой работы. Порядок выдачи задания, выполнения и оценки курсовой работы - в соответствии с Положением о курсовой работе (курсовом проекте) ТГУ

Критерии оценки

«Отлично» – курсовая работа выполнена в срок в полном объеме без замечаний.

«Хорошо» – курсовая работа выполнена с незначительными замечаниями по содержанию, срокам, или объёму.

«Удовлетворительно» – курсовая работа выполнена с существенными замечаниями по содержанию, срокам, или объёму.

«Неудовлетворительно» – курсовая работа не выполнена.

7.3. Оценочные средства для промежуточной аттестации по итогам освоения дисциплины

7.3.1. Вопросы к промежуточной аттестации

Семестр 5

№ п/п	Вопросы к экзамену
1	Понятие системного подхода к проектированию ПО .
2	Понятие проекта и проектирования. Цель проектирования.
3	Особенности проектирования ПО.
4	Основные области знаний программной инженерии.
5	Организационные области программной инженерии.
6	Процессы создания ПО по ГОСТ РИСО/МЭК 12207–99.
7	Сущность ГОСТ РИСО/МЭК 12207–99.
8	Основные процессы ЖЦ ПО ГОСТ РИСО/МЭК 12207–99.
9	Вспомогательные процессы ЖЦ ПО ГОСТ РИСО/МЭК 12207–99.
10	Организационные процессы ЖЦ ПО ГОСТ РИСО/МЭК 12207–99.
11	Основные процессы ЖЦ ПО стандарта ISO/IEC 12207.
12	Вспомогательные процессы ЖЦ ПО стандарта ISO/IEC 12207.
13	Разработка ЖЦ ПО с задачами и действиями для процесса тестирования.
14	Задачи процесса тестирования ПО.
15	Каскадная модель ЖЦ ПО.
16	Инкрементная модель ЖЦ ПО.
17	Спиральная модель ЖЦ ПО.
18	Эволюционная модель ЖЦ ПО.
19	Модель быстрой разработки приложений RAD.
20	Модель эволюционного прототипирования.
21	Факторы риска при проектировании ПО.
22	Модель стандартного ЖЦ ПО.
23	Процессы стандартного ЖЦ ПО.
31	Понятие предметной области. Объектный анализ предметной области.
35	Понятие модели и моделирования ПО.
36	Визуальное моделирование ПО.
37	Графические (визуальные модели) архитектуры ПО.
38	Понятие архитектуры ПО.
39	Графические языки моделирования ПО.
40	Виды моделей , используемых при проектировании ПО.
41	Структурные методы анализа и проектирования ПО. Понятие «черного ящика».
42	Понятие иерархической структуры ПО.
43	Особенности метода структурного анализа проектируемого ПО.

№ п/п	Вопросы к экзамену
44	Базовые принципы структурного подхода к проектированию ПО.
45	Метод функционального моделирования SADT (Structured Analysis and Design Technique) (IDEF0). Основные концепции.
46	Правила проектирования ПО, регламентируемые методом SADT.
47	Понятие диаграммы по методу SADT.
48	Порядок построения SADT-модели.
49	Виды связей между функциями в диаграммах SADT.
57	Диаграммы потоков данных DFD.
58	Основные компоненты диаграмм потоков данных.
59	Принципы построения объектной модели ПО.
60	Основные элементы объектной модели ПО.
61	Методы объектно-ориентированных методов анализа и построения моделей ПО.
62	Основные понятия методов объектного анализа проектной области.
63	Проектирование ПО и архитектура системы.
64	Этапы стандартного подхода к проектированию.
65	Уровни архитектуры ПО.
66	Архитектурные схемы проектируемого ПО.
67	Виды диаграмм в соответствии с языком UML.
68	Объектно-ориентированный подход к разработке ПО.
69	UML-метод моделирования ПО.
70	Виды диаграмм в объектно-ориентированном программировании.
71	Компонентный подход к разработке ПО.

7.3.2. Критерии и нормы оценки

Семестр	Форма проведения промежуточной аттестации	Критерии и нормы оценки	
5	Экзамен (по накопительному рейтингу)	«отлично»	Набрано 85 и более баллов
		«хорошо»	Набрано 70..84 баллов
		«удовлетворительно»	Набрано 55..69 баллов
		«неудовлетворительно»	Набрано менее 55 баллов

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Обязательная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1	О. Б. Петрова	Петрова, О. Б. Разработка и анализ требований проектирования программного обеспечения: практикум : учебное пособие / О. Б. Петрова. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2022. — 37 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/279218	Учебное пособие	2022	ЭБС "Лань"
2	А. Куликов, В. Т. Матчин, А. В. Сеницын, В. В. Литвинов	Инструментальное программное обеспечение разработки и проектирования информационных систем : учебное пособие / А. А. Куликов, В. Т. Матчин, А. В. Сеницын, В. В. Литвинов. — Москва : РТУ МИРЭА, 2022. — 263 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/311003 (дата обращения: 17.06.2023). — Режим доступа: для авториз. пользователей.	Учебное пособие	2022	ЭБС "Лань"
3	Мкртычев, С.В.	Мкртычев, С.В. Проектирование информационных систем. Выполнение курсовой работы : электронное учебно-методическое пособие / С.В. Мкртычев [и др.]. — Тольятти : изд-во ТГУ, 2019. — 1оптический диск. - ISBN 978-5-8259-1458-9	Учебно-методическое пособие	2019	Репозиторий ТГУ

8.2. Дополнительная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно- методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1	Л. А. Вдовенко	Влацкая И.В. Проектирование и реализация прикладного программного обеспечения [Электронный ресурс]: учебное пособие/ Влацкая И.В., Заельская Н.А., Надточий Н.С.— Электрон. текстовые данные.— Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2015.— 119 с.— Режим доступа: http://www.iprbookshop.ru/54145.html .— ЭБС «IPRbooks»	учебное пособие	2015	ЭБС «IPRbooks»
2	В. А. Астапчук, П. В. Терещенко	Алексеев В.А. Паттерны проектирования программных систем [Электронный ресурс]: методические указания к проведению лабораторных работ по курсу «Архитектура программных систем»/ Алексеев В.А.— Электрон. текстовые данные.— Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2016.— 33 с.— Режим доступа: http://www.iprbookshop.ru/74412.html .— ЭБС «IPRbooks»	учебное пособие	2016	ЭБС «IPRbooks»
3	Пальмов С. В.	Пальмов С.В. Методы и средства моделирования программного обеспечения [Электронный ресурс]: методические указания к лабораторным	практикум	2016	ЭБС «IPRbooks»

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
		работам/ Пальмов С.В.— Электрон. текстовые данные.— Самара: Поволжский государственный университет телекоммуникаций и информатики, 2016.— 33 с.— Режим доступа: http://www.iprbookshop.ru/71854.html .— ЭБС «IPRbooks»			
4	Пальмов С. В.	Пальмов С.В. Методы и средства моделирования программного обеспечения [Электронный ресурс]: конспект лекций/ Пальмов С.В.— Электрон. текстовые данные.— Самара: Поволжский государственный университет телекоммуникаций и информатики, 2016.— 105 с.— Режим доступа: http://www.iprbookshop.ru/71855.html .— ЭБС «IPRbooks»	конспект лекций	2016	ЭБС «IPRbooks»

8.3. Перечень профессиональных баз данных и информационных справочных систем:

№ пп	Наименование	Ссылка
1	Springer Nature (Полнотекстовая коллекция журналов)	https://www.springernature.com/gp/products
2	Springer eBooks (Полнотекстовая коллекция электронных книг издательства Springer Nature)	https://link.springer.com/
3	«Кодекс»	https://kodeks.ru/
4	ELIBRARY.RU (электронная библиотека научных публикаций)	http://elibrary.ru
5	"Гарант"	https://www.garant.ru/
6	"КонсультантПлюс"	https://www.consultant.ru/
7	Техэксперт	https://cntd.ru/

8.4. Перечень программного обеспечения

№ п/п	Наименование ПО	Реквизиты договора (дата, номер, срок действия)
1	WinPro 10 RUS Upgrd OLP NL Acdmc	Договор № 757 от 04.07.2018, срок действия - бессрочно; Контракт № 1653 от 14.12.2018, срок действия – бессрочно
2	Office Stdandard 2016 Russian OLP NL AcademicEdition	Договор № 757 от 04.07.2018, срок действия – бессрочно
3	Ramus	Свободно распространяемое ПО
4	StarUML	Свободно распространяемое ПО
5	BPMNstudio	Свободно распространяемое ПО
6	VisualParadigm	Свободно распространяемое ПО

8.5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)	Перечень основного оборудования
1	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для проведения лабораторных работ. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций Учебная аудитория для	Компьютер (монитор Samsung Sync Master 943n 19", системный блок Intel (R) Core 2 Quad 2,40 GHz 1 Gb), столы лабораторные, стулья, доска 3-х секционная (меловая), стол преподавательский.

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)	Перечень основного оборудования
	проведения занятий текущего контроля и промежуточной аттестации. (УЛК-407)	
2	Помещение для самостоятельной работы обучающихся (УЛК-105).	Столы, стулья, стеллажи (в т.ч. выставочные) с книгами, компьютеры, мобильные рабочие места.
3	Помещение для самостоятельной работы обучающихся (УЛК-406).	Столы компьютерные, стулья, микрокомпьютеры raspberry pi 32 bit.