Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 09.12.2025 12:51:11
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДАЮ
Декан факультета
«Информационные технологии»
_____ / Д.И. Демидов /
«__» _____ 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Архитектура и дизайн программного обеспечения»

Направление подготовки/специальность
09.03.03 Прикладная информатика

Профиль/специализация
«Разработка и интеграция бизнес-приложений»

Квалификация
бакалавр

Формы обучения
очная

Москва, 2025 г.

Разработчик(и):

ст. преподаватель М.В. Даньшина /

ст.преподаватель /И.В.Кулибаба/

Согласовано:
Заведующий кафедрой «Инфокогнитивные технологии»,
к.т.н., доцент / Е.А. Пухова /

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 5
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость для очной формы обучения 5
3.2 Тематический план изучения дисциплины для очной формы обучения 6
3.3 Содержание дисциплины 6
3.4 Тематика лабораторных занятий 7
4 Учебно-методическое и информационное обеспечение 7
4.1 Нормативные документы и ГОСТы 7
4.2 Основная литература 8
4.3 Дополнительная литература 8
4.4 Электронные образовательные ресурсы 8
4.5 Лицензионное и свободно распространяемое программное обеспечение 8
4.6 Современные профессиональные базы данных и информационные справочные системы 8
5 Материально-техническое обеспечение 8
6 Методические рекомендации 8
6.1 Методические рекомендации для преподавателя по организации обучения 8
6.2 Методические указания для обучающихся по освоению дисциплины 9
7 Фонд оценочных средств 9
7.1 Методы контроля и оценивания результатов обучения 9
7.2 Шкала и критерии оценивания результатов обучения 10
7.3 Оценочные средства 11

1 Цели, задачи и планируемые результаты обучения по дисциплине

Целью учебной дисциплины является изучения свойств систем автоматического контроля и управления и технологических объектов управления; анализ и синтез управляющих устройств; оценка систем автоматического управления на устойчивость и качество; ознакомление с методами расчета типовых законов регулирования и многоконтурных систем управления.

Задачами учебной дисциплины являются формирование и закрепление знаний у студентов о системах автоматического управления, их синтезе и анализе.

Обучение по дисциплине «Архитектура и дизайн программного обеспечения» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	**Индикаторы достижения компетенции**
ПК-3. Способен разрабатывать требования и проектировать программное обеспечение	ИПК-3.1. Знает возможности существующей программно-технической архитектуры; возможности современных и перспективных средств разработки программных продуктов, технических средств; методологии разработки программного обеспечения и технологии программирования; методологии и технологии проектирования и использования баз данных; языки формализации функциональных спецификаций; методы и приемы формализации задач; методы и средства проектирования программного обеспечения; методы и средства проектирования программных интерфейсов; методы и средства проектирования баз данных; принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; методы и средства проектирования программного обеспечения и баз данных; методы и средства проектирования программных интерфейсов. ИПК-3.2. Умеет проводить анализ исполнения требований; вырабатывать варианты реализации требований; проводить оценку и обоснование

	рекомендуемых решений; осуществлять коммуникации с заинтересованными сторонами; выбирать средства реализации требований к программному обеспечению; вырабатывать варианты реализации программного обеспечения; проводить оценку и обоснование рекомендуемых решений; осуществлять коммуникации с заинтересованными сторонами; использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; осуществлять коммуникации с заинтересованными сторонами. ИПК-3.3. Владеет современным инструментарием и средами разработки программного кода; современным инструментарием и средами проектирования программного кода, методами тестирования ПО.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений блока Б1.2, модуля «Разработка в области информационных технологий» и междисциплинарно связана с поддерживающими дисциплинами: основы программирования, алгоритмы и структуры данных и последующими дисциплинами: методы машинного обучения.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы (144 часа).

3.1 Виды учебной работы и трудоемкость для очной формы обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			6	
1	**Аудиторные занятия**	**72**	72	
	В том числе:			
1.1	Лекции	36	36	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	36	36	
2	**Самостоятельная работа**	**72**	72	

3	**Промежуточная аттестация**			
	Диф. зачет		Диф.зачет	
	Итого:	**144**	144	

3.2 Тематический план изучения дисциплины для очной формы обучения

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Введение в архитектуру программного обеспечения	16	4		4		8
2	Гибкие методологии и архитектура программного обеспечения	16	4		4		8
3	Jobs To Be Done. Технические средства и инструменты обеспечивающие процесс проектирования ПО.	16	4		4		8
4	Архитектура программного обеспечения при объектно-ориентированном подходе к программированию. UML. Принципы ООП	16	4		4		8
5	Функциональная структура	16	4		4		8
6	Архитектурные шаблоны	16	4		4		8
7	Проектирование систем	16	4		4		8
8	Управление качеством программного обеспечения	16	4		4		8
9	Проектирование интерфейсов пользователей	16	4		4		8
Итого		**144**	**36**		**36**		**72**

3.3 Содержание дисциплины

1. Введение в архитектуру и дизайн программного обеспечения
2. Принципы проектирования программного обеспечения
3. Модульность и разделение ответственности
4. Повторное использование кода и компонентов
5. Архитектурные шаблоны и стили
6. Клиент-серверная архитектура
7. Многоуровневая архитектура
8. Шаблон MVC (Model-View-Controller)

9. Проектирование интерфейсов пользователя
10. Диаграммы классов и объектов
11. Диаграммы последовательности и взаимодействия
12. Диаграммы компонентов и развертывания
13. Проектирование баз данных в контексте разработки архитектуры сложного программного обеспечения
14. Архитектура сетевых и распределенных систем
15. Архитектурные стили корпоративных приложений
16. Сервис-ориентированные архитектуры
17. Облачные архитектуры
18. Тестирование и отладка программного обеспечения
19. Обеспечение качества программного обеспечения
20. Управление изменениями и конфигурацией
21. Разработка масштабируемых и поддерживаемых систем
22. Работа в команде и коллаборативная разработка
23. Тенденции и новые направления в архитектуре программного обеспечения

3.4 Тематика лабораторных занятий

Введение в архитектуру программного обеспечения/

Гибкие методологии и архитектура программного обеспечения/

Jobs To Be Done. Технические средства и инструменты, обеспечивающие процесс проектирования ПО.

Архитектура программного обеспечения при объектно-ориентированном подходе к программированию. UML. Принципы ООП

Функциональная структура

Архитектурные шаблоны

Проектирование систем

Управление качеством программного обеспечения

Проектирование интерфейсов пользователей

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);

2. Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика, утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 922.

3. Приказ Министерства образования и науки РФ от 05 апреля 2017 г. № 301 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры;

4. Порядок проведения государственной итоговой аттестации по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры, утвержденный приказом Минобрнауки России от 29 июня 2015 г. № 636;

5. Положение о практической подготовке обучающихся, утвержденное приказом Министерства науки и высшего образования Российской Федерации и Министерства просвещения Российской Федерации от 5 августа 2020 г. № 885/390.

4.2 Основная литература

Соснин, П. И. Архитектурное моделирование автоматизированных систем / П. И. Соснин. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 180 с. — ISBN 978-5-507-46075-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/297017 (дата обращения: 24.09.2023). — Режим доступа: для авториз. пользователей.

4.3 Дополнительная литература

Зорин, Л. Б. Архитектурное моделирование информационных систем предприятий и организаций : учебное пособие / Л. Б. Зорин. — Москва : РТУ МИРЭА, 2021. — 74 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/218723 (дата обращения: 24.09.2023). — Режим доступа: для авториз. пользователей.

4.4 Электронные образовательные ресурсы

ЭОР https://online.mospolytech.ru/enrol/index.php?id=11863

ЭБС IPR SMART (https://www.iprbookshop.ru/)

ЭБС Лань (https://e.lanbook.com/)

Образовательная платформа Юрайт. Для вузов и ссузов (https://urait.ru/)

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Visual Studio Code
2. Браузеры Chrome, Edge, Firefox
3. MS Visio Система проектирования

4.6 Современные профессиональные базы данных и информационные справочные системы

https://www.elibrary.ru/

5 Материально-техническое обеспечение

Для проведения лабораторных работ и самостоятельной работы студентов подходят аудитории, оснащенные компьютерами с программным обеспечением в соответствии со списком в пункте 4.5 и подключенные к интернету.

Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов.

Рабочее место преподавателя должно быть оснащено компьютером с подключенным к нему проектором или иным аналогичным по функциональному назначению оборудованием.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются лекции и самостоятельная работа.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;
- оформление материала в соответствии с требованиями.

Приветствуется обсуждение самих заданий с другими студентами: можно как давать, так и получать советы по общей стратегии выполнения и изучения материала, давать и получать помощь в отладке. Однако писать код студент должен самостоятельно. Делиться кодом или писать его совместно запрещено.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

Приведенные ниже правила выставления оценок и опозданий могут быть изменены, если преподаватель сочтет это необходимым. Важно, чтобы студенты регулярно просматривали план курса, выложенный в СДО, на предмет его обновления или изменения.

Достижение компетенций оценивается с помощью лабораторных работ и рубежных контролей. Индикаторы ИПК-3.1, ИПК-3.2, ИПК-3.3 заложены в темах 1-12.

В соответствии с планом на дисциплины студентам выдаются задания на лабораторные работы. Помимо требований и описания функционала в работе указан крайний срок сдачи. Для сдачи лабораторной работы студенту необходимо прислать ссылку на репозиторий в GitHub Classroom и на хостинг, где размещен результат работы с реализованным функционалом, описанным в задании. Работа считается сданной если в ней реализовано 80% и более требований и функционала, описанного в задании.

Каждый студент имеет право на 6 дней опоздания, которые могут быть потрачены на любые задания в течение семестра. Опоздания предназначены для решения особых ситуаций, таких как болезнь или чрезвычайные семейные обстоятельства.

Когда использованы все дни опоздания за каждый день просрочки начисляется штраф в размере 25% от максимального результата за задание. Задания, присланные позже, чем 4 дня, не будут оцениваться. В связи с зависимостью между работами студентам может потребоваться все равно выполнить предыдущие работы, даже если они не оцениваются.

После сдачи лабораторной работы студент должен ее защитить. Во время защиты лабораторной работы преподаватель проверяет репозиторий, хостинг и выполнение критериев и требований задания, а студент отвечает на вопросы преподавателя по его коду, а также теоретических вопросов, приведенных после текста задания лабораторной работы. Если студент отказывается отвечать на вопросы, или дает полностью неверные ответы, или ответы не по теме, то работа может считаться сданной, но при этом она не оценивается.

Работа должна быть выполнена студентом самостоятельно: в репозитории в системе контроля версий студента содержатся коммиты только за его авторством, по этим коммитам можно проследить как велась работа, студент может объяснить свой код и ход выполнения работы, если эти правила не соблюдаются, то работа не считается сданной и не оценивается.

Рубежные контроли пишутся в аудитории индивидуально по варианту задания, выданному преподавателем в назначенные дни. При отсутствии студента в день написания контрольной работы ему дается еще один шанс ее написать на последнем занятии в семестре, но обязательно очно.

Студенты должны заранее сообщать о том, что у них могут возникнуть трудности со своевременной сдачей задания или проекта. При наличии реальных причин задержки студентам следует как можно скорее связаться с преподавателем и обсудить возможные условия.

7.2 Шкала и критерии оценивания результатов обучения

Лабораторная работа оценивается в процентах степени выполнения следующих критериев и для выставления оценки суммируются проценты за каждый из четырех критериев:

1. Полнота выполнения практического задания (30%): соответствует ли функциональность заданным требованиям и целям, насколько точно и без ошибок код выполняет поставленные задачи, насколько эффективно задание отвечает требованиям целевой аудитории и обеспечивает приятное восприятие.

2. Качество и структура кода (10%): качество, читаемость и организация кода, рациональность выполнения задания, последовательность именования и соблюдение лучших практик.

3. Творчество и инновации (10%): творческий подход студентов к выполнению заданий, насколько студенты вышли за рамки основных требований и реализовали дополнительные возможности или использовали уникальные решения.

4. Ответы на вопросы по коду студента и теории (50%):

Дает краткий ответ, содержащий ошибки или неточности. На наводящие вопросы отвечает неправильно (10% из 50%)

Дает развернутый ответ, содержащий ошибки или неточности. На наводящие вопросы отвечает неверно (20% из 50%)

Дает развернутый ответ, содержащий ошибки или неточности. На наводящие вопросы отвечает правильно (30% из 50%)

Дает правильные и развернутые ответы на вопросы (50% из 50%).

R лабораторные рассчитывается как среднее результатов за все лабораторные работы. За полное и безошибочное выполнение всех лабораторных работ в срок и их защиту можно получить максимум 100 баллов (R лабораторные).

Рубежный контроль оценивается по следующим критериям:

Полнота выполнения практического задания: соответствует ли функциональность заданным требованиям и целям, насколько точно и без ошибок код выполняет поставленные задачи.

Качество и структура кода: качество, читаемость и организация кода, рациональность выполнения задания, последовательность именования и соблюдение лучших практик.

Творчество и инновации: творческий подход студентов к выполнению заданий, насколько студенты вышли за рамки основных требований и реализовали дополнительные возможности или использовали уникальные решения.

Пользовательский опыт: отзывчивость, доступность, насколько эффективно задание отвечает требованиям целевой аудитории и обеспечивает приятное восприятие.

Самостоятельность решения: в репозитории студента есть коммиты только за его авторством, по коммитам в репозитории можно проследить как велась работа, студент может объяснить свой код и ход выполнения работы, если эти правила не соблюдаются, то работа не считается сданной.

Более подробное описание критериев дается в тексте задания рубежного контроля.

За полностью выполненные рубежные контроли также можно получить 100 баллов (R контроль).

Также имеется коэффициент сданных работ K сданные, который равен 1 если все работы сданы и 0 если хотя бы одна работа не сдана.

Итоговый балл рассчитывается по формуле: R сем = (0,5 x R лабораторные + 0,5 x R контроль) * K сданные.

Итоговый балл пересчитывается по шкале ниже и на основании полученной оценки фиксируется результат промежуточной аттестации.

Соответствие баллов в 100 балльной рейтинговой системе оценке по 4-бальной шкале:
0-54 - неудовлетворительно
55-69 - удовлетворительно
70-84 - хорошо
85-100 – отлично

7.3 Оценочные средства

7.3.1 Текущий контроль

Примерный список вопросов

1. Для чего определяются высокоуровневые требования к системе? Какой документ для этого используется? Каково содержание данного документа?
2. На основании чего формулируются потребности пользователей?
3. Какую роль в формировании требований играет изучение проблем предприятия?
4. Каким образом производится анализ пользователей будущей ИС?
5. Какие ограничения могут быть наложены на процесс проектирования ИС?
6. На основании каких показателей определяется применимость ИС?
7. На основании каких показателей определяется надежность ИС?
8. Для чего определяются специальные требования к системе? Какой документ для этого используется? Каково содержание

данного документа?

9. В чем отличи специальных требований от высокоуровневых?
10. Какие показатели определяют функциональность системы?
11. На основании чего определяется производительность будущей ИС?
12. Какие показатели определяют пригодность к эксплуатации ИС?
13. Что такое эксплуатационные требования к ИС? Что они определяют?
1. Что такое верификация требований к ИС?
2. Какая типизация требований используется?
3. Каким образом производится количественное оценивание требований?
4. По каким критериям оцениваются требования к ИС?

5. Какой документ используется для верификации требований? На основании чего оценивается качество создания этого документа?

7.3.2 Промежуточная аттестация

1. Понятие и проблемы программной инженерии.
2. Типы проектов информационных систем.
3. Этапы жизненного цикла (ЖЦ) программного обеспечения (ПО).
4. Каскадные и итеративные модели ЖЦ ПО.
5. Этапы и процессы ЖЦ ПО.
6. Стандарт ГОСТ Р ИСО/МЭК 12207.
7. Модель зрелости процессов создания ПО.
8. Стандарты ISO 15288 и ISO 15504. Показатели качества ПО.
9. Стандарт ISO 9126. Управление конфигурацией ПО.
10. Методология обеспечения качества ПО.
11. Документирование ПО в соответствии с ЕСПД. основные положения "Руководства к своду знаний по программной Инженерии" SWEBOK.
12. Стили и параметры разработки ПО.
13. Методологии программирования.
14. Особенности разработки сложных программных систем: иерархичность, групповая разработка, сборочное проектирование.
15. Преимущества и недостатки объектно-ориентированного подхода.
16. Разработка, управляемая моделями.
17. Унифицированный язык моделирования UML.
18. Структурные и поведенческие диаграммы.
19. Методология быстрой разработки приложений (RAD), методологии унифицированного процесса разработки Rational Unified Process (RUP) и экстремального программирования.
20. История автоматизации проектирования ПО.
21. Инструментальным средства и системы (CASE) разработки программного обеспечения.
22. Классификация CASE-систем и их сравнительная характеристика.
23. Средства программирования, управления программным проектом, верификаторы, документаторы.
24. Тестирование программ. Разработка тестов. Оценка тестируемости ПО.
25. Виды тестирования. Структурное и функциональное тестирование.
26. Особенности объектно-ориентированного тестирования.
27. Примеры инструментальных систем разработки ПО.
28. Виды параллельного взаимодействия. Многопоточность и многозадачность.
29. Интерфейс прикладного программирования OpenMP.
30. Технология программирования для параллельных компьютеров с распределенной памятью MPI.
31. Методика параллельного программирования PVM

7.3.3. Экзаменационные вопросы

1. Что такое архитектура программного обеспечения?
2. Какие принципы проектирования программного обеспечения вы знаете?
3. Что такое модульность и как она связана с архитектурой программного обеспечения?
4. Какие архитектурные шаблоны вы знаете и для чего они используются?
5. Что такое клиент-серверная архитектура и как она работает?
6. Какие архитектурные стили используются в корпоративных приложениях?
7. Что такое сервис-ориентированная архитектура (SOA)?
8. Какие преимущества и недостатки облачных архитектур?
9. Что такое анализ требований и как он связан с архитектурой программного обеспечения?
10. Какие диаграммы используются для визуализации архитектуры программного обеспечения?
11. Какие методы тестирования программного обеспечения вы знаете?
12. Что такое обеспечение качества программного обеспечения и как оно связано с архитектурой?
13. Как управлять изменениями и конфигурацией в архитектуре программного обеспечения?
14. Как разрабатывать масштабируемые и поддерживаемые системы?
15. Какие новые тенденции и направления существуют в архитектуре программного обеспечения?
16. Какие основные принципы модульности в программировании?
17. Что такое архитектурный стиль и как он влияет на разработку программного обеспечения?
18. Какие преимущества и недостатки имеет многоуровневая архитектура?
19. Что такое шаблон проектирования MVC и как он применяется в разработке программного обеспечения?
20. Какие методы анализа требований используются при проектировании программного обеспечения?
21. Что такое диаграмма классов и как она помогает в проектировании программного обеспечения?
22. Какие принципы разработки интерфейсов пользователя следует учитывать при проектировании программного обеспечения?
23. Что такое диаграмма последовательности и как она помогает в проектировании программного обеспечения?
24. Какие принципы тестирования программного обеспечения следует учитывать при разработке архитектуры?
25. Что такое обеспечение качества программного обеспечения и какие методы используются для его достижения?
26. Какие инструменты используются для управления изменениями и конфигурацией в архитектуре программного обеспечения?
27. Какие принципы разработки масштабируемых и поддерживаемых систем следует учитывать при проектировании архитектуры?
28. Какие методы коллаборативной разработки программного обеспечения существуют?
29. Какие новые технологии и подходы в архитектуре программного обеспечения стали популярными в последнее время?
30. Что такое микросервисная архитектура и как она отличается от других архитектурных стилей?
31. Какие принципы безопасности следует учитывать при проектировании архитектуры программного обеспечения?
32. Что такое архитектура сетевых и распределенных систем и как она влияет на проектирование программного обеспечения?

33. Какие принципы разработки мобильных приложений следует учитывать при проектировании архитектуры?
34. Что такое архитектура реактивных систем и как она применяется в разработке программного обеспечения?
35. Какие принципы разработки игровых приложений следует учитывать при проектировании архитектуры?
36. Что такое архитектура интернета вещей (IoT) и как она влияет на проектирование программного обеспечения?
37. Какие принципы разработки веб-приложений следует учитывать при проектировании архитектуры?
38. Что такое архитектура блокчейна и как она влияет на проектирование программного обеспечения?
39. Какие принципы разработки искусственного интеллекта следует учитывать при проектировании архитектуры?
40. Что такое архитектура машинного обучения и как она влияет на проектирование программного обеспечения?
41. Какие принципы разработки робототехнических систем следует учитывать при проектировании архитектуры?
42. Что такое архитектура виртуализации и как она влияет на проектирование программного обеспечения?
43. Какие принципы разработки автономных систем следует учитывать при проектировании архитектуры?
44. Что такое архитектура графических приложений и как она влияет на проектирование программного обеспечения?
45. Какие принципы разработки систем управления базами данных следует учитывать при проектировании архитектуры?
46. Что такое архитектура распределенных вычислений и как она влияет на проектирование программного обеспечения?
47. Какие принципы разработки систем машинного зрения следует учитывать при проектировании архитектуры?
48. Что такое архитектура систем управления проектами и как она влияет на проектирование программного обеспечения?
49. Какие принципы разработки систем управления контентом следует учитывать при проектировании архитектуры?
50. Что такое архитектура систем управления процессами и как она влияет на проектирование программного обеспечения?

7.3.4. Типовые практические задания

1. Разработка архитектуры для веб-приложения, учитывая требования к масштабируемости и безопасности.
2. Проектирование архитектуры для распределенной системы, включая выбор протоколов и механизмов коммуникации между компонентами.
3. Создание диаграмм классов и последовательностей для системы управления базами данных.
4. Разработка архитектуры для мобильного приложения, учитывая ограничения ресурсов и требования к производительности.
5. Проектирование архитектуры для системы обработки больших данных, включая выбор подходящих технологий и алгоритмов.
6. Создание диаграмм компонентов и развертывания для системы электронной коммерции.

7. Разработка архитектуры для системы управления проектами, включая модули для планирования, отслеживания и отчетности.
8. Проектирование архитектуры для системы управления контентом, учитывая требования к гибкости и масштабируемости.
9. Создание диаграмм состояний и активностей для системы автоматизации процессов бизнеса.
10. Разработка архитектуры для системы мониторинга и управления сетевой инфраструктурой.

7.3.5 Типовой экзаменационный билет

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1
по дисциплине
«Архитектура и дизайн программного обеспечения»
направление подготовки 09.03.03 Прикладная информатика

ВОПРОСЫ:

1. Что такое микросервисная архитектура и как она отличается от других архитектурных стилей?
2. Какие методы коллаборативной разработки программного обеспечения существуют?
3. Что такое сервис-ориентированная архитектура (SOA)?

Утверждено: _____ / _____ / «__» _____ 20__г.