

Документ подписан простой электронной подписью
Информация о владельце: **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 33.10.2023 13:09:16
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

УТВЕРЖДЕНО

Декан факультета

Информационных технологий



/ А.Ю. Филиппович /

«31» июня 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Внедрение и сопровождение программного обеспечения

Направление подготовки/специальность

09.03.03

Прикладная информатика

Профиль/специализация

Корпоративные информационные системы

Квалификация

Бакалавр

Формы обучения

очная

Москва, 2020 г.

1 Цели, задачи и планируемые результаты обучения по дисциплине

Цели:

- Понимать основы межсистемных интеграций и технологий для использования в работе;
- Готовить интеграционные решения, отвечающие потребностям бизнеса;
- Готовить качественные интеграционные требования с учетом технических и бизнес-ограничений;
- Использовать инструменты аналитики для проработки межсистемного взаимодействия;
- Использовать методы и подходы к интеграции, применяя инструменты и шаблоны.

Задачи: сформировать навыки, которые дают возможность участвовать в проектировании интеграционного взаимодействия ИТ-систем с пониманием шаблонов и технологий интеграции и использованием подходящих инструментов.

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

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	ОПК-2.1. Знает современные информационные технологии и методы их использования при решении задач профессиональной деятельности. ОПК-2.2. Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности. ОПК-2.3. Владеет способами применения необходимых информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.
ПК-1 Способен разрабатывать требования и проектировать программное обеспечение	ПК-1.1. 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,

	<p>используемые при разработке программного обеспечения; методы и средства проектирования программного обеспечения и баз данных; методы и средства проектирования программных интерфейсов.</p> <p>ПК-1.2. Умеет проводить анализ исполнения требований; выработать варианты реализации требований; проводить оценку и обоснование рекомендуемых решений; осуществлять коммуникации с заинтересованными сторонами; выбирать средства реализации требований к программному обеспечению; выработать 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</p> <p>ПК-1.3. Владеет современным инструментарием и средами разработки программного кода; современным инструментарием и средами проектирования программного кода</p>
--	---

2 Место дисциплины в структуре образовательной программы

Дисциплина «Внедрение и сопровождение программного обеспечения» относится к блоку Б1.2.ЭД1.

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками:

- Основы Проектирования бизнес - процессов и структур;
- Шаблоны проектирования приложений;
- Проектная деятельность.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных(е) единиц(ы) (144_ часа).

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			7	
1	Аудиторные занятия	72	72	
	В том числе:			

1.1	Лекции			
1.2	Семинарские/практические занятия	72	72	
1.3	Лабораторные занятия			
2	Самостоятельная работа	72	72	
3	Промежуточная аттестация			
	Курсовой проект	КП	КП	
	Зачет/диф.зачет/экзамен	экзамен	экзамен	
	Итого:	144	144	

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					Самостоятельная работа
		Всего	Аудиторная работа				
			Лекции	Семинарские/практические занятия	Лабораторные занятия	Практическая подготовка	
1	Основы интеграции информационных систем			8			8
2	Анализ требований			8			8
3	Модели архитектуры информационных систем			8			8
4	Данные/Анализ моделей и потоков данных			8			8
5	Интеграция/Проектирование межсистемного взаимодействия			8			8
6	Проектирование интеграции через шину, брокер			10			10
7	Проектирование интеграции через REST-API			10			10
8	Курсовой проект			12			12
Итого		144		72			72

3.3 Содержание дисциплины

1 Основы интеграции информационных систем

1.1 Общее представление о понятии интеграции ИТ систем. Определения понятия.

1.2 Виды интеграционных решений

1.3 Основные шаблоны интеграции ИТ систем: файловый обмен, RPCI, общая база данных, интеграционные платформы

2 Анализ требований

2.1 Виды требований

2.2 Анализ требований и выделение User Stories, связанных с интеграциями.

2.3 Анализ требований и Разработка текстовых сценариев интеграции (Use Cases)

2.4 Разработка функциональных требований к интеграции

- 2.5 Разработка нефункциональных требований к интеграции
- 3. Модели архитектуры информационных систем
 - 3.1 Классификация архитектурных решений
 - 3.2 Верхнеуровневое проектирование архитектуры. Нотация C4.
- 4 Данные/Анализ моделей и потоков данных
 - 4.1 Анализ концептуальной модели данных
 - 4.2 Разработка диаграмм потоков данных (Data Flow Diagram)
 - 4.3 Описание преобразований данных (Data Mapping)
- 5 Интеграция/Проектирование межсистемного взаимодействия
 - 5.1 Разработка диаграммы последовательности (UML Sequence Diagram)
 - 5.2 Описание запросов и ответов SOAP
 - 5.3 Форматы XML и XSD, описание WSDL
 - 5.4 Описание форматов данных JSON
- 6 Проектирование интеграции через шину, брокер
 - 6.1 Интеграционные шины (ESB)
 - 6.2 Брокеры сообщений, Rabbit MQ, Apache Kafka
- 7 Проектирование интеграции через REST-API
 - 7.1 Виды API для интеграции ИС. Конечная точка. Сравнение SOAP с REST и GraphQL
 - 7.2 Механизмы авторизации и аутентификации. OAuth и OIDC
 - 7.3 Принципы и правила проектирования API

3.4 Тематика семинарских/практических и лабораторных занятий.

3.4.1 Лабораторные занятия

- 4 Знакомство с основами интеграции информационных систем
- 5 Анализ требований
- 6 Изучение модели архитектуры информационных систем
- 7 Данные/Анализ моделей и потоков данных
- 8 Интеграция/Проектирование межсистемного взаимодействия
- 9 Проектирование интеграции через шину, брокер
- 10 Проектирование интеграции через REST-API

3.4.2 Тематика курсовых проектов (курсовых работ)

Общая тематика формулируется следующим образом: Создание приложения на микросервисной Архитектуре

Выбор предметной области – свободный. Ограничений по стеку (язык программирования/фреймворк/база данных/фронтенд) – нет, внешнее API может быть реализовано с помощью JSON-API/GraphQL.

4 Учебно-методическое и информационное обеспечение

4.1. Основная литература

Трутнев, Д. Р. Архитектуры информационных систем. Основы проектирования : учебное пособие / Д. Р. Трутнев. — Санкт-Петербург : НИУ ИТМО, 2012. — 66 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/70810> . — Режим доступа: для авториз. пользователей.

4.2. Дополнительная литература

Баранова, О. М. Интеграция информационных систем : учебно-методическое пособие / О. М. Баранова. — Москва : МИСИ – МГСУ, 2020. — 47 с. — ISBN 978-5-7264-3096-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/342467> . — Режим доступа: для авториз. пользователей.

4.3. Лицензионное и свободно распространяемое программное обеспечение

Для выполнения лабораторных работ и самостоятельной работы необходимо следующее программное обеспечение:

Microsoft Windows.
Офисные приложения, Microsoft Office.
Веб-браузер, Chrome.
Microsoft Visio.

4.4. Современные профессиональные базы данных и информационные справочные системы

1. Электронно-библиотечные системы (ЭБС) Московского политеха

<https://mospolytech.ru/obuchauschimsya/biblioteka/> Это бесплатный индивидуальный неограниченный доступ к электронно-библиотечным системам (ЭБС). Включают в себя учебники, учебные пособия, практикумы, монографии, периодические издания по всем отраслям знаний, художественную литературу. Условия доступа: Регистрация пользователей в ЭБС осуществляется с любого компьютера, или мобильного устройства, имеющих выход в интернет. В ЭБС размещены также учебные видео, аудиоиздания, имеются различные сервисы для преподавателей и студентов

- [Образовательная платформа «Юрайт»](#)
- [Электронно-библиотечная система IPR SMART](#)
- [Электронно-библиотечная система «Лань»](#)

Доступ осуществляется с компьютеров библиотеки университета

- [Научная электронная библиотека eLIBRARY.RU](#)
- [ЭБС Polpred.com](#)

Отечественные электронные ресурсы

- [Национальная электронная библиотека](#)

Доступ к НЭБ осуществляется с компьютеров библиотеки университета

- [Президентская библиотека им. Б.Н. Ельцина](#)

2. Федеральная государственная информационная система - Национальная электронная библиотека (НЭБ) <https://нэб.рф>

5. Материально-техническое обеспечение

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащено современным компьютером с подключенной к нему электронной доской.

6. Методические рекомендации

6.1. Методические рекомендации для преподавателя по организации обучения

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 для лекций, задачи для лабораторных работ и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

При проверке работ и отчетов следует учитывать не только правильность выполнения заданий Лабораторных работ, 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 аудиторские занятия и лекции, материалы лабораторных работ.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторских занятий, дорабатывают конспекты и записи, готовятся к проведению и обрабатывают результаты лабораторных работ, 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 в области Веб-технологий.

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 аудиторских занятиях, промежуточный контроль осуществляется в письменной (устной) форме.

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.

7. Фонд оценочных средств

7.1. Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- Опросы разделов
- Контрольные вопросы разделов
- Подготовка к Лабораторным работам и их защита
- Выполнение и защита курсового проекта
- Подготовка к Экзамену

Образцы заданий для проведения текущего контроля, банка тестовых заданий приведены в Разделе 7.3.1.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой дисциплины.

Образцы вопросов для проведения промежуточных аттестаций приведены в Разделе 7.3.2.

7.2. Шкала и критерии оценивания результатов обучения

Шкала оценивания промежуточной аттестации

Шкала оценивания	Описание
Отлично	Выполнены все обязательные условия подготовки студента к промежуточной аттестации , предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при переносе знаний и умений на новые, нестандартные задачи.
Хорошо	Выполнены все обязательные условия подготовки студента к промежуточной аттестации , предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками. В то же время при ответе допускает несущественные погрешности, задачи решает с недочетами, не влияющими на общий ход решения.

Удовлетворительно	Выполнены все обязательные условия подготовки студента к промежуточной аттестации , предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками. Но показывает неглубокие знания, при ответе не допускает грубых ошибок или противоречий, однако в формулировании ответа отсутствует должная связь между анализом, аргументацией и выводами, в решении задач могут содержаться грубые ошибки. Для получения правильного ответа требуются уточняющие вопросы.
Неудовлетворительно	Не выполнены обязательные условия подготовки студента к промежуточной аттестации , предусмотренные программой дисциплины, ИЛИ студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями

7.3. Оценочные средства

7.3.1 Текущий контроль

Оценка за каждую лабораторную выставляется исходя из фактического выполнения всех поставленных задач по шкале "зачет/незачет".

7.3.1. Курсовой проект

Задание

Тема для ИС – свободная. Ограничений по стеку (язык программирования/фреймворк/база данных/фронтенд) – нет, внешнее API может быть реализовано с помощью JSON-API/GraphQL. Требования к архитектуре Сервисы должны взаимодействовать между собой синхронно и асинхронно. Синхронное взаимодействие между сервисами реализуется с помощью JSON RPC 2.0, асинхронное взаимодействие – по средствам некоторого databus (Apache Kafka/RabbitMQ). Запуск сервисов происходит только в Docker-контейнерах, а в качестве оркестратора контейнеров используется Minikube или Docker-compose. Маршрутизация внешних запросов на внутренние сервисы должна быть ограничена с помощью единого Nginx, который включает в себя маршрутизацию на все публичные сервисы. База данных не должна являться подом в сервисе, а поднята как отдельный контейнер.

Требования к отдельному сервису

Сервис должен иметь:

- минимум 1 реляционную БД, например PostgreSQL;
- горячий кэш/Key-Value хранилище (Redis/MongoDB)1;
- собственный nginx;
- систему очередей2 (обработка задач, с отложенным запуском);
- описание всех методов с помощью OpenAPI 3, с разделением на внутреннее и внешнее API.

Оценивание:

«Удовлетворительно»

- Реализовано 3 сервиса, 1 из которых является потребителем 2х других
- Взаимодействие между сервисами происходит только по HTTP (JSON-RPC)
- Шлюз предоставляет внешнее API

- Сделан образ Docker для запуска сервиса (БД может быть запущена вне контейнера)
- Набор сервисов выполняет хотя бы 1 сложный бизнес-процесс (например: хранение заметок – 1й сервис хранит заметки и может отдавать их в различных форматах для отображения, 2й сервис хранит пользователей и права доступа, а 3й сервис реализует sharing заметок между пользователями, создание публичных ссылок и тп)

«Хорошо»

- Реализовано более 3х сервисов, 1 из которых является потребителем 2х других
- Взаимодействие между сервисами происходит только по HTTP (JSON-RPC)
- Шлюз предоставляет внешнее API
- Сделан образ Docker для запуска сервиса (БД может быть запущена вне контейнера)
- Набор сервисов выполняет хотя бы 1 сложный бизнес-процесс (например: хранение заметок – 1й сервис хранит заметки и может отдавать их в различных форматах для отображения, 2й сервис хранит пользователей и права доступа, а 3й сервис реализует sharing заметок между пользователями, создание публичных ссылок и тп)

«Отлично»

Выполнены все требования к курсовому проекту в полном объеме, студент показывает владение материалов на высоком уровне

7.3.2. Примерные вопросы к экзамену

8. Общее представление о понятии интеграции ИТ систем. Определения понятия.
9. Виды интеграционных решений
10. Основные шаблоны интеграции ИТ систем: файловый обмен, RPCI, общая база данных, интеграционные платформы
11. Виды требований
12. Анализ требований и выделение User Stories, связанных с интеграциями.
13. Анализ требований и Разработка текстовых сценариев интеграции (Use Cases)
14. Разработка функциональных требований к интеграции
15. Разработка нефункциональных требований к интеграции
16. Модели архитектуры информационных систем
17. Классификация архитектурных решений
18. Верхнеуровневое проектирование архитектуры. Нотация C4.
19. Данные/Анализ моделей и потоков данных
20. Анализ концептуальной модели данных
21. Разработка диаграмм потоков данных (Data Flow Diagram)
22. Описание преобразований данных (Data Mapping)
23. Интеграция/Проектирование межсистемного взаимодействия
24. Разработка диаграммы последовательности (UML Sequence Diagram)
25. Описание запросов и ответов SOAP
26. Форматы XML и XSD, описание WSDL
27. Описание форматов данных JSON
28. Проектирование интеграции через шину, брокер
29. Интеграционные шины (ESB)

30. Брокеры сообщений, Rabbit MQ, Apache Kafka
31. Проектирование интеграции через REST-API
32. Виды API для интеграции ИС. Конечная точка. Сравнение SOAP с REST и GraphQL
33. Механизмы авторизации и аутентификации. OAuth и OIDC
34. Принципы и правила проектирования API