Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 13.11.2023 18:00:24
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

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

Факультет информационных технологий

УТВЕРЖДАЮ
Декан факультета
«Информационные технологии»
_____/Д.Г.Демидов/
«18» мая 2022

Рабочая программа дисциплины

«Технологии конструирования программного обеспечения»

Направление подготовки
09.03.01 Информатика и вычислительная техника

Профиль
«Системная и программная инженерия»

Квалификация
Бакалавр

Формы обучения
очная

Москва, 2022 г.

Разработчик(и):

старший преподаватель кафедры «Информационная безопасность»

/А.Ю. Гневшев/

Согласовано:
И.о. заведующего кафедрой «Информационная безопасность»,

/А.Ю. Гневшев/

Руководитель образовательной программы,

/А.Ю. Гневшев/

Заведующий кафедрой «Инфокогнитивные технологии»,

доцент, к.т.н. /Е.А.Пухова/

1 Цели, задачи и планируемые результаты обучения по дисциплине

Цель изучения дисциплины – ознакомить студентов с основными принципами и методами разработки программного обеспечения. В ходе изучения этой дисциплины студенты получают знания о различных технологиях, инструментах и подходах, которые используются при проектировании, разработке и тестировании программного обеспечения.

Обучение по дисциплине «Технологии конструирования программного обеспечения» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ИОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы. ИОПК-8.2. Умеет проектировать блок-схемы алгоритмов, оценивать производительность алгоритмов и затраты памяти на работу алгоритма, разрабатывать программы на основе спроектированного алгоритма и проводить отладку программы, применять методы системного анализа и математического моделирования при разработке и эксплуатации ИС, проводить структурный анализ, функциональный анализ, объектно-ориентированный анализ иерархии классов. осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы. ИОПК-8.3. Владеет навыками разработки программ, построения блок-схем алгоритмов и оценки производительности алгоритмов, работы с унифицированным языком визуального моделирования, составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к базовой части основной профессиональной образовательной программы бакалавриата.

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

- Управление разработкой программного обеспечения;
- Анализ и реверс инжиниринг программных средств.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зачетных единицы, т.е. 180 академических часа (из них 108 часов – самостоятельная работа студентов и 72 часа – аудиторные занятия).

Разделы дисциплины изучаются в пятом семестре обучения, т.е. на четвертом курсе. Форма контроля – экзамен.

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			5	
1	**Аудиторные занятия**	**72**	72	
	В том числе:			
1.1	Лекции			
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	72	72	
2	**Самостоятельная работа**	**108**	108	
3	**Промежуточная аттестация**			
	Экзамен		экзамен	
	Итого:	**180**	180	

3.2 Тематический план изучения дисциплины (по формам 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	Управление жизненным циклом программного обеспечения (SDLC)	45			18		27
2	Методологии разработки программного обеспечения	45			18		27
3	Инструменты разработки	45			18		27
4	Управление конфигурацией и DevOps	45			18		27
Итого		**180**			**72**		**108**

3.3 Содержание дисциплины

1. Введение в технологии конструирования программного обеспечения.
2. Жизненный цикл программного обеспечения (SDLC).
3. Методологии разработки программного обеспечения, такие как водопадная модель, гибкая разработка (Agile) и RAD (Rapid Application Development).
4. Требования к программному обеспечению и их сбор.
5. Проектирование программного обеспечения.
6. Разработка программного обеспечения с использованием различных языков программирования, таких как Python, C++ и других.
7. Тестирование программного обеспечения и методы обеспечения качества.
8. Управление конфигурацией программного обеспечения.
9. Инструменты разработки программного обеспечения, такие как интегрированные среды разработки (IDE) и системы контроля версий.
10. Развертывание и обслуживание программного обеспечения.
11. DevOps и его роль в разработке программного обеспечения.
12. Архитектура программного обеспечения.
13. Моделирование и проектирование баз данных.
14. Управление проектами разработки программного обеспечения.
15. Анализ и управление рисками в разработке программного обеспечения.
16. Качество программного обеспечения и тестирование производительности.
17. Разработка пользовательского интерфейса.
18. Веб-разработка и технологии.
19. Безопасность программного обеспечения.
20. Тенденции и новые направления в области технологий конструирования программного обеспечения.

4 Учебно-методическое и информационное обеспечение

4.1 Основная литература

1. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2022. — 176 с. — (Высшее образование). — ISBN 978-5-534-14383-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/520097
2. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2022. — 147 с. — (Высшее образование). — ISBN 978-5-534-09172-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/513696
3.

4.2 Дополнительная литература

1. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 248 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/534336

5 Материально-техническое обеспечение

5.1 Требования к оборудованию и помещению для занятий

Практические занятия (семинары) и самостоятельная работа студентов должна проводиться в специализированных аудиториях с комплектом мультимедийного оборудования и/или доской для записей материалов. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов.

5.2 Требования к программному обеспечению

Для проведения практических занятий (семинаров) специального программного обеспечения для освоения дисциплины не требуется.

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 аудиторные занятия, семинары и практики.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, 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, промежуточный контроль осуществляется на зачете 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:

- экзамен.

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения				
ИОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы. ИОПК-8.2. Умеет проектировать блок-схемы алгоритмов, оценивать производительность алгоритмов и затраты памяти на работу алгоритма, разрабатывать программы на основе спроектированного алгоритма и проводить отладку программы, применять методы	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины.	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Свободно оперирует приобретенными знаниями.

системного анализа и математического моделирования при разработке и эксплуатации ИС, проводить структурный анализ, функциональный анализ, объектно-ориентированный анализ иерархии классов. осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы. ИОПК-8.3. Владеет навыками разработки программ, построения блок-схем алгоритмов и оценки производительности алгоритмов, работы с унифицированным языком визуального моделирования, составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.				

Шкала оценивания результатов промежуточной аттестации и её описание:

Форма промежуточной аттестации: экзамен.

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 по дисциплине.

Шкала оценивания	Описание
Отлично	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 5. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 4. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 3. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не достигнуто пороговое значение хотя бы для одного уровня формируемых на момент проведения аттестации компетенций. Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7.3 Оценочные средства

7.3.1 Экзаменационные вопросы

1. Введение в технологии конструирования программного обеспечения.
2. Жизненный цикл программного обеспечения (SDLC).
3. Методологии разработки программного обеспечения.
4. Требования к программному обеспечению и их сбор.
5. Проектирование программного обеспечения.

6. Разработка программного обеспечения с использованием различных языков программирования.
7. Тестирование программного обеспечения и методы обеспечения качества.
8. Управление конфигурацией программного обеспечения.
9. Инструменты разработки программного обеспечения.
10. Развертывание и обслуживание программного обеспечения.
11. DevOps и его роль в разработке программного обеспечения.
12. Архитектура программного обеспечения.
13. Моделирование и проектирование баз данных.
14. Управление проектами разработки программного обеспечения.
15. Анализ и управление рисками в разработке программного обеспечения.
16. Качество программного обеспечения и тестирование производительности.
17. Разработка пользовательского интерфейса.
18. Веб-разработка и технологии.
19. Безопасность программного обеспечения.
20. Тенденции и новые направления в области технологий конструирования программного обеспечения.
21. Методы оценки и выбора технологий разработки программного обеспечения.
22. Управление изменениями в программном обеспечении.
23. Разработка мобильных приложений.
24. Тестирование безопасности программного обеспечения.
25. Автоматизация процессов разработки и тестирования.
26. Облачные технологии и их применение в разработке программного обеспечения.
27. Разработка распределенных систем.
28. Методы и инструменты для улучшения процесса разработки программного обеспечения.
29. Разработка программного обеспечения для мобильных устройств.
30. Тестирование производительности и оптимизация программного обеспечения.
31. Разработка интернет-приложений и веб-сервисов.
32. Методы и инструменты для управления проектами разработки программного обеспечения.
33. Разработка программного обеспечения для IoT (интернета вещей).
34. Тестирование пользовательского интерфейса и юзабилити.
35. Разработка программного обеспечения для машинного обучения и искусственного интеллекта.
36. Методы и инструменты для контроля качества программного обеспечения.
37. Разработка игр и развлекательного программного обеспечения.
38. Тестирование мобильных приложений.
39. Разработка программного обеспечения для электронной коммерции.
40. Методы и инструменты для управления требованиями к программному обеспечению.
41. Разработка программного обеспечения для автоматизации бизнес-процессов.
42. Тестирование масштабируемости и нагрузочное тестирование программного обеспечения.
43. Разработка программного обеспечения для систем управления базами данных.
44. Методы и инструменты для анализа и моделирования бизнес-процессов.
45. Разработка программного обеспечения для робототехники и автоматизации.
46. Тестирование безотказности и восстановления программного обеспечения.
47. Разработка программного обеспечения для систем управления проектами.
48. Методы и инструменты для анализа и оптимизации производительности программного обеспечения.

49. Разработка программного обеспечения для систем управления ресурсами предприятия.
50. Тестирование совместимости и интеграции программного обеспечения.

7.3.2 Типовые практические задания

1. Разработка простого веб-приложения с использованием языка программирования Python.
2. Создание базы данных и разработка SQL-запросов для управления данными.
3. Разработка архитектуры программного обеспечения для конкретного проекта.
4. Написание кода для реализации определенной функциональности в программном обеспечении.
5. Тестирование программного обеспечения и отладка ошибок.
6. Разработка пользовательского интерфейса для программного обеспечения.
7. Использование инструментов управления версиями, таких как Git, для контроля изменений в коде программного обеспечения.
8. Создание документации для программного обеспечения, включая описание функций и инструкции по использованию.
9. Разработка мобильного приложения с использованием соответствующих технологий и фреймворков.
10. Проектирование и разработка системы управления базами данных.

7.3.3 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
«Технологии конструирования программного обеспечения»
направление подготовки 09.03.01 Информатика и вычислительная техника

ВОПРОСЫ:

1. DevOps и его роль в разработке программного обеспечения.
2. Анализ и управление рисками в разработке программного обеспечения.
3. Разработка игр и развлекательного программного обеспечения.

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