Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 03.11.2023 16:46:52
Уникальный программный ключ:
8db180d1a3f02ac9e60521a672ea7c3a1fd39e1e6b1a0

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования
Московский политехнический университет

УТВЕРЖДАЮ
И.о. декана транспортного факультета
_____ /М.Н. Лукьянов/
«__» 08 2022 г.

Рабочая программа дисциплины
Методы и языки программирования

Направление

01.03.02 Прикладная математика и информатика

Профиль подготовки (образовательная программа)
«Интеллектуальные системы управления транспортом»

Квалификация (степень) выпускника
бакалавр

Форма обучения
Очная

Москва 2022

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 3
2 Место дисциплины в структуре образовательной программы 3
3 Структура и содержание дисциплины 3
3.1 Виды учебной работы и трудоемкость 3
3.2 Тематический план изучения дисциплины 5
4 Учебно-методическое и информационное обеспечение 6
4.1 Основная литература 6
4.2 Дополнительная литература 6
5 Материально-техническое обеспечение 6
6 Методические рекомендации 7
6.1 Методические рекомендации для преподавателя по организации обучения 7
6.2 Методические указания для обучающихся по освоению дисциплины 7
7 Фонд оценочных средств 7
7.1 Методы контроля и оценивания результатов обучения 7
7.2 Шкала и критерии оценивания результатов обучения 7
7.3 Оценочные средства 11
7.3.1. Список вопросов к экзамену 11

1 Цели, задачи и планируемые результаты обучения по дисциплине

К **основным целям** освоения дисциплины «Методы и языки программирования» следует отнести:

- изучение современных технологий и методов программирования;
- получение навыков проектирования и разработки алгоритмического и программного обеспечения.

К **основным задачам** освоения дисциплины «Методы и языки программирования» следует отнести:

- изучение методологии и средств разработки ПО;
- изучение методов проектирования ПО;
- изучение оценки качества программного обеспечения;
- изучение тестирования и отладки программного обеспечения;
- изучение принципов, методов и средств сопровождения ПО.

Обучение по дисциплине **«Методы и языки программирования»** направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ИОПК-2.1 Использует и адаптирует существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач

2 Место дисциплины в структуре образовательной программы

Дисциплина «Методы и языки программирования» относится к числу профессиональных учебных дисциплин обязательной части цикла (Б.1) основной образовательной программы (Б1.1.13).

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 9 зачетных единицы, т.е. **324** академических часов, форма контроля – экзамен в 1 семестре, экзамен во 2 семестре

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			1	2
1	**Аудиторные занятия**	**152**	80	72
	В том числе:			
1.1	Лекции		16	
1.2	Семинарские/практические занятия		64	72
1.3	Лабораторные занятия			

2	**Самостоятельная работа**	**172**	100	72
3	**Промежуточная аттестация**			
	Экзамен		экзамен	экзамен
	Итого:	**324**	180	144

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	Жизненный цикл ПО. Модели жизненного цикла ПО.	4	2	2			10
2	Стандарты в сфере разработки ПО (ISO, IEEE, SEI, ГОСТ Р).	8	2	6			10
3	Процессы жизненного цикла ПО (стандарт ISO/IEC 12207).	8	2	6			10
4	Модель зрелости предприятия (CMM, CMMI).	8	2	6			10
5	Основные этапы разработки сложных программных систем.	8	2	6			10
6	Структура и состав технического задания в соответствии ГОСТ.	10	2	8			10
7	Этапы разработки ПО в соответствии с ГОСТ.	12	2	10			20
8	Методологии разработки ПО: RUP, MSF, XP.	12	2	10			20
Итого за первый семестр		**80**	**16**	**54**			**100**
9	Показатели качества ПО. Стандарт ISO/IEC 9126	8		4			4
10	Основы планирования разработки программного обеспечения. Подходы к оценке сложности и времени разработки ПО.	8		4			4
11	Размерно- и функционально-ориентированные метрики оценки сложности и времени разработки ПО: LOC, FP.	16		8			8
12	Конструктивная модель оценки ПО COCOMO, CO-COMO II.	16		8			8
13	Принципы организации разработки ПО. Организация коллективов разработчиков ПО.	16		8			8
14	Функции и роли разработчиков в программных проектах, совмещение ролей в рамках одного проекта.	16		8			8
15	Средства поддержки планирования и контроля процесса разработки ПО.	16		8			8

16	Планирование и организация разработки ПО в соответствии с методологиями RUP, MSF, XP.	16		8			8
17	Требования к программному продукту: определения, характеристики требований, типы требований, способы организации взаимодействия разработчика и заказчика.	16		8			8
18	Требования надежности, безопасности, защищенности.	16		8			8
Итого второй семестр		**144**		**72**			**72**

4 Учебно-методическое и информационное обеспечение

4.1 Основная литература

1. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2022. — 150 с. — (Высшее образование). — ISBN 978-5-534-16942-3.
   URL: https://urait.ru/bcode/532055
2. Черпаков, И. В. Основы программирования : учебник и практикум для вузов / И. В. Черпаков. — Москва : Издательство Юрайт, 2022. — 219 с. — (Высшее образование). — ISBN 978-5-9916-9983-9.
   URL: https://urait.ru/bcode/511703

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
   URL: https://urait.ru/bcode/534336

4.3 Электронные образовательные ресурсы

LMS «Методы и языки программирования»:
https://online.mospolytech.ru/local/crw/course.php?id=10813

5 Материально-техническое обеспечение

Для проведения всех видов занятий необходимо презентационное оборудование (мультимедийный проектор, экран) – 1 комплект.

Для проведения практических занятий необходимо наличие компьютерных классов оборудованных современной вычислительной техникой из расчета одно рабочее место на одного обучаемого.

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, задачи 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, 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.

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

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:

- выполнение лабораторных работ;
- 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 (модулю):

ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач				
Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**

знать: современные Методы и языки программирования; показатели качества программного обеспечения; методологии и методы проектирования программного обеспечения; методы тестирования и отладки ПО; принципы организации документирования разработки, процесса сопровождения программного обеспечения;	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: .современные Методы и языки программирования показатели качества программного обеспечения; методологии и методы проектирования программного обеспечения; методы тестирования и отладки ПО; принципы организации документирования разработки, процесса сопровождения программного обеспечения;	Обучающийся демонстрирует неполное соответствие следующих знаний: современные Методы и языки программирования; показатели качества программного обеспечения; методологии и методы проектирования программного обеспечения; методы тестирования и отладки ПО; принципы организации документирования разработки, процесса сопровождения программного обеспечения; 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: ,современные Методы и языки программирования ;показатели качества программного обеспечения; методологии и методы проектирования программного обеспечения; методы тестирования и отладки ПО; принципы организации документирования разработки, процесса сопровождения программного обеспечения; но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний: современные Методы и языки программирования; показатели качества программного обеспечения; методологии и методы проектирования программного обеспечения; методы тестирования и отладки ПО; принципы организации документирования разработки, процесса сопровождения программного обеспечения; свободно оперирует приобретенными знаниями.
уметь: формировать требования и разрабатывать внешние спецификации для разрабатываемого программного обеспечения; планировать разработку сложного программного обеспечения; проектировать структуру и архитектуру программного обеспечения с использованием современных методологий и средств	Обучающийся не умеет или в недостаточной степени умеет формировать требования и разрабатывать внешние спецификации для разрабатываемого программного обеспечения; планировать разработку сложного программного обеспечения; проектировать структуру и архитектуру программного обеспечения с использованием	Обучающийся демонстрирует неполное соответствие следующих умений: формировать требования и разрабатывать внешние спецификации для разрабатываемого программного обеспечения; планировать разработку сложного программного обеспечения; •проектировать структуру и архитектуру программного обеспечения с использованием	Обучающийся демонстрирует частичное соответствие следующих умений: . формировать требования и разрабатывать внешние спецификации для разрабатываемого программного обеспечения; планировать разработку сложного программного обеспечения; проектировать структуру и архитектуру программного	Обучающийся демонстрирует полное соответствие следующих умений: формировать требования и разрабатывать внешние спецификации для разрабатываемого программного обеспечения; планировать разработку сложного программного обеспечения; проектировать структуру и архитектуру программного

автоматизации проектирования программного обеспечения; проводить комплексное тестирование и отладку программных систем; проектировать и кодировать алгоритмы с соблюдением требований к качественному стилю программирования; работать с интегрированной средой разработки программного обеспечения;	современных методологий и средств автоматизации проектирования программного обеспечения; проводить комплексное тестирование и отладку программных систем; проектировать и кодировать алгоритмы с соблюдением требований к качественному стилю программирования работать с интегрированной средой разработки программного обеспечения;	современных методологий и средств автоматизации проектирования программного обеспечения; проводить комплексное тестирование и отладку программных систем; проектировать и кодировать алгоритмы с соблюдением требований к качественному стилю программирования; работать с интегрированной средой разработки программного обеспечения; . Допускаются значительные ошибки, проявляется недостаточность умений.	обеспечения с использованием современных методологий и средств автоматизации проектирования программного обеспечения; проводить комплексное тестирование и отладку программных систем; проектировать и кодировать алгоритмы с соблюдением требований к качественному стилю программирования ;работать с интегрированной средой разработки программного обеспечения; Умения освоены, но допускаются незначительные ошибки, неточности.	обеспечения с использованием современных методологий и средств автоматизации проектирования программного обеспечения; проводить комплексное тестирование и отладку программных систем; проектировать и кодировать алгоритмы с соблюдением требований к качественному стилю программирования; работать с интегрированной средой разработки программного обеспечения;. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
владеть: навыками проектирования программного обеспечения с использованием средств автоматизации; навыками разработки, документирования, тестирования и отладки программного обеспечения в соответствии с современными технологиями и методами программирования; навыками разработки программной документации.	Обучающийся не владеет или в недостаточной степени владеет навыками проектирования программного обеспечения с использованием средств автоматизации; навыками разработки, документирования , тестирования и отладки программного обеспечения в соответствии с современными технологиями и методами программирования ; навыками разработки	Обучающийся владеет навыками проектирования программного обеспечения с использованием средств автоматизации; навыками разработки, документирования, тестирования и отладки программного обеспечения в соответствии с современными технологиями и методами программирования; навыками разработки программной документации. , но допускаются значительные ошибки,	Обучающийся частично владеет навыками проектирования программного обеспечения с использованием средств автоматизации; навыками разработки, документирования, тестирования и отладки программного обеспечения в соответствии с современными технологиями и методами программирования ; навыками разработки программной документации.	Обучающийся в полном объеме владеет • навыками проектирования программного обеспечения с использованием средств автоматизации; • навыками разработки, документирования, тестирования и отладки программного обеспечения в соответствии с современными технологиями и методами программирования; навыками разработки программной документации.

	программной документации.	проявляется недостаточность владения	, навыки освоены, но допускаются незначительные ошибки, неточности, затруднения.	, свободно применяет полученные навыки в ситуациях повышенной сложности.

Шкалы оценивания результатов промежуточной аттестации и их 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».

Шкала оценивания	**Описание**
Отлично	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	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	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 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. Список вопросов к экзамену

1. Жизненный цикл ПО. Модели жизненного цикла ПО.
2. Стандарты в сфере разработки ПО (ISO, IEEE, SEI, ГОСТ Р).
3. Процессы жизненного цикла ПО (стандарт ISO/IEC 12207).
4. Модель зрелости предприятия (CMM, CMMI).
5. Основные этапы разработки сложных программных систем.
6. Структура и состав технического задания в соответствии ГОСТ.
7. Этапы разработки ПО в соответствии с ГОСТ.
8. Методологии разработки ПО: RUP, MSF, XP.
9. Показатели качества ПО. Стандарт ISO/IEC 9126
10. Основы планирования разработки программного обеспечения. Подходы к оценке
11. сложности и времени разработки ПО.
12. Размерно- и функционально-ориентированные метрики оценки сложности и времени разработки ПО: LOC, FP.
13. Конструктивная модель оценки ПО COCOMO, CO-COMO II.
14. Принципы организации разработки ПО. Организация коллективов разработчиков ПО.
15. Функции и роли разработчиков в программных проектах, совмещение ролей в рамках одного проекта.
16. Средства поддержки планирования и контроля процесса разработки ПО.
17. Планирование и организация разработки ПО в соответствии с методологиями RUP,
18. MSF, XP.
19. Требования к программному продукту: определения, характеристики требований,
20. типы требований, способы организации взаимодействия разработчика и заказчика.
21. Требования надежности, безопасности, защищенности.
22. Стандарты в сфере формирования требований к программному обеспечению.
23. Внешнее проектирование программного обеспечения. Методы и средства внешнего
24. проектирования.
25. Структурное проектирование ПО - основные принципы, понятие структурной
26. декомпозиции.
27. CASE-технологии. Методология структурного анализа и проектирования (SADT).
28. Моделирование потоков данных (DFD).
29. Понятия архитектуры и структуры ПО. Проектирование архитектуры ПО.
30. Требования к качественной архитектуре, характеристики качества архитектуры и
31. структуры ПО.
32. Виды программных архитектур.
33. Объектно-ориентированные анализ (ООА) и проектирование (ООП).
34. Основные понятия ООА и ООП. Объектная декомпозиция. Нотации ООА и ООП.
35. UML - цели создания, история развития, преимущества в области разработки
36. программных систем. Нотация UML. Расширения UML.
37. Стандарт CORBA: структура и содержание, версии стандарта.

Темы для подготовки к практическим занятиям

1. Основные команды git. Слияние веток, разрешение конфликтов, откат, перемещение по истории, восстановление изменений, копирование коммита, определение автора строки, перемещение по веткам, перенос ветки, синхронизация репозиториев, коммит не в ту ветку.
2. Написание unit test для консольной функции.
3. Создание документации проекта в Doxygen
4. Работа с Docker
    a. Написание Dockerfile.txt, сборка и демонстрация работы построенного контейнера.
    b. Запуск приложения с параметрами
    c. Установки модулей
    d. Перенос файлов из текущей папки в заданную папку контейнера
    e. Просмотр запущенных контейнеров
    f. Команды bash, echo
    g. Просмотр статуса
    h. Навигация внутри контейнера

Исходный проект и исходный репозиторий будут выданы.

Список тем теоретических вопросов

1. Команды git
2. Команды docker
3. Теги doxygen
4. UseCase, диаграмма классов, типы связей на UMLдиаграммах
5. Виды тестирования
6. Модели жизненного цикла программного обеспечения
7. Управление требованиями, виды требования, свойства требований
8. Паттерны проектирования (GoF)
9. ГОСТы для документации ПО vs АС