



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ
Проректор по учебной работе
Е.К. Самаров
« 17 » 2021г.



**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И
ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ
СИСТЕМ**

**РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ**

«ВВЕДЕНИЕ В ПРОГРАММНУЮ ИНЖЕНЕРИЮ»

Направление подготовки: 01.03.02. Прикладная математика и информатика

Направленность (профиль): Искусственный интеллект и управление в ракетно-космических системах

Уровень высшего образования: бакалавриат

Форма обучения: очная

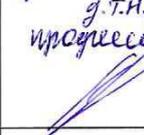
Королев
2021

Автор: Сидорова Н.П. Рабочая программа дисциплины: «Введение в программную инженерию». – Королев МО: МГОТУ, 2021.

Рецензент: к.т.н., доцент Исаева Г.Н.

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки бакалавров 01.03.02. Прикладная математика и информатика Учебного плана, утвержденного Ученым советом «МГОТУ». Протокол № 13 от 22.06. 2021 года.

Рабочая программа рассмотрена и одобрена на заседании кафедры:

Заведующий кафедрой (ФИО, ученая степень, звание, подпись)	Артюшенко В.М. д.т.н. профессор 	Артюшенко В.М. д.т.н., профессор 	Артюшенко В.М. д.т.н., профессор 	
Год утверждения (переутверждения)	2021	2022	2023	2024
Номер и дата протокола заседания кафедры	№15 от 02.06.21	№13 от 03.06.22	№14 от 09.05.23	

Рабочая программа согласована:

Руководитель ОПОП _____



к.т.н., доц. Бугай И.В

Рабочая программа рекомендована на заседании УМС:

Год утверждения (переутверждения)	2021	2022	2023	2024
	№7 от 15.06.21	№5 от 21.06.22	№6 от 16.05.23	

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОПОП

Целью изучения дисциплины является:

- получение знаний и умений в области индустриальных способов (инженерных методов и средств) создания программного обеспечения (ПО);
- получение практических навыков работы с инструментальными средствами разработки программного обеспечения.

В процессе обучения студент приобретает и совершенствует следующие компетенции:

общефессиональные компетенции (ОПК):

- Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-4);

профессиональные компетенции (ПК):

- Способность принимать участие в управлении проектами создания информационных систем и программных комплексов на стадиях их коллективной разработки ПО (ПК-7).

Основными **задачами** дисциплины являются:

1. ознакомление с понятием качества программных продуктов, жизненным циклом программных продуктов и его моделями;
2. изучение методов анализа и проектирования ПО;
3. получение навыков применения современных CASE - средств поддержки этапов жизненного цикла ПО;
4. изучение методов верификации ПО;
5. приобретение навыков тестирования ПО в конкретной среде программирования.

После завершения освоения данной дисциплины студент должен:

Знать:

- технологии создания программных продуктов и программных комплексов;
- методы и средства проектирования программного обеспечения;
- базовые понятия программной инженерии;

Уметь:

- применять методы и средства проектирования программного обеспечения;
- использовать технологии создания программных продуктов и программных комплексов

Владеть:

- навыками использования CASE-средств для анализа и проектирования ПО;
- навыками применения методов и технологии проектирования и производства программного продукта,

– практическими навыками разработки ПО.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Программная инженерия» относится к обязательной части основной профессиональной образовательной программы подготовки бакалавров по направлению подготовки 01.03.02. «Прикладная математика и информатика».

Изучение данной дисциплины базируется на дисциплинах: «Языки высокого уровня», «Основы информатики», «Архитектура ПК» и компетенциях: ОПК-2, ПК-2, ПК-3, ПК-4.

Знания и компетенции, полученные при освоении дисциплины, являются базовыми при изучении дисциплин: «Человеко-машинный интерфейс систем управления», «Системы компьютерного моделирования», «Программные и компьютерные средства обработки изображений» и др., выполнения выпускной квалификационной работы бакалавра.

3. Объем дисциплины и виды учебной работы

Общая трудоемкость модуля составляет 3 зачетные единицы, 108 часов.

Таблица 1

Виды занятий	Всего часов	Семестр ...	Семестр 3	Семестр тр	Семестр
Общая трудоемкость	108		108		
ОЧНАЯ ФОРМА ОБУЧЕНИЯ					
Аудиторные занятия	48		48		
Лекции (Л)	16		16		
Практические занятия (ПЗ)	32		32		
Лабораторные работы (ЛР)	-		-		
Самостоятельная работа	60		60		
Курсовые, расчетно-графические работы	-				
Контрольная работа, домашнее задание	+ -		+ -		
Текущий контроль знаний	Тест		+		
Вид итогового контроля	Экзамен		Экзамен		

4. Содержание дисциплины (модуля)

4.1. Темы дисциплины и виды занятий

Таблица 2

Наименование тем	Лекции, час.	Практические занятия, час	Занятия в интерактивной форме, час	Код компетенций
Тема 1. Жизненный цикл ПО	2	4	1	ОПК-4, ПК-7
Тема 2. Методы анализа и проектирования ПО	4	8	2	
Тема 3. Язык UML	4	6	2	
Тема 4. Качество, надежность и эффективность ПО	2	6	2	
Тема 5. Управление процессом разработки ПО	4	8	2	
Итого:	16	32	10	

4.2. Содержание тем дисциплины

Тема 1. Жизненный цикл ПО

Системный подход к проектированию ПО. Понятие программного обеспечения, проекта и проектирования ПО. Программная инженерия. Основные требования, предъявляемые к программному изделию, как к продукту производственно-технического назначения.

Стандарты в области программного обеспечения. Сертификация и оценка процессов создания ПО. Определение жизненного цикла ПО. Стандарты, определяющие этапы и процессы жизненного цикла (ЖЦ). Модели и стадии ЖЦ ПО. Состав работ процессов ЖЦ.

Тема 2. Методы анализа и проектирования ПО

Проблемы проектирования ПО. Архитектура ПО. Архитектурный стиль. Требования к ПО. Анализ требований к ПО.

Классификация методов анализа и проектирования ПО. Структурные методы анализа и проектирования ПО. Метод функционального моделирования

SADT (IDEF0). Метод моделирования процессов IDEF3. Моделирование потоков данных DFD. Моделирование данных.

Основные принципы и свойства объектно-ориентированного проектирования. Основные элементы объектной модели. Объекты, свойства объектов. Понятие классов. Общая характеристика классов. Пример использования объектно-ориентированного подхода. Сопоставление и взаимосвязь структурного и объектно-ориентированного подходов.

Тема 3. Язык UML

Язык UML: Основные понятия. Классификация диаграмм UML. Применение UML для выполнения этапов анализа и проектирования. Унифицированный процесс разработки ПО.

Тема 4. Качество, надежность и эффективность ПО

Понятие качества программного продукта. Показатели качества программного продукта средств. Пути повышения качества программных продуктов. Числовые характеристики объектно-ориентированных программ.

Основные понятия и показатели надежности программных средств. Классификация программных ошибок. Принципы и методы обеспечения надежности программных средств.

Отладка и тестирование ПО. Методы тестирования ПО. ГОСТ Р ИСО/МЭК 12119-2000. Работы по тестированию. Протоколы тестирования. Отчет о тестировании.

Тема 5. Управление процессом разработки ПО

Задачи управления проектами по созданию и внедрению ПО. Процессы управления созданием ПО. Планирование проекта. Управление рисками. Управление требованиями. Управление версиями.

Модель зрелости CMM. Модель процессов управления услугами - библиотека ITIL.

Методологии управления разработкой ПО: RUP, гибкие методологии. Методология SCRUM. Информационные технологии поддержки управления процессом разработки ПО.

5. Перечень учебно-методического обеспечения для самостоятельной работы по дисциплине

1. «Методические указания для обучающихся по освоению дисциплины «Программная инженерия»

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Структура фонда оценочных средств приведена в Приложении 1.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. В.А. Антипов, А.А. Бубнов, А.Н. Пылькин, В.К. Столчнев. Введение в программную инженерию: Учебник. М.: КУРС: ИНФРА-М, 2019. — 336 с. ЭБС Знаниум. Режим доступа: <http://znanium.com/bookread2.php?book=1035160>
2. Гагарина Л.Г., Кокорева Е.В., Сидорова-Виснадул Б.Д. Технология разработки программного обеспечения: учеб. пособие / под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2018. — 400 с. / ЭБС «Знаниум», режим доступа: <http://znanium.com/bookread2.php?book=924760>
3. Введение в программную инженерию: Учебник / В.А. Антипов, А.А. Бубнов, А.Н. Пылькин, В.К. Столчнев. — М.: КУРС: ИНФРА-М, 2017. — 336 с. / ЭБС «Знаниум». Режим доступа: <http://znanium.com/bookread2.php?book=850951>

Дополнительная литература:

1. Влацкая И.В., Заельская Е.А., Надточий Е.С. Проектирование и реализация прикладного программного обеспечения. – Оренбург: ОГУ. – 2015. – 119 с. ЭБС «Университетская библиотека онлайн». Режим доступа: http://biblioclub.ru/index.php?page=book_view_red&book_id=439107
2. Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул. Технология разработки программного обеспечения: Учеб. пос. / Под ред. проф. Л.Г. Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2019. - 400 с. ЭБС Знаниум. Режим доступа: <http://znanium.com/bookread2.php?book=1011120>

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Интернет-ресурсы:

1. <http://www.intuit.ru> - образовательный портал,
2. <http://www.infojournal.ru> - научно-образовательный портал,
3. <http://msdn.microsoft.com/ru-ru/library/> - портал разработчика Microsoft,
4. <http://www.interface.ru/> - научно-образовательный портал.
5. <http://www.biblioclub.ru> – электронная библиотечная система
6. <http://znanium.com> – электронная библиотечная система

9. Методические указания для обучающихся по освоению дисциплины

Методические указания для обучающихся по освоению дисциплины приведены в Приложении 2 к настоящей рабочей программе.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения: Microsoft Visio, Microsoft Visual Studio, Modelio Open Source.

Информационные справочные системы:

1. Ресурсы информационно-образовательной среды «МГОТУ».
2. Рабочая программа и методическое обеспечение по дисциплине «Программная инженерия».

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

Лекционные занятия:

- аудитория, оснащенная презентационной техникой (проектор, экран);
- комплект электронных презентаций / слайдов;
- программные продукты Microsoft Visual Studio, Microsoft Visio, Ramus.

Прочее:

- рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;
- рабочие места студентов, оснащенные компьютерами с доступом в Интернет.

Практические занятия:

- Аудитория, оснащенная мультимедийными средствами (интерактивная доска).
- рабочее место преподавателя, оснащенное ПК с доступом в глобальную сеть Интернет;
- рабочие места студентов, оснащенные компьютерами с доступом в глобальную сеть Интернет и установленным программным обеспечением.

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И
ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ
СИСТЕМ**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ**

«ВВЕДЕНИЕ В ПРОГРАММНУЮ ИНЖЕНЕРИЮ»

(Приложение 1 к рабочей программе)

Направление подготовки: 01.03.02. Прикладная математика и информатика

Направленность (профиль): Искусственный интеллект и управление в ракетно-космических системах

Уровень высшего образования: бакалавриат

Форма обучения: очная

**Королев
2021**

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№ п/п	Индекс компетенции	Содержание компетенции (или ее части)*	Раздел дисциплины, обеспечивающий формирование компетенции (или ее части)	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции (или ее части), обучающийся должен:		
				Знать	уметь	владеть
1.	ОПК-4	Способен решать задачи профессиональной деятельности с использованием существующих информационных коммуникационных технологий и с учетом основных требований информационной безопасности;	Темы 1-5.	базовые понятия программной инженерии; технологии создания программных продуктов и программных комплексов	применять методы и средства проектирования программного обеспечения;	навыками применения методов и технологий проектирования и производства программного продукта,
2.	ПК-7	Способность принимать участие в управлении проектами создания информационных систем и программных комплексов на стадиях их коллективной разработки ПО	Темы 1-5.	методы и средства проектирования программного обеспечения	применять методы и средства проектирования программного обеспечения	навыками использования CASE-средств для анализа и проектирования ПО; практическими навыками разработки ПО.

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Код компетенции	Инструменты, оценивающие сформированность компетенции	Показатель оценивания компетенции	Критерии оценки
ОПК-4, ПК-7	Практическое задание	<p>А) полностью сформирована (компетенция освоена на высоком уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> •компетенция освоена на продвинутом уровне – 4 балла; •компетенция освоена на базовом уровне – 3 балла; <p>В) не сформирована (компетенция не освоена) – 2 и менее баллов</p>	<p>1. Проводится в форме письменной работы</p> <p>2.Время, отведенное на процедуру – 10 - 15 мин. Неявка – 0.</p> <p>Критерии оценки:</p> <p>1.Соответствие ответа заявленной тематике (0-5 баллов).</p> <p>Максимальная сумма баллов - 5 баллов.</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал.</p>
ОПК-4, ПК-7	Доклад в форме презентации	<p>А) полностью сформирована (компетенция освоена на высоком уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> •компетенция освоена на продвинутом уровне – 4 балла; •компетенция освоена на базовом уровне – 3 балла; <p>В) не сформирована (компетенция не освоена) – 2 и менее баллов</p>	<p>Проводится устно с использованием мультимедийных систем, а также с использованием технических средств</p> <p>Время, отведенное на процедуру – 10 - 15 мин. Неявка – 0.</p> <p>Критерии оценки:</p> <p>1.Соответствие представленной презентации заявленной тематике (1 балл).</p> <p>2.Качество источников и их количество при подготовке доклада и разработке презентации (1 балл).</p> <p>3.Владение информацией и способность отвечать на вопросы аудитории (1 балл).</p> <p>4.Качество самой представленной презентации (1 балл).</p> <p>5.Оригинальность подхода и всестороннее раскрытие выбранной тематики (1 балл).</p> <p>Максимальная сумма баллов - 5 баллов.</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после</p>

			проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал.
ОПК-4, ПК-7	Контрольная работа (проводится в качестве закрепления освоенного курса и компетенций)	<p>А) полностью сформирована (компетенция освоена на высоком уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> •компетенция освоена на продвинутом уровне – 4 балла; •компетенция освоена на базовом уровне – 3 балла; <p>В) не сформирована (компетенция не освоена) – 2 и менее баллов</p>	<p>Критерии оценки контрольной работы:</p> <ol style="list-style-type: none"> 1.Соответствие содержания контрольной работы заявленной тематике (1 балл). 2.Качество источников и их количество при подготовке работы (1 балл). 3.Владение информацией и способность отвечать на вопросы аудитории (1 балл). 4.Качество самой представленной работы (1 балл). 5.Оригинальность подхода и всестороннее раскрытие выбранной тематики (1 балл). <p>Максимальная сумма баллов - 5 баллов Оценка проставляется в журнал</p>

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Тематика практического задания

1. Разработать IDEF0-модель для процесса записи к врачу и описать её.
2. Разработать DFD-модель для процесса записи к врачу и описать её.
3. Разработать IDEF0-модель для процесса записи в школу и описать её.
4. Разработать DFD-модель для процесса записи в школу и описать её.
5. Разработать IDEF0-модель для процесса бронирования номеров в гостинице и описать её.
6. Разработать DFD-модель для процесса бронирования номеров в гостинице и описать её.
7. Разработать IDEF0-модель для процесса заказа товаров и описать её.
8. Разработать DFD-модель для процесса заказа товаров и описать её.
9. Составить на языке UML диаграмму вариантов использования для программной системы удаленной записи к врачу и описать её.
10. Составить на языке UML диаграмму вариантов использования для программной системы удаленной записи в школу и описать её.
11. Составить на языке UML диаграмму вариантов использования для программной системы бронирования номеров в гостинице и описать её.
12. Составить на языке UML диаграмму вариантов использования для программной системы заказа товаров и описать её.

13. Составить на языке UML диаграмму деятельности для программной системы удаленной записи к врачу и описать её.
14. Составить на языке UML диаграмму деятельности для программной системы удаленной записи в школу и описать её
15. Составить на языке UML диаграмму деятельности для программной системы бронирования номеров в гостинице и описать её заказа товаров и описать её
16. Составить на языке UML диаграмму деятельности для программной системы
17. Описать классы предметной области и построить диаграмму классов UML для программной системы удаленной записи к врачу.
18. Описать классы предметной области и построить диаграмму классов UML для программной системы бронирования номеров в гостинице
19. Описать классы предметной области и построить диаграмму классов UML для программной системы заказа товаров
20. Описать классы предметной области и построить диаграмму классов UML для программной системы Call-центра.
21. Разработать тесты и провести тестирование заданного компонента методом «белого ящика».
22. Разработать тесты и провести тестирование заданного компонента методом «черного ящика».
23. Разработать тесты и провести функциональное тестирование заданного компонента.
24. Разработать прототип интерфейса пользователя на основе диаграммы вариантов использования для программной системы удаленной записи к врачу в заданной инструментальной среде программирования.
25. Разработать прототип интерфейса пользователя на основе диаграммы вариантов использования для программной системы удаленной записи в школу в заданной инструментальной среде программирования.

Тематика докладов

1. CASE-средства проектирования моделей ПО.
2. Инструментальные средства построения моделей ПО.
3. Технологии информационного моделирования.
4. Применение диаграмм UML для решения практической задачи.
5. Документирование диаграмм UML.
6. Процессы программного проекта
7. Методы определения программных требований
8. Программные средства тестирования ПО
9. Метрики и атрибуты качества ПО.
10. Методы отладки в различных программных средах.
11. Характеристика метода нисходящего тестирования.
12. Пошаговое и монолитное тестирование модулей.
13. Технология создание проектов с использованием среды Microsoft Visual Studio.
14. Назначение и описание класса
15. Сравнительный анализ сред программирования
16. Обзор методологий разработки ПО

Тематика контрольной работы

Приведена в Приложении 2

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Формой контроля знаний по дисциплине «Программная инженерия» являются две текущие аттестации в виде тестов и итоговая аттестация в форме экзамена

Неделя текущего контроля	Вид оценочного средства	Код компетенций, оценивающий знания, умения, навыки	Содержание оценочного средства	Требования к выполнению	Срок сдачи (неделя семестра)	Критерии оценки по содержанию и качеству с указанием баллов
В соответствии с графиком учебного процесса	Тестирование	ОПК-4, ПК-7	20 вопросов	Компьютерное тестирование; время, отведенное на процедуру - 30 минут	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка - Удовлетворительно - от 51% правильных ответов. Хорошо - от 70%. Отлично – от 90%.
	Экзамен	ОПК-4, ПК-7	2 вопроса и практическое задание	Экзамен проводится в устной форме, путем ответа на вопросы. Время, отведенное на процедуру – 30 минут.	Результаты предоставляются в день проведения экзамена	Критерии оценки: «Отлично»: <ul style="list-style-type: none"> • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответ на вопросы билета полные; • правильно решено

						<p>практическое задание.</p> <p>«Хорошо»:</p> <ul style="list-style-type: none"> • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответы на вопросы билета • практическое задание решено частично <p>«Удовлетворительно»:</p> <ul style="list-style-type: none"> • демонстрирует частичные знания по темам дисциплин; • незнание неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; <p>«Неудовлетворительно»:</p> <ul style="list-style-type: none"> • демонстрирует частичные знания по темам дисциплин; • незнание основных понятий предмета; • неумение использовать и применять полученные
--	--	--	--	--	--	--

						знания на практике; • не работал на практических занятиях; • не отвечает на вопросы • не решено практическое задание.
--	--	--	--	--	--	--

4.1. Тестовые задания

Тесты используются в режиме промежуточного контроля. По форме заданий выбраны закрытые тесты (с выборочным ответом).

Тема 1: Жизненный цикл ПО

1. Определите понятие жизненного цикла ПО
2. Модель жизненного цикла – это
3. Назовите модели жизненного цикла ПО
4. Укажите правильную последовательность этапов при каскадной модели жизненного цикла.
5. В чем достоинства и недостатки спиральной модели?
6. Определите специфику спиральной модели жизненного цикла
7. Назовите группы процессов жизненного цикла в соответствии со стандартом ГОСТ Р ИСО/МЭК 12207.

Тема 2. Методы анализа и проектирования ПО

8. Проектирование ПО в основном рассматривается как...
9. Что из приведенного не является одним из методов проектирования программного обеспечения?
10. Техническое задание — это
11. Архитектура программной системы — это
12. Как называется процесс разбиения одной сложной задачи на несколько простых подзадач?
13. Назначение интерфейс заключается в...?
14. При конструировании программного обеспечения на этапе разработки или выбора алгоритма решения реализуется следующее...
15. Какое свойство определяет процедуры внесения изменений в требования?
16. Что такое управление сборками?
17. Какова основная задача удовлетворения потребителя

Тема 3. Язык UML

18. UML — это ...
19. Определите назначение диаграммы прецедентов.
20. Назовите основные элементы диаграммы прецедентов
21. Для чего применяются варианты использования (Use Case)?
22. Кто может быть актером в Use Case-диаграммах?

23. Что является характеристикой точки зрения?
24. Какие диаграммы используются для моделирования аппаратной части системы, с которой связано ПО?

Тема 4. Качество, надежность и эффективность ПО

25. Что означает легкость применения программного обеспечения?
26. Мобильность программного обеспечения это:
27. Что определяет устойчивость программного обеспечения?
28. Определите задачи этапа тестирования
29. На этапе тестирования пользователь выполняет следующее:
30. Что из приведенного является критериями оценки удобства интерфейсов?
31. Что означает валидация ПО?
32. При выполнении какого вида тестирования две и более компонент тестируются на совместимость?
33. Что такое интеграционное тестирование?

Тема 5. Интегрированные среды проектирования и разработки ПО

34. Определите структуру проекта в Microsoft Visual Studio
35. Что означает значение public модификатора доступа?
36. Что означает значение модификатора доступа?
37. Для чего используются индексы?
38. Приведенное ниже описание задает...
39. Что реализует приведенных ниже запрос на LINQ
40. Назовите преимущества использования LINQ

Тема 6. Управление процессом разработки ПО

41. Назовите задачи управления проектами по созданию ПО.
42. Определите базовые процессы управления созданием ПО.
43. Назовите задачи процесса планирования ИТ-проекта.
44. Укажите классы рисков, возникающих при реализации ИТ-проекта.
45. Какие задачи решаются в рамках процесса управления требованиями.
46. Назовите уровни зрелости проектной организации в модели CMM.
47. Назовите принципы использования библиотеки ITIL.
48. Что такое артефакт в RUP?
49. Назовите базовые принципы гибких методологий.
50. Определите назначение product backlog в методологии SCRUM.
51. Какую модель жизненного цикла поддерживает методология SCRUM.

4.3. Вопросы, выносимые на экзамен

1. Подходы к разработке программных средств. Их краткая характеристика.
2. Жизненный цикл программного обеспечения. Основные понятия.
3. Модели жизненных циклов программного обеспечения, их характеристики и области применения.
4. Особенности модели жизненного цикла «спираль».
5. Специфика инкрементной модели жизненного цикла.
6. Классификация процессов жизненного цикла в стандарте ИСО Р 12207

7. Содержание этапов анализа и проектирования при разработке программного обеспечения по методу функциональной декомпозиции.
8. Краткая характеристика объектно-ориентированного подхода к разработке программного обеспечения. Понятия «Класс» и «объект».
9. Виды контроля программ; статические и динамические методы контроля
10. Тестирование и отладка, подходы к тестированию.
11. Методы функционального тестирования.
12. Методы структурного тестирования. Тестирование путей, цикломатическая сложность программы.
13. Совместное использование функционального и структурного тестирования.
14. Тестирование программных комплексов, построенных методом функциональной декомпозиции
15. Тестирование программ и жизненный цикл программного продукта.
16. Общая характеристика и назначение языка UML.
17. Диаграммы вариантов использования, назначение, компоненты, отношения между компонентами.
18. Диаграмма последовательности и кооперативная диаграмма, их назначение, компоненты.
19. Диаграмма классов, характеристики класса
20. Диаграмма классов, типы и характеристики отношений.
21. Диаграммы состояний, их назначение, компоненты.
22. Диаграммы деятельности, их назначение, компоненты.
23. Диаграммы компонентов и размещения, их назначение, составные части.
24. Унифицированный процесс разработки программных средств. Общая характеристика и этапы.
25. Основополагающие принципы разработки программных средств по унифицированному процессу.
26. Процесс определения требований к программным средствам. Документирование требований с помощью диаграмм.
27. Процесс анализа и структурирования требований при разработке программных средств.
28. Процесс проектирования при разработке программных средств. Основные отличия моделей анализа и проектирования.
29. Особенности тестирования программных средств, построенных по объектно-ориентированной методике. Тестирование классов.
30. Тестирование взаимодействия классов. Контрактное и защитное программирование. Тестирование иерархии классов.
31. Средства тестирования в Microsoft Visual Studio 2010.
32. Сравнение объектно-ориентированного и процедурного программирования.
33. Задачи управления проектами по созданию и внедрению ПО.
34. Процессы управления созданием ПО.
35. Планирование проекта.
36. Классификация рисков ИТ-проекта.

37. Классификация требований к ПО.
38. Модель зрелости CMM. Уровни зрелости проектной организации.
39. Модель процессов управления услугами - библиотека ITIL.
40. Методологии управления разработкой ПО: RUP
41. Методологии управления разработкой ПО: гибкие методологии.
42. Базовые понятия методологии SCRUM.

Тематика экзаменационного практического задания

Задача 1. Разработать диаграмму вариантов использования для программной системы по её описанию.

Программная система предназначена для системы учёта заказов мебельного магазина. С программной системой работают менеджеры, кладовщик, заведующий магазином.

Система должна обеспечивать следующие возможности: добавление новых заказов, изменение ранее введённых заказов, учёт выполнения заказов, учёт сведений об текущих остатках на складе. При получении нового заказа система должна также послать сообщение бухгалтерской системе, которая выписывает счёт заказчику. Любой заказ может содержать одну или более товарных позиций. Для каждой позиции заказа указывается наименование товара и его количество. Заполненный заказ получает кладовщик, который начинает сборку заказа. Если для каждой позиции товара на складе находится товар в достаточном количестве, то товар резервируется, и заказ помечается выполненным. Если требуемого товара нет на складе, то заказ может быть отменён, либо выполнение заказа задерживается до поступления товара на склад. По результатам работы за день заведующий магазином составляет отчет о состоянии принятых в этот день заказов.

Задача 2. Разработать диаграмму классов для программной системы по её описанию.

Программная система предназначена для системы учёта заказов мебельного магазина. С программной системой работают менеджеры, кладовщик, заведующий магазином.

Система должна обеспечивать следующие возможности: добавление новых заказов, изменение ранее введённых заказов, учёт выполнения заказов, учёт сведений об текущих остатках на складе. При получении нового заказа система должна также послать сообщение бухгалтерской системе, которая выписывает счёт заказчику. Любой заказ может содержать одну или более товарных позиций. Для каждой позиции заказа указывается наименование товара и его количество. Заполненный заказ получает кладовщик, который начинает сборку заказа. Если для каждой позиции товара на складе находится товар в достаточном количестве, то товар резервируется, и заказ помечается выполненным. Если требуемого товара нет на складе, то заказ может быть отменён, либо выполнение заказа задерживается до поступления товара на склад. По результатам работы за день заведующий магазином составляет отчет о состоянии принятых в этот день заказов.

Задача 3. Разработать контекстную диаграмму для процесса заданного его описанием.

Процесс учёта заказов мебельного магазина осуществляется по следующим правилам. При получении нового заказа менеджер передает его в бухгалтерию для оформления счёта заказчику. Любой заказ может содержать одну или более товарных позиций. Для каждой позиции заказа указывается наименование товара и его количество. Заполненный заказ получает кладовщик, который начинает сборку заказа. Если для каждой позиции товара на складе находится товар в достаточном количестве, то товар резервируется, и заказ помечается выполненным. Если требуемого товара нет на складе, то заказ может быть отменён, либо выполнение заказа задерживается до поступления товара на склад. По результатам работы за день заведующий магазином составляет отчет о состоянии принятых в этот день заказов.

Задача 4. Построить управляющий граф программы по её тексту (язык C++)

```
2. #include <cmath> // подключаем sqrt
3. using namespace std;
4. int main()
   {
5. double a,b,c,d,x1,x2; // Объявляем переменные с плавающей точкой.
6. cout << "Введите a\n"; // Вводим исходные данные
7. cin >> a;
8. cout << "Введите b\n";
9. cin >> b;
10. cout << "Введите c\n";
11. cin >> c;
12. d = b * b - 4 * a * c; // Рассчитываем дискриминант
13. if (d > 0) // Условие при дискриминанте больше нуля
    {
14. x1 = ((-b) + sqrt(d)) / (2 * a);
15. x2 = ((-b) - sqrt(d)) / (2 * a);
16. cout << "x1 = " << x1 << "\n";
17. cout << "x2 = " << x2 << "\n";
    }
18. if (d == 0) // Условие для дискриминанта равного нулю
    {
19. x1 = -(b / (2 * a));
20. cout << "x1 = x2 = " << x1 << "\n";
    }
21. if (d < 0) // Условие при дискриминанте меньше нуля
22. cout << "D < 0, Действительных корней уравнения не существует";
    }
```

Задача 5. Разработать тесты для функционального тестирования на основе тестирования классов выходных данных для программы, реализующей поиск корней квадратного уравнения.

Входные данные: a, b, c

Результаты: x1, x2.

Аномальные ситуации: a=0 или b=0

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И
ТЕХНОЛОГИЙ**

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И УПРАВЛЯЮЩИХ СИСТЕМ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ**

«ВВЕДЕНИЕ В ПРОГРАММНУЮ ИНЖЕНЕРИЮ»

(Приложение 2 к рабочей программе)

Направление подготовки: 01.03.02. Прикладная математика и информатика

Направленность (профиль): Искусственный интеллект и управление в ракетно-космических системах

Уровень высшего образования: бакалавриат

Форма обучения: очная

**Королев
2021**

1. Общие положения

Целью изучения дисциплины является:

- получение знаний и умений в области индустриальных способов (инженерных методов и средств) создания программного обеспечения (ПО);
- получение практических навыков работы с инструментальными средствами разработки программного обеспечения.

Задачи дисциплины:

1. ознакомление с понятием качества программных продуктов, жизненным циклом программных продуктов и его моделями;
2. изучение методов анализа и проектирования ПО;
3. получение навыков применения современных CASE - средств поддержки этапов жизненного цикла ПО;
4. изучение методов верификации ПО;
5. приобретение навыков тестирования ПО в конкретной среде программирования.

2. Указания по проведению практических занятий

Тема: Жизненный цикл ПО

Практическое занятие 1.

Вид практического занятия: *смешанная форма практического занятия.*

Образовательная технология: *круглый стол*

Тема и содержание практического занятия:

Цель работы: Сформировать практические навыки реализации работ по созданию ПО.

Основные положения темы занятия:

1. Этапы жизненного цикла
2. Задачи этапа анализа

Вопросы для обсуждения:

1. Модели жизненного цикла.
2. Виды работ на каждом этапе жизненного цикла
3. CASE-средства анализа.
4. Стандартизация процессов жизненного цикла.
5. Состав стандартов ИСО
6. ГОСТ Р ИСО/МЭК 9126-93.
7. ГОСТ Р ИСО МЭК 12207

Продолжительность занятия – **4 ч.**

Тема: Методы анализа и проектирования ПО Практическое занятие 2

Вид практического занятия: *смешанная форма практического занятия.*
Образовательная технология: *командная работа*
Тема и содержание практического занятия:
Цель работы: методы структурного проектирования

Основные положения темы занятия:

1. Принципы структурного анализа и проектирования ПО.
 2. Функциональное моделирование процессов.
 3. Модели потоков данных.
 4. Моделирование BPMN.
 5. CASE-средства моделирования.
 6. Технология разработки моделей процессов.
 7. Технология разработки модели потоков данных.
 8. Нисходящее проектирование алгоритмов и программ
 9. Разработка программы методом нисходящего проектирования
- Продолжительность занятия - 6 ч.

Практическое занятие 3

Вид практического занятия: *смешанная форма практического занятия.*
Образовательная технология: *командная работа*
Тема и содержание практического занятия:
Цель работы: объектно-ориентированные методы анализа и проектирования.

Основные положения темы занятия:

1. Принципы объектно-ориентированного анализа и проектирования.
2. Анализ задачи и выделение классов.
3. Продолжительность занятия - 2 ч.

Тема: Язык UML

Практическое занятие 4

Вид практического занятия: *смешанная форма практического занятия.*
Образовательная технология: *командная работа*
Тема и содержание практического занятия: Разработка UML диаграмм

Основные положения темы занятия:

1. Язык UML: состав и структура.
2. Виды и назначение диаграмм UML.
3. Документирование диаграмм.
4. Инструментальные средства разработки диаграмм.
5. Технологии разработки диаграммы вариантов использования.
6. Технологии разработки диаграммы классов.
7. Диаграмма деятельности.
8. Диаграмма последовательностей.

Продолжительность занятия - 6 ч.

Тема: Качество, надежность и эффективность ПО

Практическое занятие 5.

Вид практического занятия: *смешанная форма практического занятия.*

Образовательная технология: *командная работа*

Тема и содержание практического занятия:

Цель работы: с методами тестирования структурных алгоритмов и программ

Основные положения темы занятия:

1. Методы верификации ПО.
2. Экспертиза ПО.
3. Классификация методов верификации.
4. Место верификации в жизненном цикле ПО.
5. Методы тестирования ПО.
6. Классификация ошибок в программе.
7. Средства отладки структурных программ
8. Стратегии черного и белого ящика.
9. Методы нахождения и исправления синтаксических ошибок
10. Статическое и динамическое тестирование.
11. Инспекция кода.
12. Тестовая база, тест, тестовый случай,
13. Тестовый монитор, тестовое покрытие и инструменты его измерения .
14. Проверка программы на нахождение ошибок с помощью метода «белого ящика».
15. Критерии разработки тестов для метода «белого ящика».
16. Критерии разработки тестов для функционального тестирования.
17. Интеграционное тестирование.
18. Регрессионное тестирование.
19. Тестирование нефункциональных требований к ПО.
20. Тестирование объектно-ориентированных программ.

Продолжительность занятия - 6 ч.

Тема 5. Управление процессом разработки ПО

Практическое занятие 6.

Методология СММ и модели процессов. Практическое использование СММ-модели.

Вид практического занятия: *смешанная форма практического занятия.*

Образовательные технологии: индивидуальная работа студентов на компьютере, оборудованном выходом в Internet

Тема и содержание практического занятия:

Цель работы: Получить практические знания по методологии СММ и моделированию процессов управления.

Основные положения темы занятия:

1. Структура модели СММ. Группа ключевых процессов.
2. Распределение групп ключевых процессов по уровням зрелости
3. Стандартный производственный процесс организации
4. Координация производственного процесса организации

Вопросы для обсуждения:

1. В чем состоит назначение СММ?
2. Как выглядят уровни зрелости СММ? В чем их смысл?
3. Что такое СППО?
4. Чем подход к улучшению процессов, предлагаемый СММ, отличается от подхода, базирующегося на внедрении процессных стандартов (например, ГОСТ Р ИСО/МЭК 12207)?
5. Какова связь между СППО СММ и процессной моделью ГОСТ Р ИСО/МЭК 12207?

Продолжительность занятия - 4 ч.

Практическое занятие 7.

Вид практического занятия: *смешанная форма практического занятия.*

Образовательная технология: *командная работа*

Тема и содержание практического занятия:

Цель работы: Получить знания и практические навыки решения задач управления ИТ-проектом.

Основные положения темы занятия:

1. Методологии управления разработкой ПО: RUP
2. Гибкие методологии.
3. Методология SCRUM.
4. Информационные технологии поддержки управления процессом разработки ПО.

Продолжительность занятия - 4 ч.

3. Указания по проведению лабораторного практикума

Не предусмотрен учебным планом.

4. Указания по проведению самостоятельной работы студентов

Цель самостоятельной работы: подготовить бакалавров к самостоятельному научному творчеству.

Задачи самостоятельной работы:

- 1.Расширить представление о технологических методах разработки ПО.
- 2.Систематизировать знания в области проектирования ПО.
- 3.Сформировать навыки работы с программной документацией при разработке ПО.

Виды самостоятельной работы представлены в таблице 1.

Таблица 1

№ п/п	Наименование блока (раздела) дисциплины	Виды СРС
1	Тема 1. Жизненный цикл ПО	Самостоятельное изучение тем с подготовкой презентаций

		<p>Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Задачи этапов жизненного цикла. 2. ISO/IEC 12207:1995 «Информационные технологии. Процессы жизненного цикла программного обеспечения» 3. ISO 15226:1999 «Техническая документация на продукцию. Модель жизненного цикла и назначение документов» 4. Стандарт ГОСТ 34.601. Автоматизированные системы. Стадии создания. 5. ISO/IEC 42010:2007 «Технология систем и программного обеспечения. Рекомендуемая практика архитектурного описания программно-интенсивных систем» 6. Характеристики качества ПО.
2	Тема 2. Методы анализа и проектирования ПО	<p>Самостоятельное изучение тем.</p> <p>Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Классификация методов проектирования ПО. 2. Функциональные модели. <p>Примерная тематика докладов с презентацией:</p> <ol style="list-style-type: none"> 1. CASE-средства проектирования моделей ПО. 2. Инструментальные средства построения моделей ПО. 3. Технологии информационного моделирования.
3	Тема 3. Язык UML	<p>Самостоятельное изучение тем.</p> <p>Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Инструментальные средства диаграмм UML 2. Элементы языка UML. <p>Примерная тематика докладов для презентаций:</p> <ol style="list-style-type: none"> 1. Применение диаграмм UML для решения практической задачи. 2. Документирование диаграмм UML
4	Тема 4. Качество, надежность и эффективность ПО	<p>Самостоятельное изучение тем.</p> <p>Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Характеристика показателей качества ПО 2. Различие между тестированием и отладкой программ. 3. Средства тестирования <p>Примерная тематика рефератов:</p> <ol style="list-style-type: none"> 1. Методы отладки в различных программных средах. 2. Характеристика метода нисходящего тестирования. 3. Пошаговое и монолитное тестирование модулей.

5	Тема 5. Управление процессом разработки ПО	<p>Самостоятельное изучение тем. Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Процессы управления ИТ-проекта 2. Классификация методологий разработки ПО <p>Примерная тематика докладов с презентацией:</p> <ol style="list-style-type: none"> 1. Классификация рисков ИТ-проекта 2. Базовые понятия SCRUM. 3. Роли в SCRUM. 4. Совмещение ролей
---	--	---

5. Указания по проведению контрольных работ для студентов

Учебным планом для бакалавров предусмотрено написание одной контрольной работы в каждом семестре, что является одним из условий успешного освоения ими основных положений данной дисциплины и служит допуском к сдаче экзамена по курсу во время зачетной сессии.

Задания в контрольной работе разрабатываются преподавателем кафедры «Информационных технологий и управляющих систем» МГОТУ.

Цель выполняемой работы: продемонстрировать знания и умения в области применения стандартов при разработке программных продуктов, а также показать умения в области систематизирования и обобщения изучаемой информации.

Основные задачи выполняемой работы:

1. Закрепление полученных теоретических знаний;
2. Получение навыков работы с технической документацией
3. Оценка применения практических навыков бакалавра в будущей практической работе;

Процесс написания контрольной работы делится на следующие этапы:

1. Определение темы контрольной работы
2. Изучение литературы, относящейся к теме контрольной работы
3. Оформление контрольной работы
4. Представление ее на кафедру для регистрации
5. Защита контрольной работы

Подготовку контрольной работы следует начинать с повторения соответствующих разделов учебника, учебных пособий, конспектов лекций.

Требования к содержанию контрольной работы:

В содержании контрольной работы необходимо показать знание рекомендованной литературы по данному заданию, при этом правильно пользоваться первоисточником и избегать чрезмерного цитирования. При использовании цитат необходимо указывать точные ссылки на используемый источник: указание автора (авторов), название работы, место издания, страницы. Кроме основной литературы рекомендуется использовать дополнительную литературу и источники сети Интернет (с детальным указанием сайта, т.е. копирование ссылки и даты обращения). Если в период выполнения контрольной работы были приняты новые законы или нормативно-правовые акты, относящиеся к излагаемой теме, их необходимо изучить и использовать при ее выполнении.

Важно обратить внимание на различные концептуальные подходы по исследуемой тематике.

Оформление библиографического списка осуществляется в соответствии с установленными нормами и правилами ГОСТ.

Порядок выполнения контрольной работы:

Контрольная работа излагается логически последовательно, грамотно, разборчиво.

Структура контрольной работы:

– титульный лист, который содержит полное название высшего учебного заведения, название кафедры, реализующей данную дисциплину, название (тема) контрольной работы, фамилию, инициалы автора, также необходимо указать номер группы, фамилию и инициалы, а также должность, ученое звание и степень научного руководителя (преподавателя), проверяющего контрольную работу.

- оглавление;
- введение;
- основная часть;
- заключение;
- список используемых источников;
- приложения.

В конце работы ставится подпись студента и дата сдачи. Страницы контрольной работы должны быть пронумерованы. Номер страницы ставится в нижнем правом углу.

Объем контрольной работы должен составлять 10-15 страниц машинописного текста. Размер шрифта №14 (Times New Roman), полуторный интервал, стандартный лист формата А4. Поля: верхнее -20 мм, нижнее-20мм, левое -30 мм, правое -15 мм.

Дополнительно контрольная работа может иметь приложения (схемы, графики, диаграммы).

По всем возникающим вопросам обучающемуся следует обращаться за консультацией на кафедру. Срок выполнения контрольной работы определяется кафедрой. Срок проверки контрольной работы – 3 дня с момента необходимой фиксированной даты сдачи.

Порядок защиты контрольной работы:

Контрольная работа подлежит обязательной защите. В установленной преподавателем срок студент должен сдать контрольную работу и быть готов ответить на вопросы и замечания. Оценка работы производится по четырех бальной системе: «ОТЛИЧНО», «ХОРОШО», «УДОВЛЕТВОРИТЕЛЬНО», «НЕУДОВЛЕТВОРИТЕЛЬНО». После сдачи работы не возвращаются и хранятся в фонде кафедры.

Тематика контрольной работы:

1. Стандартизация разработки ПО
2. CASE-средства анализа и проектирования ПО.
3. Характеристики инструментальной среды программирования.
4. Технологии доступа к данным.

5. Технологии быстрой разработки ПО
6. Методы отладки программ в различных программных средах.
7. Характеристика метода нисходящего тестирования.
8. Пошаговое и монолитное тестирование модулей.
9. Сравнительный анализ интегрированных сред программирования.
10. Методология применения диаграмм UML для разработки программной системы.
11. CASE-средства проектирования моделей ПО.
12. Инструментальные средства построения моделей ПО.
13. Технологии информационного моделирования.
14. Сравнение моделей жизненного цикла программных систем.
15. Гибкие методологии разработки программных продуктов.
16. Задачи и проблемы программной инженерии.
17. Методологии разработки ПО.
18. Свойства сложных программных систем.
19. Структура программного обеспечения ИС
20. Проблемы оценки качества ПО
21. Средства информационной поддержки программных проектов и изделий (CALS) технологий
22. Классификация стандартов, определяющих разработку ПО.
23. Обзор стандарта ГОСТ Р ИСО/МЭК 12207-99.
24. Обзор стандарта ГОСТ Р ИСО 15288-2005
25. Обзор стандарта ГОСТ Р ИСО/МЭК ТО 15271-2002.
26. Стратегии разработки ПО.
27. Гибкие методологии разработки ПО.
28. Методология Microsoft разработки ПО
29. Методология IBM разработки ПО.
30. Методологии быстрой разработки ПО
31. Модели жизненного цикла.
32. Виды работ на каждом этапе жизненного цикла
33. CASE-средства анализа.
34. Принципы структурного проектирования
35. Нисходящее проектирование алгоритмов и программ
36. Разработка программы методом нисходящего проектирования
37. Объектно-ориентированный анализ.
38. Характеристика унифицированного процесса (Rational Unified Process, RUP) разработки ПО.
39. Характеристика Personal Software Process.
40. Уровни зрелости организаций.
41. Методологии проектирования и реализации распределенных приложений.
42. Преимущества объектно-ориентированного подхода к разработке ПО.
43. Сравнение стратегий разработки ПО
44. CASE-средства анализа и проектирования ПО.
45. Характеристики инструментальной среды программирования Microsoft Visual Studio.
46. Эволюция технологий доступа к данным.

47. Характеристика технологии .NET
48. Тенденции развития технологий программной инженерии.
49. Понятие проекта в Microsoft Visual Studio.
50. Структура проекта в Microsoft Visual Studio.

6. Перечень основной и дополнительной учебной литературы

Основная литература:

1. В.А. Антипов, А.А. Бубнов, А.Н. Пылькин, В.К. Столчнев. Введение в программную инженерию: Учебник. М.: КУРС: ИНФРА-М, 2017. — 336 с. ЭБС Знаниум. Режим доступа: <http://znanium.com/bookread2.php?book=850951>
2. Гагарина Л.Г., Кокорева Е.В., Сидорова-Виснадул Б.Д. Технология разработки программного обеспечения : учеб. пособие / под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ» : ИНФРА-М, 2018. — 400 с. / ЭБС «Знаниум». Режим доступа: <http://znanium.com/bookread2.php?book=924760>
3. Введение в программную инженерию : Учебник / В.А. Антипов, А.А. Бубнов, А.Н. Пылькин, В.К. Столчнев. — М.: КУРС: ИНФРА-М, 2017. — 336 с. / ЭБС «Знаниум». Режим доступа: <http://znanium.com/bookread2.php?book=850951>

Дополнительная литература:

1. Влацкая И.В., Заельская Е.А., Надточий Е.С. Проектирование и реализация прикладного программного обеспечения. – Оренбург: ОГУ. – 2015. – 119 с. ЭБС «Университетская библиотека онлайн». Режим доступа: http://biblioclub.ru/index.php?page=book_view_red&book_id=439107
2. Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул. Технология разработки программного обеспечения: Учеб. пос. / Под ред. проф. Л.Г. Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2019. - 400 с. ЭБС Знаниум. Режим доступа: <http://znanium.com/bookread2.php?book=1011120>

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Интернет-ресурсы:

1. <http://www.intuit.ru> - образовательный портал,
2. <http://www.infojournal.ru> - научно-образовательный портал,
3. <http://msdn.microsoft.com/ru-ru/library/> - портал разработчика Microsoft,
4. <http://www.interface.ru/> - научно-образовательный портал.
5. <http://www.biblioclub.ru> – электронная библиотечная система
6. <http://znanium.com> – электронная библиотечная система

8. Перечень информационных технологий

Перечень программного обеспечения: Microsoft Visio, Microsoft Visual Studio, Modelio Open Source.

Информационные справочные системы:

1. Ресурсы информационно-образовательной среды МГОТУ
2. Рабочая программа и методическое обеспечение по дисциплине «Программная инженерия».