


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
 федеральное государственное автономное образовательное учреждение высшего  
 образования  
 "САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
 АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 41

«УТВЕРЖДАЮ»

Руководитель направления

проф. д.п.н. проф.  
 (должность, уч. степень, звание)

 А.Г. Степанов  
 (подпись)

«15» \_\_\_\_\_ 06\_\_ 2020г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Программная инженерия»  
 (Наименование дисциплины)

Код направления подготовки	09.03.03
Наименование направления подготовки	Прикладная информатика
Наименование направленности	Прикладная информатика в инновационной деятельности
Форма обучения	очная

Лист согласования рабочей программы дисциплины

Программу составил (а)

доц. к.т.н.  
 (должность, уч. степень, звание)

 20.05.20г.  
 (подпись, дата)

Е.Л. Турнецкая  
 (инициалы, фамилия)

Программа одобрена на заседании кафедры № 41

«20» мая 2020 г, протокол № 10-2019/20

2 Заведующий кафедрой № 41

д.т.н., проф.  
 (уч. степень, звание)

 20.05.20г.  
 (подпись, дата)

Г.А. Коржавин  
 (инициалы, фамилия)

Ответственный за ОП ВО 09.03.03(05)

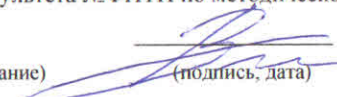
к.т.н., доц.  
 (должность, уч. степень, звание)

  
 (подпись, дата)

В.А. Галанина  
 (инициалы, фамилия)

Заместитель Декана факультета №ФПТИ по методической работе

доц. к.т.н. доц.  
 (должность, уч. степень, звание)

  
 (подпись, дата)

В.А. Голубков  
 (инициалы, фамилия)

## Аннотация

Дисциплина «Программная инженерия» входит в образовательную программу высшего образования по направлению подготовки/ специальности 09.03.03 «Прикладная информатика» направленности «Прикладная информатика в инновационной деятельности». Дисциплина реализуется кафедрой «№41».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ПК-1 «Способен формулировать задачи на проектирование и проектировать программное обеспечение»

ПК-3 «Способен разрабатывать технические спецификации на программные компоненты и их взаимодействие»

ПК-5 «Способен осуществлять организационное обеспечение процесса создания программного продукта»

ПК-7 «Способен разрабатывать планы модернизации программного продукта»

ПК-8 «Способен к проведению работ по обработке и анализу научно-технической информации и результатов исследований»

Содержание дисциплины охватывает круг вопросов, связанных с изучением организационных, инженерных и технических принципов, моделей и методов создания эффективно функционирующего и качественного программного обеспечения. Приведены примеры современных научных и практических достижений при разработке современных программных продуктов. Особое внимание уделено реализации веб-систем и программных продуктов, функционирующих в среде Интернет.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, практические занятия, самостоятельная работа обучающегося

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 4 зачетных единицы, 144 часа.

Язык обучения по дисциплине «русский»

## 1. Перечень планируемых результатов обучения по дисциплине

### 1.1. Цели преподавания дисциплины

Целью преподавания дисциплины является изложение принципов, методов и метрик, используемых в инженерном цикле создания сложных программных систем.

В ходе изучения дисциплины рассматриваются вопросы, связанные с анализом предметной области и выделением требований к разрабатываемой программной системе. Изучаются базовые архитектурные модели программного обеспечения, например, сервисно-ориентированная архитектура, и основные проектные характеристики: модульность, информационная закрытость, сложность, связность, сцепление и метрики для их оценки. Рассматриваются проектированию качественных и удобных для пользователей человеко-машинных интерфейсов. Изучаются базовые понятия тестирования программного обеспечения. Акцентируется внимание студентов на том, что спецификой информационной индустрии является коллективный труд, вовлекающий в производство специалистов разного профиля и уровня подготовки. Показывается развитие международной системы стандартов производства информационных систем. Слушателей ориентируют на то, что обеспечение качественного процесса производства программного обеспечения гарантирует необходимый уровень надежности и качества создаваемого информационного продукта. Значительное внимание уделяется реализации веб-систем.

1.2. Дисциплина входит в состав части, формируемой участниками образовательных отношений, образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Профессиональные компетенции	ПК-1 Способен формулировать задачи на проектирование и проектировать программное обеспечение	ПК-1.3.1 знать методы и приемы формализации задач; возможности современных и перспективных средств разработки программных продуктов; современные объектно-ориентированные языки программирования; современные структурные языки программирования ПК-1.У.1 2 уметь проводить анализ исполнения требований и выработать варианты их реализации ; создавать эффективные программные реализации с использованием современных программных сред; тестировать результаты работы программного продукта; осуществлять коммуникации с заинтересованными сторонами

Профессиональные компетенции	ПК-3 Способен разрабатывать технические спецификации на программные компоненты и их взаимодействие	ПК-3.У.1 уметь выбирать средства реализации требований к программному обеспечению; вырабатывать варианты реализации программного обеспечения; проводить оценку и обоснование рекомендуемых решений
Профессиональные компетенции	ПК-5 Способен осуществлять организационное обеспечение процесса создания программного продукта	ПК-5.3.1 знать инструменты и методы верификации структуры программного кода; регламенты кодирования на языках программирования; требования к документированию программного продукта; инструменты и методы управления заинтересованными сторонами
Профессиональные компетенции	ПК-7 Способен разрабатывать планы модернизации программного продукта	ПК-7.У.1 уметь анализировать, оценивать и корректировать программный продукт; оценивать риски
Профессиональные компетенции	ПК-8 Способен к проведению работ по обработке и анализу научно-технической информации и результатов исследований	ПК-8.3.1 знать актуальную нормативную документацию в области разработки программных продуктов; методы анализа научных данных ; методы проведения экспериментов и наблюдений, обобщения и обработки информации ПК-8.У.1 уметь применять актуальную нормативную документацию; оформлять результаты исследований; применять методы проведения экспериментов; анализировать научные проблемы по тематике проводимых исследований и разработок

## 2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- «Управление проектами»,
- «Технологии программирования».

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и могут использоваться при изучении других дисциплин:

- «Информационный менеджмент»,
- «Проектирование информационных систем».

### 3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам
		№6
1	2	3
<b>Общая трудоемкость дисциплины, ЗЕ/ (час)</b>	4/ 144	4/ 144
<b>Аудиторные занятия, всего час.</b>	51	51
в том числе:		
лекции (Л), (час)	17	17
практические/семинарские занятия (ПЗ), (час)	17	17
лабораторные работы (ЛР), (час)	17	17
курсовой проект (работа) (КП, КР), (час)		
экзамен, (час)	36	36
<b>Самостоятельная работа, всего (час)</b>	57	57
<b>Вид промежуточной аттестации:</b> зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз.	Экз.

Примечание: \*\* кандидатский экзамен

### 4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
<b>Семестр 6</b>					
Раздел 1. Основные понятия программной инженерии. Методологии разработки программного обеспечения.	2	2			5
Раздел 2. Анализ предметной области и требований к программному обеспечению.	2	2	2		10
Раздел 3. Основы проектирования и реализации программного обеспечения.	2	2	11		10
Раздел 4. Объектно-ориентированное проектирование и реализация программного обеспечения.	3				10
Раздел 5. Верификация и тестирование программного обеспечения	3	8	2		10
Раздел 6. Сервисно-ориентированная архитектура	2				4
Итого	17	17	17	0	57

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	<p><i>Основные понятия программной инженерии. Методологии разработки программного обеспечения.</i>                      Проблемы разработки сложных программных систем. Жизненный цикл и этапы разработки программного обеспечения. Эволюция моделей жизненного цикла программного обеспечения. Классическая технология. Технологии быстрой разработки. Современные технологии проектирования. Agile-методологии.  <i>Предусмотрена интерактивная форма проведения занятия: комбинированная лекция с элементами лекции-беседы и лекции с демонстрацией слайдов.</i></p>
2	<p><i>Анализ предметной области и требований к программному обеспечению.</i>                      Вопросы, связанные с анализом предметной области и выделением требований к разрабатываемой программной системе. Визуальные средства проектирования программных систем.                      Диаграммы UML. Диаграмма вариантов использования. Диаграмма классов. Классы анализа. Диаграммы взаимодействия. Диаграммы размещения и компонентов.  <i>Предусмотрена интерактивная форма проведения занятия: лекция с демонстрацией слайдов.</i></p>
3	<p><i>Основы проектирования программного обеспечения.</i>                      Понятие образца проектирования. Структурирование системы. Декомпозиция подсистем на модули. Информационная закрытость. Связность модуля. Сцепление модулей. Характеристики иерархической структуры программной системы. Некоторые широко используемые примеры образцов анализа и архитектурных стилей.  <i>Предусмотрена интерактивная форма проведения занятия: лекция с демонстрацией слайдов.</i></p>
4	<p><i>Объектно-ориентированное проектирование и реализация программного обеспечения.</i>                      Мышление в стиле паттернов. Принципы построения паттернов. Паттерны Наблюдатель, Компоновщик, Команда. Проектирование пользовательского интерфейса. Сущностная эффективность. Согласованность задач. Наблюдаемость задач. Визуальная связность.  <i>Предусмотрена интерактивная форма проведения занятия: лекция с демонстрацией слайдов.</i></p>
5	<p><i>Верификация и тестирование программного обеспечения.</i>                      Структурное тестирование программного обеспечения. Функциональное тестирование программного обеспечения. Особенности тестирования методами «черного» и «белого» ящиков. Статическое и динамическое тестирование. Тестирование элементов. Тестирование интеграции. Отладка программного продукта. Верификация. Метрики объектно-ориентированного тестирования.  <i>Предусмотрена интерактивная форма проведения занятия: лекция с демонстрацией слайдов.</i></p>
6	<p><i>Сервисно-ориентированная архитектура</i>                      Основные понятия SOA. XML-стандарты для веб-служб. Веб-службы и REST-подход. Разработка служб.  <i>Предусмотрена интерактивная форма проведения занятия: лекция с демонстрацией слайдов.</i></p>

7	<i>Облачные технологии</i> Основные понятия облачных технологий. Модели представления облачных технологий. Архитектура облачного SaaS-приложения. Сервер приложений для облачного SaaS-сайта.
8	<i>Обеспечение качества программных систем</i> Определение качества ПО. Факторы качества ПО. Верификация и валидация. План обеспечения качества.

#### 4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	№ раздела дисциплины
<b>Семестр 6</b>				
1.	Методологии разработки программного обеспечения	Занятие проводится в интерактивной форме: кейс, мозговой штурм.	2	1
2.	Анализ предметной области и требований к программному обеспечению	Занятие проводится в интерактивной форме: решение ситуационных задач, мозговой штурм	1	2
3.	Проектирование пользовательского интерфейса	Занятие проводится в интерактивной форме: решение ситуационных задач, мозговой штурм.	2	2,3,4
4.	Проведение технического аудита веб-приложения	Занятие проводится в интерактивной форме: решение ситуационных задач.	2	2,5
5.	Тестирование клиентской части веб-приложения	Занятие проводится в интерактивной форме: решение ситуационных задач.	2	5,8
6.	Тестирование производительности	Занятие проводится в интерактивной форме: решение ситуационных задач.	2	5,8
7.	Тестирование безопасности	Занятие проводится в интерактивной форме: решение ситуационных задач.	2	5,8
8.	Тестирование сетевых протоколов	Занятие проводится в интерактивной форме: решение ситуационных задач.	2	5,8
9.	Создание тестовой документации	Занятие проводится в интерактивной форме: решение ситуационных задач.	2	5,8
<b>Всего</b>			<b>17</b>	

#### 4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	№ раздела
-------	---------------------------------	---------------------	-----------

			дисциплины
Семестр 6			
1.	Проведение многокритериального анализа предметной области	2	2,8
2.	Проектирование логической структуры веб-приложения	1	3
3.	Разработка клиентской части. Язык гипертекстовой разметки HTML	2	3
4.	Разработка клиентской части. Каскадные таблицы стилей CSS	2	3
5.	Разработка серверной части. Язык программирования PHP	2	3
6.	Разработка серверной части. Базы данных и СУБД MySQL	2	3
7.	Разработка интерактивных страниц веб-системы. Язык программирования JavaScript и технология AJAX	2	3,4
8.	Разработка интерактивных страниц веб-системы. Работа с социальными сетями	2	3,5
9.	Работа с системами контроля версий GIT	2	3
Всего		17	

4.5. Курсовое проектирование/ выполнение курсовой работы  
Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся  
Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 6, час
1	2	3
Изучение теоретического материала дисциплины (ТО)	32	32
Подготовка к текущему контролю успеваемости (ТКУ)	15	15
Подготовка к промежуточной аттестации (ПА)	10	10
Всего:	57	57

5. Перечень учебно-методического обеспечения  
для самостоятельной работы обучающихся по дисциплине (модулю)  
Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий  
Перечень печатных и электронных учебных изданий приведен в таблице 8.  
Таблица 8– Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке
--------------------	--------------------------	-------------------------------------



		(кроме электронных экземпляров)
ЭБС Юрайт	Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2019. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <a href="https://urait.ru/bcode/438444">https://urait.ru/bcode/438444</a> (дата обращения: 15.07.2020).	
ЭБС Юрайт	Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 147 с. — (Высшее образование). — ISBN 978-5-534-09172-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <a href="https://urait.ru/bcode/452749">https://urait.ru/bcode/452749</a> (дата обращения: 15.07.2020).	
ЭБ ГУАП	Разработка веб-приложений средствами языка PHP : [ Электронный ресурс ] : учебное пособие / А. В. Аграновский [и др.] ; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - Электрон. текстовые дан. - СПб. : Изд-во ГУАП, 2018. - 121 с	
004.9 А 25	Основы интернет-программирования : учебное пособие / А. В. Аграновский, В. С. Павлов, Е. Л. Турнецкая ; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - СПб. : Изд-во ГУАП, 2018. - 135 с.	13
ЭБ ГУАП	Тестирование веб-приложений : [ Электронный ресурс ] : учебное пособие / А. В. Аграновский [и др.] ; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - Электрон. текстовые дан. - СПб. : Изд-во ГУАП, 2020.	
ЭБ ГУАП	Программирование интерактивных веб-приложений : [ Электронный ресурс ] : учебное пособие / А. В. Аграновский [и др.] ; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - Электрон. текстовые дан. - СПб. : Изд-во ГУАП, 2019. - 91 с.	
ЭБС Юрайт	Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <a href="https://urait.ru/bcode/452137">https://urait.ru/bcode/452137</a> (дата обращения: 15.07.2020).	
ЭБС Юрайт	Лаврищева, Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. — Москва : Издательство Юрайт, 2020. — 280 с. — (Высшее образование). — ISBN 978-5-534-01056-5. — Текст :	

	электронный // ЭБС Юрайт [сайт]. — URL: <a href="https://urait.ru/bcode/452156">https://urait.ru/bcode/452156</a> (дата обращения: 15.07.2020).	
ЭБС Znanium	Введение в программную инженерию : Учебник / В.А. Антипов, А.А. Бубнов, А.Н. Пылькин, В.К. Столчнев. — Москва : КУРС: ИНФРА-М, 2019. — 336 с. - ISBN 978-5-906923-22-6. - Текст : электронный. - URL: <a href="https://znanium.com/catalog/product/1035160">https://znanium.com/catalog/product/1035160</a> (дата обращения: 15.07.2020). – Режим доступа: по подписке.	

#### 7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
<a href="https://lms.guap.ru/new/course/view.php?id=932">https://lms.guap.ru/new/course/view.php?id=932</a>	ЭОС ГУАП среда LMS Дисциплина Программная инженерия
<a href="http://lib.aanet.ru/">http://lib.aanet.ru/</a>	ЭБ ГУАП
<a href="https://ligurio.github.io/swebok-ru/">https://ligurio.github.io/swebok-ru/</a>	Документация Swebok

#### 8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
	Не предусмотрено

8.2. Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

#### 9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	

2	Специализированная лаборатория «Название»	
---	-------------------------------------------	--

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Экзамен	Список вопросов к экзамену; Экзаменационные билеты; Тесты.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 – Критерии оценки уровня сформированности компетенций

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«отлично» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся глубоко и всесторонне усвоил программный материал;</li> <li>– уверенно, логично, последовательно и грамотно его излагает;</li> <li>– опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления;</li> <li>– умело обосновывает и аргументирует выдвигаемые им идеи;</li> <li>– делает выводы и обобщения;</li> <li>– свободно владеет системой специализированных понятий.</li> </ul>
«хорошо» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы;</li> <li>– не допускает существенных неточностей;</li> <li>– увязывает усвоенные знания с практической деятельностью направления;</li> <li>– аргументирует научные положения;</li> <li>– делает выводы и обобщения;</li> <li>– владеет системой специализированных понятий.</li> </ul>
«удовлетворительно» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы;</li> <li>– допускает несущественные ошибки и неточности;</li> <li>– испытывает затруднения в практическом применении знаний направления;</li> <li>– слабо аргументирует научные положения;</li> <li>– затрудняется в формулировании выводов и обобщений;</li> <li>– частично владеет системой специализированных понятий.</li> </ul>
«неудовлетворительно» «не зачтено»	<ul style="list-style-type: none"> <li>– обучающийся не усвоил значительной части программного материала;</li> <li>– допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении;</li> <li>– испытывает трудности в практическом применении знаний;</li> <li>– не может аргументировать научные положения;</li> <li>– не формулирует выводов и обобщений.</li> </ul>

10.3. Типовые контрольные задания или иные материалы.  
Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена
1.	Основные понятия и определения программной инженерии. Процесс проектирование программного обеспечения.
2.	Основные понятия, определения и особенности программной инженерии.
3.	Основные методы программной инженерии.
4.	Раскройте понятие «методология проектирования».
5.	Основные стандарты программной инженерии.
6.	Основные свойства программного обеспечения.
7.	Виды процессов разработки программного обеспечения.
8.	Фазы процесса проектирования. Этапы процесса проектирования.
9.	Виды деятельности при разработке ПО.
10.	Стратегии совершенствования процесса разработки ПО.
11.	Жизненный цикл и этапы разработки программного обеспечения.
12.	Стратегии конструирования ПО. Однократный проход. Водопадная модель.
13.	Стратегии конструирования ПО. Макетирование.
14.	Стратегии конструирования ПО. Инкрементный подход. Экстремальное программирование.
15.	Стратегии конструирования ПО. Инкрементный подход. Быстрая разработка приложений (RAID).
16.	Стратегии конструирования ПО. Agile-технологии. SCRAM.
17.	Стратегии конструирования ПО. Agile-технологии. Канбан.
18.	Стратегии конструирования ПО. Эволюционная стратегия. Спиральная модель.
19.	Оценка программного проекта. Объектно-ориентированные метрики.
20.	Оценка программного проекта. Функционально-ориентированные метрики.
21.	Оценка программного проекта. Анализ чувствительности программного проекта. Модель COSOMO.
22.	Определение требований к программному обеспечению. Функциональные требования. Множественность точек зрения.
23.	Определение требований к программному обеспечению. Анализ предметной области.
24.	Определение требований к программному обеспечению. Системные требования.
25.	Определение требований к программному обеспечению. Разработка Технического задания.
26.	Проектирование программных систем. Этапы проектирования.
27.	Проектирование программных систем. Предварительное проектирование. Структурирование системы.
28.	Проектирование программных систем. Предварительное проектирование. Моделирование управления.
29.	Модульный принцип построения программных систем. Типы связности модулей. Информационная и временная связности.

30.	Модульный принцип построения программных систем. Типы связности модулей. Логическая и коммуникативная связности.
31.	Модульный принцип построения программных систем. Типы связности модулей. Процедурная и функциональная связности.
32.	Модульный принцип построения программных систем. Информационная закрытость.
33.	Структурное тестирование программного обеспечения. Основные понятия и принципы.
34.	Тестирование программного обеспечения. Тестирование «черного ящика».
35.	Тестирование программного обеспечения. Тестирование «белого ящика».
36.	Классическая методика тестирования программных систем. Тестирование элементов.
37.	Классическая методика тестирования программных систем. Тестирование интеграций.
38.	Классическая методика тестирования программных систем. Тестирование правильности.
39.	Классическая методика тестирования программных систем. Системное тестирование. Тестирование восстановления.
40.	Классическая методика тестирования программных систем. Стрессовое тестирование.
41.	Отладка программного обеспечения.
42.	Проблема контроля версий.
43.	Системы контроля версий. Системы управления версиями (VCS - Version Control System).
44.	Системы контроля версий. Системы контроля ревизий (RCS - Revision Control System).
45.	Системы контроля версий. Системы управления исходным кодом (SCM – Source Code Management).
46.	Составление программной документации.

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета
	Учебным планом не предусмотрено

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения курсовой работы
	Учебным планом не предусмотрено

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов
1.	К какому типу проектов относятся проекты по разработке ПО: а) и к творческим, и к промышленным проектам +

	<p>б) к промышленным проектам в) к творческим проектам</p>
2.	<p>Что реализуют модели, представленные диаграммами UML: а) вид деятельности б) фазу разработки ПО в) точку зрения на программную систему +</p>
3.	<p>При использовании какого метода тестирования код программы доступен тестировщикам: а) при использовании любого метода тестирования б) при использовании метода белого ящика + в) при использовании метода черного ящика</p>
4.	<p>Что такое нагрузочное тестирование: а) тестирование системы на устойчивость к непредвиденным ситуациям б) тестирование системы на корректную работу с большими объемами данных + в) тестирование всей системы в целом, как правило, через ее пользовательский интерфейс</p>
5.	<p>Что определяют варианты использования: а) как функции, так и требования + б) только функции системы в) только требования к системе</p>
6.	<p>Какой этап следует за созданием требований к продукту при использовании метода Scrum: а) планирование итерации + б) анализ результатов, пересмотр требований в) выполнение итерации</p>
7.	<p>Укажите правильную последовательность этапов при каскадной модели жизненного цикла: а) Определение требований -&gt; Тестирование -&gt; Реализация; б) Проектирование -&gt; Реализация -&gt; Тестирование; в) Проектирование -&gt; Определение требований -&gt; Реализация.</p>
8.	<p>Проектирование ПО в основном рассматривается как а) архитектурное проектирование; + б) коммуникационные методы; в) детальные методы.</p>
9.	<p>Интерфейс пользователя — это а) набор методов взаимодействия компьютерной программы и пользователя этой программы; + б) набор методов для взаимодействия между программами; в) способ взаимодействия между объектами.</p>
10.	<p>Архитектура программной системы — а) декомпозиция решения для выделенного спектра задач домена на подсистемы или иерархию подсистем; б) определение системы в терминах вычислительных составляющих (подсистем) и интерфейсов между ними, которое отражает правила декомпозиции проблемы на составляющие; + в) соответствующие вариации состава выделенных компонент.</p>
11.	<p>Агрегация — а) отношения, утверждает наличие связи между понятиями, не уточняя зависимости их содержания и объемов; б) возможность для некоторого класса находиться одновременно в связи с одним элементом из определенного множества классов;</p>

	в) объединение нескольких понятий в новое понятие, существенные признаки нового понятия при этом могут быть либо суммой компонент или существенно новыми (отношение «доля — целое»). +
12.	Ассоциация — а) возможность для некоторого класса находиться одновременно в связи с одним элементом из определенного множества классов; б) объединение нескольких понятий в новое понятие, существенные признаки нового понятия о этом могут быть либо суммой компонент или существенно новыми (отношение «доля — целое»); в) самое общее отношение, утверждает наличие связи между понятиями, не уточняя зависимости их содержания и объемов. +
13.	Валидация — а) обеспечение соответствия разработки требованиям ее заказчиков. + б) проверка правильности трансформации проекта в код реализации; в) выявление всех ошибок.
14.	Верификация — а) обеспечение соответствия разработки требованиям ее заказчиков; б) проверка правильности трансформации проекта в программу; + в) действия на каждой стадии жизненного цикла с проверки и подтверждения соответствия стандартам.
15.	Понятность — это а) атрибут функциональности, указывающий на возможность предотвращать несанкционированный доступ; б) атрибут надежности, который указывает на способность программы к перезапуску для повторного выполнения; в) атрибут удобства, определяющий усилия, необходимые для распознавания логических концепций и условий их применения. +
16.	Артефакт — это а) любой продукт деятельности специалистов по разработке программного обеспечения; + б) результат ошибок разработчика во входных или проектных спецификациях; в) графическое представление элементов моделирования системы.

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
	Не предусмотрено

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

11.1. Методические указания для обучающихся по освоению лекционного материала .

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в

рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

Раздел 1. Основные понятия программной инженерии. Методологии разработки программного обеспечения.

Раздел 2. Анализ предметной области и требований к программному обеспечению.

Раздел 3. Основы проектирования и реализации программного обеспечения.

Раздел 4. Объектно-ориентированное проектирование и реализация программного обеспечения.

Раздел 5. Верификация и тестирование программного обеспечения

Раздел 6. Сервисно-ориентированная архитектура

Раздел 7. Облачные технологии

Раздел 8. Обеспечение качества программных систем

*Методические указания по освоению лекционного материала имеются в виде электронных ресурсов библиотеки ГУАП и системы LMS <https://lms.guap.ru/new/course/view.php?id=932>*

1. Разработка веб-приложений средствами языка PHP : [ Электронный ресурс ] : учебное пособие / А. В. Аграновский [и др.] ; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - Электрон. текстовые дан. - СПб. : Изд-во ГУАП, 2018. - 121 с

2. Основы интернет-программирования : учебное пособие / А. В. Аграновский, В. С. Павлов, Е. Л. Турнецкая ; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - СПб. : Изд-во ГУАП, 2018. - 135 с.

3. Тестирование веб-приложений : [ Электронный ресурс ] : учебное пособие / А. В. Аграновский [и др.] ; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - Электрон. текстовые дан. - СПб. : Изд-во ГУАП, 2020.



## 11.2. Методические указания для обучающихся по прохождению практических занятий

Практическое занятие является одной из основных форм организации учебного процесса, заключающаяся в выполнении обучающимися под руководством преподавателя комплекса учебных заданий с целью усвоения научно-теоретических основ учебной дисциплины, приобретения умений и навыков, опыта творческой деятельности.

Целью практического занятия для обучающегося является привитие обучающимся умений и навыков практической деятельности по изучаемой дисциплине.

Планируемые результаты при освоении обучающимся практических занятий:

- закрепление, углубление, расширение и детализация знаний при решении конкретных задач;
- развитие познавательных способностей, самостоятельности мышления, творческой активности;
- овладение новыми методами и методиками изучения конкретной учебной дисциплины;
- выработка способности логического осмысления полученных знаний для выполнения заданий;
- обеспечение рационального сочетания коллективной и индивидуальной форм обучения.

### Требования к проведению практических занятий

Студентам требуется выполнить 9 практических работ в соответствии с планом выполнения, изложенном в учебном пособии [1].

1. Методологии разработки программного обеспечения
2. Анализ предметной области и требований к программному обеспечению
3. Проектирование пользовательского интерфейса
4. Проведение технического аудита веб-приложения
5. Тестирование клиентской части веб-приложения
6. Тестирование производительности
7. Тестирование безопасности
8. Тестирование сетевых протоколов
9. Создание тестовой документации

По практическим работам 4-9 необходимо создать отчет. Пример отчета приводится в материалах ЭОС ГУАП.

Практическое занятие состоит из следующих элементов: вводная часть, основная и заключительная. Вводная часть обеспечивает подготовку студентов к выполнению задания на занятии. В нее входят: формулировка темы, цели и задач занятия, обоснование его значимости в профессиональной подготовке студентов; изложение теоретических основ работы; характеристика состава и особенностей заданий работы и объяснение методов (способов, приемов) их выполнения; характеристика требований к результату работы; проверка готовности студентов выполнять задания. Основная часть предполагает самостоятельное выполнение заданий студентами. Она может сопровождаться разъяснениями по ходу работы, устранением трудностей при выполнении работы, текущим контролем и оценкой результатов отдельных студентов, ответами на вопросы студентов. Возможно пробное выполнение задания(ий) под руководством преподавателя. Заключительная часть содержит: подведение общих итогов занятия; оценку результатов работы отдельных студентов; ответы на вопросы студентов; выдачу рекомендаций по устранению пробелов в системе знаний и умений студентов, по улучшению результатов работы; изложение сведений о подготовке к выполнению следующей работы.

В зависимости от уровня и полноты раскрытия темы задания студенты могут получить от 1 до 5 баллов. По заданиям требуется оформление отчета. Оформление

необходимо производить в соответствии с Правилами оформления текстовых документов по ГОСТ 7.32 – 2017 (<https://guap.ru/standart/doc>).

*Методические указания по освоению практических работ имеются в виде электронных ресурсов библиотеки ГУАП и системы LMS <https://lms.guap.ru/new/course/view.php?id=932>*

Тестирование веб-приложений : [ Электронный ресурс ] : учебное пособие / А. В. Аграновский [и др.] ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - Электрон. текстовые дан. - СПб. : Изд-во ГУАП, 2020.

### 11.3. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

#### Задание и требования к проведению лабораторных работ

Студентам требуется выполнить 9 лабораторных работ в соответствии с планом выполнения, изложенном в учебном пособии [1].

1. Проведение многокритериального анализа предметной области
2. Проектирование логической структуры веб-приложения
3. Разработка клиентской части. Язык гипертекстовой разметки HTML.
4. Разработка клиентской части. Каскадные таблицы стилей CSS
5. Разработка серверной части. Язык программирования PHP.
6. Разработка серверной части. Базы данных и СУБД MySQL
7. Разработка интерактивных страниц веб-системы. Язык программирования JavaScript и технология AJAX
8. Работа с социальными сетями
9. Работа с системами контроля версий GIT

#### Структура и форма отчета о лабораторной работе

1. Титульный лист
2. Задание на выполнение лабораторной работы
3. Порядок выполнения работы.

При этом текстовая часть работы дополняется скриншотами, показывающими все этапы выполнения задания на лабораторную работу.

4. Выводы о проделанной работе в формате эссе

## 5. Список использованных источников

### Требования к оформлению отчета о лабораторной работе

Оформление необходимо производить в соответствии с Правилами оформления текстовых документов по ГОСТ 7.32 – 2017 (<https://guap.ru/standart/doc>).

*Методические указания по освоению материала лабораторных работ имеются в виде электронных ресурсов библиотеки ГУАП и системы LMS <https://lms.guap.ru/new/course/view.php?id=932>.*

### 11.4. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

*Методические указания по освоению материала для самостоятельного изучения имеются в виде электронных ресурсов библиотеки ГУАП и системы LMS <https://lms.guap.ru/new/course/view.php?id=932>.*

### 11.5. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Текущий контроль успеваемости проводится в соответствии со стандартом организации ГУАП системы менеджмента качества 3.76 «Положение о текущем контроле успеваемости и промежуточной аттестации студентов и аспирантов ГУАП, обучающихся по образовательным программам высшего образования» на основании приказа ГУАП № 483-17 от 28.12.2017

Для проведения текущего контроля выбраны следующие методы:

- защита отчетов по лабораторным работам;
- выполнение практических работ;
- проведение контрольных работ;
- тестирование;
- контроль курсового проектирования и выполнения курсовых работ.

Для получения аттестации по текущему контролю студенту необходимо:

1. защитить не менее 25% отчетов от всех лабораторных семестра и выложить их в личный кабинет;

2. выполнить и защитить отчеты не менее 25% практических заданий.

11.6. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

– экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Промежуточный контроль успеваемости проводится в соответствии со стандартом организации ГУАП системы менеджмента качества 3.76 «Положение о текущем контроле успеваемости и промежуточной аттестации студентов и аспирантов ГУАП, обучающихся по образовательным программам высшего образования» на основании приказа ГУАП № 483-17 от 28.12.2017.

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой