

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

Кафедра № 82

УТВЕРЖДАЮ

Руководитель направления

проф., д. пед. н., доц.

(должность, уч. степень, звание)

А.Г. Степанов

(инициалы, фамилия)



(подпись)

«22» июня 2023 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Технологии программирования»  
(Наименование дисциплины)


Код направления подготовки/ специальности	09.03.03
Наименование направления подготовки/ специальности	Прикладная информатика
Наименование направленности	Интеллектуальные информационные системы и технологии
Форма обучения	заочная

Санкт-Петербург– 2023

Лист согласования рабочей программы дисциплины

Программу составил (а)

проф. д. пед. н. доц.  
(должность, уч. степень, звание)

  
(подпись, дата)

А.Г. Степанов  
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 82

«18» мая 2023 г, протокол № 10

Заведующий кафедрой № 82

д.э.н., доц.  
(уч. степень, звание)

  
(подпись, дата)

А.С. Будагов  
(инициалы, фамилия)

Ответственный за ОП ВО 09.03.03(02)


Ст. преподаватель  
(должность, уч. степень, звание)

  
(подпись, дата)

Н.В. Зуева  
(инициалы, фамилия)

Заместитель директора института №8 по методической работе

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

  
(подпись, дата)

Л.В. Рудакова  
(инициалы, фамилия)

## Аннотация

Дисциплина «Технологии программирования» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 09.03.03 «Прикладная информатика» направленности Интеллектуальные информационные системы и технологии. Дисциплина реализуется кафедрой «№82».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

УК-2 «Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений»

ПК-2 «Способность проектировать, разрабатывать и тестировать программные модули»

Содержание дисциплины охватывает круг вопросов, связанных с формирования информационной культуры и имеет целью обучение студентов языку программирования Python и базовым методам программирования, способам применения современных вычислительных средств и программных систем в практической деятельности специалиста.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа студента, консультации, курсовое проектирование.

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

Общая трудоемкость освоения дисциплины составляет 6 зачетных единиц, 216 часов.

Язык обучения по дисциплине «русский»

## 1. Перечень планируемых результатов обучения по дисциплине

### 1.1. Цели преподавания дисциплины

Целью преподавания дисциплины является формирование у обучающихся способности определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения исходя из действующих правовых норм, имеющихся ресурсов и ограничений, а также способность проектировать, разрабатывать и тестировать программные модули.

В результате обучаемый имеет возможность использовать стандартные технологии программирования, разрабатывать, внедрять и адаптировать прикладное программное обеспечение, проводить описание прикладных процессов и информационного обеспечения решения прикладных задач, программировать приложения и создавать программные прототипы и осуществлять поддержку информационного обеспечения решения прикладных задач.

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
Универсальные компетенции	УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.В.3 владеть навыками использования цифровых средств для решения поставленной задачи
Профессиональные компетенции	ПК-2 Способность проектировать, разрабатывать и тестировать программные модули	ПК-2.3.1 знать методы и приемы алгоритмизации поставленных задач, стандартные алгоритмы и области их применения, технологии программирования, особенности выбранной среды программирования, методы и средства проверки работоспособности программного обеспечения ПК-2.У.1 уметь писать программный код на выбранном языке программирования, составлять программу тестирования компонентов программного обеспечения, проводить анализ исполнения требований к программному обеспечению ПК-2.В.1 владеть навыками применения методов и средств

		проверки работоспособности программного обеспечения
--	--	--

## 2. Место дисциплины в структуре ОП

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

- «Информатика»,
- «Основы цифровой грамотности»,
- «Дискретная математика»,
- «Компьютерная графика»,
- «Основы программирования».

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

- «Математические основы искусственного интеллекта»,
- «Методы обработки больших данных».

## 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	
		№5	№6
1	2	3	4
<b>Общая трудоемкость дисциплины, ЗЕ/ (час)</b>	6/ 216	3/ 108	3/ 108
<b>Из них часов практической подготовки</b>	19	10	9
<b>Аудиторные занятия, всего час.</b>	42	16	26
в том числе:			
лекции (Л), (час)	14	6	8
практические/семинарские занятия (ПЗ), (час)	8		8
лабораторные работы (ЛР), (час)	20	10	10
курсовой проект (работа) (КП, КР), (час)	*		*
экзамен, (час)	9	9	
<b>Самостоятельная работа, всего (час)</b>	165	83	82
<b>Вид промежуточной аттестации:</b> зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз., Дифф. Зач.	Экз.	Дифф. Зач.

Примечание: \*\* кандидатский экзамен

\* - часы , не входящие в аудиторную нагрузку

#### 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)
Семестр 5					
1. Введение в дисциплину	1				20
2. Введение в язык программирования Python	1		2		20
3. Атомарные типы данных языка Python.	2		4		20
4. Ссылочные типы данных и последовательности в Python.	2		4		21
Итого в семестре:	6		10		83
Семестр 6					
5. Операторы языка Python.	2		2		25
6. Встроенные и пользовательские функции и классы.	2		4		25
7. Модули в Python.	2		4		20
8. Общий подход к проектированию программ.	2				12
Выполнение курсовой работы				0	
Итого в семестре:	8	8	10		82
Итого	14	8	20	0	165

##### 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.	Введение в дисциплину. Литература. Основные направления развития средств автоматизации программирования
2.	Введение в язык программирования Python. Интегрированная среда разработки Python. Комментарии. Идентификаторы и переменные
3.	Атомарные типы данных языка Python. Изменяемые и неизменяемые типы данных. Организация хранения данных в памяти и объявление переменных. Операции ( побитовые, сравнение, логические, арифметические. Литералы и строки. Встроенные методы строк. Форматированные строки.
4.	Ссылочные типы данных и последовательности в Python. Списки, кортежи, словари, множества и замороженные множества.
5.	Операторы языка Python. Операторы выбора последовательности выполнения программы (if, match, for, while, continue, break, pass
6.	Встроенные и пользовательские функции и классы. Область видимости переменной. Декораторы. Лямбда –функции. Обработка исключений. Менеджеры контекста. Работа с файлами.
7.	Модули в Python. Стандартные библиотеки и их установка.
8.	Общий подход к проектированию программ. Структурная декомпозиция и структурное программирование. Объектно-ориентированная декомпозиция Разработка общего алгоритма. Стиль программирования.

#### 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)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 6					
1	Проектирование программы	Мозговой штурм	2	2	8
2	Разработка структуры программы	Мозговой штурм	2	2	8
3	Разработка технологии тестирования программных модулей	Мозговой штурм	2	2	8
4	Разработка технологии итогового тестирования	Мозговой штурм	2	2	8
Всего			8		

#### 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)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 5				
1	Начальный запуск IDLE	2	2	2
2	Базовые операции со строками	2	2	3
3	Операции со списками	2	2	4
4	Операции с кортежами	2	2	4,5
5	Операции со словарями	2	2	4,5
Семестр 6				
6	Операции с множествами	4	4	4,5
7	Создание и выполнение функций	2	2	6
8	Работа с файлами	2	2	7
9	Подключение модулей и работа с библиотеками	2	2	7
Всего		20		

#### 4.5. Курсовое проектирование/ выполнение курсовой работы

Цель курсовой работы: формирование умения писать программный код на выбранном языке программирования, составлять программу тестирования компонентов программного обеспечения, проводить анализ исполнения требований к программному обеспечению, овладение навыками использования цифровых средств для решения поставленной задачи, а также создания средств проверки работоспособности программного обеспечения

Часов практической подготовки: 8

Примерные темы заданий на курсовую работу приведены в разделе 10 РПД.

#### 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 5, час	Семестр 6, час
1	2	3	4
Изучение теоретического материала дисциплины (ТО)	50	40	10
Курсовое проектирование (КП, КР)	52		52
Подготовка к текущему контролю успеваемости (ТКУ)	20	10	10
Домашнее задание (ДЗ)	13	13	
Контрольные работы заочников (КРЗ)		0	
Подготовка к промежуточной аттестации (ПА)	30	20	10
Всего:	165	83	82

#### 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)
	Резова, Н. Л. Технология программирования: учебное пособие / Н. Л. Резова, Г. Ш. Шкаберина. — Красноярск: СибГУ им. академика М. Ф. Решетнёва, 2019. — 94 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/147448">https://e.lanbook.com/book/147448</a> (дата обращения: 25.07.2022). — Режим доступа: для авториз. пользователей.	
	Копырин, А. С. Программирование на Python: учебное пособие / А. С. Копырин, Т. Л. Салова. — Сочи: СГУ, 2018. — 48 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/147665">https://e.lanbook.com/book/147665</a> (дата обращения: 25.07.2022). — Режим доступа: для авториз. пользователей.	
	Демидова, Л. А. Кластерный анализ. Python: учебное пособие / Л. А. Демидова. — Москва: РТУ МИРЭА, 2022. — 103 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/240092">https://e.lanbook.com/book/240092</a> (дата обращения: 25.07.2022). — Режим доступа: для авториз. пользователей.	
	Янцев, В. В. Web-программирование на Python: учебное пособие для вузов / В. В. Янцев. — Санкт-Петербург: Лань, 2022. — 180 с. — ISBN 978-5-8114-9461-3. — Текст:	



Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
	электронный // Лань: электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/233264">https://e.lanbook.com/book/233264</a> (дата обращения: 25.07.2022). — Режим доступа: для авториз. пользователей.	

#### 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="http://lms.guap.ru/">http://lms.guap.ru/</a>	Единая электронная образовательная среда ГУАП

#### 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
1.	Интерпретатор Python

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	14-05, 14-15, 24-15, 24-16 ЛС. 52-18 БМ, 12-03 Гаст.
2	Специализированная лаборатория	14-06 – 14-11 ЛС

## 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
Дифференцированный зачет	Список вопросов ;
Выполнение курсовой работы	Экспертная оценка на основе требований к содержанию курсовой работы по дисциплине.

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> </ul>

Оценка компетенции	Характеристика сформированных компетенций
5-балльная шкала	
	<ul style="list-style-type: none"> <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	Код индикатора
1.	Опишите три основные составляющие информатики	ПК-2.3.1
2.	Что представляет собой классическая структура аппаратуры Джона фон Неймана?	ПК-2.3.1
3.	Чем транслятор отличается от компилятора и интерпретатора?	ПК-2.3.1
4.	Что понимается под парадигмой программирования?	ПК-2.3.1
5.	Чем императивная парадигма программирования отличается от декларативной?	ПК-2.3.1
6.	Что такое алгоритм?	ПК-2.3.1
7.	Что понимается под сценарным языком?	ПК-2.3.1
8.	Как можно использовать режим калькулятора интегрированной среды разработки Python?	УК-2.В.3
9.	Как в Python обозначаются строки, литералы и комментарии?	УК-2.В.3
10.	Каковы правила составления идентификаторов в Python?	УК-2.В.3
11.	Какие атомарные типы данных поддерживает Python и как объявляются переменные?	УК-2.В.3
12.	Какая разница между изменяемыми и неизменяемыми типами данных Python?	УК-2.В.3
13.	Как в Python можно преобразовать число в строку и наоборот?	УК-2.В.3
14.	Какие операции поддерживает Python и чем оператор отличается от операции?	УК-2.В.3
15.	Что такое коллекция в Python ? и какие методы работы с ними существуют	УК-2.В.3
16.	Какие операции со строками поддерживает Python? Зачем нужны методы строк?	УК-2.В.3
17.	Как Python хранит данные и строки в памяти?	УК-2.В.3
18.	Что такое ссылочные типы данных в Python?	УК-2.В.3
19.	Как работает механизм памяти в Python?	УК-2.В.3
20.	Зачем нужны счетчик ссылок и сборщик мусора в Python?	УК-2.В.3

21.	Что такое последовательность и какие операции могут выполняться с ними в Python?	ПК-2.3.1
22.	Что такое список в Python? Как списки организованы в памяти? Зачем нужны методы списков?	ПК-2.3.1
23.	Что такое временная сложность алгоритма в Python?	ПК-2.3.1
24.	Как можно использовать вложенные списки в Python и что такое связанные и рекурсивные списки?	ПК-2.3.1
25.	Как работает стек, очередь, дерево и куча в Python?	ПК-2.3.1
26.	Каковы особенности копирования списков в Python	ПК-2.3.1
27.	Что такое кортеж в Python и какие методы работы с ним существуют?	ПК-2.3.1
28.	Что такое словарь в Python и какие методы работы с ним существуют?	ПК-2.3.1
29.	Что такое множество в Python и какие методы работы с ним существуют? Что такое замороженное множество?	ПК-2.3.1
30.	Что такое хеширование в Python?	ПК-2.3.1
31.	Что такое оператор if в Python? Как указать последовательность операторов, которая должна выполняться в зависимости от значения условия?	ПК-2.У.1
32.	Как работает оператор цикла while с предусловием в Python? Как указать последовательность операторов, которая должна выполняться в зависимости от значения условия?	ПК-2.У.1
33.	Как работает оператор цикла while с постусловием в Python? Как указать последовательность операторов, которая должна выполняться в зависимости от значения условия?	ПК-2.У.1
34.	Как работает оператор цикла for в Python? Как указать последовательность операторов, которая должна выполняться в зависимости от значения условия?	ПК-2.У.1

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	Код индикатора
1.	Какие виды функций есть в Python?	ПК-2.У.1
2.	Как создать пользовательскую функцию в Python? Как указать последовательность операторов, которая должна выполняться в определении функции?	ПК-2.У.1
3.	Область видимости и способы возврата параметров функции в Python.	ПК-2.У.1
4.	Как указать последовательность операторов, которая должна выполняться в цикле?	ПК-2.У.1
5.	Что такое правило распознавания имен и в чем его смысл в Python?	ПК-2.У.1
6.	Как выполняется обработка исключений в Python?	ПК-2.У.1
7.	Как создать класс в Python? Каков состав модуля класса? Как создать объект на основе класса?	ПК-2.В.1

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
8.	Что такое модуль в Python и как можно его использовать?	ПК-2.В.1
9.	Какие библиотеки Python называют стандартными?	ПК-2.В.1
10.	Как осуществляется работа с файлами в Python?	ПК-2.В.1
11.	Как осуществляется работа с отладчиком в Python?	ПК-2.В.1

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
1.	Программирование рядов и вычисление специальных функций
2.	Программирование табличных вычислений
3.	Решение уравнений и операции с многочленами
4.	Математика комплексных чисел
5.	Задачи интерполяции, аппроксимации, экстраполяции, численного интегрирования и дифференцирования
6.	Спектральный анализ
7.	Предложенная студентом и согласованная с руководителем

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	Код индикатора
1.	Переведите двоичное число 001001 в десятичную систему счисления Ответ: <input type="text"/>	УК-2.В.3
2.	Поставьте в соответствие числовую константу с ее типом 266      Ответ 1 <input type="text" value="Выберите..."/> 0xFFAA    Ответ 2 <input type="text" value="Выберите..."/> 0341      Ответ 3 <input type="text" value="Выберите..."/>	ПК-2.У.1
3.	Целью тестирования является отыскание Выберите один или несколько ответов: <input type="checkbox"/> а. семантических ошибок программы <input type="checkbox"/> б. орфографических ошибок программы <input type="checkbox"/> в. синтаксических ошибок программы <input type="checkbox"/> г. алгоритмических ошибок программы	УК-2.В.3

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

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
1.	Процедурно-ориентированное программирование
2.	Программирование на ассемблере
3.	Структурное программирование
4.	Визуальное конструирование
5.	Управление данными
6.	Логико-ориентированное программирование
7.	Объектно-ориентированное программирование
8.	Декларативное программирование
9.	Императивное программирование
10.	Классификация языков программирования

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.

Структура предоставления лекционного материала соответствует таблице 4. Лекционный материал сопровождается демонстрацией слайдов Презентация хранится в системе управления обучением на странице курса.

### 11.2. Методические указания для обучающихся по участию в семинарах Семинарские занятия не предусмотрены.

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

Практические занятия по этой дисциплине проводятся только со студентами заочной формы обучения. Практическое занятие заключается в выполнении обучающимися под руководством преподавателя комплекса учебных заданий с по разработке структуры программного средства и методов его тестирования с целью усвоения научно-теоретических основ учебной дисциплины, приобретения умений и навыков, опыта творческой деятельности.

#### Планируемые результаты при освоении обучающимися 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

При подготовке к практическим занятиям студент должен выбрать и утвердить тему своей курсовой работы. На первом занятии он должен описать математический метод, который лежит в основе его будущей программы. На втором занятии студент должен разработать общую структуру программного изделия и составить его общий алгоритм. На третьем занятии студент должен предложить технологию тестирования модуля решающего основную математическую задачу. На четвертом занятии осуждается предлагаемая студентом общая технология тестирования разрабатываемого программного обеспечения.

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

Для проведения лабораторных занятий используется рукопись учебно-методического пособия Программирование на Python размещенное на странице дисциплины в системе LMS. Оно содержит методические указания, задание и требования к проведению лабораторных работ, структуру и форму отчета о лабораторной работе, требования к оформлению отчета о лабораторной работе и контрольные вопросы.

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.

Выполнение лабораторных работ является обязательным для студента. Результаты выполнения работ оформляются в виде отчета, защищаются, утверждаются преподавателем и являются основанием для допуска студента к промежуточной аттестации в период экзаменационной сессии.

#### 11.5. Методические указания для обучающихся по прохождению курсового проектирования/выполнения курсовой работы

Выполнение курсовой работы начинается с самого начала семестра и представляет собой последовательность оговоренных в задании на курсовой проектирование этапов с заранее объявленными сроками их завершения. .

Для выполнения курсовой работы студентами дневной формы обучения в системе LMS создана отдельная страница, а при обучении по заочной форме используется страница основного курса. На странице размещен набор заданий, которые студенты должны выполнить в процессе работы. Страница обеспечивает контроль за сроками выполнения заданий. Опоздание с их выполнением оказывает существенное влияние на итоговую оценку.



Существуют Методические указания для выполнения курсовой работы Объектно-ориентированный анализ и программирование: методические указания по выполнению курсовой работы / С.-Петербург. гос. ун-т аэрокосм. приборостроения; сост.: А. Г. Степанов, В. М. Космачев, Н. В. Зуева. - СПб. : Изд-во ГУАП, 2017. - 31 с. Они содержат необходимые методические указания, предлагаемые варианты тем курсовых работ, структуру пояснительной записки курсовой работы, требования по ее оформлению и другие необходимые материалы.

При выполнении курсовой работы студенты приобретают начальные навыки освоения и использования требуемых компетенций.

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

Методические указания для самостоятельной работы студентов находятся на странице дисциплины в LMS. 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 дополнительно включает 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.

Выполнение самостоятельной работы является обязательным для студента. Результаты работы оформляются в виде отчета, утверждаются преподавателем и являются основанием для допуска студента к промежуточной аттестации в период экзаменационной сессии.

#### 11.7. 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.

Контроль освоения теоретического материала проводится в форме компьютерного тестирования в конце каждого месяца изучения дисциплины с учетом выполнения заданий и получения отчетов о самостоятельной работе студента.

Кроме этого, контроль освоения практической составляющей дисциплины проводится по материалам лабораторных работ по мере их защиты. Выполнение лабораторного практикума в полном объеме является обязательным для студента и является основанием для допуска его к промежуточной аттестации в период экзаменационной сессии.

#### 11.8. 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».

– зачет – это форма оценки знаний, полученных обучающимся в ходе изучения учебной дисциплины в целом или промежуточная (по окончании семестра) оценка знаний обучающимся по отдельным разделам дисциплины с аттестационной оценкой «зачтено» или «не зачтено».

Экзамен и зачет по дисциплине проводится в форме компьютерного тестирования в вычислительных лабораториях университета по расписанию сессии. В необходимых случаях допускается дистанционное проведение компьютерного тестирования. Результаты промежуточной аттестации оформляются в виде ведомости.

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