

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

Кафедра № 82

УТВЕРЖДАЮ

Руководитель образовательной программы

д.э.н., проф.

(должность, уч. степень, звание)

А.С. Будагов

(инициалы, фамилия)

(подпись)

«20» февраля 2025 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Технологии программирования»  
(Наименование дисциплины)

|   |   |
|---|---|
| Код направления подготовки/<br>специальности          | 09 03 03  |
| Наименование направления<br>подготовки/ специальности | Прикладная информатика                                  |
| Наименование<br>направленности                        | Интеллектуальные информационные системы и<br>технологии |
| Форма обучения  | очно-заочная  |
| Год приема  | 2025  |

Лист согласования рабочей программы дисциплины

Программу составил (а)

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

(подпись, дата)

А.Г. Степанов  
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 82

«13» февраля 2025 г, протокол № 6

Заведующий кафедрой № 82

д.э.н., проф.  
(уч. степень, звание)

(подпись, дата)

А.С. Будагов  
(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<br>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<br>ПК-2.В.1 владеть навыками применения методов и средств |

|  |  |  |
|--|--|--|
|  |  | проверки работоспособности<br>программного обеспечения |
|--|--|--|

## 2. Место дисциплины в структуре ОП

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

- «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 |       |
|---|--------|---------------------------|-------|
|   |        | №4                        | №5    |
| 1   | 2      | 3                         | 4     |
| <b>Общая трудоемкость дисциплины, ЗЕ/ (час)</b>   | 6/ 216 | 4/ 144                    | 2/ 72 |
| <b>Из них часов практической подготовки</b>   | 59     | 51                        | 8     |
| <b>Аудиторные занятия, всего час.</b>   | 102    | 85                        | 17    |
| в том числе:  |        |                           |       |
| лекции (Л), (час)   | 34     | 34                        |       |
| практические/семинарские занятия (ПЗ), (час)  | 17     | 17                        |       |
| лабораторные работы (ЛР), (час)   | 34     | 34                        |       |
| курсовой проект (работа) (КП, КР), (час)  | 17     |                           | 17    |
| экзамен, (час)  | 36     | 36                        |       |
| <b>Самостоятельная работа, всего (час)</b>  | 78     | 23                        | 55    |
| <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<br>(час) | ПЗ (СЗ)<br>(час) | ЛР<br>(час) | КП<br>(час) | СРС<br>(час) |
|--|-----------------|------------------|-------------|-------------|--------------|
| Семестр 4                                  |                 |                  |             |             |              |
| 1. Введение в дисциплину                   | 4               | 2                |             |             |              |
| 2. Введение в язык программирования Python | 4               | 2                |             |             |              |
| 3. Атомарные типы данных языка Python.     | 4               | 2                | 2           |             |              |

| Разделы, темы дисциплины                                | Лекции<br>(час) | ПЗ (СЗ)<br>(час) | ЛР<br>(час) | КП<br>(час) | СРС<br>(час) |
|---|-----------------|------------------|-------------|-------------|--------------|
| 4. Ссылочные типы данных и последовательности в Python. | 4               | 2                | 2           |             |              |
| 5. Операторы языка Python.                              | 4               | 2                | 2           |             |              |
| 6. Встроенные и пользовательские функции и классы.      | 4               | 2                | 2           |             |              |
| 7. Модули в Python.                                     | 4               | 5                | 2           |             | 10           |
| 8. Общий подход к проектированию программ.              | 8               |                  |             |             | 13           |
| Итого в семестре:                                       | 34              | 17               | 10          |             | 23           |
| Семестр 6   |                 |                  |             |             |              |
| Выполнение курсовой работы                              |                 |                  |             | 17          | 55           |
| Итого в семестре:                                       |                 |                  |             | 17          | 55           |
| Итого   | 34              | 17               | 10          | 17          | 78           |

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

#### 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. Стил 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 4 |   |                            |                     |                                       |                      |
| 1         | Работа с текстом  | групповые дискуссии        | 4                   |                                       | 7                    |
| 2         | Структуры данных  | групповые дискуссии        | 4                   |                                       | 7                    |
| 3         | Алгоритмы   | групповые дискуссии        | 4                   |                                       | 7                    |
| 4         | Параллельные вычисления: процессы, потоки и сопрограммы | групповые дискуссии        | 5                   |                                       | 7                    |
| Всего     |   |                            | 17                  |                                       |                      |

## 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 4 |   |                     |                                       |                      |
| 1         | Начальный запуск IDLE                       | 2                   | 2                                     | 2                    |
| 2         | Базовые операции со строками                | 2                   | 2                                     | 3                    |
| 3         | Операции со списками                        | 4                   | 4                                     | 4                    |
| 4         | Операции с кортежами                        | 4                   | 4                                     | 4,5                  |
| 5         | Операции со словарями                       | 4                   | 4                                     | 4,5                  |
| 6         | Операции с множествами                      | 4                   | 4                                     | 4,5                  |
| 7         | Создание и выполнение функций               | 4                   | 4                                     | 6                    |
| 8         | Работа с файлами                            | 4                   | 4                                     | 7                    |
| 9         | Подключение модулей и работа с библиотеками | 6                   | 6                                     | 7                    |
| Всего     |   | 34                  | 34                                    |                      |

## 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 4, час | Семестр 5, час |
|---|------------|----------------|----------------|
| 1   | 2          | 3              | 4              |
| Изучение теоретического материала дисциплины (ТО) |            |                |                |
| Курсовое проектирование (КП, КР)                  | 55         |                | 55             |
| Расчетно-графические задания (РГЗ)                |            |                |                |
| Выполнение реферата (Р)                           |            |                |                |
| Подготовка к текущему контролю успеваемости (ТКУ) | 5          | 1              |                |
| Домашнее задание (ДЗ)                             | 5          | 2              |                |
| Контрольные работы заочников (КРЗ)                | 10         | 10             |                |
| Подготовка к промежуточной аттестации (ПА)        | 10         | 10             |                |
| Всего:  | 78         | 23             | 55             |

#### 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/<br>URL<br>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/<br>URL<br>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;<br>Тесты.  |
| 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                    | Характеристика сформированных компетенций   |
|---------------------------------------|---|
| 5-балльная шкала                      |   |
| «отлично»<br>«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»<br>«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»<br>«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»<br>«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   | 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 |
| 35. | Как осуществляется работа с отладчиком в Python?  | ПК-2.В.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 |
|-------|---|----------------|
|       | 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                   |
|-------|--|
| 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.  | Прочитайте текст, выполните необходимые вычисления и напишите ответ. Вычислите результат выполнения операции взятия модуля числа 56 по основанию 10 и введите его с клавиатуры   | УК-2.В.3          |            |                            |  |   |  |   |   |  |  |          |
| 2.  | Прочитайте текст и напишите ответ. Парадигма программирования, предназначенная для задания или описания наших знаний того, что мы знаем наверняка, называется  | УК-2.В.3          |            |                            |  |   |  |   |   |  |  |          |
| 3.  | Прочитайте текст и установите соответствие. Установите соответствие между цифровыми средствами и их назначением: <table><tr><td>Цифровое средство</td><td>Назначение</td></tr><tr><td>1. Текстовый редактор</td><td>a. Создание и редактирование изображений</td></tr><tr><td>2. Электронная таблица</td><td>b. Обработка и анализ структурированных данных</td></tr><tr><td>3. Система управления базами данных</td><td>c. Создание и редактирование текстовых документов</td></tr><tr><td>4. Графический редактор</td><td>d. Хранение и управление данными</td></tr></table>                             | Цифровое средство | Назначение | 1. Текстовый редактор      | a. Создание и редактирование изображений         | 2. Электронная таблица                    | b. Обработка и анализ структурированных данных             | 3. Система управления базами данных     | c. Создание и редактирование текстовых документов   | 4. Графический редактор                  | d. Хранение и управление данными                               | УК-2.В.3 |
| Цифровое средство                         | Назначение   |                   |            |                            |  |   |  |   |   |  |  |          |
| 1. Текстовый редактор                     | a. Создание и редактирование изображений   |                   |            |                            |  |   |  |   |   |  |  |          |
| 2. Электронная таблица                    | b. Обработка и анализ структурированных данных   |                   |            |                            |  |   |  |   |   |  |  |          |
| 3. Система управления базами данных       | c. Создание и редактирование текстовых документов  |                   |            |                            |  |   |  |   |   |  |  |          |
| 4. Графический редактор                   | d. Хранение и управление данными   |                   |            |                            |  |   |  |   |   |  |  |          |
| 4.  | Прочитайте текст и установите соответствие. <table><tr><td>Цифровое средство</td><td>Назначение</td></tr><tr><td>1. Система контроля версий</td><td>a. Написание, отладка и запуск программного кода</td></tr><tr><td>2. Интегрированная среда разработки (IDE)</td><td>b. Управление изменениями и совместная работа над проектом</td></tr><tr><td>3. Система автоматического тестирования</td><td>c. Автоматизация процесса тестирования программного</td></tr><tr><td>4. Облачный сервис для совместной работы</td><td>d. Хранение и обмен файлами, совместная работа над документами</td></tr></table> | Цифровое средство | Назначение | 1. Система контроля версий | a. Написание, отладка и запуск программного кода | 2. Интегрированная среда разработки (IDE) | b. Управление изменениями и совместная работа над проектом | 3. Система автоматического тестирования | c. Автоматизация процесса тестирования программного | 4. Облачный сервис для совместной работы | d. Хранение и обмен файлами, совместная работа над документами | УК-2.В.3 |
| Цифровое средство                         | Назначение   |                   |            |                            |  |   |  |   |   |  |  |          |
| 1. Система контроля версий                | a. Написание, отладка и запуск программного кода   |                   |            |                            |  |   |  |   |   |  |  |          |
| 2. Интегрированная среда разработки (IDE) | b. Управление изменениями и совместная работа над проектом   |                   |            |                            |  |   |  |   |   |  |  |          |
| 3. Система автоматического тестирования   | c. Автоматизация процесса тестирования программного  |                   |            |                            |  |   |  |   |   |  |  |          |
| 4. Облачный сервис для совместной работы  | d. Хранение и обмен файлами, совместная работа над документами   |                   |            |                            |  |   |  |   |   |  |  |          |
| 5.  | Прочитайте текст и установите последовательность. Расставьте этапы разработки программного обеспечения в правильной последовательности:<br>1. Анализ требований<br>2. Проектирование архитектуры<br>3. Написание кода<br>4. Тестирование<br>5. Развертывание и эксплуатация  |                   |            |                            |  |   |  |   |   |  |  |          |
| 6.  | Прочитайте текст и установите последовательность. Расставьте этапы разработки веб-сайта в правильной последовательности:<br>1. Разработка дизайна и макета<br>2. Верстка HTML и CSS<br>3. Программирование на стороне клиента (JavaScript)<br>4. Программирование на стороне сервера (PHP, Python, Ruby и т.д.)<br>5. Тестирование и отладка   | УК-2.В.3          |            |                            |  |   |  |   |   |  |  |          |
| 7.  | Прочитайте текст и выберите правильный ответ. Какой из перечисленных инструментов наиболее подходит для разработки мобильных приложений на платформе Android?<br>1. Microsoft Visual Studio<br>2. PyCharm<br>3. Android Studio<br>4. Xcode   | ПК-2.3.1          |            |                            |  |   |  |   |   |  |  |          |

| № п/п   | Примерный перечень вопросов для тестов  | Код индикатора    |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
|---|---|-------------------|----------|--------------------------------------|-------------------------|---|-----------------------------|-------------------------------------|------------|--|--------------|-------------------------------------|-----------|---------------------------------------|-------|--|-----------|----------|--|----------|
| 8.  | Прочитайте текст и выберите один или несколько ответов. Какие из перечисленных цифровых средств могут быть использованы для организации совместной разработки программного обеспечения?<br>1. Система контроля версий (Git)<br>2. Облачный сервис для хранения файлов (Dropbox)<br>3. Система управления проектами (Jira)<br>4. Мессенджер для командной коммуникации (Slack)<br>5. Интегрированная среда разработки (IDE)  |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 9.  | Прочитайте текст и напишите ответ. Парадигма программирования, предназначенная для создания алгоритмов узнавания того, что мы не знаем непосредственно, но хотим и умеем (считаем, что умеем) узнавать, называется...   | УК-2.В.3          |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 10.   | Прочитайте текст и выберите один или несколько ответов. При разработке программы на языке Python использовались следующие идентификаторы. Отметьте те идентификаторы, которые соответствуют требованиям языка Python<br>1. val1<br>2. ib2!<br>3. px __m<br>4. tt6   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 11.   | Прочитайте текст и установите соответствие <table><tr><th>Название операции</th><th>Символ</th></tr><tr><td>1. Разность множеств символ -&gt;;</td><td>a. -</td></tr><tr><td>2. Симметричная разность множеств символ;</td><td>b. ^</td></tr><tr><td>3. Объединение множеств символ;</td><td>c.  </td></tr><tr><td>4. Равенство множеств символ;</td><td>d. ==</td></tr><tr><td>5. Пересечение множеств символ;</td><td>e. &amp;</td></tr><tr><td>6. Проверка на вхождение в множество;</td><td>f. in</td></tr><tr><td>7. Проверка на не вхождение в множество;</td><td>g. not in</td></tr></table> | Название операции | Символ   | 1. Разность множеств символ ->;      | a. -                    | 2. Симметричная разность множеств символ;                 | b. ^                        | 3. Объединение множеств символ;     | c.         | 4. Равенство множеств символ;          | d. ==        | 5. Пересечение множеств символ;     | e. &      | 6. Проверка на вхождение в множество; | f. in | 7. Проверка на не вхождение в множество; | g. not in | ПК-2.3.1 |  |          |
| Название операции   | Символ  |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 1. Разность множеств символ ->;                           | a. -  |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 2. Симметричная разность множеств символ;                 | b. ^  |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 3. Объединение множеств символ;                           | c.  |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 4. Равенство множеств символ;                             | d. ==   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 5. Пересечение множеств символ;                           | e. &  |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 6. Проверка на вхождение в множество;                     | f. in   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 7. Проверка на не вхождение в множество;                  | g. not in   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 12.   | Прочитайте текст и установите соответствие. Поставьте в соответствие данные языка Python их типу <table><tr><th>Данные</th><th>Тип</th></tr><tr><td>1. { 10,20,30,40,50,60,70,80,90,100}</td><td>a. словарь</td></tr><tr><td>2. { 1:10,2:20,3:30,4:40,5:50,6:60,7:70,8:80,9:90,10:100}</td><td>b. строка</td></tr><tr><td>3. [10,20,30,40,50,60,70,80,90,100]</td><td>c. список;</td></tr><tr><td>4. '{ 10,20,30,40,50,60,70,80,90,100}'</td><td>d. множество</td></tr><tr><td>5. (10,20,30,40,50,60,70,80,90,100)</td><td>e. кортеж</td></tr></table>  | Данные            | Тип      | 1. { 10,20,30,40,50,60,70,80,90,100} | a. словарь              | 2. { 1:10,2:20,3:30,4:40,5:50,6:60,7:70,8:80,9:90,10:100} | b. строка                   | 3. [10,20,30,40,50,60,70,80,90,100] | c. список; | 4. '{ 10,20,30,40,50,60,70,80,90,100}' | d. множество | 5. (10,20,30,40,50,60,70,80,90,100) | e. кортеж | ПК-2.3.1                              |       |  |           |          |  |          |
| Данные  | Тип   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 1. { 10,20,30,40,50,60,70,80,90,100}                      | a. словарь  |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 2. { 1:10,2:20,3:30,4:40,5:50,6:60,7:70,8:80,9:90,10:100} | b. строка   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 3. [10,20,30,40,50,60,70,80,90,100]                       | c. список;  |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 4. '{ 10,20,30,40,50,60,70,80,90,100}'                    | d. множество  |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 5. (10,20,30,40,50,60,70,80,90,100)                       | e. кортеж   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 13.   | Прочитайте текст и установите соответствие. Поставьте в соответствие типы данных языка Python их свойствам <table><tr><th>Тип данных</th><th>Свойство</th></tr><tr><td>1. set</td><td>a. изменяемые (mutable)</td></tr><tr><td>2. dict</td><td>b. неизменяемый (immutable)</td></tr><tr><td>3. list;</td><td></td></tr><tr><td>4. bool;</td><td></td></tr><tr><td>5. float</td><td></td></tr><tr><td>6. tuple l</td><td></td></tr><tr><td>7. frozenset</td><td></td></tr><tr><td>8. int</td><td></td></tr></table>  | Тип данных        | Свойство | 1. set                               | a. изменяемые (mutable) | 2. dict   | b. неизменяемый (immutable) | 3. list;                            |            | 4. bool;                               |              | 5. float                            |           | 6. tuple l                            |       | 7. frozenset                             |           | 8. int   |  | ПК-2.3.1 |
| Тип данных  | Свойство  |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 1. set  | a. изменяемые (mutable)   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 2. dict   | b. неизменяемый (immutable)   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 3. list;  |   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 4. bool;  |   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 5. float  |   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 6. tuple l  |   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 7. frozenset  |   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |
| 8. int  |   |                   |          |                                      |                         |   |                             |                                     |            |  |              |                                     |           |                                       |       |  |           |          |  |          |

| №<br>п/п                       | Примерный перечень вопросов для тестов   |  |  | Код<br>индикатора |            |                           |   |                                |   |                           |   |                |   |          |
|--------------------------------|--|--|--|-------------------|------------|---------------------------|---|--------------------------------|---|---------------------------|---|----------------|---|----------|
|                                | 9. unicode   |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
|                                | 10. str  |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
| 14.                            | Прочитайте текст и установите соответствие. Установите соответствие между алгоритмическими структурами и их описанием: <table><tr><td>Вид структуры</td><td>Описание</td></tr><tr><td>1. Линейная</td><td>а. Алгоритм, в котором команды выполняются в определенной последовательности</td></tr><tr><td>2. Разветвляющаяся</td><td>б. Алгоритм, в котором выполнение команд зависит от выполнения некоторого условия</td></tr><tr><td>3. Циклическая</td><td>в. Алгоритм, в котором команды повторяются определенное количество раз или до выполнения некоторого условия</td></tr><tr><td>4. Рекурсивная</td><td>г. Алгоритм, в котором функция вызывает сама себя для решения подзадачи</td></tr></table> |  |  | Вид структуры     | Описание   | 1. Линейная               | а. Алгоритм, в котором команды выполняются в определенной последовательности            | 2. Разветвляющаяся             | б. Алгоритм, в котором выполнение команд зависит от выполнения некоторого условия | 3. Циклическая            | в. Алгоритм, в котором команды повторяются определенное количество раз или до выполнения некоторого условия | 4. Рекурсивная | г. Алгоритм, в котором функция вызывает сама себя для решения подзадачи | ПК-2.У.1 |
| Вид структуры                  | Описание   |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
| 1. Линейная                    | а. Алгоритм, в котором команды выполняются в определенной последовательности   |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
| 2. Разветвляющаяся             | б. Алгоритм, в котором выполнение команд зависит от выполнения некоторого условия  |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
| 3. Циклическая                 | в. Алгоритм, в котором команды повторяются определенное количество раз или до выполнения некоторого условия  |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
| 4. Рекурсивная                 | г. Алгоритм, в котором функция вызывает сама себя для решения подзадачи  |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
| 15.                            | Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов. Кортеж как элемент коллекции Python обладает свойством:<br>1. изменяемости;<br>2. индексированности;<br>3. уникальности.   |  |  | ПК-2.У.1          |            |                           |   |                                |   |                           |   |                |   |          |
| 16.                            | Прочитайте текст и установите последовательность действий. Укажите последовательность появления языков программирования: Варианты ответов:<br>1. fortran;<br>2. algol;<br>3. C;<br>4. pascal.<br>5. C++;<br>6. python;<br>7. HTML;<br>8. Java.   |  |  | ПК-2.В.1          |            |                           |   |                                |   |                           |   |                |   |          |
| 17.                            | Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов. Словарь как элемент коллекции Python обладает свойством :<br>1. изменяемости ключей;<br>2. уникальности значений;<br>3. индексированности;<br>4. уникальности ключей;<br>5. изменяемости значений;<br>6. уникальности элементов;<br>7. изменяемости элементов.   |  |  | ПК-2.В.1          |            |                           |   |                                |   |                           |   |                |   |          |
| 18.                            | Прочитайте текст и установите соответствие. Установите соответствие между методами тестирования программного обеспечения и их описанием: <table><tr><td>Тип тестирования</td><td>Назначение</td></tr><tr><td>1. Модульное тестирование</td><td>а. Тестирование взаимодействия между интегрированными модулями или компонентами системы</td></tr><tr><td>2. Интеграционное тестирование</td><td>б. Тестирование отдельных программных модулей или единиц кода</td></tr><tr><td>3. Системное тестирование</td><td>в. Тестирование всей системы в целом, проверка соответствия требованиям</td></tr><tr><td>4. Приемочное</td><td>г. Тестирование системы конечным</td></tr></table>                          |  |  | Тип тестирования  | Назначение | 1. Модульное тестирование | а. Тестирование взаимодействия между интегрированными модулями или компонентами системы | 2. Интеграционное тестирование | б. Тестирование отдельных программных модулей или единиц кода                     | 3. Системное тестирование | в. Тестирование всей системы в целом, проверка соответствия требованиям                                     | 4. Приемочное  | г. Тестирование системы конечным  | ПК-2.В.1 |
| Тип тестирования               | Назначение   |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
| 1. Модульное тестирование      | а. Тестирование взаимодействия между интегрированными модулями или компонентами системы  |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
| 2. Интеграционное тестирование | б. Тестирование отдельных программных модулей или единиц кода  |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
| 3. Системное тестирование      | в. Тестирование всей системы в целом, проверка соответствия требованиям  |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |
| 4. Приемочное                  | г. Тестирование системы конечным   |  |  |                   |            |                           |   |                                |   |                           |   |                |   |          |

| №<br>п/п                                  | Примерный перечень вопросов для тестов  |  | Код<br>индикатора |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |
|---|---|--|-------------------|--------|---------------------------------|------|----------------------------------|------|----------------------------------|----|--------------------------------|-------|----------------------------------|------|--|-------|---|-----------|----------|
|   | тестирование  | пользователем или заказчиком для приемки |                   |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |
| 19.                                       | <p>Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов      Расставьте этапы тестирования программного обеспечения в правильной последовательности:</p> <p>1. Планирование тестирования<br/>2. Разработка тестовых сценариев<br/>3. Выполнение тестирования<br/>4. Анализ результатов и отчетность<br/>5. Исправление дефектов</p> <p>Прочитайте текст и установите соответствие. Поставьте в соответствие названию операции языка Python соответствующий ей символ</p> <table><tr><th>Название операции</th><th>Символ</th></tr><tr><td>8. Разность множеств символ -&gt;;</td><td>h. -</td></tr><tr><td>9. Симметричная разность символ;</td><td>i. ^</td></tr><tr><td>10. Объединение множеств символ;</td><td>j.  </td></tr><tr><td>11. Равенство множеств символ;</td><td>k. ==</td></tr><tr><td>12. Пересечение множеств символ;</td><td>l. &amp;</td></tr><tr><td>13. Проверка на вхождение в множество;</td><td>m. in</td></tr><tr><td>14. Проверка на не вхождение в множество;</td><td>n. not in</td></tr></table> |  | Название операции | Символ | 8. Разность множеств символ ->; | h. - | 9. Симметричная разность символ; | i. ^ | 10. Объединение множеств символ; | j. | 11. Равенство множеств символ; | k. == | 12. Пересечение множеств символ; | l. & | 13. Проверка на вхождение в множество; | m. in | 14. Проверка на не вхождение в множество; | n. not in | ПК-2.В.1 |
| Название операции                         | Символ  |  |                   |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |
| 8. Разность множеств символ ->;           | h. -  |  |                   |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |
| 9. Симметричная разность символ;          | i. ^  |  |                   |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |
| 10. Объединение множеств символ;          | j.  |  |                   |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |
| 11. Равенство множеств символ;            | k. ==   |  |                   |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |
| 12. Пересечение множеств символ;          | l. &  |  |                   |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |
| 13. Проверка на вхождение в множество;    | m. in   |  |                   |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |
| 14. Проверка на не вхождение в множество; | n. not in   |  |                   |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |
| 20.                                       | <p>Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов      Расставьте этапы тестирования программного обеспечения в правильной последовательности:</p> <p>1. Планирование тестирования<br/>2. Разработка тестовых сценариев<br/>3. Выполнение тестирования<br/>4. Анализ результатов и отчетность<br/>5. Исправление дефектов</p>   |  | ПК-2.В.1          |        |                                 |      |                                  |      |                                  |    |                                |       |                                  |      |  |       |   |           |          |

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.

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.-Петерб. 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 курсового 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.<br>Подпись внесшего изменения | Содержание изменений и дополнений | Дата и № протокола заседания кафедры | Подпись зав. кафедрой |
|---|-----------------------------------|--------------------------------------|-----------------------|
|   |                                   |                                      |                       |
|   |                                   |                                      |                       |
|   |                                   |                                      |                       |
|   |                                   |                                      |                       |
|   |                                   |                                      |                       |