МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики



«28» мая 2021 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.05«Интерпретируемые языки программирования»

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль) Математическое и программное обеспечение компьютерных технологий

Форма обучения очная

Квалификация бакалавр

Краснодар 2021

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат:
4E23D21DF7CC6DF6D9D04D6AEAE74F342DA725D9
Владелец: Хагуров Темыр Айтечевич
Действителен: с 13.05.2021 до 13.08.2022

Рабочая программа дисциплины «Интерпретируемые языки программирования» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.02Фундаментальная информатика и информационные технологии.

Программу составил(и):
Харченко Анна Владимировна, старший преподаватель _____

Рабочая программа дисциплины «Интерпретируемые языки программирования» утверждена на заседании кафедры информационных технологий протокол №15 от «20» мая 2021 г.
Заведующий кафедрой (разработчика)
В. В. Подколзин
_____
подпись

Рабочая программа обсуждена на заседании кафедры вычислительных технологий протокол №_____ от «_____» мая 2021 г.
Заведующий кафедрой (выпускающей)
Ю. М. Вишняков
_____
подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол №1 от «21» мая 2021 г.
Председатель УМК факультета
А. В. Коваленко
_____
подпись

Рецензенты:
Бегларян М. Е., зав. кафедрой социально-гуманитарных и естественнонаучных дисциплин СКФ ФГБОУВО «Российский государственный университет правосудия», канд. физ.-мат. наук, доцент

Рубцов Сергей Евгеньевич, кандидат физико-математических наук, доцент кафедры математического моделирования ФГБГОУ «КубГУ»

1 Цели и задачи изучения дисциплины (модуля)
1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

К группе интерпретируемых языков программирования относят языки, в которых операторы программы друг за другом отдельно транслируются и сразу выполняются (интерпретируются) с помощью специальной программы-интерпретатора (что противопоставляется компилируемым языкам, в которых все операторы программы заранее оттранслированы в объектный код). Такой язык может реализовывать конструкции, позволяющие динамические изменения на этапе времени выполнения (модификация существовавших или создание новых подпрограмм). Эти конструкции затрудняют компиляцию и трансляцию на компилируемый язык.

Язык программирования Python – это универсальный язык программирования с динамической типизацией, который позволяет разрабатывать программы в соответствии с разными парадигмами: процедурным программированием, объектно-ориентированным, параметрическим, функциональным и метапрограммированием. В данном курсе рассматриваются все основные возможности языка Python и их применение при разработке программ. Дается описание библиотек языка Python, необходимых для создания широкого круга программ.

Воспитательная цель: формирование свободного и творческого подхода к программированию на современных языках высокого уровня, интереса к наблюдению за тенденциями и новостями в области средств разработки программного обеспечения.

Отбор материала основывается на необходимости ознакомить студентов со следующей современной научной информацией:

– об особенностях и последних достижениях в области разработки кроссплатформенного ПО;

– о различных подходах к программированию, реализованному в языке Python.

Содержательное наполнение дисциплины обусловлено общими задачами в подготовке бакалавра.

Научной основой для построения программы данной дисциплины является теоретико-прагматический подход в обучении.

1.2 Задачи дисциплины

- изучение языка программирования Python как мультипарадигменного языкового средства, достаточно полно отражающего современные концепции разработки ПО;
- формирование навыков создания приложений на языке Python;
- совершенствование и углубление навыков объектно-ориентированного и функционального программирования;
- знакомство с основными аспектами многопоточного программирования с использованием библиотек Python;
- знакомство с основами создания приложений для взаимодействия с базами данных на основе технологии DB API 2.0;
- изучение принципов написания веб-приложений на Python; библиотек, реализующие шаблоны веб-страниц;
- изучение инструментов обработки данных с помощью языка Python.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Интерпретируемые языки программирования» относится к «Обязательная часть» Блока 1 «Дисциплины (модули)» учебного плана.

Входными знаниями для освоения данной дисциплины являются знания, умения и опыт, накопленный студентами в процессе изучения дисциплин «Основы программирования», «Методы программирования». Обучающийся должен:

- иметь базовые навыки в написании программ на процедурных и объектно-ориентированных языках;
- быть знакомым с наиболее часто встречающимися структурами данных, уметь ими пользоваться и знать внутреннюю организацию.

1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

ПК-5 **Способен применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии**

Знать Типовые алгоритмические и программные решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения

Основные алгоритмические и программные решения современных систем управления базами данных

Уметь Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов с использованием основных алгоритмических и программных решений в области информационно-коммуникационных технологий

Анализировать входные данные при реализации алгоритмических и программных решений в области информационно-коммуникационных технологий

Владеть Проектирование программных интерфейсов при реализации алгоритмических и программных решений в области информационно-коммуникационных технологий

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 3 зач. ед. (108часов), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)					
		5					
Контактная работа, в том числе:	**58,2**	**58,2**					

Аудиторные занятия (всего):		**52**	**52**					
Занятия лекционного типа		**18**	18					
Лабораторные занятия		**34**	34					
Занятия семинарского типа (семинары, практические занятия)								
Иная контактная работа:		**6,2**	**6,2**					
Контроль самостоятельной работы (КСР)		**6**	6					
Промежуточная аттестация (ИКР)		**0,2**	0,2					
Самостоятельная работа, в том числе:		**49,8**	**49,8**					
Проработка учебного (теоретического) материала		**14,8**	**14,8**					
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)		**25**	**25**					
Подготовка к текущему контролю								
Контроль:		**10**	**10**					
Подготовка к экзамену								
Общая трудоемкость	**час.**	**108**	**108**					
	в том числе контактная работа	**58,2**	**58,2**					
	зач. ед	**3**	**3**					

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 5 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
1.	Интерпретируемые языки программирования и их особенности		2		2	4
2.	Функции, модули и пакеты		2		4	4
3.	Классы и объектно-ориентированное программирование		2		4	4
4.	Потоки, процессы, асинхронное программирование		2		4	4
5.	Сетевое программирование		2		2	3,8
6.	Работа с базами данных		2		4	6
7.	Фреймворк Django		4		8	12
8.	Хранение данных и их обработка		2		6	12
ИТОГО по разделам дисциплины			**18**		**34**	**49,8**
Контроль самостоятельной работы (КСР)		6				
Промежуточная аттестация (ИКР)		0,2				
Подготовка к текущему контролю						
Общая трудоемкость по дисциплине		**108**				

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	Интерпретируемые языки программирования и их особенности	История Python. Типы данных. Синтаксис, операторы и управляющие конструкции. IPython и Jupyter-блокноты. PyCharm и Google Collab	К
2.	Функции, модули и пакеты	Области видимости. Аргументы. Рекурсия. Аннотация функций. Лямбда-функции. Декораторы. Генераторы. Создание, использование модулей и пакетов.	К, Т
3.	Классы и объектно-ориентированное программирование	Определение класса. Методы класса. Приватные методы и переменные. Статические методы. Наследование. Множественное наследование	К
4.	Потоки, процессы, асинхронное программирование	Многопоточное программирование. Модуль threading и класс Thread. Потоки Timer. Класс RLock. Синхронизация потоков. Модуль multiprocessing и класс Process.	К, Т
5.	Сетевое программирование	Модуль socket. Клиент и сервер, работающие без установления соединения. Клиент и сервер, работающие с установлением соединения. Фреймворк socketserver.	К
6.	Работа с базами данных	Создание базы данных SQLite. Транзакции. Уровни изоляции.	К, Т
7.	Фреймворк Django	Модели. Миграции. Запись данных.	К, Т
8	Фреймворк Django	Шаблоны и статические данные. Формы	К, Т
9	Хранение данных и их обработка	NymPy: массивы и векторные вычисления. Pandas. Kaggle	К, Т

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.2 Занятия семинарского типа

Не предусмотрены

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Интерпретируемые языки программирования и их особенности	Знакомство с python. IPython и Jupyter-блокноты. PyCharm и Google Collab	РЗ
2.	Интерпретируемые языки программирования и их особенности	Типы данных. Списки. Множества. Словари.	РЗ
3.	Функции, модули и пакеты	Лямбда-функции. Генераторы. Итераторы	РЗ
4.	Классы и объектно-ориентированное программирование	Определение класса. Методы класса. Приватные методы и переменные. Статические методы.	РЗ
5.	Классы и объектно-ориентированное программирование	Наследование. Множественное наследование. Декораторы в классах	РЗ
6.	Потоки, процессы, асинхронное программирование	Многопоточное программирование. Multiprocessing.	РЗ

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
7.	Потоки, процессы, асинхронное программирование	Асинхронное программирование	РЗ
8.	Сетевое программирование	CGI-сервер. TCP-сервер	РЗ
9.	Работа с базами данных	Создание таблиц. Запросы с позиционными и именованными параметрами.	РЗ
10.	Работа с базами данных	Транзакции. Уровни доступа	РЗ
11.	Фреймворк Django	Модели. Миграции	РЗ
12.	Фреймворк Django	URL. Представления.	РЗ
13.	Фреймворк Django	Шаблоны. Формы	РЗ
14.	Фреймворк Django	Администрирование проекта	РЗ
15.	Хранение данных и их обработка	Чтение и запись файлов в различных форматах. Очистки и подготовка данных	РЗ
16.	Хранение данных и их обработка	Агрегирование данных и групповые операции	РЗ
17.	Хранение данных и их обработка	Построение графиков и визуализация.	РЗ

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.4 Примерная тематика курсовых работ (проектов)

Не предусмотрены

……..

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа,

– в форме аудиофайла,

– в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа,

– в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

– Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

– Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

– Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

– Информационно-коммуникационные технологии (ИКТ) - расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

– Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

– Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

– Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

– Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

– Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

– Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

– Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

– работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

– проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

– анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

– развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	
Итого			

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «название дисциплины».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме тестовых заданий, разноуровневых заданий и **промежуточной аттестации** в форме вопросов и заданий к зачету.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Интерпретируемые языки программирования и их особенности	**ПК-5** ИПК-5.1(06.001 D/03.06 Зн.2) ИПК-5.10 (06.015 B/16.5 Зн.6) ИПК-5.13 (06.001 D/03.06 У.2) ИПК-5.15 (06.016 A/30.6 У.1) ИПК-5.18(06.001 D/03.06 Тд.4)	Типовые контрольные задания 1-3 Типовые тестовые задания 1-2	Задание для самостоятельной работы и промежуточной аттестации 1-3
2	Функции, модули и пакеты	**ПК-5** ИПК-5.1(06.001 D/03.06 Зн.2) ИПК-5.10 (06.015 B/16.5 Зн.6) ИПК-5.13 (06.001 D/03.06 У.2) ИПК-5.15 (06.016 A/30.6 У.1) ИПК-5.18(06.001 D/03.06 Тд.4)	Типовые контрольные задания 4-6 Типовые тестовые задания 3-4	Задание для самостоятельной работы и промежуточной аттестации 4-6

3	Классы и объектно-ориентированное программирование	**ПК-5** ИПК-5.1(06.001 D/03.06 Зн.2) ИПК-5.10 (06.015 B/16.5 Зн.6) ИПК-5.13 (06.001 D/03.06 У.2) ИПК-5.15 (06.016 A/30.6 У.1) ИПК-5.18(06.001 D/03.06 Тд.4)	Типовые контрольные задания 7-8 Типовые тестовые задания 5-7	Задание для самостоятельной работы и промежуточной аттестации 7-8
4	Потоки, процессы, асинхронное программирование	**ПК-5** ИПК-5.1(06.001 D/03.06 Зн.2) ИПК-5.10 (06.015 B/16.5 Зн.6) ИПК-5.13 (06.001 D/03.06 У.2) ИПК-5.15 (06.016 A/30.6 У.1) ИПК-5.18(06.001 D/03.06 Тд.4)	Типовые контрольные задания 9-10 Типовые тестовые задания 8-9	Задание для самостоятельной работы и промежуточной аттестации 9-10
5	Сетевое программирование	**ПК-5** ИПК-5.1(06.001 D/03.06 Зн.2) ИПК-5.10 (06.015 B/16.5 Зн.6) ИПК-5.13 (06.001 D/03.06 У.2) ИПК-5.15 (06.016 A/30.6 У.1) ИПК-5.18(06.001 D/03.06 Тд.4)	Типовые контрольные задания 11-12 Типовые тестовые задания 10-11	Задание для самостоятельной работы и промежуточной аттестации 11-12
6	Работа с базами данных	**ПК-5** ИПК-5.1(06.001 D/03.06 Зн.2) ИПК-5.10 (06.015 B/16.5 Зн.6) ИПК-5.13 (06.001 D/03.06 У.2) ИПК-5.15 (06.016 A/30.6 У.1) ИПК-5.18(06.001 D/03.06 Тд.4)	Типовые контрольные задания 13 Типовые тестовые задания 12-14	Задание для самостоятельной работы и промежуточной аттестации 13
7	Фреймворк Django	**ПК-5** ИПК-5.1(06.001 D/03.06 Зн.2) ИПК-5.10 (06.015 B/16.5 Зн.6) ИПК-5.13 (06.001 D/03.06 У.2) ИПК-5.15 (06.016 A/30.6 У.1) ИПК-5.18(06.001 D/03.06 Тд.4)	Типовые контрольные задания 14 Типовые тестовые задания 15-17	Задание для самостоятельной работы и промежуточной аттестации 14
8	Хранение данных и их обработка	**ПК-5** ИПК-5.1(06.001 D/03.06 Зн.2) ИПК-5.10 (06.015 B/16.5 Зн.6) ИПК-5.13 (06.001 D/03.06 У.2) ИПК-5.15 (06.016 A/30.6 У.1) ИПК-5.18(06.001 D/03.06 Тд.4)	Типовые контрольные задания 15-16 Типовые тестовые задания 18-23	Задание для самостоятельной работы и промежуточной аттестации 15-16

Показатели, критерии и шкала оценки сформированных компетенций

Соответствие **пороговому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **удовлетворительно /зачтено**):

ПК-5 **Способен применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии**

Знать Типовые алгоритмические и программные решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения на языке программирования Python

Основные алгоритмические и программные решения современных систем управления базами данных, и средства разработки программ PyCharm, Google Collab

Уметь Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов с использованием основных алгоритмических и программных решений в области информационно-коммуникационных технологий

Анализировать входные данные при реализации алгоритмических и программных решений в области информационно-коммуникационных технологий

Владеть Проектирование программных интерфейсов при реализации алгоритмических и программных решений в области информационно-коммуникационных технологий

Соответствие **базовому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **хорошо /зачтено**):

ПК-5 **Способен применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии**

Знать Типовые алгоритмические и программные решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения на языке программирования Python

Современные основные алгоритмические и программные решения современных систем управления базами данных, и средства разработки программ PyCharm, Google Collab

Уметь Применять современные методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов с использованием основных алгоритмических и программных решений в области информационно-коммуникационных технологий

Анализировать входные данные при реализации алгоритмических и программных решений в области информационно-коммуникационных технологий

Владеть Проектирование современных программных интерфейсов при реализации алгоритмических и программных решений в области информационно-коммуникационных технологий

Соответствие **продвинутому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **отлично /зачтено**):

ПК-5 **Способен применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных,**

операционные системы, электронные библиотеки и пакеты программ, сетевые технологии

Знать Типовые алгоритмические и программные решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения на языке программирования Python

Современные основные алгоритмические и программные решения современных систем управления базами данных, и средства разработки программ PyCharm, Google Collab

Уметь Применять современные методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов с использованием основных алгоритмических и программных решений в области информационно-коммуникационных технологий на высоком профессиональном уровне

Эффективно анализировать входные данные при реализации алгоритмических и программных решений в области информационно-коммуникационных технологий

Владеть Проектирование современных программных интерфейсов при реализации алгоритмических и программных решений в области информационно-коммуникационных технологий на высоком профессиональном уровне

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

Типовые тестовые задания

1) Запись вида a=[i for i in range(11)] дает аналогичный результат с:
1. *a=list(range(11))*
2. *a=tuple(range(11))*
3. *a=[1 for i in range(11)]*
4. *a=[0,1,2,3,4,5,6,7,8,9,10]*

2) Для вывода пар ключ значение из словаря можно использовать следующие выражения:
1. *for key,val in a.items():print(key,val)*
2. *for key,val in a.items():print(key,a[key])*
3. *for key in a.keys():print(key,a[key])*
4. *for key,val in a.keys():print(key,a[key])*
5. *for key,val in a.keys():print(key,val)*

3) Каким образом можно использовать lambda для возведения в степень 2 всех элементов списка a, состоящего из чисел?
1. *z=lambda x: x*x*
a=list(map(z,a))
2. *a=list(map(lambda x:x*x,a))*
3. *a=list(map(lambda x:x**2,a))*
4. *z=lambda x: x**2*
a=list(map(z,a))

5. *a=lambda a:a*a*

4) a=list(map(lambda x:x*x,list(range(6)))) a будет равно:
1. *map object*
2. *lambda object*
3. *function \(\lt\)lambda\(\gt\)*
4. *[1, 4, 9, 16, 25]*
5. *[0, 1, 4, 9, 16, 25]*

5) При переопределении класса
```
class A(object):
    x=5
```
от которого уже создан экземпляр a=A(), на новый
```
class A(object):
    x=7
```
какое свойство x будет у объекта a
1. *5*
2. *7*
3. *При переопределении класса выбросится ошибка*
4. *При обращении к свойству x возникнет ошибка*

6) При переопределении класса
```
class A(object):
    x=7
    def xx(self):
        print(self.x)
```
от которого уже создан экземпляр a=A(), на новый
```
 class A(object):
    x=5
    def xx(self):
        print(self.x)
```
какой результат выдаст использование метода a.xx()
1. *5*
2. *7*
3. *При переопределении класса выбросится ошибка*
4. *При обращении к методу xx возникнет ошибка*

7) Даны два класса
```
class A(object):
    x=5
    def xx(self):
        return self.x
class AB(A):
    y=7
    def xx(self):
        return (self.x,self.y)
```
получен экземпляр класса ab=AB() какой результат выдаст использование метода ab.xx()
1. *(5,7)*
2. *5,7*
3. *5*
4. *7*

8) Какие данные будут выведены на экран при после выполнения следующей программы:
```
import threading
from bs4 import BeautifulSoup as BS
import requests
```

```
def read(n):
        print(threading.currentThread().getName())
        page=requests.get('https://www.djangoproject.com/weblog/?page={}'.format(n))
        res=BS(page.text, features='html.parser').h1
        print(res)

if __name__=='__main__':
        thr_names=['hello','python','django','flask','thread']
        for i in range(5):
                my_thread=threading.Thread(target=read, name=thr_names[i], args=(i,))
                my_thread.start()


        for i in range(5):
                my_thread.join()
```

1. *Сначала будет выведено имя потока, затем информация о заголовке 1 уровня и так 5 раз*
2. *Сначала будут выведены имена потоков 5 раз, потом информация о заголовках 1 уровня на страницах 5 раз верно*
3. *Сначала будет выведено имя потока, затем информация о заголовке 1 уровня и так 4 раза*
4. *Сначала будут выведены имена потоков 4 раза, потом информация о заголовках 1 уровня на страницах 4 раза*
5. *Ошибка*

9) Какие данные будут выведены на экран при после выполнения следующей программы:

```
from multiprocessing import Process
from bs4 import BeautifulSoup as BS
import requests
import os

def read(n):
        print(os.getpid())
        page=requests.get('https://www.djangoproject.com/weblog/?page={}'.format(n))
        res=BS(page.text, features='html.parser').h1
        print(res)

if __name__=='__main__':
        pages=[5,10,15,20,25,30]
        procs=[]
        for i,n in enumerate(pages):
                proc=Process(target=read,args=(n,))
                procs.append(proc)
                proc.start()

        for proc in procs:
                proc.join()
```

1. *Будет выведены ид 6 процессов, затем 6 заголовков 1 уровня верно*
2. *Ошибка*
3. *Будет выведен ид процесса и заголовок 1 уровня и так 6 раз*
4. *Будет выведен ид процесса и содержимое внутри заголовка 1 уровня и так 6 раз*
5. *Будет выведены ид 6 процессов, затем содержимое 6 заголовков 1 уровня*

10) При создании сервера, используя модуль http.server и описании класса, унаследованного от BaseHTTPRequestHandler, send_response(200) используется для:

1. *Ответа об ошибке*
2. *Ответа об ошибке запроса*
3. *Ответа ОК*
4. *Ответа о отсутствии содержимого*
5. *Ответа о запрете доступа*

11) При создании сервера, используется код:

```
from http.server import HTTPServer
from http.server import BaseHTTPRequestHandler
class ServerWorking(BaseHTTPRequestHandler):
…
server=HTTPServer(server_address,ServerWorking)
```

server_address может представлять из себя:

1. *server_address=('', 9000)*
2. *server_address=('localhost', 9000)*
3. *server_address = ('127.0.0.1', 9000)*
4. *server_address = (9000,'127.0.0.1')*
5. *server_address=(9000,'localhost')*

12) Для создания файла БД достаточно выполнить:

1. *import sqlite3*
2. *aiter=iter((1,2,3,4))*
3. *import sqlite3*

con=sqlite3.connect('путь к файлу БД')

4. *con=sqlite3.connect('путь к файлу БД')*
5. *import sqlite3*

con=sqlite3.connect('путь к файлу БД')
con.close()

13) Для создания БД из нескольких таблиц после создания соединения и курсора, удобнее пользоваться:

1. *con.execute(скрипт)*
2. *con.executemany(скрипты)*
3. *описанием скрипта и выполнением con.executescript(sql)*
4. *cur.execute(скрипml)*
5. *описанием скрипта и выполнением cur.executescript(sql)*

14) Для заполнения полей таблицы можно использовать для записи данные следующего вида:

1. *var1=("Роман",)*
2. *var2=(2,"Рассказ")*
3. *var3={"id":3, "name":"Стихотворение"}*
4. *var4=("Поэма")*
5. *var5=('Роман в стихах')*

15) Для передачи параметров на страницу index.html из views.py можно использовать:

1. *return render(request,"index.html",{})*
2. *return render(request,{})*
3. *return HttpResponse("index.html",{})*
4. *return HttpResponseNotFound("index.html",{})*
5. *return render("index.html",{})*

16) Для создания новых миграций при изменении в моделях данных используют следующую команду в консоли:

1. *python manage.py makemigrations*
2. *python manage.py syncdb*
3. *python manage.py migrate*
4. *python manage.py sqlmigrate*
5. *python manage.py dbshell*

17) Для отображения модели приложения в части администратора можно прописать:

1. *Импортировать модель в admin.py и ее зарегистрировать*
2. *Добавление названия приложения в INSTALLED_APPS в apps.py*
3. *Добавление названия приложения в settings.py*
4. *Добавление названия приложения в apps.py*

18) Метеоролог хочет построить модель, прогнозирующую температуру воздуха на завтра. К какому типу относится данная задача?

1. кластеризация
2. классификация
3. ранжирование
4. **регрессия**

19) Доктор хочет построить модель, прогнозирующую возможность развития осложнений у пациента после операции - все ли будет благополучно в течение нескольких следующих месяцев или нет. К какому типу относится данная задача?

1. ранжирование
2. регрессия
3. кластеризация
4. **классификация**

20) Стоимость акции некоторой компании на бирже является…

1. числовым признаком
2. порядковым признаком
3. категориальным признаком
4. бинарным признаком

21) Наличие балкона в квартире является…

1. числовым признаком
2. категориальным признаком
3. бинарным признаком
4. порядковым признаком

22) Возраст человека…

1. может считаться как числовым, так и порядковым признаком
2. может считаться только числовым признаком
3. может считаться только категориальным признаком
4. может считаться только порядковым признаком

23) Цвет глаза человека является…

1. числовым признаком
2. категориальным признаком
3. бинарным признаком
4. порядковым признаком

Типовые контрольные задания

1. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите количество строгих локальных максимумов в этой последовательности. (Элемент последовательности называется локальным максимумом, если он строго больше предыдущего и последующего элемента последовательности. Первый и последний элемент последовательности не являются локальными максимумами.)

2. Определите наименьшее расстояние между двумя локальными максимумами последовательности натуральных чисел, завершающейся числом 0. Если в последовательности нет двух локальных максимумов, выведите число 0.

3. Даны два списка чисел, которые могут содержать до 10000 чисел каждый. Выведите все числа, которые входят как в первый, так и во второй список в порядке возрастания.

4. Дан файл, содержащий текст на русском языке и некоторые два слова. Определить, сколько раз они встречаются в тексте и сколько из них — непосредственно друг за другом.

5. Даны файл, содержащий текст на русском языке, и некоторые буквы. Найти слово, содержащее наибольшее количество указанных букв.

6. Даны файл, содержащий текст на русском языке, и некоторая буква. Подсчитать, сколько слов начинается с указанной буквы.

7. Создать класс с полями, указанными в индивидуальном задании. Реализовать в классе методы: конструктор по умолчанию; функции обработки данных (1 и 2) функцию формирования строки информации об объекте. Создать класс-потомок с полями, указанными в индивидуальном задании. Реализовать в классе-потомке методы: конструктор; функцию обработки данных; функцию формирования строки информации об объекте.

Класс 1: Дата (три числа): день, месяц, год

Определить, является ли год високосным (кратным 4)

Увеличить дату на 5 дней

Класс 2: Человек: ФИО, телефон, дата рождения,

Количество дней до дня очередного рождения

8. Создать класс с полями, указанными в индивидуальном задании. Реализовать в классе методы: конструктор по умолчанию; функции обработки данных (1 и 2) функцию формирования строки информации об объекте. Создать класс-потомок с полями, указанными в индивидуальном задании. Реализовать в классе-потомке методы: конструктор; функцию обработки данных; функцию формирования строки информации об объекте.

Класс 1: Дата (три числа): день, месяц, год

Увеличить год на 1

Уменьшить дату на 2 дня

Класс 2: Работник: ФИО, дата поступления на предприятие

Количество лет работы на предприятии

9. Реализуйте подсчет определенного интеграла на интервале [a,b] для функции *f(x)* методом левых и правых прямоугольников.

10. Реализуйте подсчет определенного интеграла на интервале [a,b] для функции *f(x)* методом центральных прямоугольников.

11. Создать функцию, которая проверяет пингуются ли заданные в файле IP-адреса. Проверка IP-адресов должна выполняться параллельно в разных потоках.

12. Создать функцию, которая отправляет одну и ту же команду show на разные устройства в параллельных потоках, а затем записывает вывод команд в файл. Вывод с устройств в файле может быть в любом порядке.

13. Создать БД по заданной тематике. БД должна содержать не менее трех таблиц. Должна присутствовать возможность экспорта/импорта из/в xml-файл, а также не менее трех статистических запросов.

а) «Направления музыки».

б) «Направления танца».

в) «Фото-видео техника».

14. Создать web-приложение по заданной тематике с помощью фрейморка Django. Приложение должно предоставлять разный доступ к данным в зависимости от роли пользователя (только просмотр, просмотр/редактирование данных)

а) «Направления музыки».
б) «Направления танца».
в) «Фото-видео техника».

15. Используя датасет (load_boston):

а) получить график того, как зависит цена от криминогенной обстановки,
б) построить модель,
в) обучить,
г) получить ошибку модели,
д) осуществить поиск по сетке гиперараметров с помощью GridSearchCV с целью подбора параметров и уменьшения ошибки.

16. Используя датасет (load_boston):

а) получить график того, как зависит цена от удаленности к реке,
б) построить модель,
в) обучить,
г) получить ошибку модели,
д) осуществить поиск по сетке гиперараметров с помощью GridSearchCV с целью подбора параметров и уменьшения ошибки.

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен/зачет)

1. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите, какое наибольшее число подряд идущих элементов этой последовательности равны друг другу.
2. Дана последовательность натуральных чисел, завершающаяся число 0. Определите наибольшую длину монотонного фрагмента последовательности (то есть такого фрагмента, где все элементы либо больше предыдущего, либо меньше).
3. Дан список чисел, который могут содержать до 100000 чисел каждый. Определите, сколько в нем встречается различных чисел.
4. Во входном файле (вы можете читать данные из файла input.txt) записан текст. Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки. Определите, сколько различных слов содержится в этом тексте.
5. Дан файл, содержащий текст на русском языке. Найти слово, встречающееся в каждом предложении, или сообщить, что такого слова нет.
6. Дан файл, содержащий зашифрованный русский текст. Каждая буква заменяется на следующую за ней (буква я заменяется на а). Получить в новом файле расшифровку данного текста.
7. Создать класс с полями, указанными в индивидуальном задании. Реализовать в классе методы: конструктор по умолчанию; функции обработки данных (1 и 2) функцию формирования строки информации об объекте. Создать класс-потомок с полями, указанными в индивидуальном задании. Реализовать в классе-потомке методы: конструктор; функцию обработки данных; функцию формирования строки информации об объекте.

Класс 1: Время (три числа): часы, минуты, секунды
Вычислить количество секунд в указанном времени
Увеличить время на 5секунд

Класс 2: Данные по движению поезда: номер поезда, направление, время отправления

Количество минут до отправления поезда с указанным номером и введенное время

8. Создать класс с полями, указанными в индивидуальном задании. Реализовать в классе методы: конструктор по умолчанию; функции обработки данных (1 и 2) функцию формирования строки информации об объекте. Создать класс-потомок с полями, указанными в индивидуальном задании. Реализовать в классе-потомке методы: конструктор; функцию обработки данных; функцию формирования строки информации об объекте.

Класс 1: Время (три числа): часы, минуты, секунды

Вычислить количество полных минут в указанном времени

Уменьшить время на 10 минут

Класс 2: Абонент мобильной связи: фамилия, оператор, текущее время

Определить, является ли время льготным для абонента (время от 0 до 8 часов)

9. Реализуйте подсчет определенного интеграла на интервале [a,b] для функции *f(x)* методом трапеций.

10. Реализуйте подсчет определенного интеграла на интервале [a,b] для функции *f(x)* методом Симпсона.

11. Создать функцию, которая отправляет разные команды show на разные устройства в параллельных потоках, а затем записывает вывод команд в файл. Вывод с устройств в файле может быть в любом порядке.

12. Создать функцию, которая отправляет список указанных команды show на разные устройства в параллельных потоках, а затем записывает вывод команд в файл. Вывод с устройств в файле может быть в любом порядке.

13. Создать БД по заданной теме. БД должна содержать не менее трех таблиц. Должна присутствовать возможность экспорта/импорта из/в xml-файл, а также не менее трех статистических запросов.

а) «Наземный транспорт».

б) «Вычислительная техника».

14. Создать web-приложение по заданной тематике с помощью фрейморка Django. Приложение должно предоставлять разный доступ к данным в зависимости от роли пользователя (только просмотр, просмотр/редактирование данных)

а) «Наземный транспорт».

б) «Вычислительная техника».

15. На основе данных с сайта Московской фьючерсной биржи по контракту SPFB.RTS-12.18 (RIZ8) с 1 сентября по 31 декабря 2020 года с минутным интервалом.

а) Для каждого минутного интервала сосчитайте среднюю цену (PRICE), усреднив цену открытия, закрытия, минимума и максима.

б) Для каждого минутного интервала вычислите оборот по совершенным сделкам, умножив среднюю цену (PRICE) на объем (VOL).

в) Вычислите суммарный оборот по совершенным сделкам за дату: 05.10.2020

16. Скачайте данные с сайта Московской фьючерсной биржи о том, как продавались акции компании BETA на бирже в период с 1 сентября по 31 декабря 2020 года с минутным интервалом.

а) Для каждого минутного интервала сосчитайте среднюю цену PRICE, усреднив цену открытия <OPEN>, закрытия <CLOSE>, минимума <LOW> и максима <HIGH>.

б) Определите среднюю цену за 10.11.2020 в 18:09:00

в) Для каждого минутного интервала вычислите оборот TOTAL по совершенным сделкам, умножив среднюю цену PRICE на объем <VOL>.

г) Определите оборот за 22.12.2020 в 15:07:00

д) Вычислите суммарный оборот по совершенным сделкам за 06.10.2020.
е) Ответ введите с точностью до целых.
ж) Вычислите количество минутных интервалов за 24.11.2020, когда цена открытия была строго больше цены закрытия.
з) Вычислите суммарный оборот по всем сделкам за все четверги ноября.

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

Методические рекомендации, определяющие процедуры оценивания тестов:

Тест проводится онлайн в системе Moodle или Google Docs и ограничен по времени. На сдачу теста дается две попытки. Тест считается успешно пройденным если студент правильно ответил на 70% вопросов.

Методические рекомендации, определяющие процедуры оценивания выполнения контрольных заданий:

Задание считается выполненным при выполнении следующих условий:
– предоставлен исходный код на Python в среде PyCharm, Google Collab
– продемонстрирована работоспособность программы
– студент понимает исходный код и отвечает на вопросы по его организации.

Методические рекомендации к сдаче зачета

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Текущий контроль успеваемости студентов проводится в целях совершенствования и непрерывного контроля качества образовательного процесса, проверки усвоения учебного материала, активизации самостоятельной работы студентов, стимулирования их учебной работы, обеспечения эффективности образовательного процесса, предупреждения рисков отчисления студентов.

Текущий контроль знаний студентов осуществляется постоянно в течение всего семестра.

Виды текущего контроля: устный (письменный) опрос на занятиях; проверка выполнения домашних заданий; проведение контрольных работ; оценка активности студента на занятии.

Студенты очной формы обучения обязаны сдать зачет до начала экзаменационной сессии.

По результатам сдачи зачета выставляется оценка «зачтено»/ «не зачтено».

Выставление зачетов для студентов очной формы обучения проводятся в период до экзаменационной сессии.

При отсутствии зачетной книжки у студента экзаменатор не имеет права принимать у него зачет/экзамен. Такой студент считается не явившимся на зачет/экзамен. В

исключительных случаях, на основании распоряжения декана преподаватель может допустить студента к зачету/экзамену при наличии документа, удостоверяющего личность.

В целях объективного оценивания знаний во время проведения зачетов и экзаменов не допускается наличие у студентов посторонних предметов и технических устройств.

Студенту, использующему в ходе зачета неразрешенные источники и средства получения информации, выставляется неудовлетворительная оценка, и он удаляется из аудитории.

Во время зачета студенты могут пользоваться утвержденной рабочей программой учебной дисциплины, которая должна быть в наличии на экзамене, а также с разрешения экзаменатора справочной литературой и другими пособиями.

Студенты, нарушающие правила поведения при проведении зачетов и экзаменов, могут быть незамедлительно удалены из аудитории, к ним могут быть применены меры дисциплинарного воздействия.

На зачете/экзамене могут присутствовать ректор, проректор по учебной работе, декан факультета, заведующий кафедрой, которая обеспечивает учебный процесс по данной дисциплине. Присутствие на экзаменах и зачетах посторонних лиц без разрешения ректора или проректора по учебной работе не допускается.

После прослушивания лекции рекомендуется выполнить упражнения, приводимые в лекции для самостоятельной работы, а также выполнить на компьютере с использованием среды Python задачи, приводимые в лекции в качестве примеров.

При самостоятельной работе студентов необходимо изучить литературу, приведенную в перечнях, для осмысления вводимых понятий, анализа предложенных подходов и методов разработки параллельных программ. Разрабатывая решение новой задачи, студент должен уметь выбрать методы решения задачи с учетом целевой аппаратной платформы, проводить отладку и профилирование программы на языке Python.

В качестве систем программирования для решения задач и изучения методов и алгоритмов, приведенных в лекциях, рекомендуется использовать на практических занятиях и при самостоятельной работе стандартную реализацию языка Python в связке со средой разработки PyCharm, Google Collab.

Оценка	
Не зачтено	**Зачтено**
• если студент правильно решил менее 70 % задач и/или не имеет представление как решать остальные задачи	• если студент правильно решил 70 % задач, имеет представление как решать остальные задачи

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Перечень учебной литературы, информационных ресурсов и технологий

5.1 Основная литература:

1. Чернышев, С. А. Основы программирования на python : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2021. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/477353 (дата обращения: 02.07.2021).

2. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2021. — 210 с. — (Профессиональное образование). — ISBN 978-5-534-12829-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/487079 (дата обращения: 02.07.2021).

3. Шелудько, В. М. Основы программирования на языке высокого уровня Python : учебное пособие : [16+] / В. М. Шелудько. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 147 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=500056 (дата обращения: 02.07.2021). – Библиогр. в кн. – ISBN 978-5-9275-2649-9. – Текст : электронный.

Для освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья имеются издания в электронном виде в электронно-библиотечных системах *«Библиоклуб» и «Юрайт».*

5.2 Дополнительная литература:

1. Сузи, Р. А. Язык программирования Python : учебное пособие : [16+] / Р. А. Сузи. – 2-е изд., испр. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ) : Бином. Лаборатория знаний, 2007. – 327 с. – (Основы информационных технологий). – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=233288 (дата обращения: 02.07.2021). – ISBN 978-5-9556-0109-0. – Текст : электронный.

2. Шелудько, В. М. Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули : [16+] / В. М. Шелудько. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 108 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=500060 (дата обращения: 02.07.2021). – Библиогр. в кн. – ISBN 978-5-9275-2648-2. – Текст : электронный.

3. Программные системы статистического анализа: обнаружение закономерностей в данных с использованием системы R и языка Python : [16+] / В. М. Волкова, М. А. Семенова, Е. С. Четвертакова, С. С. Вожов. – Новосибирск : Новосибирский государственный технический университет, 2017. – 74 с. : ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=576496 (дата обращения: 02.07.2021). – Библиогр.: с. 48. – ISBN 978-5-7782-3183-2. – Текст : электронный.

5.3. Периодические издания:

1. Базы данных компании «Ист Вью» http://dlib.eastview.com
2. Электронная библиотека GREBENNIKON.RU https://grebennikon.ru/

5.4. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» https://urait.ru/
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» www.biblioclub.ru
3. ЭБС «BOOK.ru» https://www.book.ru
4. ЭБС «ZNANIUM.COM» www.znanium.com
5. ЭБС «ЛАНЬ» https://e.lanbook.com

5.5.Профессиональные базы данных:

1. Web of Science (WoS) http://webofscience.com/
2. Scopus http://www.scopus.com/
3. ScienceDirect www.sciencedirect.com
4. Журналы издательства Wiley https://onlinelibrary.wiley.com/
5. Научная электронная библиотека (НЭБ) http://www.elibrary.ru/
6. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН http://archive.neicon.ru
7. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) https://rusneb.ru/
8. Президентская библиотека им. Б.Н. Ельцина https://www.prlib.ru/
9. Электронная коллекция Оксфордского Российского Фонда https://ebookcentral.proquest.com/lib/kubanstate/home.action
10. Springer Journals https://link.springer.com/
11. Nature Journals https://www.nature.com/siteindex/index.html
12. Springer Nature Protocols and Methods https://experiments.springernature.com/sources/springer-protocols
13. Springer Materials http://materials.springer.com/
14. zbMath https://zbmath.org/
15. Nano Database https://nano.nature.com/
16. Springer eBooks: https://link.springer.com/
17. "Лекториум ТВ" http://www.lektorium.tv/
18. Университетская информационная система РОССИЯ http://uisrussia.msu.ru

5.6. Информационные справочные системы:

1. Консультант Плюс - справочная правовая система (доступ по локальной сети с компьютеров библиотеки)

5.7. Ресурсы свободного доступа:

1. Американская патентная база данных http://www.uspto.gov/patft/
2. Полные тексты канадских диссертаций http://www.nlc-bnc.ca/thesescanada/
3. КиберЛенинка (http://cyberleninka.ru/);

4. Министерство науки и высшего образования Российской Федерации https://www.minobrnauki.gov.ru/;
5. Федеральный портал "Российское образование" http://www.edu.ru/;
6. Информационная система "Единое окно доступа к образовательным ресурсам" http://window.edu.ru/;
7. Единая коллекция цифровых образовательных ресурсов http://school-collection.edu.ru/ .
8. Федеральный центр информационно-образовательных ресурсов (http://fcior.edu.ru/);
9. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" https://pushkininstitute.ru/;
10. Справочно-информационный портал "Русский язык" http://gramota.ru/;
11. Служба тематических толковых словарей http://www.glossary.ru/;
12. Словари и энциклопедии http://dic.academic.ru/;
13. Образовательный портал "Учеба" http://www.ucheba.com/;
14. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety
15. Система организации конкурсов по исследованию данных, а также социальная сеть специалистов по обработке данных и машинному обучению https://www.kaggle.com
16. Портал открытых данных Российской Федерации https://data.gov.ru/
17. Финам.ру — финансовый портал: новости фондового рынка ценных бумаг и экономики, прогнозы и анализ. Инвестиционная компания Финам https://www.finam.ru

5.8.Собственные электронные образовательные и информационные ресурсы КубГУ:

1. Среда модульного динамического обучения http://moodle.kubsu.ru
2. База учебных планов, учебно-методических комплексов, публикаций и конференций http://mschool.kubsu.ru/
3. Библиотека информационных ресурсов кафедры информационных образовательных технологий http://mschool.kubsu.ru;
4. Электронный архив документов КубГУ http://docspace.kubsu.ru/
5. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" http://icdau.kubsu.ru/

6. Методические указания для обучающихся по освоению дисциплины (модуля)

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

7. Материально-техническое обеспечение по дисциплине (модулю)

По всем видам учебной деятельности в рамках дисциплины используются аудитории, кабинеты и лаборатории, оснащенные необходимым специализированным и лабораторным оборудованием.

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения

2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Практические занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
4.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
6.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.