

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Клочков Юрий Сергеевич
Должность: Ректор
Дата подписания: 06.08.2026 12:02:28
Уникальный программный ключ:
3beb265d5d589e7ff4c954946f3ad99aef70ac12

1

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ”
ИНСТИТУТ ДОПОЛНИТЕЛЬНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

УТВЕРЖДАЮ

Директор _____ А.В. Медведёв

«» 2025г.

Дополнительная профессиональная программа
(программа профессиональной переподготовки)
Программирование на Python в области добывающей промышленности и
природопользования

2025г.

Лист согласования

И.А. Анохин
ФИО

Заместитель генерального директора по
управлению персоналом и общим вопросам
ООО «Газпром межрегионгаз Север»

должность, наименование компании
(организации)



дата, подпись, печать

Аннотация

Дополнительная профессиональная программа профессиональной переподготовки ИТ-профиля (далее – Программа) предназначена для обучающихся по специальностям и направлениям подготовки не отнесенным к ИТ - сфере по следующим направлениям подготовки:

1) бакалавриат:

05.03.01 Геология

20.03.01 Техносферная безопасность

21.03.01 Нефтегазовое дело

21.03.02 Землеустройство и кадастры

2) специалитет:

21.05.01 Прикладная геодезия

21.05.02 Прикладная геология

21.05.03 Технология геологической разведки

21.05.04 Горное дело

21.05.06 Нефтегазовые техника и технологии

Программа разработана в целях обеспечения условий для формирования цифровых компетенций и навыков использования цифровых технологий у обучающихся, в том числе студентов не ИТ-специальностей в рамках проекта «Цифровые кафедры».

Целью профессиональной переподготовки является получение профессиональных цифровых компетенций, необходимых для выполнения нового вида профессиональной деятельности в области разработки компьютерного программного обеспечения.

В рамках программы обучающиеся осваивают новый вид профессиональной деятельности в области разработки компьютерного программного обеспечения и приобретение новой квалификации «Программист».

Программа не предусматривает возможность выбора обучающимися модулей для освоения.

Нормативный срок освоения программы 256 часов.

Форма подготовки - очно-заочная с применением дистанционных образовательных технологий.

Авторы и преподаватели:

№ п/п	ФИО, должность	Модули (темы, лекции)	Часов, всего
1	Спирин Игорь Сергеевич, к.п.н., доцент кафедры математики и прикладных информационных технологий, ТИУ	Обзор существующих парадигм и языков программирования	6
		Основные элементы программирования	6
		Управляющие конструкции	6

		Разработка модуля	10
		Создание пакета модулей	8
		Стандартные и распределяемые пакеты и модули	8
		Практика/ стажировка программы профессиональной переподготовки	8
2	Базанов Артем Владимирович, к.т.н., доцент кафедры сервиса автомобилей и технологических машин, научно-педагогический работник, кафедры интеллектуальных систем и технологий, ТИУ	Список. Основные операции со списком	6
		Кортеж. Основные операции с кортежем	6
		Множества. Основные операции с множеством	4
		Словарь. Основные операции со словарем	8
		Практика/ стажировка программы профессиональной переподготовки	10
3	Сидоров Андрей Юрьевич, руководитель НП "АРИТ ТО"	Множества. Основные операции с множеством	2
4	Козин Евгений Сергеевич, к.т.н., доцент кафедры сервиса автомобилей и технологических машин, научно-педагогический работник, ТИУ	Список. Объявление и вызов функции	6
		Основы функционального программирования	6
		Обработка коллекций	8
		Структуры данных и алгоритмы	6
		Абстракция, классы и инкапсуляция	6
		Наследование	4
		Полиморфизм подтипов	6
		Обработка исключений	8
5	Басинский Константин Юрьевич, доцент кафедры математики и прикладных информационных технологий, научно-педагогический работник, ТИУ	Наследование	2
6	Ахмадулин Руслан Камильевич, к.т.н., Тюменский нефтяной научный центр	Практика/ стажировка программы профессиональной переподготовки	2

I. Общие положения

1.1 Нормативная правовая основа Программы

– Федеральный закон от 29 декабря 2012 года №273-ФЗ «Об образовании в Российской Федерации»;

– постановление Правительства Российской Федерации от 11 октября 2023 г. № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;

– приказ Минобрнауки России от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по

дополнительным профессиональным программам» (с изменениями, внесенными приказом Минобрнауки России от 15 ноября 2013 г. № 1244 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденный приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499»);

- приказ Минобрнауки России от 19 октября 2020 г. № 1316 «Об утверждении порядка разработки дополнительных профессиональных программ, содержащих сведения, составляющие государственную тайну, и дополнительных профессиональных программ в области информационной безопасности;

- федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 11.03.02 Инфокоммуникационные технологии и системы связи;

- профессиональный стандарт 06.001 "Программист" Утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20.07.2022 № 424н.

1.2. Термины и определения, используемые в Программе

Итоговая аттестация (аттестация) – оценка степени и уровня освоения обучающимися ДПП ПП или ИТ-модуля в формате демонстрационного экзамена, предусматривающая выполнение обучающимся профессиональных задач и оценку результатов и (или) процесса выполнения – проверку сформированности цифровых компетенций в ходе обучения по ДПП ПП.

Демонстрационный экзамен – аттестационное испытание, предусматривающее выполнение профессиональных задач и оценку результатов и (или) процесса выполнения профессиональных задач для подтверждения применения обучающимися цифровых компетенций на практике.

Дополнительная профессиональная программа профессиональной переподготовки (Программа) – комплекс основных характеристик образования (объем, содержание, планируемые результаты) и организационно-педагогических условий, который представлен в виде учебного плана, календарного учебного графика, рабочих программ учебных курсов, дисциплин (модулей), оценочных и методических материалов, а также программ учебной и производственной практик, стажировок и форм аттестации, иных компонентов и обеспечивает приобретение дополнительной квалификации. Программа может разрабатываться с учетом положений профессиональных стандартов, федеральных государственных образовательных стандартов, требований рынка труда (индустрии).

Знание (З) – информация о свойствах объектов, закономерностях процессов и явлений, правилах использования этой информации для принятия решений,

присвоенная обучающимся на одном из уровней, позволяющих выполнять над ней мыслительные операции.

Матрица цифровых компетенций – матрица компетенций, актуальных для цифровой экономики, с приоритетом компетенций в ИТ-сфере, разработанная Университетом Иннополис при участии ИТ-компаний и университетов-участников программы «Приоритет-2030», представляющая собой перечень компетенций, структурированный по сферам применения, типу компетенций, уровням их сформированности и характеристикам.

Междисциплинарный курс (МДК) – структурный элемент Программы или программы профессионального модуля, предназначенный для формирования знаний и умений, объединенных по прагматическим основаниям с нарушением академических границ отраслей знаний.

Опыт практической деятельности (ОПД) – образовательный результат, включающий выполнение обучающимся деятельности, завершающейся получением результата / продукта (элемента продукта), значимого при выполнении трудовой функции, в условиях реального производства или в модельной ситуации.

Оценочные средства (ОС) – дидактические средства для оценки качества подготовленности обучающихся.

Практика (практическая подготовка) – форма организации образовательной деятельности при освоении образовательной программы в условиях выполнения обучающимися определенных видов работ, связанных с будущей профессиональной деятельностью и направленных на формирование, закрепление, развитие практических навыков и компетенции по профилю соответствующей образовательной программы.

Профессиональный модуль (ПМ) – структурный элемент Программы, предназначенный для формирования определенных компетенций.

Рабочая программа – нормативный документ в составе Программы, регламентирующий взаимодействие преподавателя и обучающихся в ходе учебного процесса при реализации структурных элементов Программы (модуль, дисциплина, курс).

Стажировка – формирование и закрепление полученных в результате теоретической подготовки профессиональных знаний и умений в рамках выполнения практических заданий (функций) на базе профильной компании (организации). Допускается заключение срочных трудовых договоров, предусматривающих прохождение обучающимися оплачиваемой стажировки. Время прохождения стажировки целесообразно учитывать в качестве учебной или производственной практики.

Умение (У) – освоенный субъектом способ выполнения действия, обеспечиваемый совокупностью приобретенных знаний и навыков; операция (действие), выполняемая определенным способом и с определенным качеством.

Учебная дисциплина (УД) – структурный элемент Программы, предназначенный для формирования знаний и умений в соответствующей сфере профессиональной деятельности.

Фонды оценочных средств (ФОС) – совокупность оценочных средств, используемых на различных этапах педагогической диагностики.

Целевой уровень сформированности компетенций – установленный Программой уровень сформированности компетенций в соответствии с Матрицей компетенций, актуальных для цифровой экономики, с приоритетом компетенций в ИТ-сфере.

Цифровая компетенция (компетенция) – образовательный результат, формируемый при освоении Программы, необходимый для приобретения дополнительной ИТ-квалификации, необходимой для нового вида профессиональной деятельности по внедрению и (или) развитию, и (или) разработке цифровых технологий, в том числе алгоритмов и программ, пригодных для практического применения, в одной из приоритетных отраслей экономики.

Электронное обучение – организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников.

1.3. Требования к поступающим

Студенты принимаются на обучение по ДПП ПП или ИТ-модулям на основании организуемого отбора, направленного на выявление и последующее обучение на «цифровой кафедре» студентов, заинтересованных в получении дополнительной ИТ-квалификации и обладающих базовыми знаниями о цифровых технологиях, применяемых в той или иной отрасли.

С этой целью разрабатывается и размещается в открытом доступе на бесплатной основе базовые курсы продолжительностью 8 часов. Тематики базовых курсов соответствуют профилю ДПП ПП или ИТ-модулям, реализуемым на «цифровой кафедре». Содержание курсов в доступной форме раскрывает основные направления применения и развития цифровых технологий в отраслях. Освоение курса завершается тестированием, результаты которого являются основанием для принятия решения о приеме обучающихся на «цифровую кафедру».

К обучению по Программе допускаются обучающиеся по очной или по очно-заочной форме за счет бюджетных средств или по договорам об оказании платных

образовательных услуг не ИТ-профиля, освоившие основную профессиональную образовательную программу (далее – ОПОП ВО), не отнесенным к ИТ-сфере:

- программы бакалавриата и/или специалитета в объеме не менее первого курса и не менее 60 зачетных единиц (бакалавры и/или специалисты 2-го курса очной формы обучения);

- по заочной форме, освоившие те же ОПОП ВО в объеме не менее первых двух курсов и не менее 120 зачетных единиц (бакалавры и/или специалисты 3 курса заочной формы обучения).

1.4. Квалификационная характеристика выпускника

Выпускникам Программы присваивается дополнительная ИТ-квалификация в области формирования цифровых компетенций и навыков их использования для выполнения нового вида профессиональной деятельности в области добывающей промышленности и природопользования.

Выпускник Программы будет готов к выполнению трудовой деятельности в области базового администрирования информационно-коммуникационных систем и пользователя офисных приложений (ОКВЭД 62.01 разработка компьютерного программного обеспечения).

Квалификационный уровень по национальной рамке квалификаций: 6

II. Планируемые результаты обучения и структура Программы

Получение дополнительной ИТ-квалификации «Программист» обеспечивается формированием приведенных в таблице цифровых компетенций:

Наименование сферы	ID и наименование компетенции	Инструменты профессиональной деятельности	Целевой уровень формирования компетенций в Программе			
			Минимальный (исходный)	Базовый	Продвинутый	Экспертный
Средства программной разработки	ID-28 Применяет языки программирования для решения профессиональных задач	Python		Применяет языки программирования для решения профессиональных задач под контролем более опытных специалистов	Самостоятельно применяет языки программирования и настраиваемые программные инструменты для автоматизации процессов в профессиональной деятельности	
СPython версии 3.x	ID-20 Пишет и структурирует код на Python, используя устоявшиеся практики			Использует Python-практики при написании кода под контролем автоматизированных средств проверки и/или под контролем более опытных коллег	Использует правильные практики написания кода без внешнего контроля. Может обосновать правильность того или иного подхода	
СPython версии 3.x	ID-57 Использует стандартную библиотеку языка Python			Использует базовую часть стандартной библиотеки языка Python без понимания альтернатив. Под внешним контролем правильно выбирает необходимые подходы, алгоритмы и структуры данных	Использует большинство частей стандартной библиотеки языка Python, понимает разницу между алгоритмами и структурами данных, эпизодически прибегая к консультации по возникающим проблемам и ограничениям	
Критическое мышление	ID-307 Выявляет причинно-следственные			Выявляет причинно-следственные связи при работе с информацией и	Выявляет причинно-следственные связи, обосновывает выводы и	

	связи при анализе информации			обосновывает выводы с посторонней помощью	принимает решения самостоятельно	
Цифровая грамотность	ID-375 Работает с программным обеспечением			Работает с программным обеспечением с посторонней помощью	Самостоятельно работает с программным обеспечением в стандартных условиях	

II.1. Структура образовательных результатов

Формирование цифровых компетенций, необходимых для получения обучающимися дополнительной ИТ-квалификации «Программист», обеспечивается последовательным формированием промежуточных образовательных результатов, начиная со знаний.

ID и формулировка целевого уровня формирования компетенций	Промежуточные образовательные результаты		
	Опыт практической деятельности (ОПД)	Умения (У)	Опыт практической деятельности (ОПД)
ID-28 Применяет языки программирования для решения профессиональных задач	ОПД 1 - составление формализованных описаний решений поставленных задач в соответствии с техническим заданием ОПД 2 - разработка алгоритмов решения поставленных задач в соответствии с техническим заданием	У1 - использовать методы и приемы формализации и алгоритмизации поставленных задач У2 - применять инструментарий для создания и актуализации исходных текстов программ	31 - Методологии разработки программного обеспечения 32 - Технологии программирования
ID-20 Пишет и структурирует код на Python, используя устоявшиеся практики	ОПД 3- создание программного кода на языке python , реализующего логику решения задачи ОПД 4 - слияние, разделение и сравнение исходных текстов программного кода ОПД 5 - Проверка корректности работы программы ОПД 6 - Следование PEP 8 в собственном коде	У3 - применять язык программирования Python для написания программного кода У4 - выявлять ошибки в программном коде на языке Python У5 - Проектировать архитектуру простого скрипта или программы для решения задачи У6 - Выбирать осмысленные имена для переменных, функций, классов, модулей согласно PEP 8	33 - Синтаксис и основные конструкции Python (переменные, типы данных, операторы, условия, циклы, функции). 34 - Основные структуры данных Python (списки, кортежи, словари, множества) и их свойства 35 - Принципы алгоритмизации (базовые алгоритмы: поиск, сортировка, обход). 36 - Руководство по стилю PEP

ID-57 Использует стандартную библиотеку языка Python	ОПД 7 - Практическое применение модулей в коде: импорт, вызов функций, использование классов	У7 - Оценивать, какой модуль лучше подходит для решения конкретной подзадачи	37 - стандартные библиотеки языка программирования Python
ID-307 Выявляет причинно-следственные связи при анализе информации	ОПД 8 - Использование Python (и библиотек типа pandas, numpy) для предварительной обработки и очистки данных. ОПД 9 - Применение методов фильтрации, сортировки, группировки данных для поиска закономерностей	У8 - Формулировать четкие вопросы к данным для выявления связей У9 - критически оценивать результаты анализа: достаточно ли данных, нет ли скрытых факторов	38 - Основные логические операции (И, ИЛИ, НЕ) и их применение в условиях. 39 - Принципы анализа данных (постановка вопроса, сбор данных, очистка, исследование, интерпретация).
ID-375 Работает с программным обеспечением	ОПД 10 - Уверенное использование выбранной IDE для разработки на Python (написание, запуск, отладка).	У10 - Выбирать подходящее ПО для конкретной задачи	310 - Назначение и базовые принципы работы используемого ПО (IDE - PyCharm/VSCode/Jupyter)

II.2. Структура Программы

Структура Программы регулирует образовательные траектории обучающихся, последовательность освоения структурных элементов (разделов) Программы, соответственно, последовательность формирования всех образовательных результатов.

Структурные элементы (разделы Программы)	Шифры образовательных результатов	Вариатив / инвариант и целевые группы обучающихся
Общепрофессиональный цикл (ОПЦ)		
Модуль 1. Обзор языка Python. Базовые конструкции Тема 1. Обзор существующих парадигм и языков программирования. Лекция: Введение в программирование. Особенности исполнения программ. Основные принципы работы интерпретатора Python. Версии языка Python. Практическое занятие: Python в различных операционных системах. Установка и запуск интерпретатора Python. Написание и запуск простой программы. Простой ввод и вывод данных. Установка и тестирование среды разработки Python. Настройка окружения. Применение онлайн-интерпретаторов. Тема 2. Основные элементы программирования Лекция: Структура программы. Переменные и выражения. Ключевые слова. Типы данных: числовой, строковый, логический, списки. Принцип динамической типизации. Арифметические операции. Операции со строками. Оформление кода. Комментарии и автоматическое документирование.	знания 33-36, 310 умения У3-У6, У10	Инвариант для всех групп обучающихся

Структурные элементы (разделы Программы)	Шифры образовательных результатов	Вариатив / инвариант и целевые группы обучающихся
Практическое занятие: Формирование навыков написания и отладки программ, использующих переменные различных типов. Написание арифметических выражений, изучение приоритета и последовательности выполнения операций. Применение программирования для решения простых расчетных задач. Использование оператора del. Знакомство с особенностями строкового типа данных и основными методами работы со строками. Тема 3. Управляющие конструкции: условия, циклы Лекция 1: Основы алгоритмизации: базовые управляющие конструкции в программировании. Логический тип данных. Ветвление и условный оператор. Базовая форма цикла. Реализация циклов с предусловием и с постусловием. Операторы break и continue. Оператор pass. Практическое занятие: Формирование навыков написания и отладки программ, использующих базовые управляющие конструкции (ветвления, циклы). Проверка свойств и значений. Изучение основных логических операций и написание программ, использующих простые и составные условия. Использование циклов. Проектирование, написание и отладка программ, использующих вложенные управляющие конструкции и условия продолжения/прерывания циклов.		
Модуль 2. Коллекции: кортежи, списки, словари, множества Тема 1. Список. Основные операции со списком Лекция: Коллекции данных: виды, особенности, назначение и примеры использования. Создание списка. Добавление элементов в список. Обращение к элементам списка. Перебор значений из списка. Удаление элементов из списка. Сортировка. Сравнение списков. Копирование списка. Объединение списков. Срезы. Вложенные списки. Практическое занятие: Формирование навыков написания и отладки программ, использующих списки. Создание и инициализация списка. Модификация списка. Доступ к элементам списка. Изучение методов для работы со списком. Решение задач на обработку информации, содержащейся в списках. Тема 2. Кортеж. Основные операции с кортежем Лекция: Отличия кортежа от списка. Операции с кортежами. Доступ к элементам кортежа. Преобразование кортежа в список. Формирование кортежа на основе списка. Получение подкортежей. Вложенные коллекции. Изучение методов для работы с кортежом. Практическое занятие: Формирование навыков написания и отладки программ, использующих кортежи. Создание и инициализация кортежа. Доступ к элементам кортежа. Изучение методов для работы с кортежом. Решение задач на обработку информации, содержащейся в кортежах и списках. Тема 3. Множества. Основные операции с множеством Лекция: Отличия множества от списка и кортежа. Операции с множествами. Проверка элементов множества (принадлежность). Преобразование множества в список и кортеж. Особенности формирования множества на основе списка и кортежа. Изучение методов для работы с множеством. Операции с множествами: объединение, пересечение, разность, подмножество. Неизменяемые множества. Вложенные коллекции.	знания 33-37, 310 умения У3-У7, У!0	Инвариант для всех групп обучающихся

Структурные элементы (разделы Программы)	Шифры образовательных результатов	Вариатив / инвариант и целевые группы обучающихся
Практическое занятие: Формирование навыков написания и отладки программ, использующих множества. Создание и инициализация множества. Доступ к элементам кортежа. Включение элементов в множество, исключение элементов. Перебор элементов множества. Изучение методов для работы с множеством. Объединение и пересечение множеств. Разность множеств. Определение подмножества и надмножества. Решение задач на обработку информации, содержащейся в списках, кортежах и множествах. Тема 4. Словарь. Основные операции со словарем Лекция: Особенности словаря. Создание словаря. Операции со словарем. Преобразование списков и кортежей в словарь. Добавление элементов в словарь. Изменение и удаление значений по ключу. Извлечение элементов из словаря. Копирование и объединение словарей. Перебор словаря. Распаковка и упаковка коллекций. Вложенные коллекции. Практическое занятие: Формирование навыков написания и отладки программ, использующих словари. Создание и инициализация словаря. Доступ к элементам словаря. Включение элементов в словарь, удаление и извлечение элементов. Перебор элементов словаря. Изучение методов для работы со словарем. Решение задач на обработку информации, содержащейся в словарях, списках, кортежах и множествах.		
Модуль 3. Функции. Алгоритмы и структуры данных Тема 1. Список. Объявление и вызов функции Лекция: Объявление функции. Вызов функции. Параметры функции, значения по умолчанию. Вызов функции с именованными и неименованными аргументами. Выход из функции, возврат значений. Практическое занятие: Формирование навыков написания и отладки программ, использующих функции. Определение функций. Использование списка параметров (аргументов) функции. Вызовы функций в функциях. Возврат значений функцией и обработка полученных результатов. Функциональная декомпозиция алгоритмов. Отладка программ с использованием функций. Тема 2. Основы функционального программирования Лекция: Функция как тип, параметр и результат другой функции. Вложенные (локальные) функции. Лямбда-функция. Рекурсия. Практическое занятие: Основы функционального программирования. Формирование навыков написания и отладки программ, использующих функции. Работа с функциональными объектами. Передача функции в функцию как параметра. Возврат функции из функции. Определение локальной функции. Определение безымянных (анонимных) функций. Написание лямбда-выражений. Рекурсивный вызов функции. Прямая и косвенная рекурсия. Отладка программ с использованием функций и лямбда-выражений. Разработка программ с использованием функционального подхода. Тема 3. Обработка коллекций Функции для работы с коллекциями: map, zip, filter, reduce. Генераторное выражение и генератор-функция. Генераторы списков, словарей, множеств	знания: 31-310, умения: У1-У10	Инвариант для всех групп обучающихся

Структурные элементы (разделы Программы)	Шифры образовательных результатов	Вариатив / инвариант и целевые группы обучающихся
Практическое занятие: Формирование навыков написания и отладки программ для обработки наборов данных. Преобразование (отображение) элементов коллекции, отбор значений, выполнение агрегирующих операций. Распаковка и упаковка коллекций. Написание генераторных выражений для порождения списков, кортежей, множеств и словарей. Генерация коллекций со сложной структурой. Проектирование и реализация программы для обработки набора данных (датасета).. Тема 4. Структуры данных и алгоритмы Способы организации данных: связанный список, стек, очередь, бинарное дерево, куча, граф. Алгоритмы организации данных и доступа к данным. Алгоритмы поиска и сортировки. Обход бинарного дерева поиска. Практическое занятие: Формирование навыков написания и отладки программ для работы со структурами данных. Организация данных в виде связанного списка. Обход списка. Поиск элемента в списке. Сортировка элементов списка. Выполнение операций агрегации над элементами списка. Работа со списком как со стеком и как с очередью. Программная реализация очереди с приоритетом. Организация данных в виде бинарного дерева поиска и доступ к элементам дерева. Организация данных в виде графа. Поиск элемента в графе.		
Модуль 4. Основы объектно-ориентированного программирования Тема 1. Абстракция, классы и инкапсуляция Лекция: Основы объектно-ориентированного программирования. Абстракция и определение классов. Инкапсуляция, атрибуты и свойства. Свойства (поля, атрибуты) и методы класса. Классы и объекты. Аннотации свойств. Конструкторы. Статические поля и методы. Практическое занятие: Написание и отладка программ с использованием классов. Проектирование класса. Определение полей и методов класса. Создание экземпляров (объектов) класса. Использование ключевого слова self. Написание конструкторов. Использование аннотации свойств. Написание классов с вложенными классами. Работа со статическими полями и методами класса. Тема 2. Наследование Лекция: Иерархия типов и наследование. Базовый класс (супер класс) и производный класс (подкласс). Переопределение функций базового класса. Особенности создания объектов производного класса. Проверка типа объекта (рефлексия). Множественное наследование. Практическое занятие: Проектирование иерархии классов, написание и отладка программ с использованием классов. Работа с объектами разных классов. Использование множественного наследования. Тема 3. Полиморфизм подтипов Лекция: Понятие полиморфизма подтипов. Использование полиморфизма. Унификация работы с объектами разных типов. Класс object. Строковое представление объекта. Практическое занятие: Написание и отладка программ с использованием базовых и производных классов. Работа с объектами разных классов (подтипов) как с объектами общего типа. Получение строкового представления объекта.	знания: 31-310 умения: У1-У10	Инвариант для всех групп обучающихся

Структурные элементы (разделы Программы)	Шифры образовательных результатов	Вариатив / инвариант и целевые группы обучающихся
Тема 4. Обработка исключений Лекция: Возможные ошибки при работе программы. Исключительные ситуации и способы реакции на них. Принципы обработки исключений. Конструкция try...except...else...finally. Встроенные типы исключений. Генерация исключения. Обработка исключений различных типов. Генерация исключений и оператор raise. Создание своих классов исключений. Практическое занятие: Написание программ с обработкой ошибок и исключений. Использование конструкции try...except...else...finally. Перехват исключения и получение информации о нем. Генерация исключений стандартных типов. Определение нового типа исключения, генерация и обработка пользовательских исключений. Анализ выполнения программ с управляемыми и неуправляемыми исключениями.		
Модуль 5. Модули и пакеты для реализации профессиональных задач нефтегазодобывающих предприятий Тема 1. Разработка модуля Лекция: Модули: назначение и правила создания. Особенности генерации и подключения модуля. Написание собственного модуля. Автоматическое документирование модуля. Импорт модуля и объектов из модуля. Использование программных объектов модуля. Практическое занятие: Формирование навыков оформления программ в виде модуля. Создание модуля. Подключение и использование модулей. Выгрузка модуля. Условная загрузка модуля. Написание программ, включающих библиотеку функций, классов, констант и формирование модулей. Написание программы, использующей множество модулей. Тема 2. Создание пакета модулей Лекция: Пакеты: назначение и правила создания. Особенности организации и подключения пакета. Написание собственного пакета. Импорт пакета, модуля и объектов из модуля. Инициализация пакета. Использование программных объектов и модулей из пакета. Практическое занятие: Создание собственного пакета. Загрузка пакета и его элементов. Написание программы, использующей множество пакетов и модулей. Тема 3. Стандартные и распространяемые пакеты и модули Основные стандартные пакеты. Подсистема rip. Сторонние модули. Практическое занятие: Подключение стандартного пакета. Импорт объектов из пакета. Установка стороннего модуля с использованием подсистемы rip из репозитория и локального каталога. Написание программы с использованием собственных, стандартных и сторонних модулей и пакетов.	знания: 3I-3I0 умения: УI-УI0	
Профессиональный цикл		
Модуль 5. Практика/ стажировка	опыт практической деятельности: ОПД1-ОПД10	Инвариант для всех групп обучающихся

III. Учебный план Программы

Объем Программы составляет – 256 часов.

Учебный план Программы определяет перечень, последовательность, общую трудоемкость разделов и формы контроля знаний.

Структурные элементы (разделы Программы)	Общая трудоемкость, часов	Учебная нагрузка		Самостоятельная работа, часов		Практики, стажировки, часов	Промежуточная аттестация, часов
		всего, часов	в т.ч. практические занятия, часов	всего, часов	в т.ч. практические занятия, часов		
Обзор языка Python. Базовые конструкции	30	18	18	10	10	-	2
Коллекции: кортежи, списки, словари, множества	38	26	26	10	10	-	2
Функции. Алгоритмы и структуры данных	38	26	26	10	10	-	2
Основы объектно-ориентированного программирования	40	26	26	12	12	-	2
Модули и пакеты для реализации профессиональных задач нефтегазодобывающих предприятий	38	26	26	10	10	-	2
Практика / стажировка	70	-	-	-	-	70	
Итоговая аттестация в формате демонстрационного экзамена	2	-	2	-	-	-	-
Итого:	256	122	124	52	52	70	10

IV. Календарный учебный график

Календарный учебный график представляет собой график учебного процесса, устанавливающий последовательность и продолжительность обучения, включая практику, и итоговой аттестации по неделям. При этом время, выделяемое на прохождение оценки сформированности цифровых компетенций, в общей трудоёмкости Программы, отражённой в Учебном плане, не учитывается.

Структурные элементы (разделы Программы) и этапы ассессмента	Учебные недели																		
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38
Модуль 1. Обзор языка Python. Базовые конструкции		+	+	+	+	+													
Модуль 2. Коллекции: кортежи, списки, словари, множества		+	+	+	+	+													
Модуль 3. Функции. Алгоритмы и структуры данных							+	+	+	+	+	+							
Модуль 4. Основы объектно-ориентированного программирования							+	+	+	+	+	+							
Модуль 5. Модули и пакеты для реализации профессиональных задач нефтегазодобывающих предприятий												+	+	+	+	+	+		
Модуль 6. Практика/стажировка												+	+	+	+	+	+		
Итоговая аттестация																		+	+

Структурные элементы (разделы Программы)	Учебные недели (учеб. Часы/СРС)																	
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36
Модуль 1. Обзор языка Python. Базовые конструкции	6/ 2	8/ 4	6/ 4	8/ 4	6/ 2	8/ 4												
Модуль 2. Коллекции: кортежи, списки, словари, множества	6/ 2	8/ 4	6/ 4	8/ 4	6/ 2	8/ 4												
Модуль 3. Функции. Алгоритмы и структуры данных							4/ 2	4/ 2	4/ 4	6/ 6	2/ 2	4/ 4						
Модуль 4. Основы объектно-ориентированного программирования							6/ 2	8/ 4	6/ 4	8/ 4	6/ 2	8/ 4						

[illegible]

V. Рабочие программы модулей (курсов, дисциплин)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ”
ИНСТИТУТ ДОПОЛНИТЕЛЬНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ**

Рабочая программа модуля 1

Обзор языка Python. Базовые конструкции

1. Область применения рабочей программы

Рабочая программа модуля 1. «Обзор языка Python. Базовые конструкции» (далее – рабочая программа) является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Python в области добывающей промышленности и природопользования» и направлена на формирование цифровых компетенций:

ID-20 Пишет и структурирует код на Python, используя устоявшиеся практики

ID-57 Использует стандартную библиотеку языка Python

ID-375 Работает с программным обеспечением

Промежуточные образовательные результаты:

1) Обладает знаниями о стандартной библиотеке языка программирования Python

2) Обладает знаниями и способен писать код используя синтаксис и основные конструкции Python.

3) Использует выбранную IDE для разработки на Python.

Освоение рабочей программы является инвариантным (обязательным) для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем, часов
1	Тема 1. Обзор существующих парадигм и языков программирования Лекция: Введение в программирование. Особенности исполнения программ. Основные принципы работы интерпретатора Python. Версии языка Python. Практическое занятие: Python в различных операционных системах. Установка и запуск интерпретатора Python. Написание и запуск простой программы. Простой ввод и вывод данных. Установка и тестирование среды разработки Python. Настройка окружения. Применение онлайн-интерпретаторов.	6
2	Тема 2. Основные элементы программирования Лекция: Структура программы. Переменные и выражения. Ключевые слова. Типы данных: числовой, строковый, логический, списки. Принцип динамической типизации. Арифметические операции. Операции со строками. Оформление кода. Комментарии и автоматическое документирование. Практическое занятие: Формирование навыков написания и отладки программ, использующих переменные различных типов. Написание арифметических выражений, изучение приоритета и последовательности выполнения операций. Применение программирования для решения простых расчетных задач.	6

№ п/п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем, часов
	Использование оператора del. Знакомство с особенностями строкового типа данных и основными методами работы со строками	
3	Тема 3. Управляющие конструкции: условия, циклы Лекция 1: Основы алгоритмизации: базовые управляющие конструкции в программировании. Логический тип данных. Ветвление и условный оператор. Базовая форма цикла. Реализация циклов с предусловием и с постусловием. Операторы break и continue. Оператор pass. Практическое занятие: Формирование навыков написания и отладки программ, использующих базовые управляющие конструкции (ветвления, циклы). Проверка свойств и значений. Изучение основных логических операций и написание программ, использующих простые и составные условия. Использование циклов. Проектирование, написание и отладка программ, использующих вложенные управляющие конструкции и условия продолжения/прерывания циклов.	6
4	Самостоятельная работа	10
5	Промежуточная аттестация в формате: зачета	2

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов	
		практические занятия	самостоятельная работа
1	Тема 1. Обзор существующих парадигм и языков программирования	6	2
2	Тема 2. Основные элементы программирования	6	4
3	Тема 3. Управляющие конструкции: условия, циклы	6	4
4	Промежуточная аттестация	2	
Итого		30	

4. Контроль и оценка результатов освоения рабочей программы

Образовательная организация высшего образования, реализующая рабочую программу, обеспечивает организацию и проведение текущего и промежуточного контроля демонстрируемых обучающимися образовательных результатов.

Текущий контроль проводится преподавателем на основе оценивания результатов практических работ и самостоятельной работы обучающихся. Промежуточный контроль проводится в форме тестирования. Формы и методы текущего и промежуточного контроля, критерии оценивания доводятся до сведения обучающихся в начале обучения.

Для текущего и промежуточного контроля создаются фонды оценочных средств (ФОС). ФОС включают в себя педагогические контрольно-измерительные материалы, предназначенные для определения соответствия (или несоответствия) индивидуальных образовательных достижений запланированным образовательным результатам.

4.1. Примеры оценочных средств

Порядок промежуточной аттестации по дисциплине

Формой промежуточной аттестации по дисциплине является – зачет.

Зачет проводится в формате выполнения тестового задания.

Примеры тестовых заданий

1. Что выведет данный код: a = 1; for i in range(4): a *= i; print(a):

- a. 0;
- b. 24;
- c. 16;
- d. ошибку.

2. Что выведет данный код: a = 0; for i in range(4): for i in range(6): a = i print(a):

- a. 5;
- b. 6;
- c. 4;
- d. ошибку.

3. Что выведет данный код: a = 40; b = 0; for a in range(5): for i in range(1): b = a; print(b):

- a. 40;
- b. 4;
- c. 5;
- d. ошибку.

4. Что выведет данный код: a = list(range(3)) print(a):

- a. 0 1 2;
- b. [0, 1, 2];
- c. 3;
- d. ошибку.

5. Что выведет данный код: a = range(3) print(a):

- a. 3;
- b. range(0, 3);
- c. ошибку;
- d. ничего.

6. Что выведет данный код: a = list("python"); b = ""; for i in range(len(a)):

b += a[(len(a) - i - 1)] print(b):

- a. python;

- b. nohtyp;
- c. thonpy;
- d. ошибку.

7. Что выведет данный код: a = 0: for i in range(5): if i % 2 == 0: a += i print(a):

- a. 6;
- b. 5;
- c. 7;
- d. ошибку.

8. Что выведет данный код: a = ['1', '2', '3']: b = ' ' for i in a: b+= a[i]:

- a. 123;
- b. 12;
- c. 23;
- d. ошибку.

27

9. Что выведет данный код: a = ' ' for i in 'str': a += i print(a):

- a. str;
- b. rt;
- c. st;
- d. ошибку.

10. Что выведет данный код: a = 1; for i in range(5): a = a * i print(a):

- a. 15;
- b. 5;
- c. 0;
- d. ошибку.

11. Что выведет данный код: while 1: print(1, break):

- a. 1;
- b. 2;
- c. ничего;
- d. ошибку.

12. Что выведет данный код: for i in 'hello world': print(i * 2, end=' '):

- a. hheellllloo wwoorrlldd;
- b. hheelloo wwoorrlldd;
- c. hello world;
- d. ошибку.

13. Что выведет данный код: for i in 'hello world': if i == 'o': break print(i * 2, end=' '):

- a. hheelllll;
- b. heell;
- c. oo;
- d. ошибку.

14. Что выведет данный код: `for i in range(1): print(i):`

- a. 0 1;
- b. 0;
- c. 1;
- d. ошибку.

15. Что выведет данный код: `for i in range(-2): print(i):`

- a. -0 -1;
- b. -1;
- c. ничего;
- d. ошибку.

16. Что выведет данный код: `for i in range(len("str")): if i != 2: print("str"[i], end="-") else: print("str"[1]):`

- a. s-t-t;
- b. s-t-r;
- c. s-t-r-;
- d. s-t-t.

17. Что выведет данный код: `for i in "str": print(i.upper(), end="."):`

- a. S.T.R.;
- b. S.T.R;
- c. S.T.T;
- d. S.T.T.

18. Что выведет данный код: `for i in 'hello world': if i == 'a': break else: print('0'):`

- a. 0;
- b. a;
- c. hello world;
- d. ошибку.

28

19. Что выведет данный код: `a = 0; while a == 0: continue a += 1 print(a):`

- a. 1;
- b. 0;
- c. ничего;
- d. ошибку.

6. Критерии оценки

Система оценок традиционная «зачтено», «не зачтено».

Оценка по результатам тестирования определяется исходя из доли (в процентном выражении) правильных ответов в общем объеме заданных заданий.

Примеры практико-ориентированных заданий (кейсов)

Практическое задание 1.

1. Напишите несколько строк кода, выводящих на экран ваше имя и почтовый адрес. Адрес напишите в формате, принятом в вашей стране. Никакого ввода от пользователя ваша первая программа принимать не будет, только вывод на экран и больше ничего.

2. Напишите программу, запрашивающую у пользователя его имя. В ответ на ввод на экране должно появиться приветствие с обращением по имени, введенному с клавиатуры ранее.

3. Представьте, что вы открыли в банке сберегательный счет под 4% годовых.

Проценты банк рассчитывает в конце года и добавляет к сумме счета. Напишите программу, которая запрашивает у пользователя сумму первоначального депозита, после чего рассчитывает и выводит на экран сумму на счету в конце первого, второго и третьего годов.

Практическое задание 2.

1. Напишите программу, запрашивающую у пользователя длину и ширину комнаты. После ввода значений должен быть произведен расчет площади комнаты и выведен на экран. Длина и ширина комнаты должны вводиться в формате числа с плавающей запятой. Дополните ввод и вывод единицами измерения, принятыми в вашей стране. Это могут быть футы или метры.

2. Объем цилиндра может быть рассчитан путем умножения площади круга, лежащего в его основе, на высоту. Напишите программу, в которой пользователь будет задавать радиус цилиндра и его высоту, а в ответ получать его объем, округленный до одного знака после запятой.

3. Создайте программу, которая запрашивает у пользователя два целых числа a и b , после чего выводит на экран результаты следующих математических операций:

- сумма a и b ;
- разность между a и b ;
- произведение a и b ;
- частное от деления a на b ;
- остаток от деления a на b ;
- десятичный логарифм числа a ;
- результат возведения числа a в степень b .

4. Площадь треугольника может быть вычислена с использованием следующей формулы, $S = b \cdot h / 2$ где b – длина основания треугольника, а h – его высота: Напишите программу, в которой пользователь сможет вводить значения для переменных b и h , после чего на экране будет отображена площадь треугольника с заявленными основанием и высотой.

5. Программа, которую вы напишете, должна начинаться с запроса у пользователя суммы заказа в ресторане. После этого должен быть произведен расчет налога и чаевых официанту. Вы можете использовать принятую в вашем регионе налоговую ставку для подсчета суммы сборов. В качестве чаевых мы оставим 18 % от стоимости заказа без учета налога. На выходе программа должна отобразить отдельно налог, сумму чаевых и итог, включая обе составляющие. Форматируйте вывод таким образом, чтобы все числа отображались с двумя знаками после запятой.

Практическое задание 3.

1. Напишите программу, запрашивающую у пользователя число и подсчитывающую сумму натуральных положительных чисел от 1 до введенного пользователем значения. Сумма первых n положительных чисел может быть рассчитана по формуле: $\text{sum} = (n*(n+1))/2$.

2. Представьте, что вы пишете программное обеспечение для автоматической кассы в магазине самообслуживания. Одной из функций, заложенных в кассу, должен быть расчет сдачи в случае оплаты покупателем наличными. Напишите программу, которая будет запрашивать у пользователя сумму сдачи в центах. После этого она должна рассчитать и вывести на экран, сколько и каких монет потребуется для выдачи указанной суммы, при условии, что должно быть задействовано минимально возможное количество монет.

Допустим, у нас есть в распоряжении монеты достоинством в 1, 5, 10, 25 центов, а также в 1 (loonie) и 2 (toonie) канадских доллара.

3. Напишите программу, которая будет запрашивать у пользователя радиус и сохранять его в переменной r . После этого она должна вычислить площадь круга с заданным радиусом и объем шара с тем же радиусом. Используйте в своих вычислениях константу π из модуля `math`.

4. Создайте программу, в которой пользователь сможет ввести временной промежуток в виде количества дней, часов, минут и секунд и узнать общее количество секунд, составляющее введенный отрезок.

5. Разработайте программу, запрашивающую у пользователя целое четырехзначное число и подсчитывающую сумму составляющих его цифр. Например, если пользователь введет число 3141, программа должна вывести следующий результат: $3 + 1 + 4 + 1 = 9$ __

5. Образцы учебно-методических материалов для обучающихся и преподавателей

Для эффективного освоения компетенций, формируемых учебной дисциплиной важно использование в учебном процессе активных и интерактивных форм проведения занятий.

Изучение учебной дисциплины предполагает наличие аудиторной и самостоятельной видов работ слушателей. В ходе практических занятий

рассматриваются бизнес-кейсы, практические задачи, наиболее сложные ситуации из практики с целью наиболее полного овладения умениями и навыками.

Лекции по учебной дисциплине призваны формировать знания, предусмотренные учебной программой, и включают теоретическую базу ведения бухгалтерского учета, на базе которой строятся прикладные аспекты.

Освоение дисциплины предполагает значительный объем самостоятельной внеаудиторной работы, которую слушатели должны выполнять как индивидуально, так и в малых группах.

Наряду с проработкой основной литературы (глав базового учебника) предусмотрено самостоятельное чтение дополнительной литературы (статей и других научных публикаций), а также проведение анализа кейсов, которые обсуждаются в ходе дискуссий на практических занятиях.

Практические занятия в малых группах и самостоятельная внеаудиторная работа направлены на выработку навыков экономического анализа деятельности предприятий и формирования профессиональных компетенций, установленных в соответствии с целями и задачами дисциплинами.

Для достижения поставленных целей преподавания дисциплины реализуются следующие средства, способы и организационные мероприятия:

- изучение теоретического материала дисциплины на лекции с использованием компьютерных технологий;
- самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, электронных библиотек, методических разработок, специальной и научной литературы;
- закрепление теоретического материала при проведении практических занятий с использованием учебного и научного оборудования, выполнения проблемно-ориентированных, поисковых, творческих заданий.

Самостоятельная работа слушателей включает:

1. Изучение учебной литературы по курсу.
2. Решение практических ситуаций и задач
3. Изучение источников управленческой информации
4. Работу с ресурсами Интернет
5. Решение практических ситуаций в виде творческих заданий
6. Изучение практических материалов деятельности конкретных предприятий
7. Изучение статистикой информации
8. Подготовку к зачету.

Цель самостоятельной работы - подготовка современного компетентного специалиста и формирование способностей и навыков к непрерывному самообразованию и профессиональному совершенствованию.

6. Кадровое обеспечение образовательного процесса

- | | |
|--|---|
| 1. Спирин Игорь Сергеевич,
к.т.н., кафедры математики и
прикладных
информационных
технологий, научно-
педагогический работник,
ТИУ | Лекция 1 Разработка модуля
Лекция 2 Создание пакета модулей
Лекция 3 Стандартные и распространяемые
пакеты и модули

Практика 1. Разработка модуля
Практика 2. Создание пакета модулей
Практика 3. Стандартные и
распространяемые пакеты и модули |
|--|---|

7. Материально-техническое обеспечение реализации рабочей программы

Аудитория	Назначение	Оборудование
509a(3)	Проведение практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной и итоговой аттестации, проведения практик	Учебная мебель для обучающихся, рабочее место преподавателя, доска, мультимедийное оборудование, персональные компьютеры в комплекте на основе операционных систем Microsoft Windows (или др.) с пакетом прикладных программ Microsoft Office (или др), с подключением к сети Интернет Вход на сайт ТИУ в Eduson, программа для организации видеоконференций BigBlueButton

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Асташова Т. А. Основы программирования: учебное пособие / Т. А. Асташова. Новосибирск: Изд-во НГТУ, 2022. 92 с.
2. Васильев А. Н. Python на примерах. Практический курс по программированию: учебное пособие / А. Н. Васильев. 3-е изд. СПб.: Наука и Техника, 2019. 432 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/139151> (дата обращения 05.04.2023).
3. Воробьев Г. А. Основы программирования на Python: учебно-методическое пособие / Г. А. Воробьев. Липецк: Липецкий ГПУ, 2022. 89 с. // Лань: электронно- библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/317075> (дата обращения 05.04.2023).

4. Кольцов Д. М. Справочник PYTHON. Кратко, быстро, под рукой: справочник / Д. М. Кольцов, Е. В. Дубовик. СПб.: Наука и Техника, 2021. 288 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/191480> (дата обращения 05.04.2023).

5. Широков А. И. Алгоритмизация и программирование на языке «Питон» (Python): методические указания / А. И. Широков. М.: МИСИС, 2021. 48 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/238331> (дата обращения 05.04.2023).

Дополнительная литература

1. Бизли Д. Python. Книга рецептов / Д. Бизли, Б. К. Джонс; перевод с английского Б. В. Уварова. М.: ДМК Пресс, 2019. 646 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131723> (дата обращения 05.04.2023).

2. Златопольский Д. М. Основы программирования на языке Python / Д. М. Златопольский. 2-ое изд., испр. и доп. М.: ДМК Пресс, 2018. 396 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131683> (дата обращения 05.04.2023).

3. Янцев В. В. Web-программирование на Python / В. В. Янцев. 2-е изд., стер. СПб.: Лань, 2023. 180 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа <https://e.lanbook.com/book/310289> (дата обращения 05.04.2023).

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ”
ИНСТИТУТ ДОПОЛНИТЕЛЬНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ**

Рабочая программа модуля 2

Коллекции: кортежи, списки, словари, множества

1. Область применения рабочей программы

Рабочая программа модуля 2 «Коллекции: кортежи, списки, словари, множества» (далее – рабочая программа) является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Python в области добывающей промышленности и природопользования» и направлена на формирование цифровых компетенций:

ID-20 Пишет и структурирует код на Python, используя устоявшиеся практики

ID-57 Использует стандартную библиотеку языка Python

Освоение рабочей программы является инвариантным (обязательным) для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем, часов
1.	Тема 1. Список. Основные операции со списком Лекция: Коллекции данных: виды, особенности, назначение и примеры использования. Создание списка. Добавление элементов в список. Обращение к элементам списка. Перебор значений из списка. Удаление элементов из списка. Сортировка. Сравнение списков. Копирование списка. Объединение списков. Срезы. Вложенные списки. Практическое занятие: Формирование навыков написания и отладки программ, использующих списки. Создание и инициализация списка. Модификация списка. Доступ к элементам списка. Изучение методов для работы со списком. Решение задач на обработку информации, содержащейся в списках.	6
2	Тема 2. Кортеж. Основные операции с кортежем Лекция: Отличия кортежа от списка. Операции с кортежами. Доступ к элементам кортежа. Преобразование кортежа в список. Формирование кортежа на основе списка. Получение подкортежей. Вложенные коллекции. Изучение методов для работы с кортежом. Практическое занятие: Формирование навыков написания и отладки программ, использующих кортежи. Создание и инициализация кортежа. Доступ к элементам кортежа. Изучение методов для работы с кортежом. Решение задач на обработку информации, содержащейся в кортежах и списках.	6

№ п/п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем, часов
3	Тема 3. Множества. Основные операции с множеством Отличия множества от списка и кортежа. Операции с множествами. Проверка элементов множества (принадлежность). Преобразование множества в список и кортеж. Особенности формирования множества на основе списка и кортежа. Изучение методов для работы с множеством. Операции с множествами: объединение, пересечение, разность, подмножество. Неизменяемые множества. Вложенные коллекции. Практическое занятие: Формирование навыков написания и отладки программ, использующих множества. Создание и инициализация множества. Доступ к элементам кортежа. Включение элементов в множество, исключение элементов. Перебор элементов множества. Изучение методов для работы с множеством. Объединение и пересечение множеств. Разность множеств. Определение подмножества и надмножества. Решение задач на обработку информации, содержащейся в списках, кортежах и множествах.	6
4	Тема 4. Словарь. Основные операции со словарем Особенности словаря. Создание словаря. Операции со словарем. Преобразование списков и кортежей в словарь. Добавление элементов в словарь. Изменение и удаление значений по ключу. Извлечение элементов из словаря. Копирование и объединение словарей. Перебор словаря. Распаковка и упаковка коллекций. Вложенные коллекции. Практическое занятие: Формирование навыков написания и отладки программ, использующих словари. Создание и инициализация словаря. Доступ к элементам словаря. Включение элементов в словарь, удаление и извлечение элементов. Перебор элементов словаря. Изучение методов для работы со словарем. Решение задач на обработку информации, содержащейся в словарях, списках, кортежах и множествах.	8
5	Самостоятельная работа	10
6	Промежуточная аттестация в формате: зачета	2

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов	
		практические занятия	самостоятельная работа
1	Тема 1. Список. Основные операции со списком	6	2
2	Тема 2. Кортеж. Основные операции с кортежем	6	2
3	Тема 3. Множества. Основные операции с множеством	6	2
4	Тема 4. Словарь. Основные операции со словарем	6	4
	Промежуточная аттестация	2	
Итого		38	

4. Контроль и оценка результатов освоения рабочей программы

Образовательная организация высшего образования, реализующая рабочую программу, обеспечивает организацию и проведение текущего и промежуточного контроля демонстрируемых обучающимися образовательных результатов.

Текущий контроль проводится преподавателем на основе оценивания результатов практических работ и самостоятельной работы обучающихся. Промежуточный контроль проводится в форме тестирования. Формы и методы текущего и промежуточного контроля, критерии оценивания доводятся до сведения обучающихся в начале обучения.

Для текущего и промежуточного контроля создаются фонды оценочных средств (ФОС). ФОС включают в себя педагогические контрольно-измерительные материалы, предназначенные для определения соответствия (или несоответствия) индивидуальных образовательных достижений запланированным образовательным результатам.

4.1. Примеры оценочных средств

Порядок промежуточной аттестации по дисциплине

Формой промежуточной аттестации по дисциплине является – зачет.

Зачет проводится в формате выполнения тестового задания.

Примеры тестовых заданий

**1. Что выведет данный код: `a = set(["p", "y"]) == set(["p", "y"])`
`print(a):`**

- a. True;
- b. False;
- c. ошибку.

**2. Что выведет данный код: `b = set(["p", "y"]) == frozenset(["p", "y"])`
`print(a):`**

- a. True;
- b. False;
- c. ошибку.

34

3. Что выведет данный код: `c = set(["y", "p"]) == frozenset(["p", "y", "y"])` `print(c):`

- a. True;
- b. False;
- c. ошибку.

4. Что выведет данный код: `d = {"p", "i", "p"} a = 0 for i in d: if i[0] == "p": a += 1`

`print(a):`

- a. 0;
- b. 1;
- c. 2;

d. ошибку.

5. Что выведет данный код: `e = set([1, 2]) == {1, 2} print(e)`:

a. True;

b. False;

c. ошибку.

6. Что выведет данный код: `a = set(["p", "y"]) == set(["y", "p"])`

`print(a)`:

a. True;

b. False;

c. ошибку.

7. Что выведет данный код: `b = set(["p", "y"]) == frozenset(["y", "p"])`

`print(a)`:

a. True;

b. False;

c. ошибку.

8. Что выведет данный код: `c = frozenset(["p", "i", "p"]) a = 0 for i in`

`c: if i[0] == "p":`

`a += 1 print(a)`:

a. 0;

b. 1;

c. 2;

d. ошибку.

9. Что выведет данный код: `d = set(["1", "2"]) == str({1, 2}) print(d)`:

a. True;

b. False;

c. ошибку.

10. Что выведет данный код: `e = set(["py"]) == set("py") print(e)`:

a. True;

b. False;

c. ошибку.

11. Что выведет данный код: `a = [1, 2, 3] print(a[1])`:

a. 1;

b. 2;

c. 3;

d. ошибку.

12. Что выведет данный код: `b = [1, 2] == (1, 2) print(b)`:

a. True;

b. False;

c. ошибку.

13. Что выведет данный код: `c = {1: 2} print(len(c))`:

a. 1;

- b. 2;
- c. 3;
- d. ошибку.

14. Что выведет данный код: `a = [] for i in range(3): a += [i] print(a):`

- a. [1, 2, 3];
- b. [0, 1, 2];
- c. [1, 2];
- d. ошибку.

15. Что выведет данный код: `e = [1, 2] print(e[len(e)]):`

- a. 2;
- b. 1;
- c. 0;
- d. ошибку.

16. Что выведет данный код: `a = (1, 2) + 4 print(a):`

- a. (1, 2, 4);
- b. (5, 6);
- c. (1, 6);
- d. ошибку.

17. Что выведет данный код: `b = len({1: 2}) == len([1, 2]) print(b):`

- a. True;
- b. False;
- c. ошибку.

18. Что выведет данный код: `a = [1, 2, 3] b = 0 for i in a: if a[i - 1] < 2: b += a[i] else: b -= 1 print(b):`

- a. 3;
- b. 1;
- c. 0;
- d. ошибку.

19. Что выведет данный код: `d = {1: 2}[1, 2, 3][0] print(d):`

- a. 2;
- b. 1;
- c. 0;
- d. ошибку.

20. Что выведет данный код: `e = [1, 2][1] + 1 print(e):`

- a. [1, 2, 2];
- b. [1, 2, 1];
- c. 3;
- d. ошибку.

Примеры практико-ориентированных заданий (кейсов)

Практическое задание 1.

1. Напишите программу, которая будет запрашивать у пользователя целочисленные значения и сохранять их в виде списка. Индикатором окончания ввода значений должен служить ноль. Затем программа должна вывести на экран все введенные пользователем числа (кроме нуля) в порядке возрастания – по одному значению в строке. Используйте для сортировки либо метод `sort`, либо функцию `sorted`.

2. Напишите программу, которая, как и в предыдущем случае, будет запрашивать у пользователя целые числа и сохранять их в виде списка. Индикатором окончания ввода значений также должен служить ноль. На этот раз необходимо вывести на экран введенные значения в порядке убывания.

3. В данном упражнении вам предстоит разработать программу, в которой у пользователя будет запрошен список слов, пока он не оставит строку ввода пустой. После этого на экране должны быть показаны слова, введенные пользователем, но без повторов, – каждое по одному разу. При этом слова должны быть отображены в том же порядке, в каком их вводили с клавиатуры. Например, если пользователь на запрос программы введет следующий список слов: `first second first third second`, программа должна вывести: `first second third`.

4. Строка называется палиндромом, если она пишется одинаково в обоих направлениях. Например, английская фраза «`Is it crazy how saying sentences backwards creates backwards sentences saying how crazy it is?`» является словесным палиндромом, поскольку если читать ее по словам, игнорируя при этом знаки препинания и заглавные буквы, в обоих направлениях она будет звучать одинаково. Еще примеры английских словесных палиндромов: «`Herb the sage eats sage, the herb`» и «`Information school graduate seeks graduate`

`school information`». Напишите программу, которая будет запрашивать строку у пользователя и оповещать его о том, является ли она словесным палиндромом. Не забывайте игнорировать знаки препинания при выявлении результата.

5. Напишите программу, которая будет запрашивать у пользователя числа, пока он не пропустит ввод. Сначала на экран должно быть выведено среднее значение введенного ряда чисел, после этого друг за другом необходимо вывести список чисел ниже среднего, равных ему (если такие найдутся) и выше среднего. Каждый список должен предваряться соответствующим заголовком.

Практическое задание 2.

1. Напишите программу, которая будет запрашивать у пользователя числа, пока он не пропустит ввод. После этого введенные значения передаются в функцию в формате кортежа. Сначала функция выводит на экран среднее значение введенного ряда чисел, после этого друг за другом выводит список чисел ниже среднего, равных ему (если такие найдутся) и выше среднего. Каждый список должен предваряться соответствующим заголовком.

2. Обычно при написании перечислений и списков мы разделяем их элементы запятыми, а перед последним ставим союз «и», как показано ниже:

- яблоки;
- яблоки и апельсины;
- яблоки, апельсины и бананы;
- яблоки, апельсины, бананы и лимоны.

Напишите функцию, которая будет принимать на вход кортеж из строк и возвращать собранную строку из его элементов в описанной выше манере. Хотя в представленном примере количество элементов списка ограничивается четырьмя, ваша функция должна уметь обрабатывать списки любой длины. В основной программе запросите у пользователя несколько слов, отформатируйте их должным образом при помощи функции и выведите на экран.

3. Напишите функцию, показывающую, отсортирован ли переданный ей в качестве параметра кортеж (по возрастанию или убыванию). Функция должна возвращать True, если список отсортирован, и False в противном случае. В основной программе запросите у пользователя последовательность чисел для кортежа, после чего выведите сообщение о том, является ли этот список отсортированным изначально.

Практическое задание 3.

1. Напишите программу, определяющую и выводящую на экран количество уникальных символов во введенной пользователем строке. Например, в строке Hello, World! содержится десять уникальных символов, а в строке zzz – один. Используйте словарь или набор для решения этой задачи.

2. Для выигрыша главного приза необходимо, чтобы шесть номеров на лотерейном билете совпали с шестью числами, выпавшими случайным образом в диапазоне от 1 до 49 во время очередного тиража. Напишите программу, которая будет случайным образом подбирать шесть номеров для вашего билета. Убедитесь в том, что среди этих чисел не будет дубликатов. Выведите номера билетов на экран по возрастанию.

3. В стандартной библиотеке языка Python присутствует функция count, позволяющая подсчитать, сколько раз определенное значение встречается в списке.

В данном упражнении вы создадите новую функцию countRange, которая будет подсчитывать количество элементов в списке, значения которых больше или равны заданному минимальному порогу и меньше максимального. Функция должна принимать три параметра: список, минимальную границу и максимальную границу. Возвращать она будет целочисленное значение, большее или равное нулю. В основной программе реализуйте демонстрацию вашей функции для нескольких списков с разными минимальными и максимальными границами. Удостоверьтесь, что программа

будет корректно работать со списками, содержащими как целочисленные значения, так и числа с плавающей запятой.

Используя множества, определите количество различающихся элементов в списке.

Практическое задание 4.

1. Напишите функцию с названием `reverseLookup`, которая будет осуществлять поиск всех ключей в словаре по заданному значению. Функция должна принимать в качестве параметров словарь и значение для поиска и возвращать список ключей (он может быть пустым) из этого словаря, соответствующих переданному значению. В основной программе продемонстрируйте работу функции путем создания словаря и поиска в нем всех ключей по заданному значению. Убедитесь, что функция работает корректно при наличии нескольких ключей для искомого значения, одного ключа и их отсутствии.

2. В данном упражнении мы будем симулировать 1000 выбрасываний игральных костей. Начнем с написания функции, выполняющей случайное выбрасывание двух обычных шестигранных костей. Эта функция не будет принимать входных параметров, а возвращать должна число, выпавшее в сумме на двух костях. В основной программе реализуйте симуляцию тысячи выбрасываний костей. Программа должна хранить все результаты с частотой их выпадения. После завершения процесса должна быть показана итоговая таблица с результатами. Выразите частоту выпадения каждого из чисел в процентах вместе с ожидаемым результатом согласно теории вероятностей.

3. Если помните, на старых мобильных телефонах текстовые сообщения набирались при помощи цифровых кнопок. При этом одна кнопка была ассоциирована сразу с несколькими буквами, а выбор зависел от количества нажатий на кнопку. Однократное нажатие приводило к появлению первой буквы в соответствующем этой кнопке списке, последующие нажатия меняли ее на следующую.

Кнопка	Символы
1	.,?!:
2	A B C
3	D E F
4	G H I
5	J K L
6	M N O
7	P Q R S
8	T U V
9	W X Y Z
0	Пробел

Напишите программу, отображающую последовательность кнопок, которую необходимо нажать, чтобы на экране телефона появился текст, введенный пользователем.

Создайте словарь, сопоставляющий символы с кнопками, которые необходимо нажать, а затем воспользуйтесь им для вывода на экран последовательности кнопок в соответствии с введенным пользователем сообщением по запросу. Например, на ввод строки Hello, World! ваша программа должна откликнуться следующим выводом: 443355

5555666110966677755531111. Удостоверьтесь, что ваша программа корректно обрабатывает строчные и прописные буквы. При преобразовании букв в цифры игнорируйте символы, не входящие в указанный перечень, такие как точка с запятой или скобки.

4. Анаграммами называются слова, образованные путем взаимной перестановки букв. В английском языке, например, анаграммами являются слова «live» и «evil», а в русском – «выбор» и «обрыв». Напишите программу, которая будет запрашивать у пользователя два слова, определять, являются ли они анаграммами, и выводить на экран ответ.

5. Карточка для игры в лото состоит из пяти колонок, в каждой из которых – пять номеров. Колонки помечены буквами B, I, N, G и O. Под каждой буквой могут быть номера в своем диапазоне из 15 чисел. А именно под буквой B могут присутствовать числа от 1 до 15, под I – от 16 до 30, под N – от 31 до 45 и т.д.

Напишите функцию, которая будет создавать случайную карточку лото и сохранять ее в словаре. Ключами словаря будут буквы B, I, N, G и O, а значениями – списки из пяти чисел, располагающихся в колонке под каждой буквой. Создайте еще одну функцию для отображения созданной карточки лото на экране со столбцами с заголовками.

В основной программе создайте карту лото случайным образом и выведите ее на экран.

Примечание. Как вы знаете, в настоящих карточках для игры в лото присутствуют пустые клетки в столбцах. Мы не будем реализовывать эту особенность в нашей программе.

5. Образцы учебно-методических материалов для обучающихся и преподавателей

Для эффективного освоения компетенций, формируемых учебной дисциплиной важно использование в учебном процессе активных и интерактивных форм проведения занятий.

Изучение учебной дисциплины предполагает наличие аудиторной и самостоятельной видов работ слушателей. В ходе практических занятий рассматриваются бизнес-кейсы, практические задачи, наиболее сложные ситуации из практики с целью наиболее полного овладения умениями и навыками.

Лекции по учебной дисциплине призваны формировать знания, предусмотренные учебной программой, и включают теоретическую базу ведения бухгалтерского учета, на базе которой строятся прикладные аспекты.

Освоение дисциплины предполагает значительный объем самостоятельной внеаудиторной работы, которую слушатели должны выполнять как индивидуально, так и в малых группах.

Наряду с проработкой основной литературы (глав базового учебника) предусмотрено самостоятельное чтение дополнительной литературы (статей и других научных публикаций), а также проведение анализа кейсов, которые обсуждаются в ходе дискуссий на практических занятиях.

Практические занятия в малых группах и самостоятельная внеаудиторная работа направлены на выработку навыков экономического анализа деятельности предприятий и формирования профессиональных компетенций, установленных в соответствии с целями и задачами дисциплинами.

Для достижения поставленных целей преподавания дисциплины реализуются следующие средства, способы и организационные мероприятия:

- изучение теоретического материала дисциплины на лекции с использованием компьютерных технологий;
- самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, электронных библиотек, методических разработок, специальной и научной литературы;
- закрепление теоретического материала при проведении практических занятий с использованием учебного и научного оборудования, выполнения проблемно-ориентированных, поисковых, творческих заданий.

Самостоятельная работа слушателей включает:

1. Изучение учебной литературы по курсу.
2. Решение практических ситуаций и задач
3. Изучение источников управленческой информации
4. Работу с ресурсами Интернет
5. Решение практических ситуаций в виде творческих заданий
6. Изучение практических материалов деятельности конкретных предприятий
7. Изучение статистикой информации
8. Подготовку к зачету.

Цель самостоятельной работы - подготовка современного компетентного специалиста и формирование способностей и навыков к непрерывному самообразованию и профессиональному совершенствованию.

6. Кадровое обеспечение образовательного процесса

1. Базанов Артем Владимирович, к.т.н., кафедры сервиса автомобилей и технологических машин, научно-педагогический работник, ТИУ
 - Лекция 1. Список. Основные операции со списком
 - Лекция 2. Тема 2. Кorteж. Основные операции с кортежем
 - Лекция 3. Множества. Основные операции с множеством
 - Лекция 4. Словарь. Основные операции со словарем
2. Сидоров Андрей Юрьевич, руководитель НП "АРИТ ТО"
 - Практика 1. Список. Основные операции со списком
 - Практика 2. Тема 2. Кorteж. Основные операции с кортежем
 - Практика 3. Множества. Основные операции с множеством
 - Практика 4. Словарь. Основные операции со словарем

7. Материально-техническое обеспечение реализации рабочей программы

Аудитория	Назначение	Оборудование
509a(3)	Проведение практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной и итоговой аттестации, проведения практик	Учебная мебель для обучающихся, рабочее место преподавателя, доска, мультимедийное оборудование, персональные компьютеры в комплекте на основе операционных систем Microsoft Windows (или др.) с пакетом прикладных программ Microsoft Office (или др), с подключением к сети Интернет Вход на сайт ТИУ в Educon, программа для организации видеоконференций BigBlueButton

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Васильев А. Н. Python на примерах. Практический курс по программированию: учебное пособие / А. Н. Васильев. 3-е изд. СПб.: Наука и Техника, 2019. 432 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/139151> (дата обращения 05.04.2023).
2. Воробьев Г. А. Основы программирования на Python : учебно-методическое пособие / Г. А. Воробьев. Липецк: Липецкий ГПУ, 2022. 89 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/317075> (дата обращения 05.04.2023).

3. Кольцов Д. М. Справочник PYTHON. Кратко, быстро, под рукой: справочник /Д. М. Кольцов, Е. В. Дубовик. СПб.: Наука и Техника, 2021. 288 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/191480> (дата обращения 05.04.2023).

4. Никитина Т. П. Программирование. Основы Python для инженеров / Т. П. Никитина, Л. В. Королев. СПб.: Лань, 2023. 156 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/302720> (дата обращения 05.04.2023).

5. Шевченко Л. Г. Программирование на PYTHON в среде IDLE: учебное пособие / Л.Г. Шевченко, Т.В. Дружинина. Новосибирск: Изд-во НГТУ, 2020. 195 с.

6. Широков А. И. Алгоритмизация и программирование на языке «Питон» (Python): методические указания / А. И. Широков. М.: МИСИС, 2021. 48 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/238331> (дата обращения 05.04.2023).

Дополнительная литература

1. Златопольский Д. М. Основы программирования на языке Python / Д. М. Златопольский. 2-ое изд., испр. и доп. М. : ДМК Пресс, 2018. 396 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131683> (дата обращения 05.04.2023).

2. Янцев В. В. Web-программирование на Python / В. В. Янцев. 2-е изд., стер. СПб.: Лань, 2023. 180 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/310289> (дата обращения 05.04.2023).

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ”
ИНСТИТУТ ДОПОЛНИТЕЛЬНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ**

Рабочая программа модуля 3

Функции. Алгоритмы и структуры данных

1. Область применения рабочей программы

Рабочая программа модуля 3 «Функции. Алгоритмы и структуры данных» в области строительства (далее – рабочая программа) является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Python в области добывающей промышленности и природопользования» и направлена на формирование цифровых компетенций:

ID-20 Пишет и структурирует код на Python, используя устоявшиеся практики

ID-57 Использует стандартную библиотеку языка Python

ID-307 Выявляет причинно-следственные связи при анализе информации

ID-375 Работает с программным обеспечением

Освоение рабочей программы является инвариантным (обязательным) для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем, часов
1.	Тема 1. Список. Объявление и вызов функции Лекция: Объявление функции. Вызов функции. Параметры функции, значения по умолчанию. Вызов функции с именованными и неименованными аргументами. Выход из функции, возврат значений. Практическое занятие: Формирование навыков написания и отладки программ, использующих функции. Определение функций. Использование списка параметров (аргументов) функции. Вызовы функций в функциях. Возврат значений функцией и обработка полученных результатов. Функциональная декомпозиция алгоритмов. Отладка программ с использованием функций.	6
2	Тема 2. Основы функционального программирования Лекция: Функция как тип, параметр и результат другой функции. Вложенные (локальные) функции. Лямбда-функция. Рекурсия. Практическое занятие: Основы функционального программирования. Формирование навыков написания и отладки программ, использующих функции. Работа с функциональными объектами. Передача функции в функцию как параметра. Возврат функции из функции. Определение локальной функции. Определение безымянных (анонимных) функций. Написание лямбда-выражений. Рекурсивный вызов функции. Прямая и косвенная рекурсия. Отладка программ с использованием функций и лямбда-выражений. Разработка программ с использованием функционального подхода.	6
3	Тема 3. Обработка коллекций Функции для работы с коллекциями: map, zip, filter, reduce. Генераторное выражение и генератор-функция. Генераторы списков, словарей,	8

№ п/ п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем, часов
	множеств Практическое занятие: Формирование навыков написания и отладки программ для обработки наборов данных. Преобразование (отображение) элементов коллекции, отбор значений, выполнение агрегирующих операций. Распаковка и упаковка коллекций. Написание генераторных выражений для порождения списков, кортежей, множеств и словарей. Генерация коллекций со сложной структурой. Проектирование и реализация программы для обработки набора данных (датасета).	
4	Тема 4. Структуры данных и алгоритмы Способы организации данных: связанный список, стек, очередь, бинарное дерево, куча, граф. Алгоритмы организации данных и доступа к данным. Алгоритмы поиска и сортировки. Обход бинарного дерева поиска. Практическое занятие: Формирование навыков написания и отладки программ для работы со структурами данных. Организация данных в виде связного списка. Обход списка. Поиск элемента в списке. Сортировка элементов списка. Выполнение операций агрегации над элементами списка. Работа со списком как со стеком и как с очередью. Программная реализация очереди с приоритетом. Организация данных в виде бинарного дерева поиска и доступ к элементам дерева. Организация данных в виде графа. Поиск элемента в графе	6
5	Самостоятельная работа	10
6	Промежуточная аттестация в формате: зачета	2

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов	
		практические занятия	самостоятельная работа
1	Тема 1. Список. Объявление и вызов функции	6	2
2	Тема 2. Основы функционального программирования	6	2
3	Тема 3. Обработка коллекций	8	2
4	Тема 4. Структуры данных и алгоритмы	6	4
	Промежуточная аттестация	2	
Итого		30	

4. Контроль и оценка результатов освоения рабочей программы

Образовательная организация высшего образования, реализующая

рабочую программу, обеспечивает организацию и проведение текущего и промежуточного контроля демонстрируемых обучающимися образовательных результатов.

Текущий контроль проводится преподавателем на основе оценивания результатов практических работ и самостоятельной работы обучающихся. Промежуточный контроль проводится в форме тестирования. Формы и методы текущего и промежуточного контроля, критерии оценивания доводятся до сведения обучающихся в начале обучения.

Для текущего и промежуточного контроля создаются фонды оценочных средств (ФОС). ФОС включают в себя педагогические контрольно-измерительные материалы, предназначенные для определения соответствия (или несоответствия) индивидуальных образовательных достижений запланированным образовательным результатам.

4.1. Примеры оценочных средств

Примеры тестовых заданий к экзамену по дисциплине:

Формой промежуточной аттестации по дисциплине является – зачет. Зачет проводится в формате выполнения тестового задания.

1. Что выведет данный код: `b = 10; a = lambda a: a + b print(a(1)):`

- a. 11;
- b. 10;
- c. 1;
- d. ошибку.

2. Что выведет данный код: `b = lambda: 10 print(b( )):`

- a. 0;
- b. 10;
- c. 1;
- d. ошибку.

3. Что выведет данный код: `def pl(a, b): return a + b print(pl(1, 2) == (lambda a, b: a + b)(1, 2)):`

- a. True;
- b. False;
- c. ошибку.

4. Что выведет данный код: `d = lambda a: (2**a) print(d(2)):`

- a) 4;
- b) 16;
- c) 2;
- d) ошибку.

5. Что выведет данный код: `e = lambda: print(10) e( ):`

- a. 10;
- b. print(10);
- c. ошибку;

d. ничего.

6. Что выведет данный код: `a = lambda x, y: [x + y, x - y] print(a(2, 1)):`

a. [2, 1];

b. [3, 1];

c. [2 + 1, 2 - 1];

d. ошибку.

41

7. Что выведет данный код: `b = lambda e, i: {e: i} print(b(1, 2)[1]):`

a. 2;

b. 1;

c. ошибку;

d. ничего.

8. Что выведет данный код: `c = lambda a b: a + b print(c(1, 2)):`

a. 3;

b. 2;

c. ошибку;

d. ничего.

9. Что выведет данный код: `d = lambda a, b: a + b print(d(1, 2, 3)):`

a. 6;

b. 3;

c. 4;

d. ошибку.

10. Что выведет данный код: `def pl(a): return a + 10 e = lambda a: pl(a) print(e(10)):`

a. 20;

b. 10;

c. ошибку;

d. ничего.

Примеры практико-ориентированных заданий (кейсов)

Практическое задание 1.

1. Напишите функцию, принимающую на вход длины двух катетов прямоугольного треугольника и возвращающую длину гипотенузы, рассчитанную по теореме Пифагора.

В главной программе должен осуществляться запрос длин сторон у пользователя, вызов функции и вывод на экран полученного результата.

2. Представьте, что сумма за пользование услугами такси складывается из базового тарифа в размере \$4,00 плюс \$0,25 за каждые 140 м поездки. Напишите функцию, принимающую в качестве единственного параметра расстояние поездки в километрах и возвращающую итоговую сумму оплаты

такси. В основной программе должен демонстрироваться результат вызова функции.

3. Интернет-магазин предоставляет услугу экспресс-доставки для части своих товаров по цене \$10,95 за первый товар в заказе и \$2,95 – за все последующие. Напишите функцию, принимающую в качестве единственного параметра количество товаров в заказе и возвращающую общую сумму доставки. В основной программе должны производиться запрос количества позиций в заказе у пользователя и отображаться на экране сумма доставки.

Практическое задание 2.

1. Напишите функцию, которая будет принимать на вход три числа в качестве параметров и возвращать их медиану. В основной программе должен производиться запрос к пользователю на предмет ввода трех чисел, а также вызов функции и отображение результата.

2. Необходимо написать функцию, принимающую на вход в качестве единственного аргумента целое число и возвращающую строковое значение, содержащее соответствующее числительное (на английском языке). Ваша функция должна обрабатывать числа в диапазоне от 1 до 12. Если входящее значение выходит за границы этого диапазона, вывод должен оставаться пустым. В основной программе запустите цикл по натуральным числам от 1 до 12 и выведите на экран соответствующие им числительные. Ваша программа должна запускаться только в том случае, если она не импортирована в виде модуля в другой файл.

3. Напишите функцию, которая будет принимать в качестве параметров строку s , а также ширину окна в символах – w . Возвращать функция должна новую строку, в которой вначале добавлено необходимое количество пробелов, чтобы первоначальная строка оказалась размещена по центру заданного окна. Новая строка должна формироваться по следующему принципу:

- если длина исходной строки s больше или равна ширине заданного окна, возвращаем ее в неизменном виде;
- в противном случае должна быть возвращена строка s с ведущими пробелами в количестве $(\text{len}(s) - w) // 2$ штук.

В вашей основной программе должен осуществляться пример вывода нескольких строк в окнах разной ширины.

Практическое задание 3.

Разбиение строки на лексемы (Tokenizing) представляет собой процесс преобразования исходной строки в список из подстрок, называемых лексемами (token).

Зачастую со списком лексем работать бывает проще, чем со всей исходной строкой, поскольку в ней могут присутствовать неравномерные

разрывы. Кроме того, иногда бывает непросто на лету определить, где заканчивается одна лексема и начинается другая.

В математических выражениях лексемами являются, например, операторы, числа и скобки. Здесь и далее мы будем причислять к списку операторов следующие: $*$, $/$, $^$, $-$ и $+$.

Операторы и скобки легко идентифицировать, поскольку эти лексемы всегда состоят ровно из одного символа и никогда не являются составной частью других лексем. Числа выделить бывает сложнее, поскольку эти лексемы могут состоять из нескольких символов. Любая непрерывная последовательность цифр должна восприниматься как одна числовая лексема.

Напишите функцию, принимающую в качестве единственного входного параметра строку, содержащую математическое выражение, и преобразующую ее в список лексем.

Каждая лексема должна быть либо оператором, либо числом, либо скобкой. Для простоты реализации в данном упражнении мы будем оперировать только целочисленными значениями.

Функция должна возвращать созданный список лексем. При решении поставленной задачи вы можете принять допущение о том, что входная строка всегда будет содержать математическое выражение, состоящее из скобок, чисел и операторов. При этом в вашей функции должно быть предусмотрено, что лексемы могут отделяться друг от друга разным количеством пробелов, а могут и не отделяться вовсе. В основной программе продемонстрируйте работу функции, запросив у пользователя исходную строку и выведя на экран список составляющих ее лексем.

Практическое задание 4.

Разработать структуру данных для хранения односвязного списка. Тип данных, хранящихся в списке, согласовать с преподавателем. Реализовать функции для работы со списком: добавление элемента в список (в заданную позицию в списке), получение данных элемента списка (по номеру), получение адреса элемента в списке (по номеру), удаление элемента (по номеру), подсчет количества элементов в списке.

Номер варианта:

1. Определить количество вхождений числа E в список. Вернуть адрес первого появления в списке.
2. Оставить в списке только первые вхождения одинаковых элементов. Вернуть количество удаленных.
3. Заменить в списке все вхождения числа A на число B . Вернуть адрес последней замены.
4. Добавить в конец списка $L1$ копии элементов списка $L2$ в обратном порядке.

5. Найти первый элемент списка, который содержит отрицательное значение.

Вернуть адрес элемента.

6. В списке из подряд идущих одинаковых элементов оставить только один. Вернуть количество удаленных.

7. Определить количество элементов списка, значения которых совпадают со значениями в первом или последнем элементах списка.

8. Создать список L3 из элементов, входящих и в список L1 и в список L2. Вернуть длину нового списка.

9. Заменить в списке все отрицательные значения элементов на их модуль. Вернуть адрес последней замены.

10. Создать список L3 из элементов, которые входят в список L1 и не входят в список

L2. Вернуть длину нового списка.

11. Найти первый элемент списка, который содержит положительное значение.

Вернуть адрес элемента.

12. Оставить в списке только последние вхождения одинаковых элементов. Вернуть количество удаленных.

13. Найти максимальный элемент списка. Вернуть адрес.

14. Добавить в список L1 за первым вхождением элемента E все элементы списка L2.

Вернуть длину нового списка.

15. Определить, есть ли в списке два одинаковых значения (вывести его).

16. «Перенести» в начало списка все отрицательные элементы. Вернуть количество переносов.

17. Найти среднее арифметическое значение элементов списка

18. Определить, входит ли список L1 в L2. Вернуть адрес начала вхождения (нулевой адрес в противном случае).

19. Поменять местами первый и последний элементы списка.

20. Проверить равенство списков L1 и L2 (вернуть 1 – если списки равны,

0 в противном случае).

5. Образцы учебно-методических материалов для обучающихся и преподавателей

Для эффективного освоения компетенций, формируемых учебной дисциплиной важно использование в учебном процессе активных и интерактивных форм проведения занятий.

Изучение учебной дисциплины предполагает наличие аудиторной и самостоятельной видов работ слушателей. В ходе практических занятий

рассматриваются бизнес-кейсы, практические задачи, наиболее сложные ситуации из практики с целью наиболее полного овладения умениями и навыками.

Лекции по учебной дисциплине призваны формировать знания, предусмотренные учебной программой, и включают теоретическую базу ведения бухгалтерского учета, на базе которой строятся прикладные аспекты.

Освоение дисциплины предполагает значительный объем самостоятельной внеаудиторной работы, которую слушатели должны выполнять как индивидуально, так и в малых группах.

Наряду с проработкой основной литературы (глав базового учебника) предусмотрено самостоятельное чтение дополнительной литературы (статей и других научных публикаций), а также проведение анализа кейсов, которые обсуждаются в ходе дискуссий на практических занятиях.

Практические занятия в малых группах и самостоятельная внеаудиторная работа направлены на выработку навыков экономического анализа деятельности предприятий и формирования профессиональных компетенций, установленных в соответствии с целями и задачами дисциплинами.

Для достижения поставленных целей преподавания дисциплины реализуются следующие средства, способы и организационные мероприятия:

- изучение теоретического материала дисциплины на лекции с использованием компьютерных технологий;
- самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, электронных библиотек, методических разработок, специальной и научной литературы;
- закрепление теоретического материала при проведении практических занятий с использованием учебного и научного оборудования, выполнения проблемно-ориентированных, поисковых, творческих заданий.

Самостоятельная работа слушателей включает:

1. Изучение учебной литературы по курсу.
2. Решение практических ситуаций и задач
3. Изучение источников управленческой информации
4. Работу с ресурсами Интернет
5. Решение практических ситуаций в виде творческих заданий
6. Изучение практических материалов деятельности конкретных предприятий
7. Изучение статистикой информации
8. Подготовку к зачету.

1) Цель самостоятельной работы - подготовка современного компетентного специалиста и формирование способностей и навыков к непрерывному самообразованию и профессиональному совершенствованию.

6. Кадровое обеспечение образовательного процесса

- | | |
|---|--|
| 1. Козин Евгений Сергеевич,
к.т.н., кафедры сервиса
автомобилей и
технологических машин,
научно-педагогический
работник, ТИУ | Лекция 1. Список. Объявление и вызов функции
Лекция 2. Основы функционального программирования
Лекция 3. Обработка коллекций
Лекция 4. Структуры данных и алгоритмы

Практика 1. Список. Объявление и вызов функции
Практика 2. Основы функционального программирования
Практика 3. Обработка коллекций
Практика 4. Структуры данных и алгоритмы |
|---|--|

7. Материально-техническое обеспечение реализации рабочей программы

Аудитория	Назначение	Оборудование
509a(3)	Проведение практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной и итоговой аттестации, проведения практик	Учебная мебель для обучающихся, рабочее место преподавателя, доска, мультимедийное оборудование, персональные компьютеры в комплекте на основе операционных систем Microsoft Windows (или др.) с пакетом прикладных программ Microsoft Office (или др), с подключением к сети Интернет Вход на сайт ТИУ в Educon, программа для организации видеоконференций BigBlueButton

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Воробьев Г. А. Основы программирования на Python: учебно-методическое пособие / Г. А. Воробьев. Липецк: Липецкий ГПУ, 2022. 89 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/317075> (дата обращения 05.04.2023).
2. Кольцов Д. М. Справочник PYTHON. Кратко, быстро, под рукой: справочник /Д. М. Кольцов, Е. В. Дубовик. СПб.: Наука и Техника, 2021. 288

с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/191480> (дата обращения 05.04.2023).

3. Никитина Т. П. Программирование. Основы Python для инженеров / Т. П. Никитина, Л. В. Королев. СПб.: Лань, 2023. 156 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/302720> (дата обращения 05.04.2023).

4. Шевченко Л. Г. Программирование на PYTHON в среде IDLE: учебное пособие / Л.Г. Шевченко, Т.В. Дружинина. Новосибирск: Изд-во НГТУ, 2020. 195 с.

5. Широбокова С. Н. Программирование на языке Python для лабораторных занятий: учебное пособие / С. Н. Широбокова, А. А. Кацупеев, А. В. Сулыз. Новочеркасск: ЮРГПУ, 2020. 104 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/180938> (дата обращения 05.04.2023).

Дополнительная литература

1. Бизли Д. Python. Книга рецептов / Д. Бизли, Б. К. Джонс; перевод с английского Б. В. Уварова. М.: ДМК Пресс, 2019. 646 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131723> (дата обращения 05.04.2023).

2. Златопольский Д. М. Основы программирования на языке Python / Д. М. Златопольский. 2-ое изд., испр. и доп. М. : ДМК Пресс, 2018. 396 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131683> (дата обращения 05.04.2023).

3. Янцев В. В. Web-программирование на Python / В. В. Янцев. 2-е изд., стер. СПб.: Лань, 2023. 180 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/310289> (дата обращения 05.04.2023).

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ”
ИНСТИТУТ ДОПОЛНИТЕЛЬНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ**

Рабочая программа модуля 4

Основы объектно-ориентированного программирования

1. Область применения рабочей программы

Рабочая программа модуля 4 «Основы объектно-ориентированного программирования» (далее – рабочая программа) является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Python в области добывающей промышленности и природопользования» и направлена на формирование цифровых компетенций:

ID-20 Пишет и структурирует код на Python, используя устоявшиеся практики

ID-28 Применяет языки программирования для решения профессиональных задач

ID-57 Использует стандартную библиотеку языка Python

ID-307 Выявляет причинно-следственные связи при анализе информации

ID-375 Работает с программным обеспечением

Освоение рабочей программы является инвариантным (обязательным) для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем, часов
1.	Тема 1. Абстракция, классы и инкапсуляция Лекция: Основы объектно-ориентированного программирования. Абстракция и определение классов. Инкапсуляция, атрибуты и свойства. Свойства (поля, атрибуты) и методы класса. Классы и объекты. Аннотации свойств. Конструкторы. Статические поля и методы. Практическое занятие: Написание и отладка программ с использованием классов. Проектирование класса. Определение полей и методов класса. Создание экземпляров (объектов) класса. Использование ключевого слова self. Написание конструкторов. Использование аннотации свойств. Написание классов с вложенными классами. Работа со статическими полями и методами класса.	6
2	Тема 2. Наследование Лекция: Иерархия типов и наследование. Базовый класс (супер класс) и производный класс (подкласс). Переопределение функций базового класса. Особенности создания объектов производного класса. Проверка типа объекта (рефлексия). Множественное наследование. Практическое занятие: Проектирование иерархии классов, написание и отладка программ с использованием классов. Работа с объектами разных классов. Использование множественного наследования.	6
3	Тема 3. Полиморфизм подтипов Лекция: Понятие полиморфизма подтипов. Использование полиморфизма. Унификация работы с объектами разных типов. Класс	6

№ п/п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем, часов
	object. Строковое представление объекта. Практическое занятие: Написание и отладка программ с использованием базовых и производных классов. Работа с объектами разных классов (подтипов) как с объектами общего типа. Получение строкового представления объекта.	
4	Тема 4. Обработка исключений Возможные ошибки при работе программы. Исключительные ситуации и способы реакции на них. Принципы обработки исключений. Конструкция try...except...else...finally. Встроенные типы исключений Генерация исключения. Обработка исключений различных типов. Генерация исключений и оператор raise. Создание своих классов исключений. Практическое занятие: Написание программ с обработкой ошибок и исключений. Использование конструкции try...except...else...finally. Перехват исключения и получение информации о нем. Генерация исключений стандартных типов. Определение нового типа исключения, генерация и обработка пользовательских исключений. Анализ выполнения программ с управляемыми и неуправляемыми исключениями.	8
5	Самостоятельная работа	12
6	Промежуточная аттестация в формате: зачет	2

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов	
		практические занятия	самостоятельная работа
1	Тема 1. Абстракция, классы и инкапсуляция	6	2
2	Тема 2. Наследование	6	2
3	Тема 3. Полиморфизм подтипов	6	4
4	Тема 4. Обработка исключений	8	4
	Промежуточная аттестация	2	
Итого		40	

4. Контроль и оценка результатов освоения рабочей программы

Образовательная организация высшего образования, реализующая рабочую программу, обеспечивает организацию и проведение текущего и промежуточного контроля демонстрируемых обучающимися образовательных результатов.

Текущий контроль проводится преподавателем на основе оценивания результатов практических работ и самостоятельной работы обучающихся. Промежуточный контроль проводится в форме тестирования. Формы и методы текущего и промежуточного контроля, критерии оценивания

доводятся до сведения обучающихся в начале обучения.

Для текущего и промежуточного контроля создаются фонды оценочных средств (ФОС). ФОС включают в себя педагогические контрольно-измерительные материалы, предназначенные для определения соответствия (или несоответствия) индивидуальных образовательных достижений запланированным образовательным результатам.

4.1. Примеры оценочных средств

Примеры тестовых заданий к экзамену по дисциплине:

1. Что выведет данный код: `a = "P".format("y") print(a):`

- a. Py;
- b. y;
- c. Y;
- d. P.

2. Что выведет данный код: `b = "{2}{0}".format("p", "y", "p") print(b):`

- a. py;
- b. yp;
- c. yy;
- d. pp.

3. Что выведет данный код: `c = "{2}{0}".format(*"ypc") print(c):`

- a. ypc;
- b. pc;
- c. cy;
- d. yc.

4. Что выведет данный код: `d = "P{}".format("y") print(d):`

- a. Yu;
- b. Yp;
- c. Py;
- d. Pp.

5. Что выведет данный код: `nums = {"1": "78", "2": "24"} e = "{1} {2}".format(**nums) print(e):`

- a. 78 24;
- b. 24 78;
- c. ошибку.

6. Что выведет данный код: `try: print(1) except Exception: print(0):`

- a. 1;
- b. 0;
- c. ошибку.

7. Что выведет данный код: try: a = 2 + '1' print(a) except TypeError: print("Error"):

- a. 3;
- b. Error;
- c. ошибку.

8. Что выведет данный код: a = int("qwerty") print(a):

- a. 123456;
- b. ValueError;
- c. TypeError;
- d. qwerty.

9. Что выведет данный код: try: b = 1 / 0 except ZeroDivisionError: b = 0 print(b):

- a. 1;
- b. 0;
- c. ошибку.

10. Что выведет данный код: c = "2" - "1" print(c):

- a. 1;
- b. ValueError;
- c. TypeError.

11. Что выведет данный код: a = 10 print(b):

- a. 10;
- b. TypeError;
- c. ValueError;
- d. NameError.

12. Что выведет данный код: try: a = "10" print(a) except SyntaxError: print(1):

- a. 10;
- b. 1;
- c. SyntaxError;
- d. TypeError.

13. Что выведет данный код: try: c = 1 + "1" except ValueError: c = 0 print(c):

- a. 2;
- b. 0;
- c. TypeError;
- d. ValueError.

14. Что выведет данный код: a -= 1 print(a):

- a. -1;
- b. 0;
- c. TypeError;
- d. NameError.

15. Что выведет данный код: `try: e = "{0}".format(1) except ValueError:`

`e = "Error" print(e):`

- a. 1;
- b. Error;
- c. TypeError.

Примеры практико-ориентированных заданий (кейсов)

Практическое задание 1.

1. Напишите класс, чтобы найти правильность строки скобок, '(', ')', '{', '}', '[', 'и'].

Эти скобки должны быть закрыты в правильном порядке, например, ")" и ") [] {}" допустимы, но "[)", "({[})" и "{{{" недопустимы.

2. Напишите класс, чтобы получить все возможные уникальные подмножества из набора различных целых чисел.

3. Напишите класс с именем Circle, построенный по радиусу, и два метода, которые будут вычислять площадь и периметр круга.

4. Напишите класс для преобразования целого числа в римскую цифру.

Практическое задание 2.

1. Напишите программу с классом Student, в котором есть три атрибута: name,

groupNumber и age. По умолчанию name = Ivan, age = 18, groupNumber = 10A. Необходимо создать пять методов: getName, getAge, getGroupNumber, setNameAge, setGroupNumber. Метод getName нужен для получения данных об имени конкретного студента, метод getAge нужен для получения данных о возрасте конкретного студента, метод setGroupNumber нужен для получения данных о номере группы конкретного студента. Метод setNameAge позволяет изменить данные атрибутов установленных по умолчанию, метод setGroupNumber позволяет изменить номер группы установленный по умолчанию. В программе необходимо создать пять экземпляров класса Student, установить им разные имена, возраст и номер группы.

2. Напишите программу с классом Car. Создайте конструктор класса Car. Создайте атрибуты класса Car — color (цвет), type (тип), year (год). Напишите пять методов. Первый

— запуск автомобиля, при его вызове выводится сообщение «Автомобиль заведен». Второй

— отключение автомобиля — выводит сообщение «Автомобиль заглушен». Третий — присвоение автомобилю года выпуска. Четвертый метод — присвоение автомобилю типа.

Пятый — присвоение автомобилю цвета.

3. Разработать и протестировать классы. Задача выполняется в 3 этапа:

I. Написать класс «Компьютерная техника»:

- Поля: фирма-производитель, цена.
- Методы: конструктор с параметрами; функция print() для печати сведений об объекте.

II. Написать класс «Монитор», являющийся производным от класса «Компьютерная техника»:

- Поля: размер экрана, наличие встроенных динамиков.
- Методы: конструктор с параметрами; переопределение функции print() для печати сведений об объекте (в том числе данных из базового класса).

III. В тестирующей программе создается по 1 объекту ранее разработанных классов и печатаются сведения обо всех объектах.

Практическое задание 3.

Разработать и протестировать классы. Задача выполняется в 3 этапа:

I. Написать класс «Бытовая техника»:

- Поля: фирма-производитель, цена.
- Методы: конструктор с параметрами; функция print() для печати сведений об объекте.

II. Написать класс «Лампочка», являющийся производным от класса «Бытовая техника»:

- Поля: мощность, тип (лампа накаливания/люминисцентная/светодиодная).
- Методы: конструктор с параметрами; функция print() для печати сведений об объекте.

III. В тестирующей программе создается по 2 объекта ранее разработанных классов и печатаются сведения обо всех объектах.

Практическое задание 4.

Напишите программу, выполняющую перевод из буквенных оценок в числовые и обратно. Программа должна позволять пользователю вводить несколько значений для перевода – по одному в каждой строке. Для начала предпримите попытку сконвертировать введенное пользователем значение из числового в буквенное. Если возникнет исключение, попробуйте выполнить обратное преобразование – из буквенного в числовое. Если и эта попытка окончится неудачей, выведите предупреждение о том, что введенное значение не является допустимым. Пусть ваша программа конвертирует оценки до тех пор, пока пользователь не оставит ввод пустым.

5. Образцы учебно-методических материалов для обучающихся и

преподавателей

Для эффективного освоения компетенций, формируемых учебной дисциплиной важно использование в учебном процессе активных и интерактивных форм проведения занятий.

Изучение учебной дисциплины предполагает наличие аудиторной и самостоятельной видов работ слушателей. В ходе практических занятий рассматриваются бизнес-кейсы, практические задачи, наиболее сложные ситуации из практики с целью наиболее полного овладения умениями и навыками.

Лекции по учебной дисциплине призваны формировать знания, предусмотренные учебной программой, и включают теоретическую базу ведения бухгалтерского учета, на базе которой строятся прикладные аспекты.

Освоение дисциплины предполагает значительный объем самостоятельной внеаудиторной работы, которую слушатели должны выполнять как индивидуально, так и в малых группах.

Наряду с проработкой основной литературы (глав базового учебника) предусмотрено самостоятельное чтение дополнительной литературы (статей и других научных публикаций), а также проведение анализа кейсов, которые обсуждаются в ходе дискуссий на практических занятиях.

Практические занятия в малых группах и самостоятельная внеаудиторная работа направлены на выработку навыков экономического анализа деятельности предприятий и формирования профессиональных компетенций, установленных в соответствии с целями и задачами дисциплинами.

Для достижения поставленных целей преподавания дисциплины реализуются следующие средства, способы и организационные мероприятия:

- изучение теоретического материала дисциплины на лекции с использованием компьютерных технологий;
- самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, электронных библиотек, методических разработок, специальной и научной литературы;
- закрепление теоретического материала при проведении практических занятий с использованием учебного и научного оборудования, выполнения проблемно-ориентированных, поисковых, творческих заданий.

Самостоятельная работа слушателей включает:

1. Изучение учебной литературы по курсу.
2. Решение практических ситуаций и задач
3. Изучение источников управленческой информации
4. Работу с ресурсами Интернет
5. Решение практических ситуаций в виде творческих заданий

6. Изучение практических материалов деятельности конкретных предприятий

7. Изучение статистикой информации

8. Подготовку к зачету.

1) Цель самостоятельной работы - подготовка современного компетентного специалиста и формирование способностей и навыков к непрерывному самообразованию и профессиональному совершенствованию.

6. Кадровое обеспечение образовательного процесса

- | | |
|---|---|
| <p>1. Козин Евгений Сергеевич, к.т.н., кафедры сервиса автомобилей и технологических машин, научно-педагогический работник, ТИУ</p> | <p>Лекция 1 Абстракция, классы и инкапсуляция
Лекция 2 Наследование
Лекция 3 Полиморфизм подтипов
Лекция 4 Обработка исключений</p> |
| <p>2. Басинский Константин Юрьевич, доцент кафедры математики и прикладных информационных технологий, научно-педагогический работник, ТИУ</p> | <p>Практика 1. Абстракция, классы и инкапсуляция
Практика 2. Наследование
Практика 3. Полиморфизм подтипов
Практика 4. Обработка исключений</p> |

7. Материально-техническое обеспечение реализации рабочей программы

Аудитория	Назначение	Оборудование
509a(3)	Проведение практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной и итоговой аттестации, проведения практик	Учебная мебель для обучающихся, рабочее место преподавателя, доска, мультимедийное оборудование, персональные компьютеры в комплекте на основе операционных систем Microsoft Windows (или др.) с пакетом прикладных программ Microsoft Office (или др), с подключением к сети Интернет Вход на сайт ТИУ в Educon, программа для организации видеоконференций BigBlueButton

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Васильев А. Н. Python на примерах. Практический курс по программированию: учебное пособие / А. Н. Васильев. 3-е изд. СПб.: Наука и Техника, 2019. 432 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/139151> (дата обращения 05.04.2023).
2. Воробьев Г. А. Основы программирования на Python : учебно-методическое пособие / Г. А. Воробьев. Липецк: Липецкий ГПУ, 2022. 89 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/317075> (дата обращения 05.04.2023).
3. Кольцов Д. М. Справочник PYTHON. Кратко, быстро, под рукой: справочник / Д. М. Кольцов, Е. В. Дубовик. СПб.: Наука и Техника, 2021. 288 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/191480> (дата обращения 05.04.2023).
4. Никитина Т. П. Программирование. Основы Python для инженеров / Т. П. Никитина, Л. В. Королев. СПб.: Лань, 2023. 156 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/302720> (дата обращения 05.04.2023).
5. Шевченко Л. Г. Программирование на PYTHON в среде IDLE: учебное пособие / Л.Г. Шевченко, Т.В. Дружинина. Новосибирск: Изд-во НГТУ, 2020. 195 с.
6. Широбокова С. Н. Программирование на языке Python для лабораторных занятий: учебное пособие / С. Н. Широбокова, А. А. Кацупеев, А. В. Сулыз. Новочеркасск: ЮРГПУ, 2020. 104 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/180938> (дата обращения 05.04.2023).
7. Широков А. И. Алгоритмизация и программирование на языке «Питон» (Python): методические указания / А. И. Широков. М.: МИСИС, 2021. 48 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/238331> (дата обращения 05.04.2023).

Дополнительная литература

1. Бизли Д. Python. Книга рецептов / Д. Бизли, Б. К. Джонс; перевод с английского Б. В. Уварова. М.: ДМК Пресс, 2019. 646 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131723> (дата обращения 05.04.2023).
2. Златопольский Д. М. Основы программирования на языке Python / Д. М. Златопольский. 2-ое изд., испр. и доп. М. : ДМК Пресс, 2018. 396 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131683> (дата обращения 05.04.2023).

3. Янцев В. В. Web-программирование на Python / В. В. Янцев. 2-е изд., стер.

СПб.: Лань, 2023. 180 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: [https ://e.lanbook.com/book/310289](https://e.lanbook.com/book/310289) (дата обращения 05.04.2023).

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ”
ИНСТИТУТ ДОПОЛНИТЕЛЬНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ**

Рабочая программа модуля 5

**Модули и пакеты для реализации профессиональных задач
нефтегазодобывающих предприятий**

1. Область применения рабочей программы

Рабочая программа модуля 5 «Модули и пакеты для реализации профессиональных задач нефтегазодобывающих предприятий» программы профессиональной переподготовки (далее – рабочая программа) является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Python в области добывающей промышленности и природопользования» и направлена на формирование цифровых компетенций:

ID-20 Пишет и структурирует код на Python, используя устоявшиеся практики

ID-28 Применяет языки программирования для решения профессиональных задач

ID-57 Использует стандартную библиотеку языка Python

ID-307 Выявляет причинно-следственные связи при анализе информации

ID-375 Работает с программным обеспечением

Освоение рабочей программы является инвариантным (обязательным) для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем, часов
1.	Тема 1. Разработка модуля Лекция: Модули: назначение и правила создания. Особенности генерации и подключения модуля. Написание собственного модуля. Автоматическое документирование модуля. Импорт модуля и объектов из модуля. Использование программных объектов модуля. Практическое занятие: Формирование навыков оформления программ в виде модуля. Создание модуля. Подключение и использование модулей. Выгрузка модуля. Условная загрузка модуля. Написание программ, включающих библиотеку функций, классов, констант и формирование модулей. Написание программы, использующей множество модулей.	10
2	Тема 2. Создание пакета модулей Лекция: Пакеты: назначение и правила создания. Особенности организации и подключения пакета. Написание собственного пакета. Импорт пакета, модуля и объектов из модуля. Инициализация пакета. Использование программных объектов и модулей из пакета. Практическое занятие: Создание собственного пакета. Загрузка пакета и его элементов. Написание программы, использующей множество пакетов и модулей.	8

№ п/п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем, часов
3	Тема 3. Стандартные и распространяемые пакеты и модули Основные стандартные пакеты. Подсистема <code>pip</code> . Сторонние модули. Практическое занятие: Подключение стандартного пакета. Импорт объектов из пакета. Установка стороннего модуля с использованием подсистемы <code>pip</code> из репозитория и локального каталога. Написание программы с использованием собственных, стандартных и сторонних модулей и пакетов.	8
4	Самостоятельная работа	10
5	Промежуточная аттестация в формате: зачет	2

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов	
		практические занятия	самостоятельная работа
1	Тема 1. Разработка модуля	10	2
2	Тема 2. Создание пакета модулей	8	4
3	Тема 3. Стандартные и распространяемые пакеты и модули	8	4
4	Промежуточная аттестация	2	
Итого		38	

4. Контроль и оценка результатов освоения рабочей программы

Образовательная организация высшего образования, реализующая рабочую программу, обеспечивает организацию и проведение текущего и промежуточного контроля демонстрируемых обучающимися образовательных результатов.

Текущий контроль проводится преподавателем на основе оценивания результатов практических работ и самостоятельной работы обучающихся. Промежуточный контроль проводится в форме тестирования. Формы и методы текущего и промежуточного контроля, критерии оценивания доводятся до сведения обучающихся в начале обучения.

Для текущего и промежуточного контроля создаются фонды оценочных средств (ФОС). ФОС включают в себя педагогические контрольно-измерительные материалы, предназначенные для определения соответствия (или несоответствия) индивидуальных образовательных достижений запланированным образовательным результатам.

4.1. Примеры оценочных средств

Порядок промежуточной аттестации по дисциплине

Формой промежуточной аттестации по дисциплине является – зачет.

Зачет проводится в формате выполнения тестового задания.

Примеры тестовых заданий

1. Какая библиотека отвечает за время:

- a. time;
- b. localtime;
- c. Time;
- d. clock.

2. Сколько библиотек можно импортировать в один проект:

- a. неограниченное количество;
- b. не более 3;
- c. не более 10;
- d. не более 5;
- e. не более 23.

3. Для чего в пакетах модулей python в файле `__init__.py` служит список `__all__`:

- a. для конструкторов классов, как и всё, что связано с `__init__`;
- b. список определяет, что экспортировать, когда происходит импорт с помощью `from *`;
- c. для перечисления переменных, которые будут скрыты для импортирования.

**4. Что выведется на экран монитора в результате выполнения программы: входные данные: a=10, b=20, c=30 : `a=int(input('a= '))`
`b=int(input('b= '))` `c=int(input('c= '))`**

**`if a==b and b==c and c==a: print('треугольник равносторонний')`
`else: print('треугольник не равносторонний')`**

- a. равносторонний;
- b. равнобедренный;
- c. не равносторонний.

5. Имеется кортеж вида `T = (4, 2, 3)`. Какая из операций приведёт к тому, что имя `T` будет ссылаться на кортеж `(1, 2, 3)`:

- a. `T[0] = 1`;
- b. `T = (1) + T[1:]`;
- c. `T = (1,) + T[1:]`;
- d. `T.startswith(1)`.

6. Для чего в Python используется встроенная функция `enumerate()`:

- a. для определения количества элементов последовательности;
- b. для одновременного итерирования по самим элементам и их индексам;
- c. для сортировки элементов по значениям `id`.

7. Что выведет интерпретатор для следующей программы: `def get_name_and_decades(name, age): print(f"My name is {name} and I'm {age / 10:.5f} decades old.")` `get_name_and_decades("Leo", 31)`:

- a. my name is Leo and I'm 31.00000 decades old;
- b. my name is Leo and I'm 3.1 decades old;
- с. исключение: перед строкой стоит лишняя буква f;
- d. my name is Leo and I'm 3.10000 decades old;
- e. my name is {name} and I'm {age / 10:.5f} decades old.

51

8. Необходимо собрать и вывести все уникальные слова из строки рекламного текста. Какой из перечисленных типов данных Python подходит лучше всего:

- a. кортеж (tuple);
- b. список (list);
- с. множество (set);
- d. словарь (dict).

9. Учёт зверей в зоопарке ведётся с помощью приведённого ниже списка словарей.

Какая из строчек кода выведет структуру, отсортированную в порядке увеличения возрастов животных: `animals = [ {'type': 'penguin', 'name': 'Stephanie', 'age': 8}, {'type': 'elephant', 'name': 'Devon', 'age': 3}, {'type': 'puma', 'name': 'Moe', 'age': 5}]` :

- a. `sorted(animals, key='age');`
- b. ни один вариант не является верным, два словаря нельзя сравнивать друг
- с другом;
- c. `sorted(animals, key=lambda animal: animal['age']);`
- d. `sorted(animals).`

10. Какой результат выведет код: `def f(a, *pargs, **kargs):`

`print(a, pargs, kargs) f(1, 2, 3, x=4, y=5):`

- a. 1, 2, 3, {'x': 4, 'y': 5};
- b. 1 (2, 3) {'x': 4, 'y': 5};
- с. будет вызвано исключение, после двоеточия обязательно нужно перейти на новую строку;
- d. 1, 2, 3, 'x=4', 'y=5';
- e. 1, 2, 3, 4, 5.

Примеры практико-ориентированных заданий (кейсов)

Практическое задание 1.

1. Подмножеством элементов, или подсписком (sublist), мы будем называть список, являющийся составной частью большего списка. Подсписок может содержать один элемент, множество элементов, а также быть пустым. Например, [1], [2], [3] и [4] являются подсписками списка [1, 2, 3, 4]. Список [2, 3] также входит в состав [1, 2, 3, 4], но при этом список [2, 4] не является

подписком [1, 2, 3, 4], поскольку в исходном списке числа 2 и 4 не соседствуют друг с другом. Пустой список может быть рассмотрен как подписок для любого списка. Таким образом, список [] является подписком [1, 2, 3, 4]. Также список является подписком самого себя, то есть [1, 2, 3, 4] – это подписок для [1, 2, 3, 4].

В рамках данного упражнения вам необходимо написать функцию `isSublist`, определяющую, является ли один список подписком другого. На вход функции должны поступать два списка – `larger` и `smaller`. Функция должна возвращать значение `True` только в том случае, если список `smaller` является подписком списка `larger`. Напишите также основную программу для демонстрации работы функции. Оформите решение в виде модуля.

2. Напишите функцию, возвращающую список, содержащий все возможные подписки заданного. Например, в число подписков списка [1, 2, 3] входят следующие: [],

[1], [2], [3], [1, 2], [2, 3] и [1, 2, 3]. Заметьте, что ваша функция должна вернуть как минимум один пустой список, гарантированно являющийся подписком для любого списка.

Напишите основную программу, демонстрирующую работу функции применительно к нескольким исходным спискам. Оформите решение в виде модуля.

Практическое задание 2.

1. Как вы знаете, в языке Python для создания комментариев в коде используется символ `#`. Комментарий начинается с этого символа и продолжается до конца строки – без возможности остановить его раньше. В данном упражнении вам предстоит написать программу, которая будет удалять все комментарии из исходного файла с кодом на языке Python. Пройдите по всем строкам в файле на предмет поиска символа `#`. Обнаружив его, программа должна удалить все содержимое, начиная с этого символа и до конца строки.

Для простоты не будем рассматривать ситуации, когда знак решетки встречается в середине строки. Сохраните новое содержимое в созданном файле. Имена файла источника и файла назначения должны быть запрошены у пользователя. Удостоверьтесь в том, что программа корректно обрабатывает возможные ошибки при работе с обоими файлами. Оформите решение в виде пакета из одного модуля.

2. При написании функций хорошей практикой считается предварение ее блоком комментариев с описанием назначения функции, ее входных параметров и возвращаемого значения. Но некоторые разработчики просто не пишут комментарии к своим функциям.

Другие честно собираются написать их когда-нибудь в будущем, но руки так и не доходят.

Напишите программу, которая будет проходить по файлу с исходным кодом на Python и искать функции, не снабженные блоком комментариев. Можно принять за аксиому, что строка, начинающаяся со слова `def`, следом за которым идет пробел, будет считаться началом функции. И если функция документирована, предшествующая строчка должна начинаться со знака `#`. Перечислите названия всех функций, не снабженных комментариями, вместе с именем файла и номером строки, с которой начинается объявление функции. Одно или несколько имен файлов с кодом на языке Python пользователь должен передать в функцию в качестве аргументов командной строки. Для файлов, которые не существуют или не могут быть открыты, должны выдаваться соответствующие предупреждения, после чего должна быть продолжена обработка остальных файлов. Дополните ранее созданный пакет еще одним модулем и протестируйте все возможности пакета.

Практическое задание 3.

1. Создание пароля посредством генерирования случайных символов может обернуться сложностью в запоминании полученной относительно надежной последовательности. Некоторые системы создания паролей рекомендуют сцеплять вместе два слова на английском языке, тем самым упрощая запоминание заветного ряда символов – правда, в ущерб его надежности. Напишите программу, которая будет открывать файл со списком слов, случайным образом выбирать два из них и сцеплять вместе для получения итогового пароля. При создании пароля исходите из следующего требования: он должен состоять минимум из восьми символов и максимум из десяти, а каждое из используемых слов должно быть длиной хотя бы в три буквы. Кроме того, сделайте заглавными первые буквы обоих слов, чтобы легко можно было понять, где заканчивается одно и начинается другое. По завершении процесса полученный пароль должен быть отображен на экране.

2. Проверка орфографии – лишь составная часть расширенного текстового анализа на предмет наличия ошибок. Одной из самых распространенных ошибок в текстах является повторение слов. Например, автор может по ошибке дважды подряд написать одно слово, как в следующем примере:

At least one value must be entered entered in order to compute the average.

Некоторые текстовые процессоры умеют распознавать такой вид ошибок при выполнении текстового анализа.

В данном упражнении вам предстоит написать программу для определения наличия дублей слов в тексте. При нахождении повтора на экран должен выводиться номер строки и дублирующееся слово. Удостоверьтесь, что программа корректно обрабатывает случаи, когда повторяющиеся слова находятся на разных строках, как в предыдущем примере. Имя файла для анализа должно быть передано программе в качестве единственного аргумента

командной строки. При отсутствии аргумента или невозможности открыть указанный файл на экране должно появляться соответствующее сообщение об ошибке.

5. Образцы учебно-методических материалов для обучающихся и преподавателей

Для эффективного освоения компетенций, формируемых учебной дисциплиной важно использование в учебном процессе активных и интерактивных форм проведения занятий.

Изучение учебной дисциплины предполагает наличие аудиторной и самостоятельной видов работ слушателей. В ходе практических занятий рассматриваются бизнес-кейсы, практические задачи, наиболее сложные ситуации из практики с целью наиболее полного овладения умениями и навыками.

Лекции по учебной дисциплине призваны формировать знания, предусмотренные учебной программой, и включают теоретическую базу ведения бухгалтерского учета, на базе которой строятся прикладные аспекты.

Освоение дисциплины предполагает значительный объем самостоятельной внеаудиторной работы, которую слушатели должны выполнять как индивидуально, так и в малых группах.

Наряду с проработкой основной литературы (глав базового учебника) предусмотрено самостоятельное чтение дополнительной литературы (статей и других научных публикаций), а также проведение анализа кейсов, которые обсуждаются в ходе дискуссий на практических занятиях.

Практические занятия в малых группах и самостоятельная внеаудиторная работа направлены на выработку навыков экономического анализа деятельности предприятий и формирования профессиональных компетенций, установленных в соответствии с целями и задачами дисциплинами.

Для достижения поставленных целей преподавания дисциплины реализуются следующие средства, способы и организационные мероприятия:

- изучение теоретического материала дисциплины на лекции с использованием компьютерных технологий;
- самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, электронных библиотек, методических разработок, специальной и научной литературы;
- закрепление теоретического материала при проведении практических занятий с использованием учебного и научного оборудования, выполнения проблемно-ориентированных, поисковых, творческих заданий.

Самостоятельная работа слушателей включает:

1. Изучение учебной литературы по курсу.
2. Решение практических ситуаций и задач

3. Изучение источников управленческой информации
4. Работу с ресурсами Интернет
5. Решение практических ситуаций в виде творческих заданий
6. Изучение практических материалов деятельности конкретных предприятий
7. Изучение статистикой информации
8. Подготовку к зачету.

Цель самостоятельной работы - подготовка современного компетентного специалиста и формирование способностей и навыков к непрерывному самообразованию и профессиональному совершенствованию.

6. Кадровое обеспечение образовательного процесса

- | | |
|---|---|
| 1. Спирин Игорь Сергеевич,
доцент кафедры математики
и прикладных
информационных
технологий, научно-
педагогический работник,
ТИУ | Лекция 1 Разработка модуля
Лекция 2 Создание пакета модулей
Лекция 3 Стандартные и распространяемые
пакеты и модули

Практика 1. Разработка модуля
Практика 2. Создание пакета модулей
Практика 3. Стандартные и
распространяемые пакеты и модули |
|---|---|

7. Материально-техническое обеспечение реализации рабочей программы

Аудитория	Назначение	Оборудование
509a(3)	Проведение практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной и итоговой аттестации, проведения практик	Учебная мебель для обучающихся, рабочее место преподавателя, доска, мультимедийное оборудование, персональные компьютеры в комплекте на основе операционных систем Microsoft Windows (или др.) с пакетом прикладных программ Microsoft Office (или др), с подключением к сети Интернет Вход на сайт ТИУ в Educon, программа для организации видеоконференций BigBlueButton

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Васильев А. Н. Python на примерах. Практический курс по программированию: учебное пособие / А. Н. Васильев. 3-е изд. СПб. : Наука и

Техника, 2019. 432 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступ: <https://e.lanbook.com/book/139151> (дата обращения 05.04.2023).

2. Воробьев Г. А. Основы программирования на Python : учебно-методическое пособие / Г. А. Воробьев. Липецк: Липецкий ГПУ, 2022. 89 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/317075> (дата обращения 05.04.2023).

3. Кольцов Д. М. Справочник PYTHON. Кратко, быстро, под рукой: справочник /Д. М. Кольцов, Е. В. Дубовик. СПб.: Наука и Техника, 2021. 288 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/191480> (дата обращения 05.04.2023).

4. Никитина Т. П. Программирование. Основы Python для инженеров / Т. П. Никитина, Л. В. Королев. СПб.: Лань, 2023. 156 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/302720> (дата обращения 05.04.2023).

5. Шевченко Л. Г. Программирование на PYTHON в среде IDLE: учебное пособие /Л.Г. Шевченко, Т.В. Дружинина. Новосибирск: Изд-во НГТУ, 2020. 195 с.

6. Широбокова С. Н. Программирование на языке Python для лабораторных занятий: учебное пособие / С. Н. Широбокова, А. А. Кацупеев, А. В. Сулыз. Новочеркасск: ЮРГПУ, 2020. 104 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/180938> (дата обращения 05.04.2023).

7. Широков А. И. Алгоритмизация и программирование на языке «Питон» (Python) : методические указания / А. И. Широков. М.: МИСИС, 2021. 48 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/238331> (дата обращения 05.04.2023).

Дополнительная литература

1. Бизли Д. Python. Книга рецептов / Д. Бизли, Б. К. Джонс; перевод с английского Б. В. Уварова. М.: ДМК Пресс, 2019. 646 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131723> (дата обращения 05.04.2023).

2. Златопольский Д. М. Основы программирования на языке Python /Д. М. Златопольский. 2-ое изд., испр. и доп. М. : ДМК Пресс, 2018. 396 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131683> (дата обращения 05.04.2023).

3. Янцев В. В. Web-программирование на Python / В. В. Янцев. 2-е изд., стер.

СПб.: Лань, 2023. 180 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/310289> (дата обращения 05.04.2023).

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ”
ИНСТИТУТ ДОПОЛНИТЕЛЬНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ**

Рабочая программа модуля 5

**Практики/ стажировки программы профессиональной
переподготовки**

1. Область применения рабочей программы

Рабочая программа модуля 5 Практики/ стажировки программы профессиональной переподготовки (далее – рабочая программа) является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Python в области добывающей промышленности и природопользования» и направлена на формирование цифровых компетенций:

ID-20 Пишет и структурирует код на Python, используя устоявшиеся практики

ID-28 Применяет языки программирования для решения профессиональных задач

ID-57 Использует стандартную библиотеку языка Python

ID-307 Выявляет причинно-следственные связи при анализе информации

ID-375 Работает с программным обеспечением

Освоение рабочей программы является инвариантным (обязательным) для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование тем, виды учебной работы и краткое содержание учебного материала	Объем часов
1	Развертывание среды программирования Python на персональном компьютере	5
2	Создание и организация файлов проекта	5
3	Разработка бизнес-требований и технического задания	20
4	Создание, запуск и отладка программы на языке Python	30
5	Презентация проекта и защита у Заказчика	10

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов	
		практические занятия	самостоятельная работа
1	Развертывание среды программирования Python на персональном компьютере	5	-
2	Создание и организация файлов проекта	5	-
3	Разработка бизнес-требований и технического задания	20	-
4	Создание, запуск и отладка программы на языке Python	10	-
5	Презентация проекта и защита у Заказчика	10	-
	Итого	70	

4. Контроль и оценка результатов освоения рабочей программы

Текущий контроль проводится преподавателем на основе оценивания результатов этапов выполнения проекта обучающимися. Итоговая оценка выставляется по результатам защиты проекта.

4.1. Примеры оценочных средств

Технический вывод интерпретатора

Стабильность системы при корректных действиях пользователя

Устойчивость системы при некорректных действиях пользователя

Проверка структуры и состава данных, подвергшихся обработке

Сравнение результата с эталонным ответом

Соответствие результата техническому заданию

Анкеты и опросники

4.2. Защита проекта

При оценивании проекта руководитель практики проверяет соответствие требованиям по оформлению и соответствие содержания индивидуальному заданию на практику.

В процессе защиты проекта должны быть:

- озвучены цель и задачи практики,
- организация места прохождения практики,
- кратко освещены основные профессиональные действия, которые выполнял слушатель или принимал участие их в проведении,
- сделаны выводы о том, какие профессиональные навыки приобретены в процессе прохождения практики,
- наличие всех требуемых разделов.

Критерии оценки промежуточной аттестации по практике (зачет с оценкой)

По результатам прохождения промежуточной аттестации по практике выставляются оценки по четырехбалльной системе («отлично», «хорошо», «удовлетворительно», «неудовлетворительно») с учетом следующих критериев:

«отлично» – цель практики выполнена полностью, слушатель аргументированно и убедительно прокомментировал проект. Проект представлен в срок, отчет полностью соответствуют требованиям.

«хорошо» – цель практики выполнена почти полностью: кратко представлены некоторые примеры и результаты деятельности. Слушатель убедительно и уверенно прокомментировал проект. Проект представлен в срок, однако в представленном проекте имеются несущественные дефекты.

«удовлетворительно» – цель практики выполнена частично: недостаточно отработаны и применены на практике профессиональные компетенции. Слушатель отвечал неполно, неуверенно прокомментировал

проект. Проект представлен в срок, однако в представленном проекте имеются существенные дефекты.

«неудовлетворительно» – слушатель не представил проект, не прошел практику по неуважительной причине, проект представлен в срок, однако является неполным, примеры и результаты деятельности отсутствуют, что свидетельствует о несформированности у слушателя надлежащих компетенций.

5. Перечень предприятий для прохождения практики/ стажировки

№ п/п	Наименование предприятий	Вид деятельности
1	ООО «Газпром Добыча Надым»	Добыча и подготовка газа и газового конденсата
2	ООО «Газпромнефть Заполярье»	Дочернее предприятие «Газпром нефти», создано для реализации крупных проектов в области добычи
3	ООО «МНП ГЕОДАТА»	Работы геолого-разведочные, геофизические и геохимические в области изучения недр и воспроизводства минерально-сырьевой базы
4	АО «Синефтемаш»	Выпуск оборудования для бурения нефтегазовых скважи
5	ООО «Газпром межрегионгаз Север»	Ресурсоснабжающая организация

6. Условия реализации рабочей программы практики/ стажировки

6.1 Требования к минимальному материально-техническому обеспечению

Реализация программы практики требует наличие учебного кабинета
Оборудование учебного кабинета:

- посадочные места по количеству обучающихся;
- рабочее место преподавателя;
- комплект учебно – методической документации по практике.

Технические средства обучения:

- компьютер с лицензионным программным обеспечением

7. Кадровое обеспечение образовательного процесса

к.т.н., доцент ТИУ

Базанов Артем Владимирович

к.п.н. доцент ТИУ

Спирин Игорь Сергеевич

к.т.н. программист ТННЦ

Ахмадулин Руслан Камильевич

8. Материально-техническое обеспечение реализации рабочей программы

Аудитория	Назначение	Оборудование
509a(3)	Проведение практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной и итоговой аттестации, проведения практик	Учебная мебель для обучающихся, рабочее место преподавателя, доска, мультимедийное оборудование, персональные компьютеры в комплекте на основе операционных систем Microsoft Windows (или др.) с пакетом прикладных программ Microsoft Office (или др), с подключением к сети Интернет Вход на сайт ТИУ в Educon, программа для организации видеоконференций BigBlueButton

9. Информационное обеспечение реализации рабочей программы

Основная литература

1. Васильев А. Н. Python на примерах. Практический курс по программированию: учебное пособие / А. Н. Васильев. 3-е изд. СПб. : Наука и Техника, 2019. 432 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступ: <https://e.lanbook.com/book/139151> (дата обращения 05.04.2023).

2. Воробьев Г. А. Основы программирования на Python : учебно-методическое пособие / Г. А. Воробьев. Липецк: Липецкий ГПУ, 2022. 89 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/317075> (дата обращения 05.04.2023).

3. Кольцов Д. М. Справочник PYTHON. Кратко, быстро, под рукой: справочник /Д. М. Кольцов, Е. В. Дубовик. СПб.: Наука и Техника, 2021. 288 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/191480> (дата обращения 05.04.2023).

4. Никитина Т. П. Программирование. Основы Python для инженеров / Т. П. Никитина, Л. В. Королев. СПб.: Лань, 2023. 156 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/302720> (дата обращения 05.04.2023).

5. Шевченко Л. Г. Программирование на PYTHON в среде IDLE: учебное пособие /Л.Г. Шевченко, Т.В. Дружинина. Новосибирск: Изд-во НГТУ, 2020. 195 с.

6. Широбокова С. Н. Программирование на языке Python для лабораторных занятий: учебное пособие / С. Н. Широбокова, А. А. Кацупеев, А. В. Сулыз. Новочеркасск: ЮРГПУ, 2020. 104 с. // Лань: электронно-

библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/180938> (дата обращения 05.04.2023).

7. Широков А. И. Алгоритмизация и программирование на языке «Питон» (Python) : методические указания / А. И. Широков. М.: МИСИС, 2021. 48 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/238331> (дата обращения 05.04.2023).

Дополнительная литература

1. Бизли Д. Python. Книга рецептов / Д. Бизли, Б. К. Джонс; перевод с английского Б. В. Уварова. М.: ДМК Пресс, 2019. 646 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131723> (дата обращения 05.04.2023).

2. Златопольский Д. М. Основы программирования на языке Python / Д. М. Златопольский. 2-ое изд., испр. и доп. М. : ДМК Пресс, 2018. 396 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/131683> (дата обращения 05.04.2023).

3. Янцев В. В. Web-программирование на Python / В. В. Янцев. 2-е изд., стер.

СПб.: Лань, 2023. 180 с. // Лань: электронно-библиотечная система. [Электронный ресурс]. Режим доступа: <https://e.lanbook.com/book/310289> (дата обращения 05.04.2023).

VI. Аттестация по Программе

После завершения обучения по Программе и прохождения итоговой оценки сформированности цифровых компетенций обучающиеся допускаются к итоговой аттестации.

Итоговая аттестация проводится с участием представителей профильных индустриальных партнёров в форме демонстрационного экзамена и предусматривает выполнение обучающимся профессиональных задач и оценку результатов и/или процесса выполнения – проверку сформированности в рамках Программы цифровых компетенций.

Задания демонстрационного экзамена разрабатываются с участием организаций-работодателей, отраслевых партнёров и профессиональных сообществ. Демонстрационный экзамен должен предусматривать выполнение (демонстрацию) обучающимся деятельности, завершающейся получением результата (продукта или его элемента), значимого при выполнении трудовой функции или трудовых действий.

Для обеспечения организации и проведения итоговой аттестации разрабатывается положение об итоговой аттестации, регулирующее требования к выполнению, оформлению и оцениванию работ, заданий, условия проведения итоговой аттестации, требования к составу

аттестационной комиссии. Состав комиссии, перечень тем итоговых аттестационных работ, портфолио, практических заданий и требований к выполнению разрабатывается и актуализируется при участии промышленных партнёров.

Примеры тем и заданий для демонстрационного экзамена

1. Интеграция HTTP API или GRPC API модуля VMS.
2. Проектирование инфраструктуры для авто-тестирования VMS.
3. Специальный калькулятор (например, по генетике).
4. Сериализация/десериализация файлов конфигурации в формате XML.
5. Массовое переименование медиафайлов.
6. Парсер текста (например, в формате XML или JSON).
7. Автоматизация установки приложения.
8. Лексикографический анализ лог-файлов.
9. Работа с потоковым видео на базе ffmpeg-python.
10. Обработка PDF-файлов.
11. Администрирование базы госучреждений или органов государственной власти.
12. Анализ результатов спортивных соревнований.
13. Работа с базой событий VMS на основе psycorg2.
14. Редактирование и анализ XLSX-файлов.
15. Эмуляция API IP-камер для внутреннего тестирования.
16. Разработка скриптов "кусочной" автоматизации в экосистеме заказчика.
17. Взаимодействие с электронной почтой.

Критерии оценки:

Отлично: 85-100

Хорошо: 70-84

Удовлетворительно: 56-69

Неудовлетворительно: 0-55

Качество доклада:

0 - Композиция доклада не выстроена, работа и результаты, не представлены в полном объёме;

5 - Композиция доклада выстроена; работа и её результаты представлены, но не в полном объёме;

10 - Выстроена композиция доклада, в нём в полном объёме представлена работа и её результаты; основные позиции проекта аргументированы; убедительность речи и убеждённость оратора.

Ответы на вопросы:

0 - Не даёт ответа на заданные вопросы;

5 - Ответы на вопросы не полные, нет убедительности, отсутствуют аргументы;

10 - Докладчик убедителен, даёт полные, аргументированные ответы на вопросы, стремится использовать ответы для раскрытия темы и сильных сторон работы.

Качество презентации:

0 – Презентация не отражает сути проекта;

5 – Презентация частично отражает основные положения проекта;

10 – Презентация полностью отражает цели, инструменты и результат проекта.

VII. Завершение обучения по Программе

Лицам, завершившим обучение по Программе и достигших целевого уровня сформированности цифровых компетенций по результатам итоговой оценки и прошедших итоговую аттестацию, присваивается дополнительная ИТ-квалификации «Программист», установленная Программой.

При освоении Программы параллельно с получением высшего образования диплом о профессиональной переподготовке выдается не ранее получения соответствующего документа об образовании и о квалификации (за исключением лиц, имеющих среднее профессиональное или высшее образование).

Лицам, не прошедшим итоговую аттестацию или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть Программы и (или) отчисленным из образовательной организации высшего образования, реализующей Программу, выдается справка об обучении или о периоде обучения по образцу, самостоятельно устанавливаемому образовательной организацией высшего образования.