Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Тлисов Азамат Борисович
Должность: Директор Северо-Кавказского института - филиала РАНХиГС
Дата подписания: 17.11.2026 21:28:15
Уникальный программный ключ:
ac2df3a37f30cf89983b8fac4daa7e874e1d10ed

Федеральное государственное бюджетное образовательное учреждение высшего образования
«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»

СЕВЕРО-КАВКАЗСКИЙ ИНСТИТУТ-ФИЛИАЛ

«УТВЕРЖДАЮ»

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат:
00D6F63EBCD06169909268096273 7E91ED
Владелец: Тлисов Азамат Борисович
Действителен: с 28.04.2025 до 22.07.2026

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА

повышения квалификации

«Программирование»

Пятигорск, 2026

Разработчики:

Кандидат экономических наук, доцент,
декан факультета экономики, управления и
права _____ Е.А. Акопян

Руководитель программы: _____ Е.А. Акопян

Программа повышения квалификации рассмотрена на заседании ученого совета Северо-Кавказского института-филиала и рекомендована к реализации, протокол № 5 от «18» 02 2026 г.

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ 4
1.1. Цель реализации 4
1.2. Нормативные правовые акты 4
1.3. Планируемые результаты обучения 5
1.4. Категория слушателей 5
1.5. Формы обучения и технологии обучения 6
1.6. Период обучения, срок освоения и режим занятий 6
1.7. Документ о квалификации 7
2. СОДЕРЖАНИЕ ПРОГРАММЫ 7
2.1. Календарный учебный график 7
2.2. Учебный план 8
2.2. Содержание программы по модулям/разделам/дисциплинам/ и (или) темам.. 10
3. ОРГАНИЗАЦИОННЫЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ 11
3.1. Материально-техническое и программное обеспечение реализации. программы 11
3.2. Учебно-методическое и информационное обеспечение программы 11
4. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ 19

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Цель реализации

Программа повышения квалификации «Программирование» является приобретение системных знаний для выполнения нового вида профессиональной деятельности в сфере написания программного кода с использованием языков программирования, определения и манипулирования данными. Программа повышения квалификации разработана с учетом потребностей широкого круга слушателей с различным базовым образованием, заинтересованных в освоении процесса создания компьютерных программ в условиях часто меняющихся требований к специалистам на рынке труда.

1.2. Нормативные правовые акты

Дополнительная профессиональная программа повышения квалификации разрабатывалась на основании следующих нормативных документов:

1. Федеральный закон от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации».
2. Приказ Минобрнауки России от 24.03.2025 N 266 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам" (Зарегистрировано в Минюсте России 22.04.2025 N 81928).
3. «Общероссийский классификатор видов экономической деятельности» (утв. Приказом Росстандарта от 31.01.2014 N 14-ст) (ред. от 11.09.2025)
4. Приказ Минздравсоцразвития РФ от 26.08.2010 №761н (ред. от 31 мая 2011) «Об утверждении Единого квалификационного справочника должностей руководителей, специалистов и служащих, раздел «Квалификационные характеристики должностей работников образования».
5. Постановление Минтруда России от 21 августа 1998 №37 (ред. от 27 марта 2018) «Квалификационный справочник должностей руководителей, специалистов и других служащих».
6. Методические рекомендации по использованию электронного обучения, дистанционных образовательных технологий при реализации дополнительных профессиональных образовательных программ Министерства образования и науки Российской Федерации от 10 апреля 2014 №06–381
7. Методические рекомендации по разработке основных профессиональных образовательных программ дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов (утв. Минобрнауки России 22 января 2015 №ДЛ-1/05вн).
8. Письмо Минобрнауки России от 21 апреля 2015 №ВК-1013/06 «Методические рекомендации по реализации дополнительных профессиональных программ с использованием дистанционных образовательных технологий, электронного обучения и в сетевой форме».
9. Постановление Правительства РФ от 11 октября 2023 г. N 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».
10. Приказ РАНХиГС от 22 сентября 2017 №01–6230 «Об утверждении положения о применении в Академии электронного обучения. Дистанционных образовательных технологий при реализации образовательных программ».
11. Приказ РАНХиГС от 02 декабря 2025 г. №02– 02669/001 «Об утверждении порядка разработки и утверждения в Академии дополнительных профессиональных программ – программ повышения квалификации, программ профессиональной переподготовки».
12. Приказ Министерства труда и социальной защиты РФ от 20 июля 2022 г. N 424н «Об утверждении профессионального стандарта «Программист»// Гарант: справочно-правовая система [Офиц. сайт]. URL: https://www.garant.ru/products/ipo/prime/doc/405083219/.

1.3. Планируемые результаты обучения

Таблица 1

Перечень профессиональных компетенций в рамках имеющейся квалификации и профессиональных компетенций, планируемых к освоению (результаты обучения)[1]

Виды деятельности	Общепрофессиональные /профессиональные компетенции ОПК, ПК или трудовые функции (ПСК и СК) (формируются и (или) совершенствуются)	Практический опыт	Умения	Знания
Разработка компьютерного программного обеспечения	ПКС-1 (A/02.3) Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных	Владеет навыками программирования на языке Python; навыками работы с разными форматами файлов с данными; навыками поиска необходимых для программирования на языке Python данных; навыками сбора и обработки данных на языке программирования Python.	Умеет применять язык программирования Python для написания программного кода; использовать возможности имеющейся технической и/ или программной архитектуры; использовать среду программирования Python; использовать средства системы управления базами данных.	Знает синтаксис языка программирования Python, особенности программирования на этом языке, стандартные библиотеки языка программирования; технологии программирования; компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними; особенности среды программирования Python; особенности системы управления базами данных.
	ПКС-2 (A/05.3) Проверка и отладка программного кода	Владеет навыками отладки программного кода.	Умеет интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов.	Знает методы и приемы отладки программного кода, типы и форматы сообщений об ошибках, предупреждений.

Приказ Министерства труда и социальной защиты РФ от 20 июля 2022 г. N 424н «Об утверждении профессионального стандарта «Программист»// Гарант: справочно-правовая система [Офиц. сайт]. URL: https://www.garant.ru/products/ipo/prime/doc/405083219/

1.4. Категория слушателей

Уровень образования: среднее профессиональное образование - программы подготовки специалистов среднего звена.

В том числе:

а) граждане в возрасте 50 лет и старше, граждане предпенсионного возраста;

б) граждане, фактически осуществляющие уход за ребенком и находящиеся в отпуске по уходу за ребенком до достижения им возраста 3 лет;

[1] Приказ Министерства труда и социальной защиты РФ от 20 июля 2022 г. N 424н «Об утверждении профессионального стандарта «Программист»// Гарант: справочно-правовая система [Офиц. сайт]. URL: https://www.garant.ru/products/ipo/prime/doc/405083219/

в) женщины, не состоящие в трудовых отношениях и имеющие детей дошкольного возраста в возрасте от 0 до 7 лет включительно;

г) инвалиды;

д) граждане, обратившиеся в органы службы занятости в целях поиска работы;

е) безработные граждане, зарегистрированные в органах службы занятости;

ж) работники, находящиеся под риском увольнения, включая введение режима неполного рабочего времени, простой, временную приостановку работ, предоставление отпусков без сохранения заработной платы, проведение мероприятий по высвобождению работников;

з) граждане Украины и лица без гражданства, постоянно проживающие на территории Украины, которые получили удостоверение беженца или свидетельство о предоставлении временного убежища на территории Российской Федерации;

и) ветераны боевых действий, принимавшие участие (содействовавшие выполнению задач) в специальной военной операции на территориях Донецкой Народной Республики, Луганской Народной Республики и Украины с 24 февраля 2022 г., на территориях Запорожской области и Херсонской области с 30 сентября 2022 г., уволенные с военной службы (службы, работы);

к) лица, принимавшие в соответствии с решениями органов публичной власти Донецкой Народной Республики, Луганской Народной Республики участие в боевых действиях в составе Вооруженных Сил Донецкой Народной Республики, Народной милиции Луганской Народной Республики, воинских формирований и органов Донецкой Народной Республики и Луганской Народной Республики начиная с 11 мая 2014 г.;

л) члены семей лиц, указанных в подпунктах "и" и "к" настоящего пункта, погибших (умерших) при выполнении задач в ходе специальной военной операции (боевых действий), члены семей лиц, указанных в подпунктах "и" и "к" настоящего пункта, умерших после увольнения с военной службы (службы, работы), если смерть таких лиц наступила вследствие увечья (ранения, травмы, контузии) или заболевания, полученного ими при выполнении задач в ходе специальной военной операции (боевых действий);

м) молодежь в возрасте до 35 лет включительно, относящаяся к категориям:

граждан, которые со дня окончания военной службы по призыву не являются занятыми в соответствии с законодательством о занятости населения в течение 4 месяцев и более;

граждан, которые со дня выдачи им документа об образовании и (или) о квалификации не являются занятыми в соответствии с законодательством о занятости населения в течение 4 месяцев и более;

граждан, находящихся под риском увольнения (планируемых к увольнению в связи с ликвидацией организации либо прекращением деятельности индивидуальным предпринимателем, сокращением численности или штата работников организации, индивидуального предпринимателя и возможным расторжением трудовых договоров);

граждан, завершающих обучение по образовательным программам среднего профессионального или высшего образования в текущем календарном году (за исключением получивших грант на обучение или обучающихся по договорам о целевом обучении), обратившихся в органы службы занятости, для которых отсутствует подходящая работа по получаемой профессии (специальности) с уровнем образования СПО, бакалавр, специалист, магистр.

1.5. Формы обучения и технологии обучения

Форма обучения очно-заочная с применением ЭО и ДОТ.

1.6. Период обучения, срок освоения и режим занятий

Продолжительность обучения составляет 12 дней очно (с применением ЭО и ДОТ) по 6 часов в день (по утвержденному графику).

Общая трудоемкость дисциплин (модулей) – 72 ак. часа (с применением ЭО и ДОТ); количество академических часов, выделенных на лекционную часть – 14 ак. часов; на практическую работу слушателей – 20 ак. часов (с применением ДОТ); самостоятельную работу – 36 ак. часов, Итоговая аттестация проводится в форме итогового зачета (тестирование), в соответствии с учебным планом – 2 ак. часа с применением ЭО.

1.7. Документ о квалификации

Удостоверение о повышении квалификации федерального государственного бюджетного образовательного учреждения высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации», установленного образца.

2. СОДЕРЖАНИЕ ПРОГРАММЫ

2.1. Календарный учебный график

Таблица 2

Период обучения – 12 дней					
1 день	2 день	3 день	4 день	5 день	6 день
УЗ ДОТ, СР	УЗ ДОТ, СР	УЗ ДОТ, СР	УЗ ДОТ, СР	УЗ ДОТ, СР	УЗ ДОТ, СР
7 день	8 ень	9 день	10 день	11 день	12 день
УЗ ДОТ, СР	УЗ ДОТ, СР	УЗ ДОТ, СР	УЗ ДОТ, СР	УЗ ДОТ, СР	УЗ ДОТ, ИА ЭО

Условные обозначения: УЗ ДОТ– учебные занятия с применением дистанционных образовательных технологий; СР – самостоятельная работа; ИА ЭО – итоговая аттестация с применением электронного обучения.

2.2. Учебный план

Таблица 3

Учебный план

№п/п[1]	Наименование раздела, модуля, дисциплины, темы, практики, стажировки2	Общая трудоемкость, час.[3]	Контактная работа, час.[4]					Самостоятельная работа, час[7]	Контактная работа (с применением дистанционных образовательных технологий, электронного обучения), час.[6]					Самостоятельная работа (в т.ч. электронное обучение (ЭО), час	Текущий контроль успеваемости	Промежуточная аттестация (	Итоговая аттестация (вид /час.)	Код компетенции
			Всего[4]	В том числе					Всего[4]	В том числе								
				Лекции / в интерактивной форме5	Практические (семинарские/лаборато рные) занятия /в интерактивной форме	Контактная самостоятельная работа, час	Индивидуальные и групповые консультации			Лекции/ в интерактивной форме	Практические /лаборат /1 семинарские занятия интерактивной форме	Контактная самостоятельная работа, час	Индивидуальные и групповые консультации					
1.	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	Введение в Python. Базовые инструменты разработчика. Переменные.	**4**							**2**	2				2				ПКС-1 ПКС-2
2	Простые диалоговые программы. Консольный ввод и вывод данных.	**10**							**6**	2		4		4				ПКС-1 ПКС-2
3	Синтаксис языка. Типы данных, структуры данных: множества, словари, списки, кортежи. Стандартная библиотека Python. Работа с пакетами.	**10**							**4**	2		2		6				ПКС-1 ПКС-2
4	Базовые алгоритмические конструкции: условные операторы, циклы. Исключения: работа с ошибками.	**18**							**10**	4		6		8				ПКС-1 ПКС-2
5	Процедурное программирование. Подпрограммы: процедуры и функции, рекурсивные функции.	**14**							**6**	2		4		8				ПКС-1 ПКС-2
6	Строки в Python и обработка текстовой информации	**14**							**6**	2		4		8				ПКС-1 ПКС-2
	Итого:	**70**							**34**	14		20		36				
	Итоговая аттестация	**2**															2 (зачет	

																	в формате тестирования)	
	Всего:	72							34	14		20		36			2	ПКС-1 ПКС-2

2.2. Содержание программы по модулям/разделам/дисциплинам/ и (или) темам.

Таблица 4

Содержание дисциплин (модулей)

Номер темы (раздела)	Содержание темы (раздела)
Введение в Python. Базовые инструменты разработчика. Переменные.	Знакомство со средой разработки IDLE. Инструменты IDE: редактор кода для набора и редактирования исходного кода программ; сервисы для проверки и запуска программного кода, расширения для решения дополнительных задач разработки.
Простые диалоговые программы. Консольный ввод и вывод данных.	Операторы ввода и вывода данных. Оператор присваивания. Встроенные функции и методы для работы с числами. Модуль Math. Математические функции. Операторы для работы с последовательностью. Приоритет выполнения операторов. Генерация случайных чисел. Модуль random.
Синтаксис языка. Типы данных, структуры данных: множества, словари, списки, кортежи. Стандартная библиотека Python. Работа с пакетами.	Структура программы. Комментарии. Основные типы данных и операций в языке Python. Основные типы переменных. Преобразование типов переменных.
Базовые алгоритмические конструкции: условные операторы, циклы. Исключения: работа с ошибками.	Программирование разветвляющихся и циклических процессов на языке Python. Операции сравнения. Операторы условного перехода. Операторы цикла. Функции range() и enumerate(). Операторы перехода на следующую итерацию и прерывания цикла. Вложенные циклы.
Процедурное программирование. Подпрограммы: процедуры и функции, рекурсивные функции.	Определение функции и ее вызов. Необязательные параметры функций и сопоставление по ключам. Переменное число параметров в функции. Анонимные функции. Функции генераторы. Декораторы функций. Глобальные и локальные переменные. Рекурсивные функции. Вложенные функции.
Строки в Python и обработка текстовой информации.	Строковый тип данных. Способы объявления строк. Индексация строк. Операторы для работы со строками. Основные встроенные функции класса str для работы со строками.

3. ОРГАНИЗАЦИОННЫЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

3.1. Материально-техническое и программное обеспечение реализации программы

Институт располагает специальными помещениями, представляющие собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Специальные помещения должны быть укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, обеспечивающие тематические иллюстрации, соответствующие примерным программам дисциплин (модулей), рабочим учебным программам дисциплин (модулей).

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду Института. В процессе обучения используются корпоративные аккаунты на платформе видеоконференцсвязи (Яндекс Телемост, МТС-Линк и др.).

3.2. Учебно-методическое и информационное обеспечение программы

Методические разработки по проведению практических занятий

Тема 2. Простые диалоговые программы. Консольный ввод и вывод данных. (4 часа)

Практическая работа №1

Контрольные вопросы:

1. Что такое алгоритм?
2. Что такое язык программирования?
3. Что такое компьютерная программа?
4. Почему можно рассматривать Python как интерпретируемый, интерактивный и объектно-ориентированный язык программирования?
5. В чем особенности Python?
6. С чем связана скорость выполнения программ на Python?
7. Что представляет собой язык программирования Python и чем он интересен?
8. Что такое Zen of Python?
9. Что такое IDLE?
10. В чем отличия работы в интерактивном и сценарном режимах среды разработки IDLE?
11. Зачем необходимо использовать комментарии в программе?
12. Как вывести сообщение на Python?
13. Какие типы комментариев существуют в Python?

Практическая работа №2

Задание 1. Написать интерактивную программу, которая будет запрашивать соответствующие данные (Ф.И.О., дата рождения, адрес (можно только город), любимый род деятельности, любимое блюдо, любимый фильм и так далее), а затем выводить их в табличном

виде на экран монитора. Для формирования таблицы можно воспользоваться выводом символов таблицы ASCII-кодов с помощью escape-последовательностей.

Задание 2. Написать программу «Рифмоплет», которая, используя дополнительные параметры функции print(), будет запрашивать у пользователя несколько рифм и заканчивать ими заранее подготовленное стихотворение.

Контрольные вопросы

1. Какими способами отображаются строки в Python?
2. Зачем в Python используют тройные кавычки (апострофы)?
3. Зачем нужны escape-последовательности?
4. Зачем нужны raw-строки и как их записать в коде?
5. Что такое таблица кодировок?
6. Чем отличается ASCII-кодировка от Unicode-кодировки?
7. Best of LUCK and remember to HAVE FUN while you're learning
8. Какой стандарт (тип кодировки) используется для кодирования символьной информации в Python 3?
9. Опишите синтаксис функции print().
10. Опишите синтаксис функции input().

Тема 3. Синтаксис языка. Типы данных, структуры данных: множества, словари, списки, кортежи. Стандартная библиотека Python. Работа с пакетами. (2 часа)

Практическая работа №3

Задание 1. Разработать интерактивную программу, демонстрирующую работу с простыми типами данных:

1.	Программа должна запрашивать имя пользователя (строка) и почтовый индекс (целое число) и записывать соответственно в переменные *Chameleon*1 и *Chameleon*2. Вывести значения и типы переменных на экран
2.	Преобразовать переменную *Chameleon*2 к строковому типу с использованием функции `str()`. Умножить полученную строку на число даты рождения пользователя и вывести результат
3.	Сложить переменные *Chameleon*1 и *Chameleon*2, результат записать в *Chameleon*3. Вывести результат суммирования и тип переменной
4.	Преобразовать переменную *Chameleon*3 к списку с использованием функции `list()`. Вывести результат и тип переменной на экран
5.	Напечатать начальный и конечный элементы списка *Chameleon*3. Первое значение списка *Chameleon*3 заменить возрастом пользователя
6.	Прибавить к списку *Chameleon*3 список, вводимый пользователем с клавиатуры. Результат вывести на экран, а затем умножить на 2 и снова вывести на экран
7.	Преобразовать переменную *Chameleon*3 к кортежу с использованием функции `tuple()`. Вывести результат и тип переменной на экран
8.	Попытаться изменить любое значение кортежа *Chameleon*3
9.	Продемонстрировать операцию сложения кортежей и умножения кортежа на число
10.	Преобразовать переменную *Chameleon*3 к множеству с использованием функции `set()`. Вывести результат и тип переменной на экран. Сделать вывод о назначении множеств
11.	Создать словарь *My_Dictionary* из пяти пар, осуществляющий перевод слов с русского на английский. Продемонстрировать обращение к элементам словаря по ключу. Вывести словарь несколько раз на экран. Сделать вывод
12.	Сделать вывод о простых типах данных языка *Python,* универсальности команд и о динамической типизации.

Задание 2. Разработать интерактивную программу «Каков мой возраст в секундах», которая на входе прининает дату рождения пользователя, рассчитывает количество прожитых пользователем секунд и выводит результат на экран монитора.

Задание 3. Необходимо написать программу для специального банкомата. Банкомат должен оптимальным способом выдавать любую веденную пользователем сумму: вначале выдаются крупные банкноты, потом меньше и так вплоть до одного рубля.
К примеру, пользователь запрашивает в банкомате сумму в 587 рублей. Банкомат ему должен выдать следующее: банкноты номиналом 500, 50, три по 10, 5 и затем монеты: 2, 1.

Задание 4. Необходимо написать приложение для разменного автомата. Работа автомата заключается в следующем: автомат принимает у пользователя купюру и выдает ему купюры меньшего номинала.
Примечание: каждая программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом.

Контрольные вопросы

1. Опишите архитектуру и основные элементы компьютера. Какой элемент является центральным при построении любой вычислительной системы?
2. Какие типы памяти доступны при разработке программ?
3. Что такое машинный код?
4. В чем отличия языка высокого уровня от языка низкого уровня?
5. Какой язык понимает и обрабатывает центральный процессор
6. Что такое транслятор и что он делает?
7. Что общего между компилятором и интерпретатором и чем они отличаются?
8. Что такое переменная и зачем она нужна в программе?
9. Что такое «соглашение по присваиванию имен» ?
10. Что такое константа?
11. Что такое hardcode («тяжелый код»)?
12. Описать стандартные простые типы данных в Python.

Тема 4. Базовые алгоритмические конструкции: условные операторы, циклы. Исключения: работа с ошибками. (6 часов)

Практическая работа №4

Задание 1. Разработать интерактивную программу «Quadric Equation» («Квадратное уравнение») для решения квадратных уравнений вида: $ax^2 + bx + c = 0$. Программа должна запрашивать соответствующие параметры a, b и c, проверять параметры и выдавать результат.

Задание 2. Написать программу, которая бы эмулировала игру в кости. Суть игры заключается в броске двух шестигранных кубиков (костей) и подсчета общей суммы очков, которые выпали на первой и второй кости. Для генерирования случайного значения на костях воспользоваться функциями из стандартного модуля random.

Задание 3. Написать программу, которая определяет настроение пользователя в текущий момент времени. Приложение генерирует случайное число, в зависимости от которого выводится одно из псевдографических «лиц», отображающее настроение.

Требования к выполнению: программа должна обязательно быть снабжена комментариями, в которых необходимо указать краткое предназначение программы. Каждая программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом.

Контрольные вопросы

1. Перечислите основные управляющие конструкции.
2. Опишите основные элементы блок-схем.
3. Как в языке Python реализуется механизм истинности-ложности? Может ли значение быть условием?
4. Опишите синтаксис простой условной конструкции if. Представьте примерную блок-схему конструкции.
5. С помощью каких операторов можно комбинировать в одной условной конструкции if несколько условий? Какой механизм оптимизации применяет интерпретатор Python для эффективного вычисления результата комбинированных условных выражений?
6. Опишите синтаксис условной конструкции if-else. Представьте примерную блок-схему конструкции.
7. Опишите синтаксис условной конструкции elif. Представьте примерную блок-схему конструкции.

Практическая работа №5

Задание 1. Разработать интерактивную программу «Попробуй угадать число», которая эмулирует игру на отгадывание числа. Суть игры сводиться к следующему: компьютер генерирует случайное число из диапазона, к примеру, от 1 до 100, а пользователь пытается отгадать число. При каждой попытке компьютер «подсказывает» игроку, как соизмеряется вариант игрока с загаданным компьютером числом: загаданное число больше или меньше указанного. Как только игрок отгадывает число, компьютер должен «поздравить» его с выводом на экран угаданного числа и количества затраченных игроком попыток. Далее компьютер может «предложить» повторно сыграть в игру или выйти.

Для универсальности можно добавить возможность выбора диапазона генерирования компьютером случайных чисел, а также задание ограничения на количество попыток. В случае, если игрок не укладывается в заданное количество попыток, программа должна выводить надпись «Игра завершена».

Задание 2. Модернизировать программы, которые были разработаны в предыдущих лабораторных работах: «Банкомат» и «Разменный аппарат», «Игра в кости» и «Датчик настроения». Их нужно переписать таким образом, чтобы у пользователя была возможность повторного выполнения программы без выхода из нее.

Задание 3. Написать программу «Орел или решка?», которая бы «подбрасывала» условно монету, к примеру, 1000 раз и сообщала, сколько раз выпал орел, а сколько – решка.

Переработать программу из основного задания таким образом, чтобы пользователь загадывал число, а компьютер, используя оптимальный алгоритм, число отгадывал.

Требования к выполнению: программа должна обязательно быть снабжена комментариями. В программах, где это необходимо, предусмотреть возможность ее повторного выполнения, а также защиты от ввода некорректны данных.

Контрольные вопросы

1. Для чего используются циклы? Что такое итерация?
2. Какие разновидности циклов существуют?
3. Описать Python-синтаксис цикла с предусловием while.
4. Чем является выражение после ключевого слова while – инициализацией, условием или обновлением?
5. Какова роль оператора break в теле цикла?
6. Какова роль оператора continue в теле цикла?
7. Какова роль оператора pass в теле цикла?

8. Может ли выражение после ключевого слова while содержать истинное значение или значение других типов данных?

Практическая работа 6

Задание 1. Разработать программы для решения следующих задач:

1) Найти сумму цифр и количество цифр заданного натурального числа.
2) Возвести число в натуральную степень n.
3) Найти количество различных цифр у заданного натурального числа.
4) Найти наибольшую цифру натурального числа.
5) Задано натуральное число. Проверить, является ли заданное натуральное число палиндромом.
6) Определить является ли заданное натуральное число простым.
7) Найти все простые делители заданного натурального числа.
8) Найти НОД и НОК двух натуральных чисел.
9) Заданы три целых числа, которые задают некоторую дату. Определить дату следующего дня.
10) Запрограммировать последовательность чисел Фибоначчи (пользователь вводит порядковый номер элемента последовательности Фибоначчи, а программа выводит на экран значение).

Контрольные вопросы

1. Опишите Python-синтаксис и принцип работы цикла for.
2. В чем особенность использования циклической конструкции for в языке Python по сравнению с использованием в других языках программирования, к примеру, в Pascal или C/C++?
3. Для чего используется стандартная функция range(..)?
4. Опишите наиболее распространенные случаи использования функции range(..).
5. Опишите сочетание функции range(..) и цикла for.

Тема 5. Процедурное программирование. Подпрограммы: процедуры и функции, рекурсивные функции. (4 часа)

Практическая работа №7

Задание 1. Рекурсивно описать функцию f(x, n), вычисляющую $x^n/n!$ при любом действительном x и любом неотрицательном целом n.

Задание 2. Рекурсивно описать функцию pow(x, n), вычисляющую x^n для любого действительного $x \neq 0$ и любого целого n.

Задание 3. Реализовать функцию, которая вычисляет N-й элемент ряда Фибоначчи с использованием рекурсивного алгоритма. На базе данной функции разработать программу, которая должна предлагать пользователю следующие возможности:

- вывод конкретного элемента последовательности;
- вывод всех элементов до указанного пользователем элемента;
- вывод части последовательности, значение последнего элемента которой не превосходит введенного пользователем значения.

Практическая работа №8

Задание 1. Найти все натуральные числа, не превосходящие N, сумма цифр каждого из которых в некоторой степени дает это число (например: 74 = 2 401, 83 = 512, 92 = 81 и т. д.).

Задание 2. Напечатать N автоморфных чисел (числа, совпадающие с младшими цифрами своего квадрата, например: 25 и 625).

Задание 3. Найти N троек чисел Пифагора (натуральные числа a, b и c называются числами Пифагора, если выполняется условие $a^2 + b^2 = c^2$).

Задание 4. Рекурсивно описать функцию divs (N) для подсчета количества всех делителей заданного целого числа N (N > 1), без учета делителей 1 и N (например: divs (7) = 0, divs (10) = 2, divs (16) = 3 и так далее).

Контрольные вопросы

1. Что такое структурированное программирование?
2. Что такое функция? Как описывается функция в Python?
3. Зачем нужны функции?
4. Для чего используется оператор return в функциях? Как возвратить из функции несколько значений?
5. Чем формальные параметры отличаются от фактических?
6. Что такое позиционные параметры?
7. Что такое параметры по умолчанию?

Тема 6. Строки в Python и обработка текстовой информации. (4 часа)

Практическая работа №9

Задание 1. Написать программу «Анаграммы». Суть игры заключается в следующем: формируется группа слов в виде кортежа, компьютер случайным образом выбирает одно из слов, и случайным образом переставляет в нем буквы, а затем представляет пользователю (игроку). Цель игрока – угадать выбранное и «перемешанное» компьютером слово.

Практическая работа №10

Задание 1. Модифицировать программу «Анаграммы» таким образом, чтобы в ней были уровни прохождения игры. К примеру, в игре «Анаграммы» можно реализовать 10 уровней, где на первых уровнях предлагаются простые слова. При этом от уровня к уровню слова становятся длиннее, и растет их уровень сложности. Дополнительно можно добавиться к игре ограниченное количество подсказок и попыток угадывания.

Требования к выполнению:
1. Программа должна обязательно быть снабжена комментариями.
2. В программах, предусмотреть возможность ее повторного выполнения, а также защиты от ввода некорректных пользовательских данных. Для этого рекомендуется разработать отдельные функции.
3. Разработанные основные функции бизнес-логики необходимо разместить в отдельном модуле, а затем подключить в другом модуле, где будет происходить тестирование данных функций.
4. При разработке программ рекомендуется придерживаться принципа единственной ответственности.

Контрольные вопросы

1. Чем характеризуется строковый тип данных в Python?
2. Какие есть способы объявления строк в Python?
3. Что такое неизменяемость строк?
4. Зачем нужна индексация строк, и как ее использовать?
5. Зачем и как используются срезы строк?

6. Какие основные операторы используются для работы со строками?
7. Какие основные встроенные функции класса str используются для работы со строками?
8. Как объявить шаблонную строку, и какие спецификаторы типов (операторы форматирования) в этой строке можно использовать?

Методические разработки для организации самостоятельной работы

Тема 1. Введение в Python. Базовые инструменты разработчика. Переменные.

Вопросы для самостоятельной работы:

14. Типы данных в Python.
15. Примеры использования переменных и типов данных.

Тема 2. Простые диалоговые программы. Консольный ввод и вывод данных.

Вопросы для самостоятельной работы:

1. С помощью какой функции в Python можно узнать подробную справочную информацию?
2. Предположим, в коде программы используется escape-последовательность '\a' для создания звука с помощью системного динамика. В каком случае при запуске программы на выполнение будет слышен звук?

Тема 3. Синтаксис языка. Типы данных, структуры данных: множества, словари, списки, кортежи. Стандартная библиотека Python. Работа с пакетами.

Вопросы для самостоятельной работы:

1. Что измеряет диапазон типа данных?
2. Что такое выражение?
3. Что такое инициализация?
4. Можно ли обратиться к переменной, если инициализировать ее чуть позже?
5. Что произойдет, если присвоить строковой переменой целочисленное значение (или наоборот)?
6. Какой из арифметических операторов не может иметь дробного операнда?
7. Какой из арифметических операторов не может иметь в качестве второго операнда значение ноль?
8. Описать все группы операций, доступные в языке Python.

Тема 4. Базовые алгоритмические конструкции: условные операторы, циклы. Исключения: работа с ошибками.

Вопросы для самостоятельной работы:

1. Чем использование elif будет отличаться от использования вложенных условных конструкций if-else?
2. Как сгенерировать случайную последовательность чисел с использованием функций randint() и randrange()? Чем они отличаются? Какие еще есть полезные функции в модуле random?
3. Что такое бесконечный цикл? Когда он применяется? Привести пример кода организации диалога на тему завершения программы, либо повторного выполнения программы.
4. Если необходимо использовать вложенные циклы while для вывода элементов прямоугольной матрицы в виде строк и столбцов, какой из циклов будет печатать строки: внутренний или внешний?
5. Обязательно ли использовать переменную цикла for внутри самого цикла?
6. Какую дополнительную конструкцию (ветку) могут иметь циклы в языке Python? В каких случаях тело данной конструкции будет выполняться, а в каких – нет?

Тема 5. Процедурное программирование. Подпрограммы: процедуры и функции, рекурсивные функции.

Вопросы для самостоятельной работы:

1. Чем отличается глобальная переменная от локальной?
2. Зачем применяются рекурсивные алгоритмы?
3. В чем заключается смысл принципа единственной ответственности (Single Responsibility Principle) при грамотной разработке единиц программного кода (в частности, функций)?

Тема 6. Строки в Python и обработка текстовой информации.

Вопросы для самостоятельной работы:

1. Что необходимо прописать в исходном коде программы, если нужно использовать различные кодировки для отображения текстовой информации?
2. Как создать всплывающую подсказку с описание вызываемой функции?

Основная литература

1. Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/567821 (дата обращения: 28.06.2025).
2. Федоров, Д. Ю. Программирование на python : учебное пособие для вузов / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 187 с. — (Высшее образование). — ISBN 978-5-534-19666-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/556864 (дата обращения: 28.06.2025).
3. Чернышев, С. А. Основы программирования на Python : учебник для среднего профессионального образования / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 349 с. — (Профессиональное образование). — ISBN 978-5-534-17056-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/567823 (дата обращения: 28.06.2025).
4. Федоров, Д. Ю. Программирование на python : учебник для среднего профессионального образования / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 187 с. — (Профессиональное образование). — ISBN 978-5-534-19654-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/563564 (дата обращения: 28.06.2025).

Дополнительная литература

5. Буйначев, С. К. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 91, [1] с.
6. Изучаем Python, 5-е издание / Лутц Марк – Пер. с англ. – СПб.: Диалектика, 2019. – 832 с.
7. Изучаем Python. Программирование игр, визуализация данных, веб-приложения / Мэтиз Эрик. - Пер. с англ. - СПб.: Питер, 2017. - 491 с.
8. Саммерфилд, М. Python на практике. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2014. — 338 с. — Режим доступа: http://e.lanbook.com/book/66480
9. Василюк Н.Н. Языки программирования. Основы web-программирования: учеб. пособие / Н. Н. Василюк; Перм. гос. нац.-исслед. ун-т. Пермь, 2019. – 103 с.

10. Дронов В. А. Django 2.1. Практика создания веб-сайтов на Python. - СПб.: БХВ-Петербург, 2019. - 672 с.
11. Сысоева М. В., Сысоев И. В. Программирование для «нормальных» с нуля на языке Python: Учебник. В двух частях. Часть 1 / Ответственный редактор: В. Л. Черный : М.: Базальт СПО; МАКС Пресс, 2018. 176 с.:ил.
12. Федоров Д. Ю. Программирование на языке высокого уровня Python. 2-е изд. – Москва: Издательство Юрайт, 2019. – 161 с.

Интернет – ресурсы

1. Электронно-библиотечная система «IPRbooks» [Электронный ресурс]. – Режим доступа: http://www.iprbookshop.ru/
2. Электронно-библиотечная система издательства «Лань» [Электронный ресурс]. – Режим доступа: https://e.la№book.com/
3. Электронно-библиотечная система издательства ЮРАЙТ [Электронный ресурс]. – Режим доступа: https://www.biblio-o№li№e.ru/

4. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Итоговая аттестация проводится в форме зачета (формат – тестирование).

Вопрос 1. Как вывести текст на экран в Python?
a) echo("Hello")
b) print("Hello")
c) console.log("Hello")
d) System.out.println("Hello")
Вопрос 2. Какой тип данных используется для хранения целых чисел?
a) float
b) int
c) str
d) bool
Вопрос 3. Как объявить переменную в Python?
a) var x = 5
b) x = 5
c) int x = 5
d) variable x = 5
Вопрос 4. Какой оператор используется для возведения в степень?
a) ^
b) **
c) //
d) %%
Вопрос 5. Как получить длину списка?
a) length(list)
b) len(list)
c) list.length()
d) list.size
Вопрос 6. Как правильно написать условный оператор if?
a) if (x > 0) {}
b) if x > 0:

c) if x > 0 then
d) if: x > 0
Вопрос 7. Какой цикл используется для итерации по последовательности?
a) while
b) for
c) loop
d) repeat
Вопрос 8. Как выйти из цикла досрочно?
a) stop
b) break
c) exit
d) return
Вопрос 9. Как пропустить текущую итерацию цикла?
a) skip
b) continue
c) next
d) pass
Вопрос 10. Какой оператор используется для обработки исключений?
a) catch
b) try-except
c) error
d) rescue
Вопрос 11. Как объявить функцию в Python?
a) function my_func()
b) def my_func():
c) func my_func()
d) void my_func()
Вопрос 12. Что будет в результате выполнения следующего действия:
а) print((14%3)*2)
ответ_____
б) print((29//3)/3)
ответ_____
Вопрос 12. Что будет выведено на экран монитора в результате программы:

```
y = 5
x=3
z = x**3+9*y
print(z)
```

ответ_____
Вопрос 13. Какое значение будет выведено на экран дисплея, если x=6.

```
y = x + 2
if y >= x:
print (x + y)
else:
print (y - x)
```

Вопрос 14. Что будет выведено на экран?

```
f = 9
h = f - 6
if f <= h:
print (h + f * 2)
else:
print (f - h * 2)
```

Вопрос 15. Что будет выведено на экран?

```
r = 8
```

```
s = r - 2
if s > r or r %2 = = 0:
print(r + 2 * s)
else:
print(s + r * 2)
```

Вопрос 16. Что будет выведено на экран?

```
a = 10
b = 1
while (a>4):
a = a - 1
b = b + 5
print(b)
```

Система оценки в ходе итоговой аттестации в форме зачета

Оценка	Требования к знаниям
зачтено	Даны правильные ответы более чем на 10 вопросов теста
не зачтено	Даны правильные ответы менее чем на 11 вопросов теста

Приложение № 1

С В Е Д Е Н И Я
о преподавательском составе и ведущих специалистах
(кадровая справка)

Реализация программы обеспечивается преподавательским составом Академии, а также высококвалифицированными специалистами из числа руководителей и ведущих специалистов государственных органов, учреждений и иных организаций (включая индивидуальных предпринимателей и самозанятых граждан).

Ф.И.О. преподавателя / ведущего специалиста	**Квалификация по диплому о высшем или среднем профессиональном образовании**	**Дополнительные квалификации**	**Ученая степень, ученое (почетное) звание**	**Стаж работы в области профессиональной деятельности**	**Наименование преподаваемой дисциплины/темы (модуля)**
1	2	3	4	5	6
Акопян Елена Александровна	Южно-Российский государственный технический университет (Новочеркасский политехнический институт), специальность - математические методы и исследование операций в экономике, квалификация - экономист-математик, 2000 г.	Организация электронного обучения и дистанционных образовательных технологий (ЭО и ДОТ) в системе дистанционного обучения (СДО) № 08139-2023-У-КУ УМУ 22.03.2023	Кандидат экономических наук, доцент	25 г.	Введение в Python. Базовые инструменты разработчика. Переменные. Простые диалоговые программы. Консольный ввод и вывод данных. Синтаксис языка. Типы данных, структуры данных: множества, словари, списки, кортежи. Стандартная библиотека Python. Работа с пакетами. Базовые алгоритмические конструкции: условные операторы, циклы. Исключения: работа с ошибками. Процедурное программирование. Подпрограммы: процедуры и функции, рекурсивные функции. Строки в Python и обработка текстовой информации.