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

Омский филиал

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 00D0FBD24264A8B913A66A4DE72D598CF7
Владелец: Боровская Ольга Игоревна
Действителен: с 10.11.2025 до 03.02.2027

«УТВЕРЖДАЮ»
Директор Омского филиала
О.И. Боровская
«20» марта 2026 г.

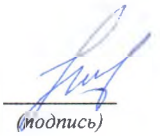
ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
повышения квалификации

«Базовый Python для анализа данных»

(наименование программы)

Омск, 2026

Разработчики:

Доцент кафедры экономики и управления Омского филиала РАНХиГС, к.т.н. —  (подпись) — И.Г. Ольгина (И.О.Фамилия)

Директор Центра дополнительного образования Омского филиала РАНХиГС —  (подпись) — Н.А. Кузина (И.О.Фамилия)

Руководитель программы
Доцент кафедры экономики и управления Омского филиала РАНХиГС, к.т.н. —  (подпись) — И.Г. Ольгина (И.О.Фамилия)

Программа повышения квалификации рассмотрена на заседании ученого совета Омского филиала РАНХиГС и рекомендована к реализации, протокол № 8 от «20» марта 2026 г.

СОДЕРЖАНИЕ

		Стр.
1.	ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ……..	4
1.1.	Цель и задачи реализации программы……..	4
1.2.	Нормативные правовые акты…….....	5
1.3.	Планируемые результаты обучения……..	7
1.4.	Категория слушателей……	11
1.5.	Формы и технологии обучения……..	11
1.6.	Период обучения, срок освоения и режим занятий….....	11
1.7.	Документ о квалификации……..	12
2.	СОДЕРЖАНИЕ ПРОГРАММЫ……	13
2.1.	Календарный учебный график……..	13
2.2.	Учебный план……..	14
2.3.	Содержание программы по разделам и темам…….....	15
3.	ОРГАНИЗАЦИОННЫЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ……	18
3.1.	Материально-техническое и программное обеспечение реализации программы……	18
3.2.	Учебно-методическое и информационное обеспечение программы……	19
4.	ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ……	21
5.	ИНДИКАТОРЫ СФОРМИРОВАННЫХ КОМПЕТЕНЦИЙ….	42

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Цель и задачи реализации программы

Программа повышения квалификации **«Базовый Python для анализа данных»** направлена на обучение основам программирования на языке Python с целью дальнейшего использования полученных навыков для решения задач обработки и анализа данных.

Программа ориентирована на тех, кто хочет освоить базовые инструменты программирования и научиться применять их для аналитики данных. Она подходит специалистам разных областей, которым необходимы знания Python для своей профессиональной деятельности, включая экономистов, маркетологов, инженеров, исследователей, студентов технических специальностей и всех желающих войти в мир Data Science.

Настоящая программа реализуется в рамках федерального проекта «Активные меры содействия занятости», входящего в состав национального проекта «Кадры» 2026 года, и направлена на дополнительное профессиональное образование отдельных категорий граждан. Реализация мероприятий по обучению осуществляется с учетом потребности работодателей в кадрах и в целях приобретения или развития гражданами имеющихся знаний, компетенций и навыков для обеспечения их занятости.

Основная **цель** реализации программы — обеспечить слушателей фундаментальными знаниями и навыками программирования на Python, необходимыми для эффективного сбора, подготовки, анализа и визуализации данных. Программа также призвана подготовить слушателей к дальнейшему освоению специализированных инструментов и библиотек Python, применяемых в области анализа данных.

Задачи реализации программы:

1. Освоение базовых конструкций языка Python (переменных, типов данных, циклов, условных операторов).
2. Изучение основных структур данных (списки, словари, множества, кортежи).
3. Овладение основными приемами работы с файлами и базовыми методами обработки текста.
4. Развитие навыков работы с библиотеками для анализа данных (NumPy, Pandas, Matplotlib).
5. Формирование умения решать практические задачи обработки и анализа данных средствами Python.
6. Понимание принципов написания качественного и читаемого программного кода.
7. Подготовка базы для освоения продвинутых методов машинного обучения и статистики.

Завершив данную программу, слушатели смогут уверенно писать код на Python и эффективно применять полученные знания для реализации простых проектов анализа данных.

1.2. Нормативные правовые акты

• Федеральный закон от 29.12.2012 N 273-ФЗ (ред. от 29.12.2025) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.03.2026);

• 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);

• Постановление Правительства РФ от 11.10.2023 N 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;

• Постановление Правительства РФ от 07.03.2025 N 291 (ред. от 30.01.2026) «Об утверждении Положения о реализации мероприятий по организации профессионального обучения и дополнительного профессионального образования отдельных категорий граждан»;

• Приказ Министерства труда и социальной защиты Российской Федерации от 29 сентября 2014 года № 667н «О реестре профессиональных стандартов (перечне видов профессиональной деятельности)» (зарегистрирован Министерством юстиции Российской Федерации 19 ноября 2014 года, регистрационный № 34779) с изменением, внесенным приказом Министерства труда и социальной защиты Российской Федерации от 9 марта 2017 г. № 254н (зарегистрирован Министерством юстиции Российской Федерации 29 марта 2017 года, регистрационный № 46168);

• Приказ Минобрнауки России от 19 сентября 2017 г. № 926. (ред. от 27.02.2023) «Об утверждении Федерального государственного образовательного стандарта высшего образования – бакалавриата по направлению подготовки 09.03.02 Информационные системы и технологии» (Зарегистрировано в Минюсте России 20.10.2017 N 48535);

• Приказ Минтруда России от 20.07.2022 N 424н «Об утверждении профессионального стандарта «Программист» (Зарегистрировано в Минюсте России 22.08.2022 N 69720);

• Методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов (утв. Минобрнауки России 22.01.2015 года № дл-1/05вн);

• <Письмо> Минобрнауки России от 22.04.2015 N ВК-1032/06 «О направлении методических рекомендаций» (вместе с «Методическими рекомендациями-разъяснениями по разработке дополнительных профессиональных программ на основе профессиональных стандартов»);

• Приказ РАНХиГС от 02 декабря 2025 года N 02-02669/001 «Об утверждении порядка разработки и утверждения в Академии дополнительных профессиональных программ - программ повышения квалификации, программ профессиональной переподготовки»;

• Приказ РАНХиГС от 13 января 2026 года N 02-00009/001 «Об утверждении Положения об итоговой аттестации слушателей дополнительных профессиональных программ в Академии»;

• Приказ РАНХиГС от 22 сентября 2017 года N 01-6230 «Об утверждении Положения о применении в Академии электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».

Нормативные документы, определяющие требования к выпускнику программы:

• «Трудовой кодекс Российской Федерации» от 30.12.2001 N 197-ФЗ (ред. от 29.12.2025, с изм. от 06.02.2026);

• Приказ Росстандарта от 16.05.2025 N 423-ст «Об утверждении Общероссийского классификатора профессий рабочих, должностей служащих и тарифных разрядов ОК 016-2025»;

• «Квалификационный справочник должностей руководителей, специалистов и других служащих» (утв. Постановлением Минтруда России от 21.08.1998 N 37) (ред. от 27.03.2018).

1.3. Планируемые результаты обучения

Таблица 1

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

Виды деятельности	Общепрофессиональные/профессиональные компетенции ОПК, ПК или трудовые функции (ПСК и СК) *(формируются и (или) совершенствуются)*	Практический опыт	Знания	Умения
1	2	3	4	5
ПСК - профессионально-специализированные компетенции				
ВД 1. Разработка и отладка программного кода	Трудовая функция А/02.3 ПСК-1 [1]. Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных	Создание программного кода в соответствии с техническим заданием (готовыми спецификациями). Оптимизация программного кода с использованием специализированных программных средств.	Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования. Методологии разработки компьютерного программного обеспечения. Технологии программирования.	Применять выбранные языки программирования для написания программного кода. Использовать выбранную среду программирования и средства системы управления базами данных. Использовать возможности имеющейся технической и/или

[1] *ПСК-1 (профессионально-специализированные компетенции) - коды компетенций сформированы на основе трудовых функций профессионального стандарта «Программист», утвержденного Приказом Минтруда России от 20.07.2022 N 424н «Об утверждении профессионального стандарта «Программист» (Зарегистрировано в Минюсте России 22.08.2022 N 69720)*

			Особенности выбранной среды программирования и системы управления базами данных.	программной архитектуры для написания программного кода.
	Трудовая функция A/03.3 ПСК-2[2]. Оформление программного кода в соответствии с установленными требованиями	Приведение наименований переменных, функций, классов, структур данных и файлов в соответствие с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода. Структурирование исходного программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода. Комментирование и разметка программного кода в соответствии с нормативно-техническими документами (стандартами и	Инструментарий для создания и актуализации исходных текстов программ Системы кодировки символов, форматы хранения исходных текстов программ.	Применять нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода. Применять инструментарий для создания и актуализации исходных текстов программ. Применять заданные стандарты и шаблоны для составления и оформления

[2] *ПСК-2 (профессионально-специализированные компетенции) - коды компетенций сформированы на основе трудовых функций профессионального стандарта «Программист», утвержденного Приказом Минтруда России от 20.07.2022 N 424н «Об утверждении профессионального стандарта «Программист» (Зарегистрировано в Минюсте России 22.08.2022 N 69720)*

		регламентами), определяющими требования к оформлению программного кода. Форматирование исходного программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода. Приведение наименований переменных, функций, классов, структур данных и файлов в соответствие с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода. Структурирование исходного программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к		технической документации[4].

[4] *ПСК-2 (профессионально-специализированные компетенции) - коды компетенций сформированы на основе трудовых функций профессионального стандарта "Программист", утвержденный Приказом Минтруда России от 20.07.2022 N 424н (Зарегистрировано в Минюсте России 22.08.2022 N 69720)*

		оформлению программного кода. Комментирование и разметка программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода. Форматирование исходного программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода.[3]		
ПК – профессиональные компетенции, ОПК – общепрофессиональные компетенции				
ПК-1. Способен разрабатывать требования и проектировать программное обеспечение[5]		Имеет навыки отладки программного кода на уровне программных модулей	Знает языки формализации функциональных спецификаций,	Умеет применять выбранные языки программирования для написания

[3] *ПСК-2 (профессионально-специализированные компетенции) - коды компетенций сформированы на основе трудовых функций профессионального стандарта "Программист", утвержденный Приказом Минтруда России от 20.07.2022 N 424н (Зарегистрировано в Минюсте России 22.08.2022 N 69720)*

[5] *ПК-1 (профессиональные компетенции) - коды компетенций сформированы на основе ФГОС ВО Приказ Минобрнауки России от 19 сентября 2017 г. № 926. (ред. от 27.02.2023) «Об утверждении Федерального государственного образовательного стандарта высшего образования – бакалавриата по направлению подготовки 09.03.02 Информационные системы и технологии»*

		Компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними.	программного кода
ПК-2. Способен разрабатывать компоненты системных программных продуктов[6]	Имеет навыки работы с технической документацией по языку программирования, системе команд процессора устройства, адресации памяти и регистров процессора устройства	Знает синтаксис, особенности программирования и стандартные библиотеки выбранного языка программирования; принципы организации, состав и схемы работы операционных систем	Уметь создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов; осуществлять отладку программных продуктов для целевой операционной системы
ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий[7]	Решение практических задач с использованием методов алгоритмизации и языков программирования.	Принципы разработки программ, языки и технологии программирования;	Разрабатывать алгоритмы и выбирать языки и технологии программирования

[6] *ПК-2 (профессиональные компетенции) - коды компетенций сформированы на основе ФГОС ВО Приказ Минобрнауки России от 19 сентября 2017 г. № 926. (ред. от 27.02.2023) «Об утверждении Федерального государственного образовательного стандарта высшего образования – бакалавриата по направлению подготовки 09.03.02 Информационные системы и технологии»*

[7] *ОПК- 6 (общепрофессиональные компетенции) - коды компетенций сформированы на основе ФГОС ВО Приказ Минобрнауки России от 19 сентября 2017 г. № 926. (ред. от 27.02.2023) «Об утверждении Федерального государственного образовательного стандарта высшего образования – бакалавриата по направлению подготовки 09.03.02 Информационные системы и технологии»*

	Участие в разработке программных продуктов. Участие в создании технической документации	принципы автономной отладки и тестирования программ	выполнять тестирование и отладку программ; оформлять программную документацию[8]

[8] *ОПК- 6 (общепрофессиональные компетенции) - коды компетенций сформированы на основе ФГОС ВО Приказ Минобрнауки России от 19 сентября 2017 г. № 926. (ред. от 27.02.2023) «Об утверждении Федерального государственного образовательного стандарта высшего образования – бакалавриата по направлению подготовки 09.03.02 Информационные системы и технологии»*

1.4. Категория слушателей

К гражданам, имеющим право принять участие в мероприятиях по обучению, относятся следующие категории:

- Граждане предпенсионного возраста
- Граждане в возрасте 50 лет и старше
- Незанятая женщина, имеющая детей в возрасте от 0 до 7 лет
- Гражданин, находящийся в декрете до 3-х лет
- Инвалид
- Незанятый гражданин в возрасте до 35 лет, окончивший военную службу по призыву
- Незанятый гражданин в возрасте до 35 лет, имеющий документ об образовании и (или) о квалификации
- Члены семей военных, погибших при исполнении в ходе специальной военной операции (боевых действий)
- Женщина, находящаяся в отпуске по уходу за ребёнком до 3-х лет
- Гражданин в возрасте до 35 лет, не имеющий СПО или ВО и не обучающийся по образовательным программам СПО и ВО
- Ветеран боевых действий в ДНР, ЛНР, Запорожье, Херсоне и на Украине, уволенный с военной службы
- Участник боевых действий в ДНР и ЛНР, начиная с 11 мая 2014 г.
- Гражданин, ищущий работу
- Безработный гражданин (НП «Кадры»)

К освоению программы допускаются: лица, имеющие среднее профессиональное и (или) высшее образование;

1.5. Формы и технологии обучения

Очно-заочная форма обучения, с применением дистанционных образовательных технологий (ДОТ).

Срок освоения – 72 академических часа, из них контактная аудиторная работа – 30 ак. час., в том числе лекций 0 ак. час., практические занятия – 30 ак. час.; лабораторные занятия – 0 ак. час., контактная работа (с применением дистанционных образовательных технологий) – 28 ак. час., в том числе лекций – 28 ак. час., практические занятия – 0 ак. час; лабораторные занятия – 0 ак. час., самостоятельная работа слушателей – 10 ак. час., итоговая аттестация – 4 ак. час.

1.6. Период обучения, срок освоения и режим занятий

Программа рассчитана на 72 академических часа.

Период обучения: 4 недели и 4 дня по 4-8 академических часа в день.

Один академический час - 45 минут. Занятия проводятся парами. Время занятий согласно расписанию: 9.00-21.20 (пн-пт), 9.00-17.40 (сб).

В течение обучения слушатели могут получить доступ к электронной информационно-образовательной среде Омского филиала РАНХиГС, пройти тестирование.

1.7. Документ о квалификации

Слушатели, успешно освоившие дополнительную профессиональную программу и прошедшие итоговую аттестацию, получают удостоверение о повышении квалификации Российской академии народного хозяйства и государственной службы при Президенте Российской Федерации.

Слушатели, не прошедшие итоговой аттестации, или получившие на итоговой аттестации неудовлетворительные результаты, а также освоившие часть дополнительной профессиональной программы, получают справку об обучении или периоде обучения установленного Академией образца.

2. СОДЕРЖАНИЕ ПРОГРАММЫ

2.1. Календарный учебный график

Таблица 2

Период обучения - 4 недели и 4 дня				
1 неделя	2 неделя	3 неделя	4 неделя	4 дня
УЗ, УЗ ДОТ, ТКУ, СР	УЗ, УЗ ДОТ, ТКУ, СР	УЗ, УЗ ДОТ, ТКУ, СР	УЗ, УЗ ДОТ, ТКУ, СР	УЗ, УЗ ДОТ, ТКУ, СР, ИА

Календарный учебный график заполняется с помощью условных обозначений:

УЗ – учебные занятия;

УЗ ДОТ – учебные занятия с применением дистанционных образовательных технологий;

ТКУ – текущий контроль успеваемости;

СР – самостоятельная работа;

ИА – итоговая аттестация.

2.1. Учебный план

Таблица 3

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
повышения квалификации
«Базовый Python для анализа данных»
Очно-заочная форма обучения (с применением ДОТ)

№п/п	Наименование раздела дисциплины, темы	Общая трудоемкость, час.	Контактная работа, час.					Самостоятельная работа, час	Контактная работа (с применением дистанционных образовательных технологий), час.					Самостоятельная работа (в т.ч. электронное обучение (ЭО), час	Текущий контроль успеваемости	Промежуточная аттестация (форма/час)	Итоговая аттестация (вид /час.)	Код компетенции
			Всего	В том числе					Всего	В том числе								
				Лекции / в интерактивной форме	Практические занятия/ в интерактивной форме	Контактная самостоятельная работа, час	Индивидуальные и групповые консультации			Лекции/ в интерактивной форме	Лабораторные занятия /в 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	Раздел 1. Основы программирования на языке Python																	
1.1	Тема 1. Язык программирования Python и среды разработки	3						1	2	2								ПК-1 ОПК-6
1.2	Тема 2. Типы данных. Операторы языка программирования	6	4		4				2	2					ПЗ			ПК-1 ПК-2
1.3	Тема 3. Коллекции. Преобразование коллекций	8	6		6				2	2					ПЗ			ПК-1 ПК-2
1.4	Тема 4. Функции.	7	4		4			1	2	2					ПЗ			ПК-1 ПК-2
1.5	Тема 5. Области видимости переменных.	4	2		2				2	2					ПЗ			ПК-1 ПК-2

1.6	Тема 6. Передача параметров.	5	2		2			1	2	2					ПЗ			ПК-1 ПК-2
1.7	Тема 7. Функции с переменным числом аргументов.	5	2		2			1	2	2					ПЗ			ПК-1 ПК-2
1.8	Тема 8. Функции высшего порядка. Рекурсия.	5	2		2			1	2	2					ПЗ			ПК-1 ПК-2
1.9	Тема 9. Библиотеки.	12	4		4			2	6	6					ПЗ			ПК-1 ПК-2
2	Раздел 2. Объектно-ориентированное программирование (ООП).																	
2.1	Тема 10. Введение в ООП.	5	2		2			1	2	2					ПЗ			ПК-1 ОПК-6
2.2	Тема 11 Основные принципы объектно-ориентированного программирования (ООП).	5	2		2			1	2	2					ПЗ			ПК-1 ПК-2
2.3	Тема 12. Проектирования программы.	3						1	2	2								ПК-1 ОПК-6
	Итого:	68	30		30			10	28	28								
	Итоговая аттестация	4															4 ЗТ	
	Всего:	72	30		30			10	28	28							4	

*Условные обозначения:
Виды текущего контроля успеваемости (ТКУ): ПЗ – практическое задание.
Виды итоговой аттестации (ИА): ЗТ – (зачет/традиционная форма проведения).

2.2. Содержание программы по разделам и темам

Содержание теоретического и практического материала раскрывается в логической последовательности изучения разделов и тем учебного плана. Содержание программы раскрывается с учетом современного развития образования и науки, техники, культуры, а также перспектив их развития (Таблица 4).

Таблица 4

Содержание программы по разделам и темам

Номер темы и наименование	Содержание темы
Раздел 1. Основы программирования на языке Python	
Тема 1. Язык программирования Python и среды разработки.	Лекция 1.1. Введение в программирование на Python. Основные особенности языка программирования Python
	Лекция 1.2. Знакомство со средами разработки. Понятие среды программирования. Основные компоненты среды разработки. Цикл разработки программы.
Тема 2. Типы данных. Операторы языка программирования.	Лекция 2.1. Основные синтаксические конструкции Python, включая переменные, типы данных, условные операторы, циклы и функции, а также особенности и области применения.
	Практическое занятие 2.1. Простейшие функции.
	Практическое занятие 2.2. Создание простых программ с циклами.
Тема 3. Коллекции. Преобразование коллекций.	Лекция 3.1. Структуры данных в Python: множества, списки, кортежи и словари. Операции индексации и среза.
	Лекция 3.2. Преобразование коллекций и хранение текстов в памяти компьютера при программировании на Python.
	Практическое занятие 3.1. Операции над множествами.
	Практическое занятие 3.2. Методы списков и строк. Работа со словарями.
Тема 4. Функции.	Лекция 4. 1. Функция как способ группировать команды и именовать участки кода. Аргументы функций. Возвращение значений из функций.
	Практическое занятие 4.1. Создание простейших функций без параметров.
	Практическое занятие 4.2. Создание функций с параметрами.
Тема 5. Области видимости переменных.	Лекция 5.1. Локальные и глобальные переменные. Области видимости. Аргументы функций как локальные переменные.
	Практическое занятие 5. 1. Использование функции main(). Использование глобальных
Тема 6. Передача параметров.	Лекция 6. 1. Отличие между переменной и значением. Функции, изменяющие значение аргумента. Изменяемость и неизменяемость объектов.
	Практическое занятие 6. 1. Пример равенства объектов. Пример идентичности объектов. Операции мелкого и глубокого копирования.

Номер темы и наименование	Содержание темы
Тема 7. Функции с переменным числом аргументов.	Лекция 7.1. Аргументы по умолчанию. Позиционные и именованные аргументы. Именованные аргументы. Функция как объект.
	Лекция 7.2. Функции высшего порядка. Комбинирование функций.
	Практическое занятие 7. Обработка коллекций. Потоковый ввод.
Тема 8. Функции высшего порядка. Рекурсия.	Лекция 8. 1. Функции высшего порядка. Комбинирование функций.
	Лекция 8. 2. Рекурсия
	Практическое занятие 8.1. Потоковый ввод stdin.
	Практическое занятие 8.2. Рекурсивное вычисление факториала.
Тема 9. Библиотеки.	Лекция 9.1. Библиотека в программировании. Библиотеки как наследие. Установка библиотек. Встроенные модули.
	Лекция 9.2. Библиотеки для анализа данных на Python.
	Практическое занятие 9.1. Импорт модулей. Работа со стандартными библиотеками.
	Практическое занятие 9.2. Работа с графическими файлами. Работа со звуком. Работа с документами.
	Практическое занятие 9.3. Создание статистических графиков. Визуализация данных.
Раздел 2. Объектно-ориентированное программирование (ООП)	
Тема 10. Введение в ООП	Лекция 10.1. Основные понятия. Создание классов. Методы классов. Инициализация экземпляров класса.
	Практическое занятие 10.1. Переопределение поведения объекта в функции.
Тема 11. Основные принципы объектно-ориентированного программирования (ООП)	Лекция 11.1. Полиморфизм. Инкапсуляция. Наследование. Иерархия классов. Механизм наследования.
	Лекция 11.2. Расширение методов. Переопределение методов. Множественное наследование.
	Практическое занятие 11.1. Использование методов наследников в базовом классе.
	Практическое занятие 11.2. Переопределение поведения объекта в функции.
Тема 12. Проектирования программы	Лекция 12.1. Проектирование и разработка классов.
	Лекция 12.2. Создание и использование своих модулей.
	Практическое занятие 12.1. Проектирование программы.
	Практическое занятие 12.2. Разработка приложения.

3. ОРГАНИЗАЦИОННЫЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

3.1. Материально-техническое и программное обеспечение реализации программы

Для проведения занятий по программе необходимо материально-техническое обеспечение учебных аудиторий в зависимости от типа занятий: лекционного типа, практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенных:

- комплектами специализированной учебной мебели;
- доска аудиторная;
- компьютерная техника;
- презентационное оборудование;
- выход в Интернет и доступ в электронную информационно-образовательную среду организации.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к Интернет и доступом в электронную информационно-образовательную среду организации.

Каждый слушатель в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом в электронно-библиотечную систему и к электронной информационно-образовательной среде Академии.

Электронно-библиотечная система и электронная информационно-образовательная среда Академии обеспечивают возможность доступа слушателя из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети «Интернет», и отвечает техническим требованиям организации, как на территории организации, так и вне ее.

Омский филиал РАНХиГС обеспечен необходимым комплектом лицензионного программного обеспечения.

Слушателям и педагогическим работникам обеспечен доступ (удаленный доступ) к современным профессиональным базам данных, базам данных научных изданий и информационным справочным системам, состав которых определяется в рабочих программах дисциплин.

В случае, если для реализации программы потребуется применение ЭО и(или) ДОТ, то используется личный кабинет слушателя в системе СДО Академии https://lk.ranepa.ru/

Вход в систему СДО по логину и паролю, который направляется слушателю при зачислении на ДПП.

Омский филиал располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работы слушателей, предусмотренных учебным планом.

3.2. Учебно-методическое и информационное обеспечение программы

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 Знаниум [сайт]. — URL: https://www.urait.ru/bcode/567821 (дата обращения: 03.10.2025).

2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2025. — 343 с. — ISBN 978-5-16-016906-4. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2186214 (дата обращения: 19.02.2026).

3. Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2024. — 216 с. + Доп. материалы [Электронный ресурс]. — ISBN 978-5-16-015638-5. - Текст : электронный. — URL: https://znanium.ru/catalog/product/213186 (дата обращения: 19.02.2026).

4. Кузьменков, Д. С. Основы языка программирования Python : практическое пособие / Д. С. Кузьменков, Е. Ю. Кузьменкова ; Гомельский гос. ун-т им. Ф. Скорины. – Гомель : ГГУ им. Ф. Скорины, 2022. – 47 с. — URL: http://elib.gsu.by/jspui/handle/123456789/45375(дата обращения: 19.02.2026).

5. Ван Хорн II, Б. М. PyCharm: профессиональная работа на Python : практическое руководство / Б. М. Ван Хорн II, К. Нгуен ; пер. с англ. И. Л. Люско. – Москва : ДМК Пресс, 2024. - 620 с. – ISBN 978-5-93700-274-7. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2205063 (дата обращения: 19.02.2026).

Дополнительная литература

6. 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://www.urait.ru/bcode/556864 (дата обращения: 03.10.2025).

7. Эрик Мэтиз. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. - Санкт-Петербург : Питер, 2021. - 512 с. - ISBN 978-5-4461-1528-0. — URL: https://ibooks.ru/bookshelf/371712/reading (дата обращения: 24.02.2026).

8. Майер Кристиан. Однострочники Python: лаконичный и содержательный код. - Санкт-Петербург : Питер, 2022. - 256 с. - ISBN 978-5-4461-2966-9. — URL: https://ibooks.ru/bookshelf/382390/reading (дата обращения: 24.02.2026).

9. Виафоре П. Надежный Python: Пер. с англ. / П. Виафоре. - Санкт-Петербург : БХВ-Петербург, 2023. - 352 с. - ISBN 978-5-9775-1174-2. - URL: https://ibooks.ru/bookshelf/389630/reading (дата обращения: 24.02.2026).

10. Программирование на языке Python. Среды разработки PyCharm и Jupyter Notebook : учебное пособие для вузов / А. А. Бровкин, Н. А. Лаптев, А. Н. Пылькин [и др.]. - Москва : Горячая линия - Телеком, 2023. - 143 с. - ISBN 978-5-9912-1001-0. - Текст : электронный. — URL: https://znanium.ru/catalog/product/2184225 (дата обращения: 24.02.2026).

11. Седер Наоми. Python. Экспресс-курс. 3-е изд. - Санкт-Петербург : Питер, 2019. - 480 с. - ISBN 978-5-4461-0908-1. — URL: https://ibooks.ru/bookshelf/365296/reading (дата обращения: 02.03.2026).

Нормативные правовые акты

1. ГОСТ 34.201-89 Виды, комплектность и обозначения документов при

2. создании автоматизированных систем.

3. ГОСТ 34.320-96 Концепции и терминология для концептуальной схемы и

4. информационной базы.

5. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления.

6. ГОСТ 34.601-90 Автоматизированные системы. Стадии создания.

7. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы.

8. ГОСТ 34.603-92 Информационная технология. Виды испытаний автоматизированных систем.

9. РД 50-34.698-90 Автоматизированные системы. Требования к содержанию документов.

10. ГОСТ Р ИСО/МЭК 8824-3-2002 Информационная технология. Абстрактная синтаксическая нотация версии один.

11. ГОСТ Р ИСО/МЭК 10746-3-2001 Управление данными и открытая распределённая обработка.

12. ГОСТ Р ИСО/МЭК 15271-02 Процессы жизненного цикла программных средств.

13. ГОСТ Р ИСО/МЭК 15910-2002 Процесс создания документации пользователя программного средства.

Электронно-библиотечные ресурсы

1. https://urait.ru –Электронно-библиотечная система [ЭБС] Юрайт

2. http://www.iprbookshop.ru – Электронно-библиотечная система [ЭБС] «IPRsmart» («Iprbooks»)

3. https://e.lanbook.com - Электронно-библиотечная система [ЭБС] «Лань»

4. https://new.znanium.com Электронно-библиотечная система [ЭБС] «Znanium.com».

5. https://ibooks.ru - Электронно-библиотечная система [ЭБС] «Айбукс.ру/ibooks.ru»

6. https://lib.ranepa.ru/ru/informatsionnye-resursy/russkoyazychnye-resursy/58-elibrary-ru - Научная электронная библиотека Elibrary.ru

7. https://dlib.eastview.com – Информационный сервис «East View»

8. http://www.edu.ru - Федеральный образовательный портал «Российское образование»

9. https://rusneb.ru - Национальная электронная библиотека

10. Справочно-правовая система «КонсультантПлюс».

Интернет-ресурсы

1. https://www.python.org/doc/ - Документация, учебные пособия и руководства по Python.

4. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Оценка качества освоения программы осуществляется с помощью комплекса форм и процедур текущего контроля успеваемости и итоговой аттестации. В структуру оценочных средств входят:

- Типовые контрольные работы — для проверки теоретических знаний и понимания ключевых концепций языка Python, структур данных, основ анализа и обработки информации.
- Тестовые задания — для оперативной проверки усвоения терминологии, синтаксиса, базовых алгоритмов и стандартных библиотек.
- Практические задания — выполняются в ходе аудиторных занятий и самостоятельной работы, направлены на закрепление умений и формирование навыков программирования, анализа данных, визуализации и работы с реальными наборами данных.
- Проектная работа — итоговое задание, в рамках которого слушатель самостоятельно решает прикладную задачу анализа данных, демонстрируя приобретённые знания, умения и практический опыт.

Методы контроля

В процессе обучения используются следующие методы контроля:

- Текущий контроль — осуществляется по итогам выполнения практических заданий на каждом занятии. Преподаватель оценивает корректность, полноту и стиль кода, а также умение применять изученные инструменты для решения задач.
- Промежуточный контроль — проводится в форме тестирования или выполнения контрольной работы по завершении крупных разделов программы.
- Итоговый контроль — защита итогового проекта, где слушатель представляет разработанную программу, обосновывает выбранные методы и демонстрирует навыки работы с Python на примере практической задачи.

Аспекты практических заданий

1. Работа с переменными, типами данных, операторами и условиями.
2. Создание и обработка списков, словарей, множеств.
3. Написание функций для решения типовых задач анализа данных.
4. Чтение и запись данных из/в файлы (CSV, Excel).
5. Использование библиотек *Pandas* и *NumpPy* для обработки и анализа табличных данных.

6. Визуализация данных с помощью *Matplotlib*.

7. Решение прикладной задачи: анализ набора данных (например, выгрузка из CRM или открытых датасетов), подготовка отчёта с выводами и визуализациями.

Примеры практических заданий.

Задание 1. Реализуйте программу, которая будет принимать сначала число строк, а затем число столбцов будущей таблицы. Элементами таблицы будут вещественные числа, равные делению текущей строки на текущий столбец. Нумерация строк и столбцов начинается с единицы. Значения элементов таблицы будут разделены знаком табуляции.

Ввод	Вывод
3 3	1.0 0.5 0.3333333333333333 2.0 1.0 0.6666666666666666 3.0 1.5 1.0

Задание 2. Простые числа

Необходимо реализовать программу, которая получает число и выводит все простые числа до заданного числа.

Ввод	Вывод
20	2 3 5 7 11 13 17 19 или 19 17 13 11 7 5 3 2

Задание 3. Задача по индексации. «Классическая игра в города»

В классической игре в города участники поочередно называют города, где каждый последующий город должен начинаться на последнюю букву предыдущего. Необходимо написать программу, которая считывает последовательно названия городов, осуществляя проверку правилам – первая буква нового города должна начинаться на последнюю букву из предыдущего. Если введённый город не согласуется с правилами, программа выводит сообщение: «Нарушение правил игры!».

Ввод	Вывод
омск киров владивосток кемерово	омск киров владивосток кемерово

москва	москва Нарушение правил игры!

Обратите внимание, что регистр букв играет роль при сравнении – это разные символы! Поэтому при вводе слов, желательно сделать регистр для всех символов одинаковым.

Задание 4. Задача по срезам

Обработка текста. Напишите программу, которая проводит первичную обработку текста. Нужно удалять сочетание «% %» в начале некоторых строк и удалить строки, начинающиеся с «****».

На вход программы поступает количество строк *n*, а затем сами строки.

ввод	вывод
4 Gkjfhu3545frkn %%peorkf8f4fFjd rkbvgrH5O ****jfbg455	Gkjfhu3545frkn peorkf8f4fFjd rkbvgrH5O

Задание 5. Задача по вложенным циклам.

Бывают ситуации, когда на горку в аквапарке стоит очередь, по технике безопасности спуск с горки в бассейн занимает 1 минуту. Необходимо написать программу, ведущую обратный отсчет для каждого посетителя в очереди, который начинается с 0 – от первого посетителя и удлиняется на минуту для следующего. На вход программы поступает количество человек в очереди, а на выходе – обратный отсчет по минутам для каждого в очереди.

Ввод	Вывод
3	Посетителю 1 Осталось ждать 0 минут Посетителю 2 Осталось ждать 1 минут Осталось ждать 0 минут Посетителю 3 Осталось ждать 2 минут Осталось ждать 1 минут Осталось ждать 0 минут

Задание 6. «Приближение к числу 1,618»

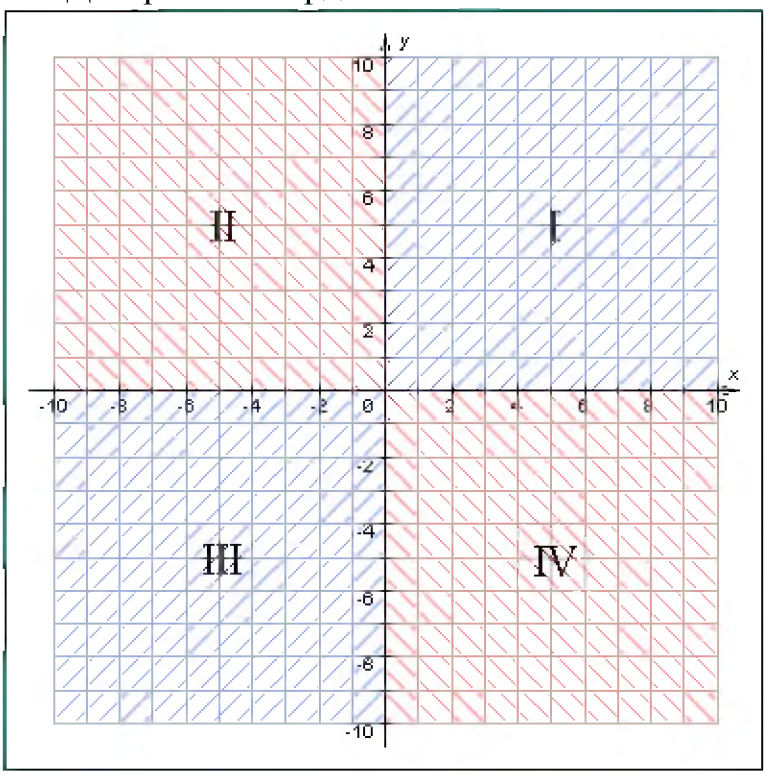
Отношение, выраженное числом 1,618, часто называют золотым сечением. Золотое сечение тесно связано с числами Фибоначчи. i-ое приближения золотого сечения вычисляется как отношение (i+1)-го числа Фибоначчи к i-му.

Необходимо написать функцию approximation(i), которая получает число i как аргумент и печатает i-ое последовательное приближение золотого сечения.

Пример:

Ввод	Вывод
approximation(10)	1.6181818181818182

Задание 7. Декартовы координаты.



Пусть у нас имеется координатная плоскость, образованную осями OX и OY. Как видно на рисунке, оси делят плоскость на 4 части, которые в тригонометрии именуются как: четверть справа вверху обозначается I, и далее против часовой стрелки соответственно II, III и IV.

Необходимо написать функцию coordinate(ox, oy) которая принимает в качестве параметра координаты точки, и печатает, в какой четверти она лежит.

Например

Ввод	Вывод
coordinate(1, 2)	I четверть

Задание 8. Предположим, что каждый напечатанный вами символ (в том числе и пробел) стоит 40 коп. Посчитайте, сколько стоит введенная вами стока.

Пример:

Ввод	Вывод
Привет! Как дела?	6 рублей 80 копеек

Подсказка. Для подсчета рублей и копеек нужно будет использовать операции получения целой части от деления и остатка от деления.

Задание 9. Напишите программу, которая на входе получает одно вещественное число, после чего выводит «положительное», «отрицательное» или «ноль», если это число – положительное, отрицательное или ноль, соответственно.

Пользователь вводит четные числа. Ввод чисел прекращается, если введено нечетное число. Необходимо определить, сколько чисел среди введенных заканчивались на ноль.

Пример:

Ввод	Вывод
2 4 10 20 0 100 1	Количество цифр, заканчивающихся на ноль, равно: 6

Задание 10. Задача по вложенным циклам.

Бывают ситуации, когда на горку в аквапарке стоит очередь, по технике безопасности спуск с горки в бассейн занимает 1 минуту. Необходимо написать программу, ведущую обратный отсчет для каждого посетителя в очереди, который начинается с 0 – от первого посетителя и удлиняется на минуту для следующего. На вход программы поступает количество человек в очереди, а на выходе – обратный отсчет по минутам для каждого в очереди.

Пример:

Ввод	Вывод
3	Посетителю 1 Осталось ждать 0 минут Посетителю 2 Осталось ждать 1 минут Осталось ждать 0 минут Посетителю 3 Осталось ждать 2 минут Осталось ждать 1 минут Осталось ждать 0 минут

Примеры заданий для самостоятельной работы

Тема 1.

Задания для самостоятельной работы.

1. Необходимо самостоятельно установить IDE Wing и PyCharm.
2. Написать простейшую программу, в которой будет содержаться приветствие от вашего лица. Например, «Здравствуйте, меня зовут Татьяна». Подсказка: print('Здравствуйте, меня зовут …').

– Необходимо реализовать эту программу в графическом пользовательском интерфейса python.

– Создать отдельный файл с расширением .py.

– Запустить созданный файл с помощью интерактивного окна и с помощью командной строки.

Тема 4.

Задания для самостоятельной работы.

1. Напишите простую функцию-приветствие. Эта функция выводит на экран приветствие: «Здравствуйте, (ИМЯ) (ФАМИЛИЯ)». Далее необходимо её вызвать:

```
print('Привет, как вас зовут?')
ваша_функция()
```

Пример

```
print('Добрый день, как вас зовут?')
…..(здесь стоит вызов вашей функции)
Здравствуйте, Ваше_имя Ваша_фамилия.
```

2. «Функция, которая здоровается»

Напишите функцию, которая читает имя пользователя из стандартного ввода, пока он не введет его в правильном формате, а затем здоровается. Приветствие должно быть в форме: **«Здравствуйте, {имя}!»**. После вывода надо не забыть перевести курсор на новую строку.

Если пользователь ввёл имя в неправильной форме, функция должна считать его снова. Корректным считается имя, в том виде, в котором мы его используем, то есть состоящее из одного слова, в котором нет символов кроме букв, первая буква заглавная, остальные - строчные.

Подсказка:

Можно воспользоваться:

– строковыми методами isalpha (проверка, что все символы в строке буквы), isupper() – приведены ли буквы/а к верхнему регистру).

– f-строкой, для заполнения имени

Тема 6.

Задания для самостоятельной работы.

1. Напишите функцию transformation(str, list), которая принимает два аргумента: строку str, содержащую числа, написанных через запятую и пробел, а также список list. Предполагается, что пользователь не будет передавать пустую строчку.

Функция должна извлечь из строки числа и добавить их в начало списка.

Пример:

Ввод	Вывод

`string = '1, 2, 3, 4, 5'`
`my_list = ['spam', 'spam2']`
`transformation(string, my_list)`
`print(*my_list)`
	1 2 3 4 5 spam spam2

Подсказка. Можно использовать метод списка insert. Чтобы список выводился без квадратных скобок, его можно распаковать оператором * (вспомните, мы об этом уже говорили):

print(*my_list) = >1 2 3 4 5 spam spam2

print(my_list) = > [1, 2, 3, 4, 5, 'spam', 'spam2']

2. Напишите функцию prompter(speech), которая будет повторять введенные реплики. Если в переменной speech находится предложение, которое уже встречалось хотя бы один раз, то функция печатает эту фразу.

Ввод	Вывод

`prompter('Привет!')`
`prompter('Как дела?')`
`prompter('Привет!')`
	`Привет!`

Тема 7.

Задания для самостоятельной работы.

1. Матрица

Напишите генератор матриц различного размера. Он представляет собой функцию matrix_generator(), которая создает, заполняет и возвращает матрицу. Функция содержит три аргумента, в зависимости от них, мы получаем разные результаты:

1) matrix_generator() – возвращает пустую матрицу [[]].

2) matrix_generator(arg1) – возвращает матрицу arg1 × arg1, заполненную единицами.

3) matrix_generator(arg1, arg2) – возвращает матрицу arg1 × arg2, заполненную единицами.

4) matrix_generator(arg1, arg2, arg3) – возвращает матрицу arg1 × arg2, заполненную значением arg3.

Пример:

Ввод	Вывод
print(matrix_generator())	[[]]
matrix_generator(5)	[1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1]
matrix_generator(2,3)	[1, 1, 1] [1, 1, 1]
matrix_generator(5,3, '*')	['*', '*', '*'] ['*', '*', '*'] ['*', '*', '*'] ['*', '*', '*'] ['*', '*', '*']

2. Позиционные и именованные аргументы, задача

Напишите функцию format_greeting, которая принимает следующие параметры:

name (обязательный позиционный аргумент) – имя человека,

age (обязательный позиционный аргумент) – возраст,

prefix (именованный аргумент, по умолчанию "Привет") – приветствие,

suffix (именованный аргумент, по умолчанию "!") – знак в конце приветствия,

uppercase (именованный аргумент, по умолчанию False) – если True, то весь результат возвращается в верхнем регистре.

Функция должна возвращать строку, вставляя имя и возраст в приветствие, например:

Ввод	Вывод
format_greeting("Иван", 30) print(format_greeting('Аня', 30, uppercase=True))	Привет, Иван! Тебе 30 лет! ПРИВЕТ АНЯ! ТЕБЕ 30 ЛЕТ.

Тема 8.

Задания для самостоятельной работы.

1. Вычисление Фибоначчи

Числа Фибоначчи – это последовательность чисел, в которой каждое следующее число является суммой двух предыдущих. Она начинается с 0 и 1, поэтому выглядит так: 0, 1, 1, 2, 3, 5, 8, 13, 21 и так далее.

Вспомним, что мы уже выводили ряд Фибоначчи с использованием кортежей:

```
n = int(input())
f1, f2 = 0, 1
for i in range(n):
    print(f1)
    f1, f2 = f2, f1 + f2
```

На вход функции поступает n, функция должна вернуть n-е число ряда фибоначи. Индексация в данном случае начинается с нуля.

База рекурсии: если n =0, то возвращаем 0, а если 1 – тогда возвращаем 1.

Шаг рекурсии: fibonacci(n-1) + fibonacci(n-2)

Пример:

Ввод	Вывод
print(fibonacci(6))	8

2. Линеаризация списка* (если останется время)

Напишите функцию linear(some_list), которая получает список произвольной вложенности, линеаризирует его и возвращает результат. Функция должна распаковать вложенный список — превратить список с вложенными списками в плоский одномерный список

Ввод	Вывод
`print(*linear([[1, 2], 4, [[2, [-4, "month", [0], [{2: "next"}]]]]]))`	1 2 4 2 -4 month 0 {2: 'next'}

Тема 9.

Задания для самостоятельной работы.

1. Установить библиотеку pillow, подготовить растровое изображение и протестировать на нём рассмотренные возможности python.

2. Нарисовать горящий фонарь в ночи с помощью функции def flashlight(file_name, width, height).

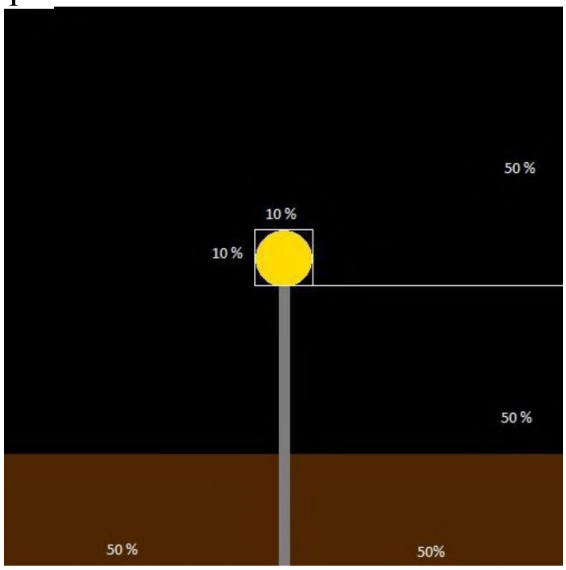
Небо черным цветом draw.rectangle, цвет "#000000".

Земля draw.rectangle, цвет "#4f2602" (высота может быть выбрана по вашему усмотрению, на примере она составляет 20% от высоты рисунка).

Столб фонаря draw.line, цвет серый.

Сам фонарь draw.ellipse, цвет "#FFDB00".

Допускается свой собственный цвет для столба, и оттенки для земли и фонаря.

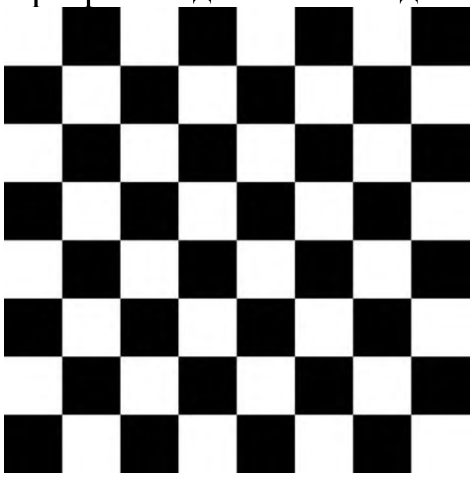


3. Создать поле для шахматной доски. Оно должно быть реализовано с помощью библиотеки PIL в функции chess_field('chess_field.jpg', 800, 800).

Так как размер поля 800 на 800, то квадратики (черные и белые), которыми будет заполняться поле, будут в 8 раз меньше основного поля. С помощью двойного цикла, задавая координаты верхнего правого и нижнего левого углов маленьких квадратиков, необходимо отрисовать (расположить) черно-белые квадратики на основном поле.

Белый цвет "#ffffff", черный "#000000".

Программа должна выводить следующее:



4. Изучить информацию о звуковых файлах из лекционного материала.

Тема 10.

Задания для самостоятельной работы.

Напишите класс MinMaxWordFinder. Класс должен уметь анализировать текст и находить в нём слова наименьшей и наибольшей длины. Текст состоит из предложений, которые добавляются в обработку методом add_sentence. Метод shortest_words возвращает список самых коротких на данный момент слов, метод longest_words — самых длинных. Слова, возвращаемые методами shortest_words и longest_words, должны быть отсортированы по алфавиту.

Если одно из самых коротких слов встретилось в исходных предложениях несколько раз, оно должно столько же раз повториться в списке самых коротких слов. Самые длинные слова, наоборот, должны входить в список без повторов.

Пример 1

Ввод	Вывод
finder = MinMaxWordFinder() finder.add_sentence('hello abc world') finder.add_sentence('def asdf qwert') print(' '.join(finder.shortest_words())) print(' '.join(finder.longest_words()))	abc def hello qwert world

Пример 2

Ввод	Вывод
finder = MinMaxWordFinder() finder.add_sentence('hello') finder.add_sentence('abc') finder.add_sentence('world') finder.add_sentence('def') finder.add_sentence('asdf') finder.add_sentence('qwert') print(' '.join(finder.shortest_words())) print(' '.join(finder.longest_words()))	abc def hello qwert world

Тема 11.

Задания для самостоятельной работы.

1. Задача на полиморфизм. Необходимо реализовать два класса Dog и Cat, а также написать методы sound() для каждого класса, которые будут выводить на печать звук животных. Для класса Dog – «Гав!», а для класса Cat – «Мяу!».

Необходимо также реализовать функцию make_sound(), которая принимает объект и вызывает метод sound этого объекта, демонстрируя полиморфизм, когда один и тот же вызов метода работает по-разному для разных классов.

Пример:

Ввод	Вывод

`dog = Dog("Шарик")`
`cat = Cat("Барсик")`
`make_sound(dog)`
`make_sound(cat)`
	`Шарик говорит: Гав!`
	`Барсик говорит: Мяу!`

2. Задача на иерархию классов

Добавим к классам Dog и Cat из предыдущего задания базовый класс Animal, от которого будут наследоваться классы Dog, Cat и новый класс Pantera – наследник класса Cat.

Интерфейс базового класса Animal состоит из двух методов: __init__(name), метод info(), который печатает название.

Интерфейс нового класса Pantera состоит из метода __init__ и нового метода fast_running(), внутри которого ничего реализовано не будет.

Необходимо реализовать иерархию классов

Пример

Ввод	Вывод

`pantera = Pantera("Багира")`
`print(pantera.name)`
`print(pantera.info())`
	`Багира`
	`Класс: Pantera`

Тема 12.

Задания для самостоятельной работы.

Название проекта «Менеджер личных финансов».

Цель проекта: получить опыт проектирования и реализации небольшого, но функционального приложения на Python с использованием структур данных, работы с файлами и организации пользовательского интерфейса в консоли.

Описание задачи:

Разработать программу на Python, которая позволит пользователю управлять своими личными финансами. Программа должна поддерживать учёт доходов и расходов, работу с несколькими категориями (например, «Продукты», «Транспорт», «Развлечения» и т. д.), а также формировать отчёты по заданным периодам.

Основные функции:

- Добавление новых записей о доходах и расходах с указанием суммы, категории и даты.
- Просмотр списка всех операций.
- Фильтрация операций по категории и/или периоду.
- Подсчёт общего баланса (доходы минус расходы).
- Формирование простого текстового отчёта за выбранный период.
- Сохранение данных в файл (например, CSV или JSON) и возможность их загрузки при запуске программы.

Дополнительные возможности:

- Реализация меню для удобной навигации по функциям.
- Валидация вводимых данных (например, сумма должна быть положительным числом).
- Возможность удаления или редактирования записей.

Пример:
Добро пожаловать в Менеджер личных финансов!
1. Добавить доход
2. Добавить расход
3. Показать все операции
4. Сформировать отчёт за период
5. Сохранить данные
6. Загрузить данные
7. Выход
Ваш выбор: 1
Введите сумму: 5000
Введите категорию: Зарплата
Введите дату (ГГГГ-ММ-ДД): 2026-04-10
Запись добавлена!

Примеры тестовых заданий

Раздел 1. Основы программирования на языке Python.

1. Языки программирования – это
1) искусственные языки
2) естественные языки
3) смешанные языки

2. Как называется средство для преобразования текстов из языков программирования высокого уровня в машинный код?
1) маршрутизатор
2) транслятор

3) автокод
4) контроллер

3. Величина, не изменяющая свое значение в процессе выполнения программы называется…
1) константа
2) переменная
3) функция
4) выражение

4. Как называется поименованная область памяти, предназначенная для временного хранения данных?
1) константа
2) переменная
3) функция
4) оперативная память

5. Программа – это
1) протокол взаимодействия компонентов компьютерной сети
2) набор команд операционной системы компьютера
3) алгоритм, записанный на языке программирования
4) законченное минимальное символьное выражение на языке программирования

6. Назначение массивов:
1) хранение данных
2) сортировка чисел
3) облегчение записи программы
4) упорядоченное хранение данных

7. Логический тип имеет значения:
1) да и нет
2) один или несколько
3) истина и ложь
4) истина

8. Строковый тип
1) хранит строку символов
2) хранит строку чисел
3) хранит число
4) хранит выражение

9. Переменные, которые доступны во всем проекте
1) переменные контейнера

2) логические
3) локальные
4) глобальные

10. Тип данных определяет…
1) множество значений, которые могут принимать объекты программы
2) множество значений, допустимых для операций программы
3) множество значений, которые могут принимать объекты программы, а также совокупность операций, допустимых над этими значениями
4) совокупность операций, допустимых над этими значениями

11. Какой оператор используется для ввода данных в Python?
1) input()
2) print()
3) def()
4) return()

12. Какие функции можно использовать для преобразования строки в число в Python?
1) int()
2) float()
3) str()
4) print()

13. Какие из перечисленных команд являются правильными?
1) name = input("Ваше имя? ")
2) x = int(input("Введите число: "))
3) print("Привет,", name, "!")
4) name = print("Ваше имя? ")

14. Какие операторы используются для преобразования данных?
1) int()
2) float()
3) str()
4) while()

15. Какие из следующих имён переменных допустимы?
1) user_name
2) user1
3) _username
4) 1user
5) name+user

Раздел 2. Объектно-ориентированное программирование (ООП).

1. Метод определения объектов, при котором производные объекты наследуют свойства от своих потомков:
1) монорфизм
2) полиморфизм
3) наследование
4) инкапсуляция

2. Свойство объектов, при котором действие с одинаковыми именами вызывает различное поведение для различных объектов:
1) полиморфизм
2) инкапсуляция
3) передача
4) монорфизм

3. Данные, характеризующие состояние объекта:
1) доли объекта
2) части объекта
3) атрибуты объекта

4. Совокупность объектов, характеризующаяся общностью методов и свойств:
1) класс
2) вид
3) род
4) тип

5. Действие, которое может выполнить объект:
1) метод
2) событие
3) свойство

6. Характеристика объекта:
1) Событие
2) Данность
3) Свойство

7. Способ защититься от использования объектов одного класса вместо другого, или по крайней мере управлять таким использованием – это
1) типизация
2) наследование
3) полиморфизм

4) инкапсуляция

8. В основе объектно-ориентированного подхода лежит понятие
1) класс
2) объект
3) инкапсуляция

9. Инкапсуляция – это
1) свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью
2) сущность в адресном пространстве вычислительной системы, выполняющая при создании экземпляра класса или копирования прототипа
3) свойство системы, позволяющее объединить данные и методы, работающие с ними в классе, и скрыть детали реализации от пользователя

10. Термин «наследование» обозначает, что …
1) в производных классах присутствует часть состояния родительского класса
2) класс B обладает всеми свойствами класса A
3) производные классы наследуют модификаторы доступа членов родительского класса

11. Как объявить класс в Python?
1) class MyClass {}
2) class MyClass:
3) def class MyClass:
4) object MyClass:

12. Какой метод вызывается при создании объекта?
1) __init__
2) __init__
3) __new__
4) __create__

13. Как обозначается приватный атрибут класса?
1) _attr
2) __attr
3) private attr
4) #attr

14. Что такое наследование?
1) Создание копии класса
2) Использование свойств родительского класса

3) Изменение существующего класса
4) Удаление методов класса

15. Как проверить, является ли объект экземпляром класса?
1) typeof(obj)
2) isinstance(obj, Class)
3) obj.type()
4) Class.check(obj)

Критерии оценки

- *Знания:* полнота и точность теоретических ответов, понимание терминологии.
- *Умения:* корректное применение инструментов Python для решения задач.
- *Навыки:* качество, читаемость и эффективность кода; самостоятельность при выполнении заданий; способность анализировать и интерпретировать результаты.

Такой подход обеспечивает объективную и всестороннюю оценку качества освоения программы, а также способствует формированию у слушателей устойчивых практических компетенций в области анализа данных с использованием Python.

Итоговая аттестация слушателей осуществляется в соответствии с локальными нормативными актами организации РАНХиГС. Итоговая аттестация проводится преподавателем(-ями), реализующим(и) данную программу.

Требования к итоговой аттестации по программе повышения квалификации «Базовый Python для анализа данных»

1. Цель итоговой аттестации

Итоговая аттестация проводится в форме зачёта для проверки и подтверждения уровня освоения слушателями программы, а также оценки сформированности у них базовых знаний, умений и практических навыков работы с языком Python для решения задач анализа данных.

2. Форма проведения

Итоговая аттестация проводится в комбинированной форме:

- Теоретическая часть — устный или письменный опрос по ключевым вопросам программы.
- Практическая часть — выполнение итогового практического задания или защита индивидуального проекта по анализу данных.

3. Критерии оценки

Критерий	Описание

Теоретические знания	Владение терминологией, понимание синтаксиса Python, принципов работы с данными, библиотек.
Практические умения	Способность самостоятельно выполнять типовые задачи: обработка данных, написание функций, визуализация.
Качество кода	Читаемость, структурированность, использование комментариев, соответствие стилю.
Аналитические навыки	Умение интерпретировать результаты анализа, делать выводы на основе данных.
Защита проекта	Краткое и логичное изложение хода решения задачи, ответы на вопросы преподавателя.

4. Порядок проведения зачёта

1. Слушатель получает задание (или представляет заранее подготовленный проект).
2. Выполняет практическую часть (при необходимости — на компьютере в присутствии преподавателя).
3. Защищает выполненную работу: объясняет выбранные методы, демонстрирует результаты, отвечает на вопросы.
4. Преподаватель оценивает работу по критериям и выставляет результат («зачтено»/ «не зачтено»).

5. Нормативы для допуска к итоговой аттестации

- Выполненные практические задания по всем разделам программы.
- Итоговый проект (отчёт по анализу данных с кодом и визуализациями).

6. Результаты итоговой аттестации

- «Зачтено» — слушатель продемонстрировал достаточный уровень теоретических знаний и практических навыков для самостоятельной работы с Python в задачах анализа данных.
- «Не зачтено» — слушатель не справился с практической частью или не ответил на основные вопросы теории. В этом случае предоставляется возможность пересдачи после дополнительной подготовки.

7. Примерные вопросы для теоретической части

- Основные типы данных в Python.
- Работа с библиотеками *Pandas* и *NumPy*.
- Основы визуализации данных (*Matplotlib*).
- Алгоритмы обработки и очистки данных.

8. Пример итогового задания

Провести анализ предоставленного набора данных: загрузить данные, провести предварительную обработку, выполнить основные аналитические операции, построить визуализации и сформулировать выводы по результатам анализа.

5. ИНДИКАТОРЫ СФОРМИРОВАННЫХ КОМПЕТЕНЦИЙ ВЫПУСКНИКА ПРОГРАММЫ

В результате освоения программы у слушателя сформированы компетенции:

Таблица 5

Компетенция (код, содержание)	Индикаторы
ПСК-1. Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных.	Профессиональные (технические) компетенции: • Умение разрабатывать программы на языке Python. Например, умение: создавать программы и корректировать их; составлять алгоритмы для решения задач; реализовывать алгоритмы на компьютере в виде программ, написанных на Python; отлаживать и тестировать программы, написанные на Python. Исследовательские навыки: • Понимание того, как проанализировать качество созданного программного продукта. Аналитические навыки: • Знания основ языка, работа с библиотеками для анализа данных. Навыки программирования: • Умение применять изученные методы и структуры данных в соответствии с технологией разработки программ; • Умение обобщать, систематизировать материал и применять его при решении практических заданий.
ПСК-2. Оформление программного кода в соответствии с установленными требованиями	Следование стандартам: • Знание основ основных стандартов объектно-ориентированного программирования и умение следовать им при создании своего программного продукта.