

Министерство науки и высшего образования Российской Федерации

**Министерство науки, высшего образования и инноваций Кыргызской
Республики**

**Межгосударственная образовательная организация высшего
образования Кыргызско-Российский Славянский университет имени
первого Президента Российской Федерации Б.Н. Ельцина.**

Фонд оценочных средств

по дисциплине (практике) Прикладное программирование

Уровень высшего образования БАКАЛАВРИАТ

Направление:

23.03.01 - РФ, 670300 - КР Технология транспортных процессов

23.03.03 - РФ, 670200 - КР Эксплуатация транспортно-технологических машин и комплексов, профиль "Автомобильный сервис"

Квалификация Бакалавр

Бишкек 2025г.

Фонд оценочных средств предназначен для контроля знаний обучающихся по направлению подготовки 23.03.01 «Технология транспортных процессов», 23.03.03 «Эксплуатация транспортно-технологических машин и комплексов», профиль "Автомобильный сервис" по дисциплине «Прикладное программирование»

Фонд оценочных средств рассмотрен и утвержден на заседании кафедры

Прикладная математика и информатика

Наименование кафедры

Протокол №1 от «28» 08. 2025г.

Заведующий кафедрой

Прикладная математика и информатика



Аширбаев Бейшембек Ыбышевич

наименование кафедры

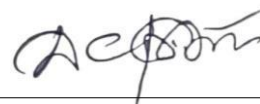
подпись

расшифровка подписи

Руководитель образовательной программы

Доулбекова Салтанат Байызбековна

Доцент



ФИО

должность

подпись

Исполнители:

к.ф.-м.н., доцент кафедры ПМИ



М.Ж. Нарматова

должность

подпись

расшифровка подписи

Доцент



Джаналиева Жылдыз Рахманкуловна

должность

подпись

расшифровка подписи

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Формируемые компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций	Виды оценочных средств/ шифр раздела в данном документе
<p>ОПК-3: Способен в сфере своей профессиональной деятельности проводить измерения и наблюдения, обрабатывать и представлять экспериментальные данные и результаты испытаний;</p>	<p>Знать: Уровень 1 Основные методы решения задач алгоритмизации и программирования производственно-технологической, расчетно-проектной, экспериментально-исследовательской деятельности. Уровень 2 Возможности пакета прикладных программ MatLab для проведения инженерных и технических расчетов Уровень 3 Методы структурного программирования MatLab.</p>	<p>Тестирование; Выполнение лабораторных работ; Реферат; Презентации.</p>
	<p>Уметь: Уровень 1 использовать возможности вычислительной техники и программного обеспечения для решения расчетных задач. Использовать возможности вычислительной техники и программного обеспечения для решения расчетных задач. Уровень 2 Работать в среде MatLab и Simulink. Уровень 3 Решать математические задачи и строить 2D и 3D графики в MatLab.</p>	<p>Тестирование; Выполнение лабораторных работ; Реферат; Презентации.</p>
	<p>Владеть: Уровень 1 Высокоуровневым языком программирования пакета MATLAB для решения расчетных задач с использованием простейших алгоритмов.</p>	<p>Тестирование; Выполнение лабораторных работ; Реферат; Презентации.</p>

2. ТЕХНОЛОГИЧЕСКАЯ КАРТА ДИСЦИПЛИНЫ/ПРАКТИКИ

Технологическая карта дисциплины
«Прикладное программирование»

Курс/семестр: 2/4

Количество кредитов (ЗЕ): 2

Отчетность: Зачет с оценкой

Название модулей дисциплины согласно РПД	Контроль	Форма контроля	зачетный минимум	зачетный максимум	график контроля
Модуль 1					
Основные сведения о Python.	Текущий контроль	Изучение теории по теме практической работы, выполнение практических работ по данному разделу	7	10	32
	Рубежный контроль	Контрольная работа №1	13	25	
Модуль 2					
Расчеты в Python.	Текущий контроль	Изучение теории по теме практической работы, выполнение практических работ по данному разделу	7	10	37
	Рубежный контроль	Контрольная работа №2	13	25	
ВСЕГО за семестр			40	70	
Промежуточный контроль (Зачет с оценкой)			20	30	
Семестровый рейтинг по дисциплине			60	100	

3. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ И ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ / ПРАКТИКЕ (ОЦЕНОЧНЫЕ СРЕДСТВА)

п/п	Наименование оценочного средства	Краткая характеристика оценочного средства
1.	Презентации	Работы, направленные на выполнение комплекса учебных и исследовательских заданий. Позволяют оценить умения обучающихся самостоятельно конструировать свои знания в процессе решения учебных задач, ориентироваться в информационном пространстве и уровень сформированности аналитических, исследовательских навыков, навыков практического и творческого мышления.
2.	Реферат	Продукт самостоятельной работы обучающегося, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной учебно-исследовательской темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее.

3.	Доклад, сообщение	Расширенное письменное или устное сообщение на основе совокупности ранее опубликованных исследовательских, научных разработок, по соответствующей отрасли научных знаний, имеющих значение для теории науки и практического применения. Представляет собой обобщённое изложение результатов проведённых исследований, экспериментов и разработок, известных широкому кругу специалистов в отрасли научных знаний.
4.	Контрольная работа	Контрольные работы проводятся с целью определения конечного результата в обучении по данной теме или разделу, позволяют контролировать знания одного и того же материала еднократно. Целесообразно проводить контрольные работы различного вида. С помощью промежуточной контрольной работы проверяется усвоение обучающимися материала в период изучения темы. Итоговая контрольная работа проводится с целью проверки знаний и умений по отдельной теме, курсу. Домашняя контрольная работа призвана систематизировать знания, позволяет повторить и закрепить материал. При ее выполнении обучающиеся не ограничены временем, могут использовать любые учебные пособия. Каждому обучающемуся дается свой вариант работы, в который включаются творческие задания для формирования обозначенных компетентностей.
5.	Конспект	Конспект позволяет формировать и оценивать умения обучающихся по переработке информации.
6.	Практическая, лабораторная работа	Практическая работа — это задание для студента, которое должно быть выполнено потеме, определенной преподавателем. Предполагается также использование рекомендованной им литературы при подготовке к практической работе и плана изучения материала. Рассматриваемое задание в ряде случаев включает дополнительную проверку знаний студента – посредством тестирования или, например, написания контрольной работы. Главная цель проведения практической работы заключается в выработке у студента практических умений, связанных с обобщением и интерпретацией тех или иных научных материалов. Кроме того, ожидается, что результаты практических занятий будут впоследствии использоваться обучающимися для освоения новых тем.
7.	Проект	Проект представляет собой учебный проект, выполняемый обучающимся в рамках одного или нескольких учебных предметов с целью продемонстрировать свои достижения в самостоятельном освоении содержания и методов избранных областей знаний и/или видов деятельности и способность проектировать и осуществлять целесообразную и результативную деятельность (учебно-познавательную, конструкторскую, социальную, художественно-творческую).
8.	Кейс-задание	Проблемное задание, в котором обучающемуся предлагают осмыслить реальную профессионально-ориентированную ситуацию, необходимую для решения данной проблемы.
9.	Тест	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.
10.	Дифференцированный зачёт	Контрольное мероприятие, которое проводится по окончании изучения дисциплины в виде, предусмотренном учебным планом.

3. МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ, ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

Контроль и оценка результатов освоения общеобразовательной дисциплины раскрываются через дисциплинарные результаты, направленные на формирование общих и профессиональных компетенций по разделам и темам содержания учебного материала. **Контроль и оценка результатов** освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий и лабораторных работ, тестирования, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Название модулей дисциплины	Форма контроля	Тип оценочных мероприятий
Основные сведения о Python.	изучение теории по теме практической работы, выполнение практических заданий по данному разделу.	Выполнение практических, Лабораторных заданий, оценка результатов устных ответов, решения задач (в том числе профессиональноориентированных)
Расчеты в Python.	изучение теории по теме практической работы, выполнение практических заданий по данному разделу.	Выполнение практических, Лабораторных заданий, оценка результатов устных ответов, решения задач (в том числе профессиональноориентированных)
Дифференцированный зачет	По всем разделам	Выполнение и защита проекта, в том числе профессиональноориентированных

Выполнение практических, Лабораторных заданий. Расчеты в Python.

Предлагается к выполнению 8 практических работ по вариантам каждому студенту.

За каждое правильно выполненную практическую работу дается 7 баллов, за 1 и 2 по 6 баллов.

Максимальная сумма баллов – 54.

Проект. Проект представляет собой учебный проект, выполняемый обучающимся в рамках одного или нескольких учебных предметов с целью продемонстрировать свои достижения в самостоятельном освоении содержания и методов избранных областей знаний и/или видов деятельности и способность проектировать и осуществлять целесообразную и результативную деятельность (учебно-познавательную, конструкторскую, социальную, художественно-творческую).

Максимальная сумма баллов – 16.

ПРЕЗЕНТАЦИЯ

Требования к презентации:

Титульный лист с заголовком

Дизайн слайдов и использование дополнительных эффектов

(смена слайдов, звук, анимации, переходы, видео, графики при необходимости)

Текст презентации написан коротко, представлены ключевые идеи каждой главы и каждого параграфа.

Сформированность идей, их ясное изложение и структурирование

Наличие собственных выводов и замечаний по соответствующему показателю

Максимальная сумма баллов – 20.

Реферат

СТРУКТУРА И ОФОРМЛЕНИЕ РАБОТЫ

1 Структура работы в соответствии с требованиями

- Титульный лист
- Содержание
- Введение
- Развитие темы (тезиса) в основной части (раскрытие основных положений через систему аргументов, подкрепленных фактами, примерами и т.д.)
- Заключение
- Список используемой литературы согласно ГОСТа, не ранее 2010 гг.

2 Оформление работы в соответствии с требованиями

Times New Roman, №14, интервал -1,5. Поля – обычные.

Не менее 10 листов формата А-4

Всего баллов Сумма баллов – 10

4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩЕГОСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ / ПРАКТИКИ И ВЫПОЛНЕНИЮ КОНТРОЛЬНЫХ ЗАДАНИЙ

МОДУЛЬНЫЙ КОНТРОЛЬ ПО ДИСЦИПЛИНЕ ВКЛЮЧАЕТ:

1. Текущий контроль: усвоение учебного материала на аудиторных занятиях (лекциях, практических занятиях, в том числе учитывается посещение и активность) и выполнение обязательных заданий для самостоятельной работы.
2. Рубежный контроль: проверка полноты знаний и умений по материалу модуля в целом. Выполнение модульных контрольных заданий проводится в письменном виде и является обязательной компонентой модульного контроля. К выполнению РК студент допускается всегда, независимо от посещаемости и выполнения других видов учебной работы.
3. Промежуточный контроль - завершенная задокументированная часть учебной дисциплины (или вся дисциплина полностью) – совокупность тесно связанных между собой зачетных модулей.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНЫХ РАБОТ СТУДЕНТОВ

1. Советы по планированию и организации времени, необходимого для изучения дисциплины. Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины: Изучение конспекта лекции в тот же день, после лекции – 10-15 минут. Изучение конспекта лекции за день перед следующей лекцией – 10-15 минут. Изучение теоретического материала по учебному пособию и конспекту – 1 час в неделю. Подготовка к практическому занятию – 2-3 час. Всего в неделю – 4 часа.
2. Описание последовательности действий студента Для понимания материала и качественного его усвоения рекомендуется такая последовательность действий:
 1. После прослушивания лекции и окончания учебных занятий, при подготовке к занятиям следующего дня, нужно сначала просмотреть и обдумать текст лекции, прослушанной сегодня (10-15 минут).
 2. При подготовке к лекции следующего дня, нужно просмотреть текст предыдущей лекции, подумать о том, какая может быть тема следующей лекции (10-15 минут).
 3. В течение недели выбрать время (2-3 часа) для работы с рекомендуемыми электронными учебными пособиями.
 4. При подготовке к практическим занятиям следующего дня, необходимо сначала прочитать основные понятия и подходы по теме домашнего задания. При выполнении упражнения или задачи нужно сначала понять, что требуется в задаче, какой теоретический материал нужно использовать, наметить план решения задачи.

3. Рекомендации по использованию материалов учебно-методического комплекса. Все рекомендуемые учебные пособия размещены на серверах компьютерных классов ИВТ в сетевой папке.
4. Рекомендации по работе с литературой. Теоретический материал курса становится более понятным, когда УП: b23030130_21_1тгп.rlx стр. 11 дополнительно к прослушиванию лекции и изучению конспекта, изучаются и книги. Легче освоить курс, придерживаясь одного учебника и конспекта. Рекомендуется, кроме «заучивания» материала, добиться состояния понимания изучаемой темы дисциплины. С этой целью рекомендуется после изучения очередного параграфа выполнить несколько простых упражнений на данную тему. Кроме того, очень полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): о чем этот параграф?, какие новые понятия введены, каков их смысл?, что даст это на практике?.
5. Советы по подготовке к рубежному и промежуточному контролю. Рубежный контроль проходит в виде тестов, контрольных и самостоятельных работ. Промежуточный контроль по данной дисциплине проходит в виде экзамена. Дополнительно к изучению конспектов лекции необходимо пользоваться учебником. При подготовке к промежуточному контролю нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить несколько типовых задач из каждой темы. При решении задач всегда необходимо уметь качественно интерпретировать итог решения.
6. Указания по организации работы с контрольно-измерительными материалами, по выполнению домашних заданий. При выполнении домашних заданий необходимо сначала прочитать основные понятия и подходы по теме задания. При выполнении упражнения или задачи нужно сначала понять, что требуется в задаче, какой теоретический материал нужно использовать, наметить план решения задачи, а затем приступить к расчетам и сделать качественный вывод.

Вопросы для проверки уровня обученности

Знать

1. Что такое m-файл?
2. Что такое mat-файл?
3. Что такое рабочее пространство?
4. Как сохранить числовые данные, определенные в рабочем пространстве?
5. Как загрузить числовые данные из файла в рабочее пространство?
6. Как определить текущую директорию и ее содержимое?
7. Как узнать, какие переменные определены в рабочем пространстве?
8. Как очистить рабочее пространство и командное окно?
9. Как узнать содержание m-файла?
10. Matlab. Справочные и управляющие команды и функции.
11. Управление переменными и рабочей средой
12. Работа с файлами и операционной системой.
13. Операторы, константы, служебные символы и переменные.
14. Арифметические операторы.
15. Операторы сравнения. Специальные символы.
16. Переменные. Константы.
17. Логические операторы. Операторы цикла.
18. Синтаксис, использование Условные операторы. Оператор ввода информации.
19. Элементарные алгебраические функции. Тригонометрические функции. Специальные математические функции.
20. Формирование векторов и подматриц. Матричные функции. Операции над матрицами.
21. Функции построения и оформления двумерных графиков.

Примерный перечень заданий для проверки уровней обученности Уметь и Владеть

1. Расставьте следующие операции в порядке возрастания их приоритетов: сложение (+), возведение в степень (^), логическое ИЛИ (||), поэлементное деление (./), унарный минус (-), меньше (<), логическое отрицание (~).
2. Перепишите выражение $(a*b \sim c) + (a > c) + (a^c <= b)$, используя вместо операторов отношения соответствующие функции отношения.
3. Проверьте правильность выполнения полученного выражения, присвоив переменным a и b конкретные числовые значения
4. Выполнить указанные преподавателем операции над матрицами
5. По заданию преподавателя составить программу с использованием вложенных циклов
6. По заданию преподавателя составить программу с использованием условных операторов
7. По заданию преподавателя составить программу с использованием векторных и матричных переменных
8. По заданию преподавателя составить программу с использованием специальных математических функций
9. По заданию преподавателя составить программу с логических функций
10. По заданию преподавателя составить программу с использованием операторов условного и безусловного перехода
11. По заданию преподавателя составить программу с использованием в комбинации операторов цикла и ветвления
12. По заданию преподавателя составить программу с использованием среды Simulink
13. По заданию преподавателя построить и оформить двумерные графики
14. По заданию преподавателя построить и оформить 3D графики
15. По заданию преподавателя построить и оформить комбинированные графики

Примерные задания для проведения контрольных работ:

Контрольная работа № 1. Прикладное программирование.

Модуль 1. Напишите: группа _____ ФИО _____

1. Возможности пакета Matlab и сфера использования: _____
2. Команда help - _____
3. Для загрузки переменных рабочей области используется команда - _____ при этом по умолчанию переменные загружаются из файла: _____
4. Режимы выполнения операций в Matlab - _____
5. Напишите назначение спец символов и операторов: = () % : , 6. Функция sum - _____
7. == - это _____
8. ~ - это _____

9. Функция `get(x, y)` -

10. Для управления режимом сохранения текущего графического окна используется:
_____ Синтаксис:

11. `contour3` _____

12. Перечислите функции Matlab для работы с графиками:

13. Напишите назначение операторов и функций: `loglog legend graficn tril prod`

14. - -

это _____

Контрольная работа № 2.

1. Виды управления в программах -

2. `switch` – синтаксис и

использование: _____

3. Оператор цикла с предусловием (синтаксис и использование):

4. оператор `break` используется для

5. `.*` -

это _____

6. Операторы отношения и выполняемые ими операции:

7. Оператор цикла со счетчиком (синтаксис и использование):

8. `meshgrid` -

9. Условные (ветвления) операторы в Matlab - синтаксис и использование:

10. `diag` –

11. `m` –файлы используются

для: _____

12. Напишите назначение операторов и функций: `switch input ones break disp`

13. Перечислите функции Matlab для работы с трехмерными графиками:

Практическая работа 1

Введение в язык программирования Python

Цель работы: Познакомиться со средой разработки Python. Изучить основные типы данных, команды ввода и вывода данных.

Язык программирования Python был создан в конце 1980-х Гвидо ван Россумом в Нидерландах как преемник языка ABC. Впервые он был представлен в 1991 году с выпуском версии 0.9.0. С тех пор Python претерпел множество изменений и улучшений, вырос в популярности и теперь занимает одно из ведущих мест среди языков программирования в мире. Ван Россум продолжал быть ведущим автором языка до 2018 года, поддерживая философию простоты и читаемости кода.

Python поддерживает несколько парадигм программирования, включая процедурное, объектно-ориентированное и функциональное программирование. Он оснащен динамической типизацией, автоматическим управлением памятью и полной стандартной библиотекой, которая автоматически включается во все установки. Используется в самых разнообразных областях, от веб-разработки до научных исследований.

Основные характеристики языка программирования Python

Характеристика	Описание
Простота	Python разработан с упором на читаемость кода, благодаря чему программирование на нем легко изучать и использовать.
Высокоуровневый	Язык абстрагируется от деталей компьютерной архитектуры, позволяя разработчикам сосредоточиться на логике программы.
Интерпретируемый	Код Python выполняется построчно, что упрощает отладку программ и ускоряет процесс разработки.
Мультипарадигменный	Поддерживает объектно-ориентированное, процедурное и функциональное программирование, предоставляя разработчикам гибкость в выборе подхода.
Динамическая типизация	Типы переменных определяются автоматически во время выполнения. Это делает код более компактным и уменьшает количество требуемого кода.
Встроенная обширная стандартная библиотека	Обширная стандартная библиотека предлагает модули и функции для различных задач, от веб-разработки до науки о данных.
Переносимость	Программы на Python выполняются на многих операционных системах без изменений кода.
Расширяемость	Возможность интеграции с другими языками программирования, такими как C и C++, для оптимизации производительности и использования специализированных библиотек.

В комплекте вместе с интерпретатором Python идет IDLE (интегрированная среда разработки). По своей сути она подобна интерпретатору, запущенному в интерактивном режиме с расширенным набором возможностей (подсветка синтаксиса, просмотр объектов, отладка и т.п.).

Для запуска IDLE в Windows необходимо перейти в папку Python в меню “Пуск” и найти там ярлык с именем “IDLE (Python 3.X XX-bit)”.

Для запуска редактора программы (кода) следует выполнить команду File->New File или сочетание клавиш Ctrl+N.

Любая Python-программа состоит из последовательности допустимых символов, записанных в определенном порядке и по определенным правилам.

Программа включает в себя:

- комментарии;
- команды;
- знаки пунктуации;
- идентификаторы;
- ключевые слова.

Комментарии необходимы для пояснения кода или временного отключения определенных участков.

Комментарии в Python обозначаются символом # и продолжаются до конца строки, при этом не допускается использование перед символом # кавычек.

Многострочные комментарии заключаются в три апострофа: ““Комментарий...””

Особенностью языка является выделение блоков кода отступами. Для каждого уровня отступов необходимо использовать по 4 пробела. Конец строки является концом инструкции. Вложенные инструкции объединяются в блоки по величине отступов.

Знаки пунктуации

В алфавит Python входит достаточное количество знаков пунктуации, которые используются для различных целей. Например, знаки "+" или "*" могут использоваться для сложения и умножения, а знак запятой "," - для разделения параметров функций.

Идентификаторы

Идентификаторы в Python это имена используемые для обозначения переменной, функции, класса, модуля или другого объекта.

Ключевые слова

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Переменные объявляются присваиванием значения. Тип переменной определяется автоматически во время выполнения программы.

Типы данных

1. None (неопределенное значение переменной)
2. Логические переменные (Boolean Type)
3. Числа (Numeric Type)
 1. int – целое число
 2. float – число с плавающей точкой
 3. complex – комплексное число
4. Списки (Sequence Type)
 1. list – список
 2. tuple – кортеж
 3. range – диапазон
5. Строки (Text Sequence Type)
 1. str

Управляющие - esc последовательности

В наборе символов, использующихся в компьютере, есть символы, воспринимающиеся как команды или интерпретирующиеся особым образом.

Escape-символ или последовательность - это знак, что последующий символ или определенное количество символов должны интерпретироваться не так, как обычно. Иногда их называют экранирующие последовательности, так как они помогают экранировать какие-либо символы

Для введения управляющих последовательностей, позволяющих получить наглядное представление некоторых символов, не имеющих графического аналога, *используется косая черта за которой следует символ:*

- \a` - звуковой сигнал встроенного динамика,
- \b` - возврат на шаг, Backspace, стираем символ слева
- \f` - перевод формата,
- \n` - переход на новую строку,
- \r` - возврат каретки,
- \` - апостроф, одинарная кавычка
- \` - двойные кавычки,
- \v` - вертикальная табуляция,
- \t` - добавляет горизонтальную табуляцию (4 отступа),
- \` - слеш - обратная косая черта,
- \?` - вопросительный знак,
- \0` - пустой символ Null (не является признаком конца строки)
- \0ddd` - восьмеричная константа,
- \0xdd` - шестнадцатеричная константа,
- \uhhhh` 16-битовый символ Юникода в 16-ричном представлении.

Синтаксис Python

- Конец строки является концом инструкции (точка с запятой не требуется).
- Вложенные инструкции объединяются в блоки по величине отступов. Отступ может быть любым, главное, чтобы в пределах одного вложенного блока отступ был одинаков. И про читаемость кода не забывайте. Отступ в 1 пробел, к примеру, не лучшее решение. Используйте 4 пробела (или знак табуляции, на худой конец).

- Вложенные инструкции в Python записываются в соответствии с одним и тем же шаблоном, когда основная инструкция завершается двоеточием, вслед за которым располагается вложенный блок кода, обычно с отступом под строкой основной инструкции.

Основная инструкция:
Вложенный блок инструкций

Ввод и вывод данных

Ввод данных `input()`

Для организации ввода данных с клавиатуры в языке Python используется функция `input()`. По умолчанию все значения принимают строковый тип.

Синтаксис:

```
имя_переменной = input([строка-приглашение])
```

- где `имя_переменной` – идентификатор переменной в которую будет записано вводимое значение;
- `строка-приглашение` – необязательный параметр, сообщение, которое будет выведено на экран при выполнении программы.

`a = input()`

В скобках функции можно указать сообщение - комментарий к вводимым данным:

```
a = input("Введите количество: ")
```

Команда `input()` по умолчанию воспринимает входные данные как строку символов. Поэтому, чтобы ввести целочисленное значение, следует указать тип данных `int()`:

```
a = int(input())
```

Для ввода вещественных чисел применяется команда

```
a=float(input())
```

Вывод данных `print()`

Функция `print` Python выводит заданные объекты на стандартное устройство вывода (экран) или отправляет их текстовым потоком в файл. Синтаксис:

```
print(*args[, sep=' ', end='\n', file=sys.stdout])
```

где `*args` – выводимые объекты, `print(<выражение 1> , <выражение 2> , ..., <выражение N> , ...`

`sep` – разделитель между выводимыми значениями (по умолчанию – пробел);

`end` – символ, которым заканчивается вывод (по умолчанию – символ новой строки);

`file` – объект, в который мы можем перенаправить вывод, который по умолчанию производится в `sys.stdout`.

- Каждая последующая команда `print()` выводит указанный текст с новой строки.
- Команда `print()` с пустым списком аргументов просто вставляет новую пустую строку.

Примеры:

```
# 1 - вывод одной переменной строкового типа
```

```
a = "Hello" print(a)
```

Результат:

Hello

```
# 2 - вывод нескольких переменных строкового типа через пробел
```

```
a = "A"
```

```
b = "B"
```

```
c = "C"
```

```
print(a,b,c)
```

Результат:

ABC

```
# 3 - вывод нескольких переменных строкового типа через знак
```

```
# с помощью параметра sep
```

```
a = "A"
```

```
b = "B"
```

```
c = "C"
```

```
print(a,b,c,sep="#")
```

Результат:

A#B#C#

```
a = 1
```

```
b = 2
```

```
print(a)
```

```
print(a + b)
```

```
print('сумма = ', a + b)
```

Существует возможность записи команд в одну строку, разделяя их через `;`. Однако не следует часто использовать такой способ, это снижает удобочитаемость:

```
a = 1; b = 2; print(a)
```

```
print (a + b)
```

```
print ('сумма = ', a + b)
```

Для команды **print** может задаваться так называемый сепаратор — разделитель между элементами вывода:

```
x=2
```

```
y=5
```

```
print ( x, "+", y, "=", x+y, sep = " ")
```

Результат отобразится с пробелами между элементами: $2 + 5 = 7$

Для форматированного вывода используется format:

Строковый метод `format()` возвращает отформатированную версию строки, заменяя идентификаторы в фигурных скобках `{}`. Идентификаторы могут быть позиционными, числовыми индексами, ключами словарей, именами переменных.

Синтаксис команды **format**:

поле замены := "{" [имя поля] ["!" преобразование] [":" спецификация] "}"

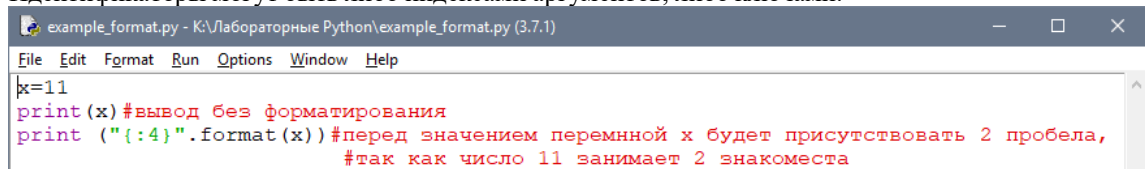
имя поля := `arg_name` ("." имя атрибута | "[" индекс "]")*

преобразование := "r" (внутреннее представление) | "s" (человеческое представление)

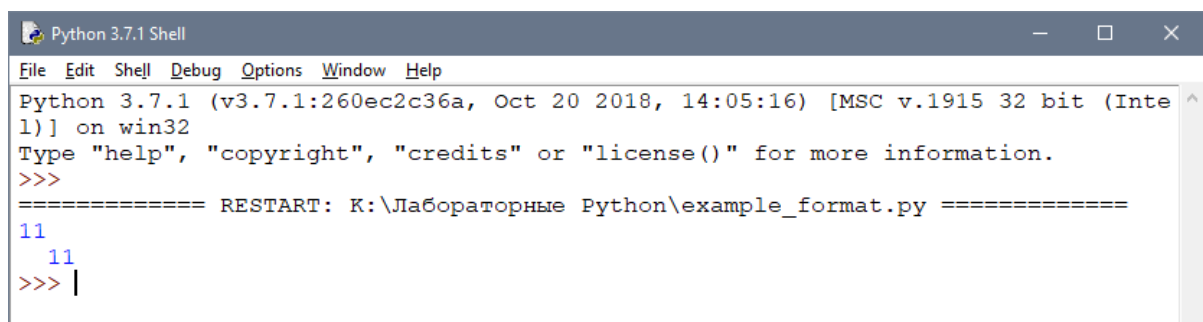
спецификация := см. ниже

Аргументов в `format()` может быть больше, чем идентификаторов в строке. В таком случае оставшиеся игнорируются.

Идентификаторы могут быть либо индексами аргументов, либо ключами:



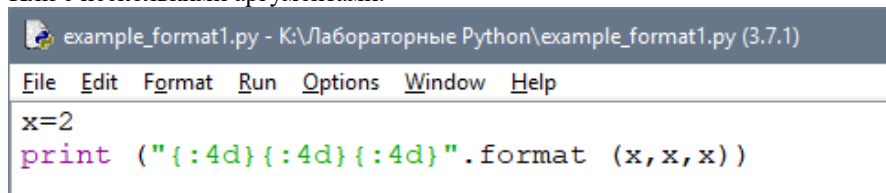
```
example_format.py - K:\Лабораторные Python\example_format.py (3.7.1)
File Edit Format Run Options Window Help
x=11
print(x) #вывод без форматирования
print ("{:4}".format(x)) #перед значением переменной x будет присутствовать 2 пробела,
                        #так как число 11 занимает 2 знакоместа
```



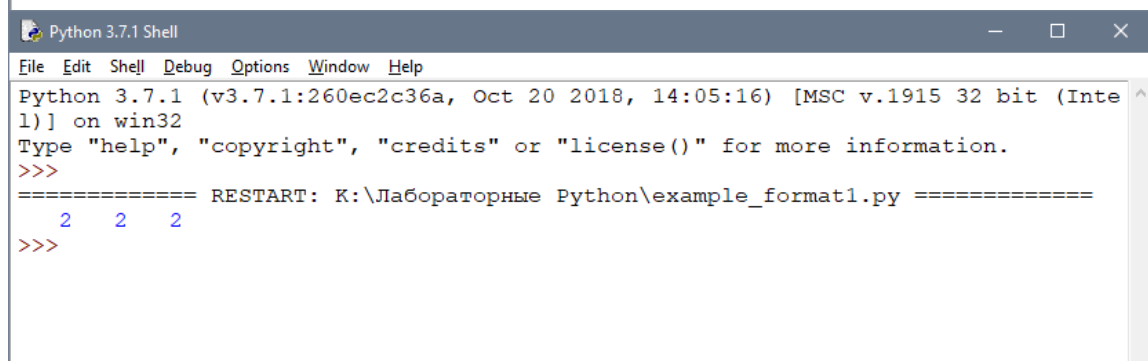
```
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: K:\Лабораторные Python\example_format.py =====
11
  11
>>> |
```

В результате выведется число 11, а перед ним два пробела, так как указано использовать для вывода четыре знакоместа.

Или с несколькими аргументами:



```
example_format1.py - K:\Лабораторные Python\example_format1.py (3.7.1)
File Edit Format Run Options Window Help
x=2
print ("{:4d}{:4d}{:4d}".format (x, x, x))
```



```
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: K:\Лабораторные Python\example_format1.py =====
  2  2  2
>>>
```

В итоге каждое из значений выводится из расчета 4 знакоместа.

Спецификация формата:

спецификация	:= [[fill]align][sign][#][0][width][.][precision][type]
заполнитель	:= символ кроме '{' или '}'
выравнивание	:= "<" ">" "=" "^"
знак	:= "+" "-" " "
ширина	:= integer
точность	:= integer
тип	:= "b" "c" "d" "e" "E" "f" "F" "g" "G" "n" "o" "s" "x" "X" "%"

Тип	Значение
'd', 'i', 'u'	Десятичное число.
'o'	Число в восьмеричной системе счисления.
'x'	Число в шестнадцатеричной системе счисления (буквы в нижнем регистре).
'X'	Число в шестнадцатеричной системе счисления (буквы в верхнем регистре).
'e'	Число с плавающей точкой с экспонентой (экспонента в нижнем регистре).
'E'	Число с плавающей точкой с экспонентой (экспонента в верхнем регистре).
'f', 'F'	Число с плавающей точкой (обычный формат).
'g'	Число с плавающей точкой. с экспонентой (экспонента в нижнем регистре), если она меньше, чем -4 или точности, иначе обычный формат.
'G'	Число с плавающей точкой. с экспонентой (экспонента в верхнем регистре), если она меньше, чем -4 или точности, иначе обычный формат.
'c'	Символ (строка из одного символа или число - код символа).
's'	Строка.
'%'	Число умножается на 100, отображается число с плавающей точкой, а за ним знак %.

Для форматирования вещественных чисел с плавающей точкой используется следующая команда:

```
print('{0:.2f}'.format(вещественное число))
```

```
format_chisla.py - K:/Лабораторные Python/format_chisla.py (3.7.1)
File Edit Format Run Options Window Help
x=10
y=7
print("{0:.2f}".format(x/y))
```

В результате выведется число с двумя знаками после запятой.

```
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 1) on win32
Type "help", "copyright", "credits"
>>>
===== RESTART: K:/Лаборатор
1.43
>>> |
```

Пример

Напишите программу, которая запрашивала бы у пользователя:

- ФИО ("Ваши фамилия, имя, отчество?")
- возраст ("Сколько Вам лет?")
- место жительства ("Где вы живете?")

После этого выводила бы три строки:

"Ваше имя"

"Ваш возраст"

"Вы живете в"

Решение

```
a=input('Введите ваши фамилию, имя, отчество ')
b=input('Сколько вам лет? ')
c=input('Где вы живёте? ')
print('Ваше имя ',a)
print('Ваш возраст ',b)
print('Вы живете в ',c)
```

```
Введите ваши фамилию, имя, отчество Иванов Иван Иванович
Сколько вам лет? 15
Где вы живёте? Уссурийск
Ваше имя Иванов Иван Иванович
Ваш возраст 15
Вы живете в Уссурийск
```

Задания для самостоятельной работы

1. История создания и автор Python.
2. Преимущества и недостатки Python.
3. Известные программы на Python, применение Python.

по вариантам:

Напишите программу, которая запрашивала бы у пользователя:

Вариант 1

Имя, Фамилию, Возраст, Место жительства

- фамилия, имя ("Ваши фамилия, имя?")
- возраст ("Сколько Вам лет?")
- место жительства ("Где вы живете?")

После этого выводила бы три строки:

"Ваши фамилия, имя"

"Ваш возраст"

"Вы живете в"

Также выводила бы эту информацию в одну строку, используя любые три символа в качестве разделителя этих строк.

Вариант 2

Имя, Дату рождения, Образование

- имя ("Ваше, имя?")
- дата рождения ("Ваша дата рождения?")
- образование ("Где Вы учитесь?")

После этого выводила бы три строки:

"Ваше имя"

"Дата рождения"

"Вы учитесь в "

Также выводила бы эту информацию в одну строку, используя любой символ в качестве разделителя этих строк.

Вариант 3

Фамилию, Место жительства

- Фамилия("Ваша фамилия?")
- место жительства ("Где Вы живете?")

После этого выводила бы две строки:

"Ваша фамилия"

"Вы живете в"

Также выводила бы эту информацию в одну строку, используя любые три символа в качестве разделителя этих строк.

Вариант 4

Фамилию, Место рождения, любимую музыку

- Фамилия, ("Ваша фамилия?")
- место рождения ("Где Вы родились?")
- музыка ("Какая музыка нравится? ")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Вы родились в"

"Ваша любимая музыка "

Также выводила бы эту информацию в одну строку, используя любые два символа в качестве разделителя этих строк.

Вариант 5

Имя, Фамилия, ФИО мамы, ФИО отца

- ФИО (например, "Ваши фамилия, имя, отчество?")
- возраст ("Сколько Вам лет?")
- место жительства ("Где Вы живете?")

После этого выводила бы три строки:

"Ваши имя, фамилия, отчество"

"Ваш возраст"

"Вы живете в"

Также выводила бы эту информацию в одну строку, используя любые четыре символа в качестве разделителя этих строк.

Вариант 6

Имя, Любимый предмет в школе, Номер класса

- имя ("Ваше имя?")
- любимый предмет ("Какой Ваш любимый предмет в школе?")
- номер класса ("В каком классе Вы учитесь?")

После этого выводила бы три строки:

"Ваше имя"

"Ваш любимый предмет в школе"

"Вы учитесь в классе номер"

Также выводила бы эту информацию в одну строку, используя любые три символа в качестве разделителя этих строк.

Вариант 7

Имя, Фамилию, Страну, Край , Город

- Фамилия, имя ("Ваши фамилия, имя?")
- страна ("В какой стране Вы живете?")
- город ("В каком городе Вы живете?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Вы живете в стране"

"Вы живете в крае"

"Вы живете в городе"

Также выводила бы эту информацию в одну строку, используя любой символ в качестве разделителя этих строк.

Вариант 8

Имя, Фамилию, Отчество, Хобби

- ФИО (например, "Ваши фамилия, имя, отчество?")
- хобби ("Чем Вы увлекаетесь?")

После этого выводила бы две строки:

"Ваши имя, фамилия, отчество"

"Ваше хобби"

Также выводила бы эту информацию в одну строку, используя любые два символа в качестве разделителя этих строк.

Вариант 9

Имя, Фамилию, любимый спорт

- Фамилия, имя ("Ваши фамилия, имя?")
- образование ("В какой школе Вы учитесь?")
- ФИО Вашего руководителя по информатики ("ФИО Вашего руководителя по информатики?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Вы учитесь в школе номер: "

"ФИО Вашего руководителя по информатике "

Также выводила бы эту информацию в одну строку, используя любые три символа в качестве разделителя этих строк.

Вариант 10

Имя, Фамилия, Любимый предмет в школе (в институте), ФИО классного руководителя (куратора)

- Фамилия, имя ("Ваши фамилия, имя?")
- любимый предмет в школе ("Какой Ваш любимый предмет в школе?")
- ФИО классного руководителя ("ФИО Вашего классного руководителя?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Ваш любимый предмет в школе "

"ФИО Вашего классного руководителя"

Также выводила бы эту информацию в одну строку, используя любые два символа в качестве разделителя этих строк.

Вариант 11

Имя, Фамилию, Возраст, Дату рождения

- Фамилия, имя ("Ваши фамилия, имя?")
- возраст ("Сколько Вам лет?")
- дата рождения ("Когда Вы родились?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Ваш возраст"

"Дата Вашего рождения"

Также выводила бы эту информацию в одну строку, используя любой символ в качестве разделителя этих строк.

Вариант 12

Имя, Фамилию, Место жительства, Место рождения

- Фамилия, имя ("Ваши фамилия, имя?")
- место рождения ("Где Вы родились?")
- место жительства ("Где Вы живете?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Вы родились в"

"Вы живете в"

Также выводила бы эту информацию в одну строку, используя любые три символа в качестве разделителя этих строк.

Вариант 13

Имя, Фамилию, Возраст, Номер телефона

- Фамилия, имя ("Ваши фамилия, имя?")
- возраст ("Сколько тебе лет?")
- номер телефона ("Номер Вашего телефона?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Ваш возраст"

"Ваш номер телефона"

Также выводила бы эту информацию в одну строку, используя любой символ в качестве разделителя этих строк.

Вариант 14

Имя, Фамилия, Страна, Край, Город

- Фамилия, имя ("Ваши фамилия, имя?")

- страна ("В какой стране Вы живете?")

- город ("В каком городе Вы живете?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Вы живете в стране"

"Вы живете в крае"

"Вы живете в городе"

Также выводила бы эту информацию в одну строку, используя любые два символа в качестве разделителя этих строк.

Вариант 15

Имя, Фамилию, ФИО Вашего классного руководителя

- Фамилия, имя ("Ваши фамилия, имя?")

- ФИО Вашего классного руководителя ("ФИО Вашего классного руководителя?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"ФИО Вашего руководителя по информатике"

"ФИО Вашего классного руководителя"

Также выводила бы эту информацию в одну строку, используя любые три символа в качестве разделителя этих строк.

Вариант 16

Имя, Фамилию, Любимый предмет

- Фамилия, имя ("Ваши фамилия, имя?")

- Ваш любимый предмет ("Любимый предмет?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"ФИО Вашего руководителя по информатике"

"ФИО Вашего классного руководителя"

Также выводила бы эту информацию в одну строку, используя любые три символа в качестве разделителя этих строк.

Вариант 17

Юноша вы или девушка, Любимое хобби, Где вы любите отдыхать

- юноша вы или девушка ("Юноша вы или девушка?")

- любимое хобби ("Любимое хобби?")

- где вы любите отдыхать ("Любимое место отдыха?")

После этого выводила бы три строки:

Я....."

Моё любимое хобби - ...

Моё любимое место отдыха - ...

Также выводила бы эту информацию в одну строку, используя любые два символа в качестве разделителя этих строк.

6. Список источников

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основная литература

- 1 Потемкин В.Г. МАТЛАБ: среда проектирования инженерных приложений: Учебное пособие М.: Диалог-МИФИ 2008
- 2 Наместников С.М. Основы программирования в MatLab: Сборник лекций УлГТУ, Ульяновск 2011 Л1.3 Трошина Г. В. Численные расчеты в среде MatLab: Учебное пособие Новосибирск: Новосибирский государственный технический университет 2020

Дополнительная литература

- 1 Половко А. М., Бутусов П. Н. МАТЛАВ для студента: Учебная литература СПб.: БХВ-Петербург 2005
- 2 Дьяконов В. П. МАТЛАВ: Полный самоучитель Саратов: Профобразование 2019

Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

- 1 Материалы по продуктам MATLAB & Toolboxes <http://matlab.exponenta.ru/>
- 2 MatLab. Руководство для начинающих <http://www.chemometrics.ru/materials/textbooks/matlab.htm>
- 3 Электронно-библиотечная система IPRbooks <https://iprbooks.ru> Э4 База знаний Matlab. <https://radiomaster.ru/matlab/>