**Лабораторная работа 1 Введение в язык программирования Python**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Цель работы:** Познакомиться со средой разработки Python. Изучить основные типы данных, команды  ввода и вывода данных.  **Краткая теория**  Python– это объектно-ориентированный, интерпретируемый, переносимый язык сверхвысокого уровня. Программирование на Python позволяет получать быстро и качественно необходимые программные модули.  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 в Python обозначаются предваряющим их символом # и продолжаются до конца строки(т.е. в Python все комментарии являются однострочными), при этом не допускается использование перед символом # кавычек:    **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**  Некоторые слова имеют в Python специальное назначение и представляют собой управляющие конструкции языка.  Ключевые слова в Python:  ['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']  **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   **Ввод и вывод данных**  Ввод данных осуществляется при помощи команды **input(**список ввода**)**:  a = input()  print(a)  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 осуществляется при помощи команды **print**(список вывода):  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  **Простые арифметические операции над  числами**   |  |  | | --- | --- | | x + y | Сложение | | x - y | Вычитание | | x \* y | Умножение | | x / y | Деление |     Пример программы на Python    Результат выполнения программы с применением простых арифметических операций  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:      В результате выведется число 11, а перед ним два пробела, так как указано использовать для вывода четыре знакоместа.  Или с несколькими аргументами:      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))    В результате выведется число с двумя знаками после запятой.    **Пример**  Напишите программу, которая запрашивала бы у пользователя:  **Вариант 0**  - 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**      **Задания для самостоятельной работы (по вариантам)**  Напишите программу, которая запрашивала бы у пользователя:  **Вариант 1**  Имя, Фамилия, 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"  **Вариант 2**  Имя, , Дата 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 "  **Вариант 3**  Фамилия, Место жительства  - Фамилия( "Ваша фамилия?")  - место жительства ("Где Вы живете?")  После этого выводила бы две строки:  "Ваша фамилия"  "Вы живете в"  **Вариант 4**  Фамилия, Место рождения, любимая музыка  - 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 "  **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"  **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" |