

Полиморфизм

Полиморфизм – свойства кода работать с разными типами данных.

При изучении основ программирования на Python, мы уже неоднократно использовали это свойство многих функций и операторов.

Например, операция + (сложения) является полиморфной:

```
print(5+2)
print(8.5+0.7)
print("Доброе"+" утро")
```

7
9.2
Доброе утро

Внутренняя реализация операции + существенно отличается для целых чисел, чисел с плавающей точкой и строк, т.е. на самом деле это три разных операции.

Для усложнения задачи, воспользуемся функцией:

```
def function(x,y):
    return x+y

print(function(5,2))
print(function(8.5,0.7))
print(function("Доброе", " утро"))
```

7
9.2
Доброе утро

In [9]:

Т.к. Python – язык с **динамической типизацией**, любое значение обрабатываемое интерпретатором несет в себе информацию о типе, что и помогает системе выбрать правильную реализацию операции +. Исходя из понятия ООП, тип данных в Python – класс объекта, и именно эта информация о классе объекта используется при выборе операции.

Рассмотрим уже ранее изученный метод `__init__`. Он выполняется при создании каждого нового экземпляра класса и инициализирует свойства нового экземпляра. Напомню, первый аргумент, `self`, он получает от интерпретатора, остальные передаются классу в круглых скобках при создании экземпляра

```
class HomeBook:
    def __init__(self,name,author):
        self.name=name
        self.author=author

    def get_name(self):
        return self.name

    def get_author(self):
        return self.author

book1=HomeBook('Изучаем программирование на Python','Пол Бэрри')
book2=HomeBook('Изучаем Python','Марк Лутц')
print ('{},{}'.format(book1.get_name(),book1.get_author()))
print ('{},{}'.format(book2.get_name(),book2.get_author()))
```

Изучаем программирование на Python,Пол Бэрри
Изучаем Python,Марк Лутц

In [16]:

При использовании кода `book1=HomeBook('Изучаем программирование на Python','Пол Бэрри')` будет создан объект, у которого до момента присваивания ссылки на него переменной вызовется метод `__init__`, создающий атрибуты `name` и `author` и задающий значение этим атрибутам.

Для того, чтобы понять, что такое **полиморфизм** и как его реализовать в ООП на Python, создадим следующую иерархию классов:

Круг



- Площадь: πr^2
- Периметр: $2\pi r$

Квадрат



- Площадь: a^2
- Периметр: $4a$

Создадим 2 класса **Circle** и **Square** для вычисления площади и параметра фигуры.

```
from math import pi
class Circle:
    def __init__(self, radius):
        self.radius = radius

    def ploshad_fig(self):
        return pi * pow(self.radius, 2)
    def perimetr_fig(self):
        return 2 * pi * self.radius

class Square:
    def __init__(self, storona):
        self.storona = storona

    def ploshad_fig(self):
        return pow(self.storona, 2)
    def perimetr_fig(self):
        return 4 * self.storona
```

Для **обоих классов** методы для расчета площади и периметра фигуры называются одинаково **ploshad_fig** и **perimetr_fig**, так же эти методы имеют **одинаковое** количество **параметров** (self) и они оба возвращают в результате работы число (может быть как целое, так и вещественное).

Для вывода данных о фигуре, создадим полиморфную функцию **fig_info**:

```
from math import pi
class Circle:
    def __init__(self, radius):
        self.radius = radius

    def ploshad_fig(self):
        return pi * pow(self.radius, 2)
    def perimetr_fig(self):
        return 2 * pi * self.radius

class Square:
    def __init__(self, storona):
        self.storona = storona

    def ploshad_fig(self):
        return pow(self.storona, 2)
    def perimetr_fig(self):
        return 4 * self.storona

def fig_info(a):
    print('Площадь={}, Периметр={}'.format(a.ploshad_fig(), a.perimetr_fig()))

s = Square(20)
fig_info(s)
s1 = Circle(20)
fig_info(s1)
```

```
Площадь=400, Периметр=80
Площадь=1256.6370614359173, Периметр=125.66370614359172
In [40]: |
```

Если аргумент функции **fig_info**—экземпляр класса **Square**, то выполняются методы, определённые в этом классе, если экземпляр **Circle**, то выполняются методы **Circle**.

Выбор конкретной реализации вычисления площади и периметра производится в момент вызова методов **ploshad_fig** и **perimeter_fig** и зависит от класса экземпляра.

Полиморфизм – это способность объектов с одним интерфейсом иметь различную реализацию.

Понятие **Утиная типизация** (Duck Typing) («Если что-то выглядит как утка, плавает как утка и крякает как утка, это наверняка и есть утка») - означает, что до тех пор, пока классы содержат одинаковые методы, интерпретатор Python не различает их, поскольку единственная проверка вызовов происходит во время выполнения.

Перегрузка метода

Перегрузка метода относится к свойству метода вести себя по-разному, в зависимости от количества или типа параметров. Ниже простой пример **перегрузки метода**.

В скрипте ниже, если метод start() вызывается передачей одного аргумента, параметр будет выведен на экран. Однако, если мы передадим 2 аргумента методу start(), он внесет оба аргумента и выведет результат суммы.

```
# создаем класс Car
class Car:
    def start(self, a, b=None):
        if b is not None:
            print(a + b)
        else:
            print(a)

car_a = Car()
car_a.start(10)
car_a.start(10, 20)
```

10
30

Задания для самостоятельной работы:

1. Создать 2 класса **Прямоугольник** и **Трапеция**, используя интерфейс предыдущих классов, протестировать.

2. Написать классы **Американская дата** и **Европейская дата**. При инициализации они должны вводить **год, месяц, число** (порядок обязателен). В обоих классах нужно реализовать методы **set_year**, **set month**, **set_day** для изменения одной из компонентов даты и **три метода** для чтения компонентов даты (названия методам задайте произвольно). Еще **один метод** (название произвольно) должен возвращать строковое представление даты в соответствии с определенным классом.

Учесть: Форматы записи дат в виде строки в США и Европе отличаются.

В США принят формат мм.дд.гггг, в Европе — дд.мм.гггг, где дд — день (дополняется нулём слева, если число меньше 10), мм — месяц (так же дополняется нулём слева), гггг — год.

Например, 10 апреля 2000 года будет записано в американском формате как 04.10.2000, а в европейском — как 10.04.2000. Все годы в задаче — четырёхзначные.

Дата должна быть введена корректно, реальная (не 30 февраля).

3.*Написать два класса **LeftPage** и **RightPage** для печати абзаца текста с выравниванием по левому и правому краю. При инициализации, экземпляры обоих классов должны принимать целое число, равное ширине поля ввода. В обоих классах нужно реализовать метод для добавления слова в абзац – **add_word** и метод **result**, выводящий полученный абзац на печать и начинающий формировать новый.

Учесть: длина слова не должна превышать ширину поля ввода.

4. Работа по вариантам:

1. Создать класс с полями, указанными в индивидуальном задании.
2. Реализовать в классе методы:
 - конструктор по умолчанию – найти примеры реализации самостоятельно;
 - деструктор для освобождения памяти (с сообщением об уничтожении объекта) – найти примеры реализации самостоятельно;
 - функции обработки данных, указанные в индивидуальном задании;
 - функцию формирования строки информации об объекте.
3. Создать проект для демонстрации работы: сформировать объекты со значениями-константами и с введенными с клавиатуры значениями полей объекта. Вывести результаты работы на экран.

Варианты заданий приведены в таблице 1.

Таблица 1

№ варианта	Класс и его поля	Функция метод 1 обработки данных	Функция метод 2 обработки данных
1	Дата (три числа): день, месяц, год	Определить, является ли год високосным (кратным 4)	Увеличить дату на 5 дней
2	Работник: фамилия, оклад, год поступления на работу	Вычислить стаж работы работника на данном предприятии	Сколько дней прошло после года поступления на работу
3	Книга: название, количество страниц, цена	Вычислить среднюю стоимость одной страницы	Увеличить цену книги в два раза, если название начинается со слова «Программирование»
4	Время (три числа): часы, минуты, секунды	Вычислить количество полных минут в указанном времени	Уменьшить время на 10 минут
5	Товар: наименование, цена, год выпуска	Определить, сколько лет назад был выпущен товар	Увеличить цену товара на 20%, если в наименовании товара есть слово «TV».
6	Дата (три числа): день, месяц, год	Увеличить год на 1	Уменьшить дату на 2 дня
7	Книга: название, автор, год издания	Вычислить, сколько лет книге	Количество дней, прошедших после года издания книги
8	Работник: фамилия, оклад, дата рождения	Вычислить возраст работника	Сколько календарных дней до исполнения работнику 50 лет
9	Время (три числа): часы, минуты, секунды	Вычислить количество секунд в указанном времени	Увеличить время на 5 секунд
10	Четыре целых числа: a, b, c, d	Вычислить среднее арифметическое чисел	Определить максимальное из чисел
11	Работник: фамилия, должность, оклад	Увеличить оклад на 15% (каждому работнику)	Работникам, у которых фамилия начинается с сочетания букв «Иван», присвоить должность «инженер».
12	Книга: название, количество страниц, цена	Увеличить количество страниц на 10	Уменьшить цену в два раза, если количество страниц больше 100 (после

			увеличения)
13	Дата (три числа): день, месяц, год	Определить, совпадают ли номер месяца и число дня	Увеличить дату на один месяц
14	Товар: наименование, цена в рублях, изготовитель	Пересчитать цену товара в евро	Увеличить цену товара в евро, если название товара содержит слово «Samsung»
15*	Время (три числа): часы, минуты, секунды	Определить количество минут до полуночи (24:00:00)	Увеличить время 100 минут
16	Правильная дробь: числитель, знаменатель	Выразить значение дроби в процентах	Найти сумму цифр значения знаменателя
17	Комната: длина, ширина, высота (в метрах)	Площадь стен (вместе с окнами и дверьми)	Площадь стен без окна (размер 2×15 м) и двери (размер 2×8 м).
18	Комплексное число: действительная (a1) и мнимая (b1) части числа	Вычислить модуль комплексного числа	Вычислить аргумент комплексного числа в градусах
19	Координаты изображения прямоугольника: x1, y1, x2, y2	Вычислить площадь прямоугольника в пикселях	y2 Вычислить длину диагонали прямоугольника в пикселях
20	Параллелепипед: длины сторон	Вычислить площадь поверхности	Вычислить объем параллелепипеда