1. **Инкапсуляция** - **это** механизм, который объединяет данные и методы, манипулирующие этими данными.

**class** **Human**:

sex = "male"

**def** **change\_sex**(self):

print( “not in this country man”)

andrey= Human()

andrey.change\_sex()

1. Наследование нужно чтобы создать подкласс на основе уже имеющегося, перенять его поля и методы с при необходимости добавить или переопределить имеющиеся

+

*1)Сокращение работы, возможность итспользовать готовые классы и на их основе базироваться*

*2)Наглядность*

*- скорость*

*Сложность*

1. *Git – Система контроля версий,*

*Github:*  веб-сервис для хостинга IT-проектов и их совместной разработки, предосталяет возможность выкладывать свои программы в открытый доступ, создавать различные ветки, ускорять сомвестные проекты

Магические методы в Python – это базовые методы, которые можно назначить любому классу:   
*1)Инициализация объекта: \_\_init\_\_*

*2) Репрезентация объекта :\_\_str\_\_*

*3+4) Магические методы для переопределение арифметических операций с объектами*

* \_\_add\_\_(self, other)  
  Сложение.
* \_\_sub\_\_(self, other)  
  Вычитание.
* \_\_mul\_\_(self, other)  
  Умножение.
* \_\_floordiv\_\_(self, other)  
  Целочисленное деление, оператор //.
* \_\_div\_\_(self, other)  
  Деление, оператор /.

class Cmplx:  
 def \_\_init\_\_(self, real, imag):  
 self.real = real  
 self.imag = imag  
  
 def \_\_add\_\_(self, other):  
 return Cmplx(self.real + other.real, self.imag + other.imag)  
  
 def \_\_sub\_\_(self, other):  
 return Cmplx(self.real - other.real, self.imag - other.imag)  
  
 def \_\_mul\_\_(self, other):  
 return Cmplx(self.real \* other.real - self.imag \* other.imag, self.imag \* other.real + self.real \* other.imag)  
  
 def \_\_truediv\_\_(self, other):  
 return Cmplx((self.real \* other.real + self.imag \* other.imag) / (other.real \*\* 2 + other.imag \*\* 2),  
 (self.imag \* other.real - self.real \* other.imag) / (other.real \*\* 2 + other.imag \*\* 2))  
  
 def \_\_str\_\_(self):  
 return f"{round(self.real, 2)}+{round(self.imag, 2)}i"

1. Статические методы это те которые не имеют доступа к полям класса

по-сути обычные функции, помещенные в класс для удобства и находящиеся в пространстве имен этого класса.

Нужны например для создания какой либо библиотеки (для примера библиотека пчитабщая площади разных фигур в класса площадь, для каждого статического метода нгужны разные входные параметры которые будут передаваться внутрь статического метода