

# Объектно-Ориентированное Программирование

---

Для вопросов по курсу:

Шевская Наталья Владимировна, [natalya.shevskaya@moevm.info](mailto:natalya.shevskaya@moevm.info)

Иванов Дмитрий Владимирович, [dmitry.ivanov@moevm.info](mailto:dmitry.ivanov@moevm.info)

Префикс в теме письма [CS\_23XX]

# Организационное

---



# Почты преподавателей лабораторных работ

2300 -- Валентина Петровна Голубева [valentina.golubeva@moevm.info](mailto:valentina.golubeva@moevm.info)

2303 -- Роман Павлович Шестопалов [roman.shestopalov@moevm.info](mailto:roman.shestopalov@moevm.info)

2304 -- Роман Павлович Шестопалов [roman.shestopalov@moevm.info](mailto:roman.shestopalov@moevm.info)

2381 -- Максим Олегович Доброхвалов [maxim.dobrokhvalov@moevm.info](mailto:maxim.dobrokhvalov@moevm.info)

2382 -- Максим Олегович Доброхвалов [maxim.dobrokhvalov@moevm.info](mailto:maxim.dobrokhvalov@moevm.info)

2383 -- Сергей Валерьевич Сорокумов [sergey.sorokumov@moevm.info](mailto:sergey.sorokumov@moevm.info)

2384 -- Сергей Валерьевич Сорокумов [sergey.sorokumov@moevm.info](mailto:sergey.sorokumov@moevm.info)

В копию можно ставить:

Дмитрий Владимирович Иванов [dmitry.ivanov@moevm.info](mailto:dmitry.ivanov@moevm.info)



# Общение с преподавателями лабораторных работ

Во время лабораторной работы по расписанию -- общение в дискорд

В остальное время -- по почте

При написании писем обязательно следуйте правилам:

[http://se.moevm.info/doku.php/inf:communication\\_rules](http://se.moevm.info/doku.php/inf:communication_rules)



# Регистрация студентов 1 курса 2023

Новеньким студентам следует заполнить гугл-форму, опубликованную на странице информатики

<http://se.moevm.info/doku.php/courses:informatics:start>

## Анкета первокурсника МОЭВМ

-  Анкета первокурсника МОЭВМ 2022/2023

Срок заполнения – до 23:59 19 февраля 2023 (*заполнение позже срока приводит к увеличению времени обработки*)



# Доступ к курсу в весеннем семестре



Ссылка: <http://e.moevm.info/course/view.php?id=69>

Курс открыт на самозапись, то есть надо перейти по ссылке и самостоятельно записаться на курс.

# Объектно-Ориентированное Программирование

---



# Основные понятия. Объект

Объект - конкретная сущность предметной области.





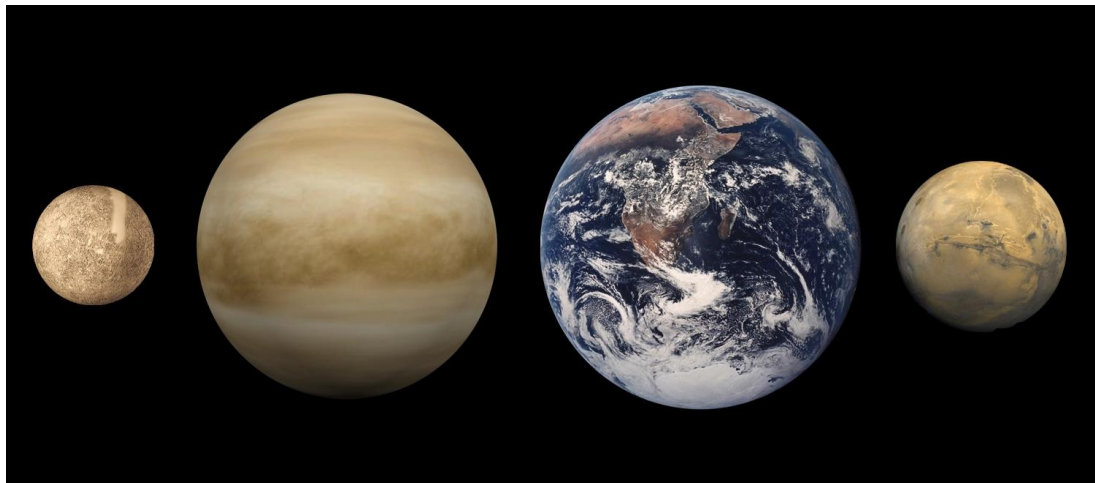


# Основные понятия. Класс

Класс - это тип объекта. Или говорят, что объект - экземпляр класса.

Класс: **Планета.**

Объекты: **Меркурий, Венера, Земля, Марс.**





# Основные понятия. Методы класса

- Метод класса -- функция, которая принадлежит классу.

`<имя объект>.<метод>(аргумент_1, аргумент_2, ...)`

- В языке Python первый аргумент метода - **self**, экземпляр класса. При описании метода пишется явно. При вызове метода передается неявно.
- Методы имеют доступ к полям экземпляра класса.
- Пример вызова метода: `Венера.получитьСреднийРадиус()`

```
>>> s = 'qwerty'
>>> s.replace('q', 'w') # self неявно
'wwerty'
```



# Основные понятия. Методы класса. Self

➤ Пример:

```
>>> s.replace('q', 'w')
```

➤ Модуль *builtins.py*: метод *replace* “внутри” класса *str*:

```
def replace(self, old, new, count=None): # real signature unknown; restored from __doc__  
    """
```

```
    S.replace(old, new[, count]) -> str
```

```
    Return a copy of S with all occurrences of substring  
    old replaced by new. If the optional argument count is  
    given, only the first count occurrences are replaced.
```

```
    """
```

```
    return
```



# Основные понятия. Конструктор

- Конструктор - метод, который вызывается при создании экземпляра класса.
- Конструктор ничего не возвращает.
- Конструктор может быть не описан, тогда создастся пустой объект.
- Конструктор может быть только один, но может иметь переменные по умолчанию.

**class** Student:

```
def __init__(self): # метод класса Student; self - текущий экземпляр класса Student  
    pass
```



# Основные понятия. Поля объекта

- Поле (атрибут) объекта -- некоторая переменная, которая лежит в области видимости объекта и доступна во внешней программе через синтаксис:

`<имя_объекта>.<поле>`

- Поля объекта устанавливаются в методах класса через обращение к экземпляру **self**, например:

```
def __init__(self):
```

```
    self.name = 'Fedor' # поле экземпляра класса Student
```



# Основные понятия. Создание простого класса

**class** Student:

**def** \_\_init\_\_(self): *# конструктор; self -- текущий экземпляр класса Student*

self.name = **'Fedor'** *# поле экземпляра класса Student*

name = **'Petr'** *# локальная переменная в конструкторе*

new\_student = Student() *# создание экземпляра класса Student*

**print**(new\_student.name)



# Некоторые основные принципы ООП

- Инкапсуляция
- Наследование
- Полиморфизм



# Инкапсуляция

- Соккрытие внутренней реализации от пользователя.

*! В языке Python нельзя создать по-настоящему приватный атрибут.*

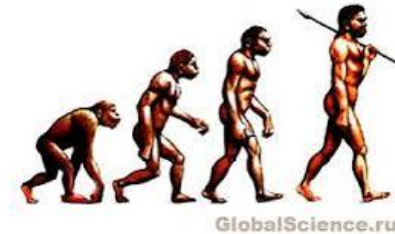
- Соккрытие деталей реализации за интерфейсом объекта.

**Пример:** работа с библиотекой Wikipedia.





# Наследование



- Повторное использование и последующее расширение одним классом атрибутов другого класса.
- Класс, определённый через наследование от другого класса, называется производным классом, классом-потомком, подклассом, классом-наследником. Класс, от которого новый класс наследуется, называется предком, базовым классом или суперклассом.
- Пример в Python:

```
class Super: # Родитель
    def __init__(self):
        self.data = 'some data'
```

```
class Subclass(Super): # Потомок
    pass
```



# Класс object

- Встроенный класс object является суперклассом для всех классов.
- В иерархии наследования класс object автоматически стоит выше остальных классов.
- Класс object предоставляет методы, которые реализуют различные операции, например, поиск длины объекта или вывод на экран. Эти методы можно переопределить.



## isinstance() и isinstance()

➤ `isinstance(obj1, class1)`

Возвращает True, если **obj1** является экземпляром класса **class1** или суперкласса класса **class1**

➤ `issubclass(class1, class2)`

Возвращает True, если **class1** является наследником класса **class2**.



# Источники и полезные ссылки

- Классы в Python, документация

<https://docs.python.org/3/tutorial/classes.html>

- Операторы языка python, которые вы можете переопределить:

<https://docs.python.org/3/reference/datamodel.html>

- Курс “Python: основы и применение” <https://stepik.org/course/512/syllabus>

- М. Лутц “Изучаем Python”, 4-е издание

# Вопросы по курсу можно задавать:

---

Шевская Наталья Владимировна  
[natalya.shevskaya@moevm.info](mailto:natalya.shevskaya@moevm.info)

Иванов Дмитрий Владимирович  
[dmitry.ivanov@moevm.info](mailto:dmitry.ivanov@moevm.info)