

Тест по блоку

Вас зовут:

1. Отметьте все исключения стандартной библиотеки Python

- `RequestException`
- `IndexError`
- `ValueError`
- `KeyboardInterrupt`

2. Какие из методов генерации исключения разрешены в Python?

- `raise ValueError("error")`
- `raise None`
- `raise "ValueError"`
- `raise ValueError`

3. Обращение к несуществующему атрибуту экземпляра

- сгенерирует исключение `AttributeError`
- вернет `False`
- вернет `None`
- сгенерирует исключение `KeyError`

4. Отметьте верные утверждения про `classmethod`

- Метод не принимает дополнительных аргументов кроме указанных программистом
- Метод первым аргументом принимает класс
- Метод первым аргументом принимает ссылку на экземпляр класса
- К этому методу можно обращаться от имени класса
- К этому методу можно обращаться от экземпляра класса

5. Отметьте верные утверждения про `staticmethod`

- Метод первым аргументом принимает ссылку на экземпляр класса
- Метод первым аргументом принимает класс
- К этому методу можно обращаться от имени класса
- Метод не принимает дополнительных аргументов кроме указанных программистом
- К этому методу можно обращаться от экземпляра класса

6. Для чего используются `@property` ?

- Чтобы делать методы приватными
- Чтобы делать атрибуты приватными
- Чтобы создать вычисляемый атрибут

7. Если имя метода в классе начинается с символа нижнего подчеркивания, например: `_get_name`

- символ нижнего подчеркивания в начале метода не добавляет никакого дополнительного значения
- если вызвать метод у экземпляра класса, то сгенерируется исключение `AttributeError`
- обращаться к методу объекта не рекомендуется, так как метод не считается публичным API класса

8. Можно ли использовать экземпляры классов в качестве ключей словаря (`dict`)?

- Да
- Нет

9. Можно ли передавать экземпляр класса как аргумент в функцию?

- Да
- Нет

10. Предположим есть базовый класс питомец - `Pet` и класс наследник - `Cat` .
Отметьте варианты, которые вернут `True`

- `issubclass(Cat, Pet)`
- `isinstance(Cat(), Cat)`
- `issubclass(Cat, object)`
- `isinstance(Cat(), Pet)`
- `issubclass(Pet, Cat)`