**Контрольные вопросы**

1. Механизм перегрузки оператора в классе Python основан на использовании специальных методов, которые начинаются и заканчиваются двумя символами подчеркивания. [Эти методы имеют специальное назначение и строго определенные имена для каждой операции](https://www.bestprog.net/ru/2021/02/13/python-overloading-operators-in-classes-general-information-ru/)[1](https://bing.com/search?q=%D0%BC%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%B7%D0%BC+%D0%BF%D0%B5%D1%80%D0%B5%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B8+%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%B0+%D0%B2+%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B5+Python)[2](https://www.bestprog.net/ru/2021/02/13/python-overloading-operators-in-classes-general-information-ru/). [Например, метод \_\_add\_\_() перегружает оператор +, а метод \_\_eq\_\_() перегружает оператор ==3](https://nbco.kz/basics-of-object-oriented-programming/operatory-peregruzki-v-python-dlya/).
2. [В основе механизма перегрузки операторов в классах Python лежат следующие принципы](https://bing.com/search?q=%d0%bc%d0%b5%d1%85%d0%b0%d0%bd%d0%b8%d0%b7%d0%bc+%d0%bf%d0%b5%d1%80%d0%b5%d0%b3%d1%80%d1%83%d0%b7%d0%ba%d0%b8+%d0%be%d0%bf%d0%b5%d1%80%d0%b0%d1%82%d0%be%d1%80%d0%b0+%d0%b2+%d0%ba%d0%bb%d0%b0%d1%81%d1%81%d0%b5+Python)[4](https://bing.com/search?q=%D0%BF%D1%80%D0%B8%D0%BD%D1%86%D0%B8%D0%BF%D1%8B+%D0%BF%D0%B5%D1%80%D0%B5%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B8+%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BE%D0%B2+%D0%B2+%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0%D1%85+Python):
   * При использовании перегрузки операторов в классе используются методы, начинающиеся и заканчивающиеся двумя символами подчеркивания (например, \_\_add\_\_(), \_\_str\_\_() и другие). Эти методы имеют специальное назначение и строго определенные имена для каждой операции.
   * Как только вызывается встроенный оператор (операция), вызов соответствующего метода происходит автоматически.
3. [В Python можно перегружать большинство операторов, таких как арифметические (+, -, \*, /, %, //, \*\*), операторы сравнения (==, !=, <, >, <=, >=), операторы присваивания (=, +=, -=, и т.д.), операторы доступа ([], [:]) и другие5](https://fullstacker.ru/peregruzka-operatorov-v-python-printsip-raboty-na-primerah). [Также можно перегрузить методы, начинающиеся и заканчивающиеся двумя символами подчеркивания, такие как \_\_add\_\_, \_\_str\_\_ и другие](https://www.bestprog.net/ru/2021/02/13/python-overloading-operators-in-classes-general-information-ru/)[1](https://bing.com/search?q=%D0%BC%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%B7%D0%BC+%D0%BF%D0%B5%D1%80%D0%B5%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B8+%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%B0+%D0%B2+%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B5+Python)[2](https://www.bestprog.net/ru/2021/02/13/python-overloading-operators-in-classes-general-information-ru/).
4. [В Python, конструктор \_\_init\_\_() является специальным методом, который вызывается автоматически при создании нового экземпляра класса](https://bing.com/search?q=%d0%bc%d0%b5%d1%85%d0%b0%d0%bd%d0%b8%d0%b7%d0%bc+%d0%bf%d0%b5%d1%80%d0%b5%d0%b3%d1%80%d1%83%d0%b7%d0%ba%d0%b8+%d0%be%d0%bf%d0%b5%d1%80%d0%b0%d1%82%d0%be%d1%80%d0%b0+%d0%b2+%d0%ba%d0%bb%d0%b0%d1%81%d1%81%d0%b5+Python)[6](https://bing.com/search?q=%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80+__init__+%D0%B2+Python). [Он используется для инициализации объекта при его создании и позволяет задать начальные значения для атрибутов класса](https://bing.com/search?q=%d0%bc%d0%b5%d1%85%d0%b0%d0%bd%d0%b8%d0%b7%d0%bc+%d0%bf%d0%b5%d1%80%d0%b5%d0%b3%d1%80%d1%83%d0%b7%d0%ba%d0%b8+%d0%be%d0%bf%d0%b5%d1%80%d0%b0%d1%82%d0%be%d1%80%d0%b0+%d0%b2+%d0%ba%d0%bb%d0%b0%d1%81%d1%81%d0%b5+Python)[6](https://bing.com/search?q=%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80+__init__+%D0%B2+Python).
5. [Деструктор в Python представляет собой специальный метод \_\_del\_\_(), который вызывается, когда объект удаляется7](https://fullstacker.ru/destruktor-v-python-udalenie-obekta-metodom-__del__). [Он обычно используется для выполнения «завершающих» операций, таких как освобождение ресурсов7](https://fullstacker.ru/destruktor-v-python-udalenie-obekta-metodom-__del__).
6. [Наследование в Python — это механизм, который позволяет создавать новый класс на основе уже существующего класса8](https://metanit.com/python/tutorial/7.3.php). [Это один из краеугольных камней объектно-ориентированного программирования, позволяющий создавать иерархии связанных классов9](https://www.bestprog.net/ru/2021/02/09/python-class-inheritance-rules-ru/).
7. [В Python, наследование реализуется путем указания имени суперкласса в круглых скобках в заголовке инструкции класса9](https://www.bestprog.net/ru/2021/02/09/python-class-inheritance-rules-ru/). [Подкласс наследует атрибуты своих суперклассов9](https://www.bestprog.net/ru/2021/02/09/python-class-inheritance-rules-ru/).
8. Операция расширения класса в Python обычно относится к процессу создания нового класса на основе существующего класса путем добавления или изменения методов и атрибутов. Это форма наследования, где новый класс (подкласс) наследует функциональность существующего класса (суперкласса) и расширяет ее.
9. Делегирование в Python — это процесс передачи работы или задачи от одного объекта к другому. Он используется для обеспечения связи и взаимодействия между объектами. В контексте объектно-ориентированного программирования, делегирование часто используется для обеспечения полиморфизма и повторного использования кода.