1. Объектно-ориентированное программирование основано на том, что программа представляется как набор взаимодействующих друг с другом классов. Все важные переменный лежат внутри классов, операции с классами – их методы.
2. **Инкапсуляция** – необходимо скрывать от прямого доступа всё, что можно. То есть не нужно заранее предполагать, что в будущем какой-то член класса понадобиться пользователю и из-за этого объявлять его в public.

**Наследование** – чтобы избегать копирования и запутываний, можно просто наследовать некоторые параметры из более общих классов. Например, класс студент может унаследовать имя, фамилию, год рождения и что-то в этом роде из класса человек, если такой имеется.

**Полиморфизм** – в некоторых местах удобнее разбить программу на несколько ветвей с разными способами обработки данных, а в некоторых для одних и тех же типов операции с разными данными может подойти один класс.



**Абстракция** – в один класс можно объединять логически никак не связанные элементы для более удобного использования.

1. Чтобы было компактнее. Посмотрев на интерфейс класса, можно сразу же понять, какие методы и какие переменные в нём есть, ведь для других программистов не столь важна реализация какой-то функции по сравнению с информацией, каков результат её работы и что за аргументы у неё, что можно узнать из интерфейса.
2. При внешнем связывании к этому элементу мы можем обратиться из любого файла проекта, то есть он доступен везде и в процессе линковки будет связан со своими реализациями в других файлах, если они имеются. При внутреннем связывании элемент доступен только в определенной области, не выходящей за пределы данного файла.
3. Можно переходить в одно пространство имен в разных частях кода (это аддитивность, кажется). Возможно создание подпространств. При обращении к элементу пространства извне необходимо писать “имя пространства”::”имя элемента” …операции с ним..; Можно использовать псевдонимы