1. Класс - совокупность методов и полей, описывающих поведение объекта; шаблон объекта.

Объект - некоторая сущность, экземпляр класса.

Объекты создаются с помощью конструктора с помощью слова new.

Тип_сслыки имя_ссылки = new имя_конструктора;

Конструкторы можно перегружать.

Если объект сериализовали, то его можно десериализовать(восстановить)

2. Java передает все по значению. При передаче примитива в метод передается копия данного значения. Изменение значения в методе не затрагивает первоначальное значение. При передаче ссылочного типа в метод мы передаем ссылку на реальный объект. Поучается, что у нас есть 2 ссылки, которые ссылаются на один объект. И при изменении одной из ссылок, другая ссылка не изменится.

3. Исключение - ошибочная ситуация в коде.

На верхушке иерархии исключений находится класс Throwable. От него наследуются классы Exception и Error. От класса Exception наследуется класс RunTimeException.

Исключения обрабатываются с помощью try, catch, finally, throw, throws.

С помощью try-catch-finally ошибка обрабатывается в коде.
throws мы указываем в сигнатуре метода, как бы предполагая, что метод может вызвать исключительную ситуацию.

Проверяемые (checked) исключения видны на этапе компиляции. К ним относится класс Exception. Такие ошибки нужно обрабатывать.

Непроверяемые (unchecked) исключения не видны на этапе компиляции. Они появляются во время запуска кода. К таким исключениям относятся классы Error и RunTimeException. Если появляются такие исключения, значит, нужно исправлять код.

4. Коллекция – это хранилище, предназначенное для накопления и упорядочивания объектов, а также обеспечивает эффективный доступ. Коллекции используются для хранения, поиска и передачи данных. Все коллекции реализуют интерфейс итератор.

Основные методы для работы с классами-коллекциями: equals(), size(), isEmpty(), contains(), add(), remove(), containsAll(), addAll(), removeAll().

5. Перегрузка – механизм использования одинакового имени метода, но с разной сигнатурой.

Например, нам нужно посчитать площадь различных фигур area(), количество входных параметров и реализация каждого метода будут разными.

При наследовании статические методы можно переопределять нестатическими, а нестатические - статическими.

Переопределение - механизм использования одинакового имени и сигнатуры метода, но с различной реализацией.

Например, в интерфейсе мы написали методы, а в классе, который реализует этот интерфейс, мы эти переопределяем эти методы.

Переопределение характерно для полиморфизма.

При наследовании статические методы не переопределяются нестатическими, а нестатические - статическими.