1. Объект-это экземпляр класса. Класс-это шаблон для создания объекта который содержит поля и методы образующие функциональный интерфейс. Пример создания объекта Сlass obj=new Class (); объявление класса class Example {}. При объявлении класса можно использовать модификаторы доступа, static.
2. Я определяю в классе поля которые мне нужны в этом классе так же если это класс описывающий нас объект то в нем будут гетеры и сетеры а остальные методы будут в классе ClassNameAction. При объявлении поля или метода к ним можно использовать атрибуты доступа и static если использовать ststic значит эта переменна/метод общий для всех экземпляров класса и он будет загружен в память при объявлении класса.
3. Конструктор — это метод класса, который инициализирует новый объект (выделяет под него память) Имя конструктора совпадает с именем класса. У конструктора нет типа возвращаемого результата (void тоже нет). Конструкторов может быть столько сколько мы захотим. Конструктор по умолчанию срабатывает каждый раз при создании любого объекта любого класса (если мы написали свой, то сработает он) в классе всегда будет конструктор по умолчанию.this говорит, что именно для этого объекта с которым мы сейчас работаем this.x = x; x данного экземпляра класса станет равен x.this() -используем тогда хотим, чтобы в конструкторе вызывался другой конструктор.
4. Поля класса инициализируются в момент создания класса а поля экземпляра класса в момент создания экземпляра класса. Поля класса инициализируются когда компоратор доходит до создания класса а поля экземпляра класса когда компоратор доходит до создания экземпляра. Значение по умолчанию зависит от того какого типа поля примитивные типы 0 ,boolean false, ссылочные null.так же значения помут присваиваться в блоках инициализации о них рассказывается дальше.
5. Перегрузка методов это когда у нас есть несколько методов с одинаковым названием но принимающие разное значение и компаратор будет решать какой метод вызывать может быть такое что значение которое мы передаем в метод не будет совпадать с теми значениями которые у нас определенны в методе в таком случае компаратор начинает несколько этапный поиск подходящего метода они будут описываться дальше. Мы можем перешружать метод базового класса но для этого нужно использовать команду using чтобы перенести ту функцию в наше пространство имен. Private методы нельзя переопределять.Перегружать конструкторы можно как и менять у них атрибуты доступа.
6. Ранее связывание производится компоратором который решает какой метод будет вызваться в данном месте кода позднее связывание происходит во время выполнения кода и обычно происходит с переопределенными метода когда мы при помощи введенных данных решаем какой метод вызывать. Перегрузка это ранее связывание так как мы сразу определяем какой метод вызывать по типу принимаемого значения. Так же во время решения какой метод вызывать компоратор смотрит на возвращаемы тип. В случае если сразу нужный метод не нашелся идет поиск в 3 этапа на первом мы идем вверх по иерархии типов напривер int стал Integer если так не нашло то происходит уменьшение типа Integer стал int если это не помогло то на последнем этапе он компоратор проходит по всей иерархии в поисках подходящего метода.
7. Неявная ссылка this это ссылка на экземпляр класса к которому мы обращаемся она невидима для нас но видна контструктору и поэтому когда мы пишем this.name он будет обращается к полю name экземпляра класса для которого мы его вызвали. Данная ссылка присутствует в методах которые вызываются для определенного экземпляра и обращаются к его полям.
8. Final поля это поля которые нельзя модифицировать. Но мы должны сразу задавать значение переменной типа final. Final модно применять как к полям класса так и к методам и даже классам в случае класса это значит что у этого класса не может быть наследников.
9. Методы и поля помеченные как static напрямую связаны с классом и будут одинаковые для всех экземпляров класса и мы можем обращаться к данным полям/методам используя название класса. В static методе все переменные должны быть static а сыми static переменные мы можем использовать и в не static методах. Static методы нельзя переопределять но можно перегружать и да они наследуются.
10. Блоки вызываются в порядке их размещения в коде. Размещены они могут быть в любом порядке и их может быть сколько душе угодно. Статический блок используется для инициализации static полей в java. Статический блок вызывается только один при загрузке класса и выполняется до вызова конструктора.Логический блок нужен для инициалинации не static полей и срабатывают каждый раз в момент создания экземпляра класса и он тоже выполняется до конструктора класса.
11. Метод с переменным числом параметров- это метод способный принимать переменное число аргументов. Это значит что он может принять как 5 аргументов так и 1 если он примет 1 то остальные 4 будут проинициализированы по умолчанию. Списки аргументов переменной длины обозначаются символом многоточия ( ... ).При перегрузке выбирается метод наиболее подходящий метод(тот в котором больше совпадений)