1. Массив — это структура данных хранящая набор значений (элементов массива). Индексация элементов массива начинается с 0. Обращаться к I элементу mas[i-1] -1 потому что отчет идет с 0.Размер массива может быть фиксированный, например, mas[12]. Массив в Джава однороден и может хранить значения только одного с ним типа.
2. Пример объявления одномерного массива int[][] twoDimArray квадратные скобки могут быть около типа массива или около названия int twoDimArray[][]. Сколько мерный массив решается количеством []. Инициализация может быть несколькими способами, например, {11,12,12} или int[][] twoDimArray = new int[3][4]; String[] seasons = {"Winter", "Spring", "Summer", "Autumn"}; пример объявления ссылочного типа данных как видим разницы нет.
3. Двумерный массив в Джава — это массив массивов, то есть в каждой ячейке блоков первого уровня массива находится массив, например, [1][5] мы обратились к пятому элементу первого массива. Рваный массив — это такой двумерный массив, когда в количество массивов в блоках первого уровня может разнится, например, в [1][2] в первой ячейке только 3 элемента может быть, а вот в [2][5] этого же массива как мы видим 5. array[0].lenght при помощи этой команды можно узнать сколько элементов в данной ячейке.
4. Ошибка java.lang.ArrayIndexOutOfBoundsException возникает в том случае если мы выходим за пределы массива (пытаемся обратится к элементу, которого нет). java.lang.ArrayStoreException данная ошибка возникает в случае если мы пытаемся добавить в массив элемент не того типа, для которого он создан (массив интов, а мы пытаемся добавить float).
5. Динамический массив-это массив, который меняет свой размер если это нужно. Пример динамической структуры данных: Arraylist, vector и т.д.
6. 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.
7. Если данное поле нужно нам только в 1 методе, то я создам его внутри метода если нужен в нескольких, то создам внутри класса (вне методов) также и с модификаторами доступа если переменная/метод нужен только внутри нашего класса, то private если нужен в других, то publick к полям и методом можно использовать модификаторы доступа что они значат уже описывалось и static который мы используем если хотим, чтобы некая переменная/метод были одинаковые для всех экземпляров класса.
8. 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.
9. Java-bean это стандарт в соответствием с которым все свойства частные (используем геттеры/сеттеры), конструктор без аргументов и должен реализовывать интерфейс serializable (Социализация — это процесс сохранения состояния объекта в последовательность байт.).Если используем булеву переменную то гетер должен зеваться ISчто-то.

Тип данных набор свойств, которыми мы можем охарактеризовать любое его значение и тип действий которые мы можем с ним делать.