1. Массив – структура данных, в которой хранятся элементы одного и того же типа. Индексация начинается с 0. Array[i-1] т.к. индексация начинается с нуля.
2. int[] myArray = new int[10]; От количества [] зависит каким будет являться массив т.е. одномерный, двумерный и т.д. String[] seasons = new String[] {"Winter", "Spring", "Summer", "Autumn"}; Выше приведен пример ссылочного и примитивного типа и как видите данные будут записаны так или иначе и все зависит от вашего желания.
3. Он похож на таблицу, у которой есть номер строки и номер столбца т.е. матрицу. Рваный массив представляет собой массив-массивов, в котором его члены имеют разные размеры. Array[i].lenght – с помощью этой команды можем узнать длину определенной ячейки.
4. Java.ArrayIndexOutOfBoundsEsception – выход за пределы массивы, т.е. обращение к элементу выходящий за массив. Java.lang.ArrayStoreException – ошибка возникающая при добавлении в массив элемента другого типа.
5. Динамическая структура- структура которая может менять свой размер в зависимости от вызванной процедуры. List, vector, ArrayList и тд.

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Epam");

1. Объект – некая модель реальной сущности, обладающая своими данными в виде полей и методами. Класс – шаблон для создания объекта со своими полями и методами. Private, public, protected, static.

class Person{

String name; // имя

int age; // возраст

}

public class Program{

public static void main(String[] args) {

Person tom = new Person();

tom.name = "Tom";

tom.age = 34;

}

}

1. Животное, в него входит кот и собака, так как они являются более узким представлением класса животное и все поля, характеризующие эти классы, будут входить в них, т.е. что-то в собаку, а что-то в кота. Модификаторы доступа используем в зависимости от ситуации, т.е. если нужен в другом классе поле или метод то ставим public если нет то private или static, который используем, чтобы переменная или метод были одинаковые для всех экземпляров класса.
2. Конструктор – метод инициализирующий объект (выделяет память + заполняет табл виртуальных методов). По имени т.е. имя конструктора совпадает с именем класса + нет возвращаемого типа. Конструкторов может быть сколько угодно. Конструктор, который срабатывает по умолчанию при создании объекта класса (он всегда присутствует в классе). Вызов другого конструктора.------------this() вызывает конструктор именно этого класса. Эта операция должна быть первой.
3. Класс должен иметь конструктор без параметра с индификатором доступа public. Поля класса должны быть доступны через set get и другие методы. Класс должен быть сериализуем. Должны быть переопределены стандартные методы equals hash-code toString. Serializebl

Чект анчект – runtime;

Класс – entity,bean