Notes Day-6 Date: 03-05-2023

Passing array as a argument

```
public class ArrayDemo {

        public static void PrintArray(int arr2[])
        {
                System.out.print("Array is:   ");
                for(int i:arr2 )
                {
                System.out.println(i);
                }

                arr2[4]=112;
        }

        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                int arr[]=new int[5]; //Decalartion of Array of int type with fixed size of 5 elements
        for(int i=0;i<arr.length;i++)
        {
                System.out.println("Enter a Int value");
                arr[i]=sc.nextInt();
        }
        ArrayDemo.PrintArray(arr); //passing array to PrintArray Method as a argument

        System.out.print("Array after calling Print Array Method:   ");
        for(int i:arr )
        {
        System.out.println(i);
        }
        }
}
```

Creating array of primitive values/reference/instances

```
public class Student {
        int RollNo;
        int fees;
        void SetData()
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter Roll No");
                RollNo=sc.nextInt();
                System.out.println("Enter Fees");
                fees=sc.nextInt();

        }
        void PrintRecord()
        {
                System.out.println("Student Information");
                System.out.println(RollNo+" "+fees);
        }

        public static void main(String[] args) {

                Student[] s1=new Student[5]; //creating five references of class student
                for(int i=0;i<s1.length;i++)
                {
                        s1[i]=new Student();      //creating instances of student class and assigning them to student ref
                }
                for(int i=0;i<s1.length;i++)
                {
                        System.out.println("Enter data for "+i+1+" Instance");
                        s1[i].SetData();
                }
                for(int i=0;i<s1.length;i++)
                {
                        s1[i].PrintRecord();
                }

        }

}
```

Pass by reference

```
public class ArrayDemo {

        public static void PrintArray(int arr2[])
        {
                System.out.print("Array is:   ");
                for(int i:arr2 )
                {
                System.out.println(i);
                }

                arr2[4]=112;
        }

        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                int arr[]=new int[5]; //Decalartion of Array of int type with fixed size of 5 elements
        for(int i=0;i<arr.length;i++)
        {
                System.out.println("Enter a Int value");
                arr[i]=sc.nextInt();
        }
        ArrayDemo.PrintArray(arr); //passing array to PrintArray Method as a argument

        System.out.print("Array after calling Print Array Method:   ");
        for(int i:arr )
        {
        System.out.println(i);
        }
        }
}
```

Pass by reference

```
public class PassByValueDemo {


        static void AddNum(int p, int q) {

                System.out.println(p+q);

                p=30;

        }
        public static void main(String[] args) {

                int a=10;
                int b=20;
                System.out.println("Value of a and b before passing"+a+" "+b);
                PassByValueDemo.AddNum(a,b);
                System.out.println("Value of a and b after passing"+a+" "+b);

        }
}
```

java.util.Arrays

- This class contains various methods for manipulating arrays (such as sorting and searching). This class also contains a static factory that allows arrays to be viewed as lists.

- Refer: https://docs.oracle.com/javase/8/docs/api/ (https://docs.oracle.com/javase/8/docs/api/)

Multi Dimensional array

- In java we can create multidimentinal array using data types.

```
public class MDArrayDemo {

        public static void main(String[] args) {

                Scanner sc= new Scanner(System.in);
                int arr[][]=new int[3][3];
                for(int i=0;i<3;i++)
                {
                        System.out.println("Enter the element for "+i+1+" row");
                        for(int j=0;j<3;j++)
                        {
                                System.out.println("Enter Element");
                                arr[i][j]=sc.nextInt();
                        }
                }
                System.out.println("Entered Array is:");
                outer: for(int i=0;i<3;i++)
                {
                        inner: for(int j=0;j<3;j++)
                        {
                                System.out.print(arr[i][j]+"    ");
                                break inner;
                        }
                        System.out.println();

                }
        }

}
```

Jagged/Ragged array

- Study pictorial representation of zaaged array.

Enum

Writing menu driven code using Enum

Major and minor elements of oops

Major Elements

1. Abstraction
2. Encapsulation
3. Modularity
4. Hierarchy

Minor Elements

1. Typing: Related to data type
2. Concurrency
3. Persistance

Simulation of pass by reference in Java

- Make the pictorial representation of passing array ref to another ref.

Abstraction and Encapsulation

- Abstraction is hiding the complex functinality and providing only required funatinality to use hided one.

Encapsulation

- Hiding the elements by using access modifiers.
- providing the defined functinality to access the hidden elements.

composition versus Aggregation

- for composition kindly check Employee and EmployeeImpl class
- Aggregation

```java
class Student
{
        private int RollNo;
        private String Name;

        public int getRollNo() {
                return RollNo;
        }
        public void setRollNo(int rollNo) {
                RollNo = rollNo;
        }
        public String getName() {
                return Name;
        }
        public void setName(String name) {
                Name = name;
        }
        Student(int rollno, String name)
        {
                this.RollNo=rollno;
                this.Name=name;
        }
        void showdata()
        {
                System.out.println("Roll No is:   "+RollNo);
                System.out.println(+RollNo+"    "+Name);
        }
}
class Teacher
{
        String TeachName;
        Student s;

        Teacher(String Name, Student s1)
        {
                s=s1;
                this.TeachName=Name;
                s.setName(s1.getName());
                s.setRollNo(s1.getRollNo());

        }
        void DisplayTeacher()
        {
                System.out.println("Teacher Informatio:");
                System.out.println(this.TeachName);
                System.out.println("Student Name:   "+s.getName());
                System.out.println("Student Roll No:   "+s.getRollNo());
        }
}
public class DemoOfAgr {

        public static void main(String[] args) {

                Student stu=new Student(1001, "Sandeep");
                //Teacher t1=new Teacher("Malkeet", stu);
                Teacher t1=new Teacher("Malkeet", new Student(1002,"Mandeep"));
                t1.DisplayTeacher();
        }
}
```

Inheritance basics

- Any Object Oriented Programming allow the programmer to use the functionality of one class into another class. it is known as inheritance.