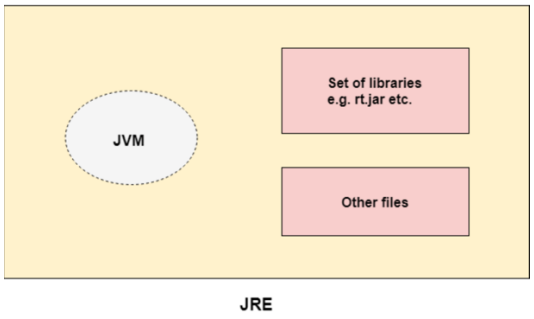
JAVA FUNDAMENTALS

JVM – Java Virtual Machine

- It is a virtual machine which doesn't exist physically.
- It provides a runtime environment in which Java byte code is executed.
- It loads code, verifies code, executes code and provides run time environment.

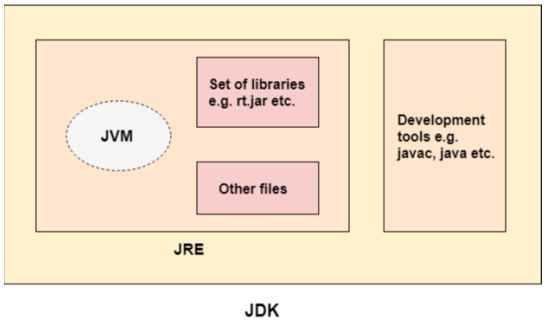
JRE – Java Runtime Environment

- Physically exist.
- It contains set of libraries and other files JVM uses.



JDK – Java Development Kit

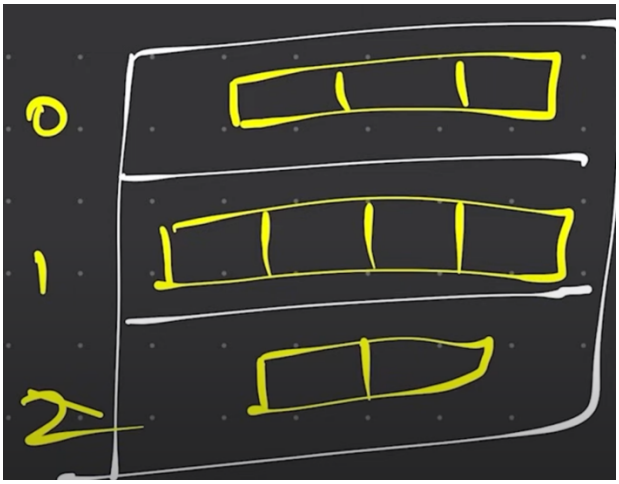
- This kit contains tools and JRE. (tools- Compiler, interpreter)
- It is a software development used to develop Java applications and applets.



Jagged Array

- It is a multidimensional array
- No need to give same size for columns

Ex - int matrix [ ][ ]= new int [3] [ ];
matrix[0]= new int [3];
matrix[1]= new int [4];
matrix[2]= new int [2];



BufferedReader

- Need to import java.io.BufferedReader and java.io.InputStreamReader.
- Have to use try catch or throws IO Exception.
- Must and should close it after use.

Ex-

```
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.util.*;
public class Main
{
    public static void main(String [] args) throws IOException
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        int number =Integer.parseInt(br.readLine());
        System.out.println(number);
       br.close();
    }
}
```

BufferedWriter

- Need to import java.io.BufferedWriter and java.io.OuputStreamWriter
- Have to use try catch or throws IO Exception.
- Must and should close it after use. (flush)

```
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.OutputStreamReader;
import java.io.OutputStreamWriter;
import java.io.BufferedWriter;
import java.util.*;

public class Main
{
    public static void main(String [] args) throws IOException
    {
        BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
        BufferedWriter bw=new BufferedWriter(new
OutputStreamWriter(System.out));

        int number =Integer.parseInt(br.readLine());
        br.close();
        bw.write(number+"\n");
        bw.flush();

    }
}
```

StringBuffer and StringBuilder

- Key difference is StringBuffer is thread safe whereas StringBuilder is not. i.e two threads can't call StringBuffer simultaneously and vice-versa for StringBuilder.
- Syntax and rest are same for both.

```
StringBuffer sb=new StringBuffer();
        sb.append("Crypto");

        StringBuilder sb1=new StringBuilder();
        sb1.append("Crypto");

```