

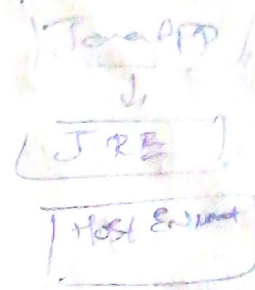
JRE  
(Java Run-time Environment)

- Creating & Running Java Apps.
- End-users requires only JRE

JDE [Integrated Development Environment]  
↳ code to compile & run it.

Package → all lowercase.

Vs JDK → feel source code & produces Java code.  
[Java Development Kit]  
↳ tools are provided platform independent  
→ Developers requires JDK



Variables / Datatypes / Math operations

↓  
int mem;

name → first character → not number  
double → 64 bit 32c float, char, Boolean

++ applies before returning value.

Operator Precedence → Prefix, Postfix, Multiplicative Additive  
Type Conversion:

long lval = 50

int lval = (int)lval

Classes → allows storage.

→ Method returns a single value.

```
Set value { this.setvalue = setvalue; }
(double a)
get value { return a; }
```

Field Initial state → zero

Field Initialization → method call.

↳ Math.random()

Constructor → No. of Parameters

↳ Type of Parameters.

Clone → Create new object instance duplicates current instance

getClass → return type information

Constructors are not inherited.

for (loop-variable-declaration; array)  
statement;

Access modifiers

Public void setA (double A)

{ this.a = a;

}

public double getA

{ return a; }

Field Initialization

↳ Initialization Block

↳ Constructor

Wrapper class

↳ treat as

'==' & equals Object

↓ Same output

for no's it is less than  
64 bit 128

Runtime Exceptions → Unchecked → Null Pointer Exception

Compile time Exceptions → Checked → IO Exception

java.lang.Comparable → Interface

compareTo → Method → used for determining relative order

Static members & fields are shared class wide

Static Initialization blocks → one-time Initialization.

↳ cannot access instance keywords.