Day 2 Basics of JAVA

Day1 Revisit

Java -version (To check the active JDK version in the system)

Intellij IDEA – (Community Edition)

JAVA

Simple HelloWorld program in notepad.

Different Flavours/Editions of JAVA (SE/EE/ME)

Features of JAVA (Security/Platform independent/ Multi purpose/Fast)

We created a project using IDE.

Core Lang Features

1. Token (Alphabets(A-Z,a-z), Symbols)
2. Keywords (Reserved words – with predefined meaning)
3. Statements/Expressions [Syntax]
4. Program [Class, Methods, variables, Parameters, Interface etc.,]
5. Projects

Variables (Declaration & Initialization)

Primitive Data types

Access & Non-Access Modifiers

String class (Special Data type) – It’s a built-in class.

Checking API documentations for different versions of JAVA.

Types of Statements (Simple, Conditional, Looping)

Setting and updating the path & Environment Variable.

echo %JAVA-HOME% - Display the value of Environment variable JAVA\_HOME

set path = %JAVA\_HOME%\bin;%path% (System Level Env Variable)

Day 2 – Basics of JAVA

Eclipse IDE –

Cloud based development

1. Eclipse che
2. Github Codespace
3. Theia etc.,

JDK – Java Development Kit ( Which contains everything to build, compile and run JAVA code) [javac]

JRE – Java Runtime Environment (java)

Package in Java is like a container.

Package name should be in lower case only.

Package declaration should be the first line in the java code.

Package naming convention – reverse of the company url - com.cams (com 🡪 cams)

Date (java.util & java.sql)

StringBuilder (No Synchronization support – should be used in single threaded application)

StringBuilder is a drop-in replacement for StringBuffer

Lexicographical comparison (Comparing strings based on ascii value)

Arrays ( Data Structure) [] – To Store the similar type of data in a consecutive memory location.

Adv : (Easy Operation) – Easy to Add data at the end, Search data easily.

Dis-adv : (Complex operation) – Fixed Length, adding data at the beginning

Types of Array (Based on Dimension)

1. Single Dimensional []
2. Multi Dimensional [][] (Matrix, 3d images)

Types of Array (Based on content)

1. Homogenous Array (Same type of data)
2. Heterogenous Array (different type of data)



Object Orient Programming (OOPs)

Procedure Oriented Programming (POP) – Security (Global variable)

Class & Objects.

Class – Blueprint for creating objects.

Object – Instance of a Class

Java is not a pure OOP bcos of primitives.(They are non objects)

Class is the pre-requisite for creating objects.

1. Abstraction – Hiding the implementation (using interface & abstract class)
2. Encapsulation – Hiding the data (using access modifiers)
3. Inheritance – Inheriting the prop & behavior of other class (Relationship)
4. Polymorphism – Reusing the same name to perform different task ( Overloading & overriding)

Class – Properties & Behavior (State/Method)

Different Types of Class

1. POJO class (Plain Old Java Object) /Simple Class /Concrete Class
2. Abstract (In-Complete) class [To provide specification]
3. Built-in/ Pre-defined/System defined class
4. User-defined/Custom Class
5. Bean Class [A class with properties, Getter, Setters & Constructors]
6. Entity Bean Class [This bean class represents a Table in DB]
7. Wrapper Class [Boolean, Byte, Character, Short, Integer, Float, Long, Double]
8. Base/Parent/Super Class
9. Sub/Derived/Child Class
10. Final Class – Can’t be inherited
11. Sealed Class
12. Static Class
13. Controller / Service/ Configuration/Util/Exception
14. Singleton (Only one instance of a class can be created)
15. Public/private/default/protected class
16. Inner class ( A class defined inside another class)
17. Anonymous Class (Nameless class)
18. Starter Class (Class with main method)

Types of Variables

1. Local Variable (Variables declared inside a method)
2. Instance variable (Variables declared outside any methods and inside a class)
3. Class/Static variable (It’s a shared variable – only one copy is shared with all objects)
4. Arguments (Variables declared inside the method signature)

Structure of a Class

<access-modifier> class <ClassName> {

<access-modifier> <data\_type> property1;

<access-modifier> <data\_type> property2;

<access-modifier> <return\_type> method\_name() -- Method signature

{ -- method body starts

} – method body ends

}

Constructor

1. It’s a special method which has similar name to the class.
2. It won’t return anything (No need to use void )
3. It helps in setting/initializing the instance variable (properties)
4. Primitives will have zero, Boolean will have false, objects will have null value

Types of Constructors

1. Default or No-Argument
2. Parameterized (Partially/Fully) or All-Argument (ReqArg Constructor)

JVM – Java Virtual Machine – Virtual Machine made up of software codes only.

Quiz Time :

1000 1 rupee coin & 10 boxes of equal size.

Rules:

1. I will full amount only.
2. One or more box (Complete box)
3. Exactly same amount

Char – 2 bytes ( 16 bits)

Char – 1 byte (8 bits) [0-255, -127 to 128]

UTF – Unicode Text Format (Unicode Text)

Java won’t support Direct pointer manipulation.

Derived data types => arrays, enums, structure, classes. (pointers)

Type casting – Is the process of converting one primitive type to another primitive type

Narrowing(explicit) boolean 🡨 byte 🡨 char 🡨short 🡨int 🡨float 🡨double 🡨long

Widening (implicit) Boolean 🡪 byte 🡪 char 🡪 short 🡪int 🡪float 🡪double 🡪 long

Boxing – Is the process of converting primitive to it’s corresponding object representation (Auto Boxing)

In Java, everything is Object.

Object is the root in JAVA.

Java.lang (default package of JAVA)

emp.equals(emp2)

OOPs concepts

1. Abstraction (Using Interface & abstract class) – Hiding the implementation

Interface = Generalization (Specification)

Concrete Class = implementation

Types of Interface

1. Marker Interface – An interface without any methods (Serializable)
2. Functional Interface – An interface with only one abstract method.

All the members of the interface are public by default.

Annotation – Meta Data – Data about data (TOC, Index in a book)

Introduced in Java version 1.5 (5)

.xml config file is needed.

All annotations start with @ symbol.

Types of annotation

1. Pre-defined/built-in Annotation (@Override, @SupressWarnings etc.,)
2. Meta-Annotation (Annotations used inside annotation - @Documented, @Target)
3. User-defined/custom annotations

Symbols

1. @ - Annotation
2. () – Methods
3. [] – Arrays
4. . – calling method using object reference
5. {} – class / block of code

Package name – com.cams

Inheritance

1. Single Inheritance (A🡪 B)
2. Multi-level Inheritance (A🡪B🡪C)

Getting user Input (in, out, err)

1. Scanner

$$$$$$$$$$$$$$$$$$$$$$$$$$ Welcome to CAMS Bank $$$$$$$$$$$$$$$$$$$$$$$$

1. Login
2. Register
3. Exit

Enter your choice[1-3] :

Enter username : sample

Enter password : $$$$$$$

Login Successful and display dashboard

1. View Accounts
2. Add Account
3. Edit Account
4. Deposit
5. Withdraw
6. Logout

Enter your choice [1-6] :

Accounts List

Add New Account

Edit Account

Deposit

Withdraw

Thanks for using CAMS Bank.!!!