**What is JDK,JRE,JVM?**  
*The JDK is the kit which provides the environment to* ***develop*** *and* ***execute*** *the java program. JDK is a kit which includes two things:  
- Development Tools (****To develop****: interpreter/loader, compiler (javac), an archiver (jar), a document generator (Javadoc), and other tools)  
- JRE (****To execute****)  
  
The JRE provides the minimum requirements for executing a Java application; it consists of the Java Virtual Machine (JVM), core classes via class loader and supporting files.  
  
The JVM is a very important part of both the JDK and JRE because it is contained or built in both. JVM is responsible for executing the Java program line by line hence it is also known as interpreter.*

**Is JVM, a compiler or interpreter ?**  
*The JVM is an interpreter.*

**Why Java don't use pointers?***- Memory access via pointer arithmetic is unsafe, Java has a robust security model and disallows pointer arithmetic for this reason.  
- Array access via pointer offsets, Java handles this by indexed array access so pointers are not needed.  
- Java provides very good automatic garbage collection which takes care of memory management.  
- Java doesn’t have pointers because it doesn’t need them for general purpose OOP programming. Adding pointers to Java would undermine security and robustness and make the language more complex.*

**What are the various types of Class loaders used by JVM ?***Bootstrap: Loads classes from rt.jar package. They are found in jdk/jre/lib/ret.jar and written in native language  
Extension: Loads classes from ext folder. Found in jdk/jre/lib/ext and written in Java  
Application/System: Which refers to the class being run. Written in Java.*

**How are classes loaded by JVM?***The JVM uses the bootstrap class loader for loading JDK internal classes like in rt.jar and other core libraries.  
The extension class loader is a child of the bootstrap class loader and takes care of loading the extensions of the standard core Java classes.  
The system/application class loader loads files found in the classpath environment.  
Class loaders are part of the JRE. When the JVM requests a class, the class loader tries to locate the class and load the class definition into the runtime using the fully qualified class name.*

**Which memory areas does instance and static variables use?***All static and instance variables are stored in PermGen space of the heap memory.*

**What is PermGen or Permanent Generation?***PermGen is a special heap space separated from the main memory heap. The JVM keeps track of loaded class metadata here.*

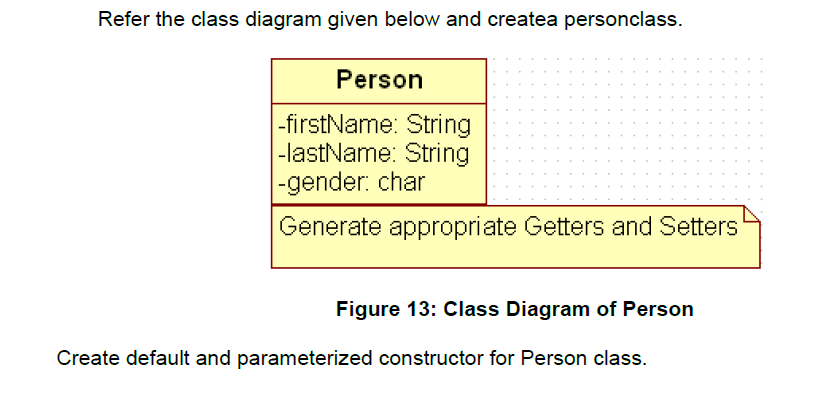
**What is metaspace?***Metaspace is a memory space that grows automatically by default. Metaspace has replaced the older PermGen memory space due to how it handles memory allocation.*

**Describe what happens when an object is created in Java.***- Memory is allocated  
- Fields are initialized to their default values  
- The first line of the chosen constructor is invoked  
- The instance initializer is executed and the fields are initialized to their requested values.  
- The rest of the constructor code is executed*

**Different types of memory used by JVM?***Method area: Stored the class code, code of the variables and methods  
Heap: Java objects are created in this area  
Java Stack: Results are stored in the stack  
Native method stacks: Similar to Java stack, native methods are executed on the Native method sacks* **Does Java Pass by Value or Pass by Reference?***Java is pass-by-value*

**What are native methods?***A native method is a Java method whose implementation is written in another programming language such as C.*

Lab work:



Also Create “PersonMain.java” program and write code for following operations:

a) Create an object of Person class and specify person details through constructor.

b): Modify to accept phone number of a person. Create a new method to implement the same and also define method for displaying person details.

c): Modify the above program, to accept only ‘M’ or ‘F’ as gender field values. Use Enumeration for implementing the same

--After spring--

d) Add a method called calculateAge which should accept person’s date of birth and calculate age of a person.

e) Add a method called getFullName(String firstName, String lastName) which should return full name of a person

f) Display person details with age and fullname.

.

|  |
| --- |
| Write a program to accept a number from user as a command line argument and check whether the given number is positive or negative number. |
| Uploaded to github |
|  |