Enums in JAVA
(by Mr. RAGHU)

FAQs:-

1. What is enum?
   enum is keyword, used to create new data type with 'SET OF POSSIBLE VALUES'.
2. In which version of Java enum concept is added?
   JDK 5 version enum concept is added in java.
3. How to create enum?
   public enum <EnumName> {
   //enum property in Upper case (or) enum members
   }
4. What are default methods added to every enum?
   values() and valueOf() are two methods added to every enum by compiler.
5. What is super type for every enum?
   java.lang.Enum is the default super type for every enum.
6. What are modifiers added to every enum property?
   Every enum type is public static final by default.
7. How to access enum property?
   It is static type. So, syntax is: **enumName.property**
8. What is ordinal of enum property? How to find it?
   Ordinal is an index number assigned to every enum property by java compiler. It starts from zero.
9. Can we write nested enums?
   Yes, it is possible. [ie enum inside another enum]
10. Can we define constructors inside enum?
    Yes, compiler provides default constructor. We can even define any parameterized constructor.
11. Can we write variables and methods in enum?

Yes, it is possible. Variables used to change internal values of properties.

12. Which method is used to read all properties of a enum?
    values() method is used to read all possible properties as Enum type array.
13. How to convert String to enum type?
    By using valueOf("_") method, we can convert String to enum type.
14. How to convert enum type to String?
    By using toString() method we can convert enum type to String type.
15. Can enum extends any enum or class?
    No. Not by programmer (or) externally. By default every enum extends java.lang.Enum
16. Can enum implements any interface?
    Yes, it is possible. But enum should implement all abstract methods.
17. Can we apply abstract or final modifiers to enum?
    No modifiers are allowed to enum name or its properties.
18. What members can we define in enum?
    Enum can have 'Enum properties' and also 'variables, blocks, constructors, methods, inners'.

Find Output/Errors:-

1.
```
package com.app;
enum TimeMode{
	AM,PM,NOON;
}
public class Test {
	public static void main(String[] args) {
		//Syntax: enumName.property
		System.out.println(TimeMode.AM);
	}
}
```
Output:_____

2.
```
package com.app;
enum Grade{
	A,B,C,D;
}
public class Test {
```

```
	public static void main(String[] args) {
		Grade[] gobs=Grade.values();
		for (Grade g:gobs) {
			System.out.println(g);
		}
	}
}
```

Output:_____

3.

```
package com.app;
enum Direction{
	EAST,WEST,NORTH,SOUTH;
}
public class Test {
	public static void main(String[] args) {
		System.out.println(Direction.EAST.ordinal());
		System.out.println(Direction.WEST.ordinal());
		System.out.println(Direction.NORTH.ordinal());
		System.out.println(Direction.SOUTH.ordinal());
	}
}
```

Output:_____

4.

```
package com.app;
enum Week{
	SUN,MON,TUE,WED,THR,FRI,SAT;
}
public class Test {
	public static void main(String[] args) {
		//ordinal-ordinal
		System.out.println(Week.MON.compareTo(Week.FRI));
	}
}
```

Output:_____

5.

```
package com.app;
enum Gender{
	MALE(10),FEMALE(20), OTHER(15);

	private int code;

	Gender(int code){
		this.code = code;
	}

	public int getCode() {
		return code;
	}

}
public class Test {
	public static void main(String[] args) {
		System.out.println(Gender.MALE);
		System.out.println(Gender.MALE.ordinal());
		System.out.println(Gender.MALE.getCode());
	}
}
```

Output:_____

6.

```
package com.app;
enum ExamResult{
	PASS(1,"GOOD"),FAIL(0,"BAD"),ABSENT(-1,"UNKNOWN");

	private int code;
	private String message;

	ExamResult(int code,String message){
		this.code=code;
		this.message=message;
	}

	public int getCode() {
		return code;
```

```
	}

	public String getMessage() {
		return message;
	}

}
public class Test {
	public static void main(String[] args) {
		ExamResult[] arr=ExamResult.values();
		for(ExamResult e:arr) {
			System.out.println(e +"="+ e.getCode()+"/"+
e.getMessage());
		}
	}
}
```

Output:_____

7.

```
package com.app;
enum SignalSystem {
	RED,GREEN,YELLOW;
}
public class Test {
	public static void main(String[] args) {
		SignalSystem  s = SignalSystem.GREEN;
		switch (s) {
		case RED:
			System.out.println("Stop vehicle!"); break;
		case GREEN:
			System.out.println("You can go now!!"); break;
		case YELLOW:
			System.out.println("Ready to stop/start!!!"); break;
		default :
			System.out.println("Invalid entry");    break;
		}
	}
}
```

Output:_____

8.

```
package com.app;
enum CricketMatch {
	ONEDAY,TEST,T20;
}
public class Test {
	public static void main(String[] args) {
		CricketMatch cm=CricketMatch.TEST;

		if(CricketMatch.ONEDAY.equals(cm)) {
			System.out.println("Enjoy one day cricket!!");
		}else if(CricketMatch.TEST.equals(cm)) {
			System.out.println("Enjoy 5 days cricket!!");
		}else if(CricketMatch.ONEDAY.equals(cm)) {
			System.out.println("Enjoy 3 hours cricket!!");
		}else {
			System.out.println("Invalid option");
		}
	}
}
```

9.

```
package com.app;
enum Choice {
	YES,NO,MAYBE;
}
public class Test {
	public static void main(String[] args) {
		Choice c=Choice.YES;
		do {
			System.out.println("Hello "+c);
			c++;
		} while (c!=Choice.NO);
	}
}
```

10.

```
package com.app;
enum Choice {
	YES,NO,MAYBE;
}
public class Test {
	public static void main(String[] args) {
		Choice c[]=Choice.values();
		for (int i = 0; i < c.length; i++) {
			Choice ch = c[i];
			System.out.println(ch);
		}
	}
}
```

Output:_____

11.

```
package com.app;
import java.util.Arrays;
enum Code {
	M,B,A,D,Z;
}
public class Test {
	public static void main(String[] args) {
		Code c[]=Code.values();
		Arrays.sort(c);
		for (int i = 0; i < c.length; i++) {
			Code ch = c[i];
			System.out.println(ch);
		}
	}
}
```

Output:_____

12.

```
package com.app;
enum Model {
	LARGE,SMALL,MEDIUM;
}
public class Test {
```

```
	public static void main(String[] args) {
		Model m=Model.valueOf("SMALL");
		System.out.println(m);
	}
}
```

Output:_____

13.

```
package com.app;
enum Model {
	LARGE,SMALL,MEDIUM;
}
public class Test {
	public static void main(String[] args) {
		String s=Model.MEDIUM.toString();
		System.out.println(s.length());
	}
}
```

Output:_____

True or False:-

1. enum is keyword
2. enum contains constants (final variables)
3. Every enum member/property contains index(ordinal) number
4. Ordinal starts from one(1).
5. Ordinal number can be changed by programmer
6. Enum property is default public static final
7. Every enum extends internally java.lang.Enum
8. We cannot add methods to enum.
9. toString() method is used to convert String to enum type
10. Enum property can be made as private type
11. Nesting of enums not possible.
12. enum can extend any other class
13. enum can implement interfaces
14. enum can be declared as abstract

15. Empty enum can be created (enum without properties)
16. enum type can be used at switch-case
17. Enums supports equals() and compareTo() method
18. compareTo() returns the difference of ordinals
19. enum can have constructors
20. Use values() method to read all enum properties
21. valueOf() method is used to convert String to enum type.
22. Enum type are called as User defined DataTypes.

Fill in the blanks:-

1. enum property default modifiers ____ , ___ and _____
2. _____ method is used to convert enum to String type
3. enum get methods added by _____
4. _____ is called index number to every enum property
5. _____ is the syntax to access enum property.
6. enum default super type is _____.
7. _____ method is used to read index number of enum property
8. _____ method is used to read all properties of enum
9. Every enum converted to _____ type by compiler.
10. _____ is the method used to convert String to enum type

Match the Following:

1. Enum	( )	a. Enum to String type
2. values()	( )	b. public static and final
3. java.lang.Enum	( )	c. Keyword
4. Ordinal	( )	d. Read all properties
5. valueOf()	( )	e. Super Type for enum
6. toString()	( )	f. Index number for properties
7. Default modifiers	( )	g. String to enum type

Programming:

1. Write an enum with name 'Months' with all possible values.
2. Write an enum with name 'BankingTypes' with all possible values.
3. Write an enum with name 'DbOperations' with all possible values.
4. Write an enum with name 'TrainTicketStatus' with possible values.
5. Write an enum with name 'CricketerType' with possible values.
6. Write an enum with name 'InputModes' having values FORM, DB. Print all values.
7. Write an enum with name 'PaymentOptions' having values CASH, CARD, CHEQUE and print all values with their ordinals using any loop.
8. Write an enum with name 'BusType' having option GENERAL, SLEEPER, SEMISLEEPER. Print one by one value with ordinals.
9. Write an enum with name 'Slot' having options MORNING, AFTERNOON, EVENING. Define one switch-case, read input from end user and print message example like 'Welcome to Morning batch'.
10. Write an enum 'SocialMedia' having values FACEBOOK(5), TWITTER(8), WHATSAPP(10) using Param constructor(int code). Print all values with their codes.
    [Hint: Use variable, param constructor, get method for variable]

FB: https://www.facebook.com/groups/thejavatemple/

email: javabyraghu@gmail.com