```java
package javaprj.ramda;
import java.util.function.Supplier;


@FunctionalInterface
interface MySupplier2<T> {
	T get( );  //  입력이 없고 반환이 있는거 !
}

public class MySupplierTest2 {
	public static void main(String[] args) {
	 //1 이름있는 클래스작성해서

	//지역내부클래스 , 매서드내에서 클래스 작성 => 대부분 익명클래스로 작성한다
	class MySupplierImp2 implements MySupplier2<Integer>{
		@Override
		public Integer get() {
			return (int) ( Math.random()* 100 ) +1;
		}
	 }

	 MySupplierImp2 imp = new MySupplierImp2();
	 int result =imp.get();
	 System.out.println( result);


	 // 2 익명클래스
	 MySupplier2<Integer> imp2 = new MySupplier2<>() {
		@Override
		public Integer get() {
			return (int) ( Math.random()* 100 ) +1;
		}
	 };

	 int result2 =imp2.get();
	 System.out.println( result2);

	 //3. 람다식 사용 ( 구현해야 할 매서드가 한 개인 경우에 한해서만 람다식 허용가능)
	 MySupplier2<Integer> imp3= ()-> (int) ( Math.random()* 100 ) +1 ;
	 int result3 =imp3.get();
	 System.out.println( result3);

	// MySupplier2<String> imp4= ()-> "hello" ;
	  MySupplier2<String> imp4= ()-> { return new String("hello"); };
	  System.out.println( imp4.get());
	 // Supplier<T> 인터페이스 제공
	 /*
	  * interface Supplier<T> {
	  *	T get( );
	  * }
	  */
	 Supplier<Integer> s = ()-> (int) ( Math.random()* 100 ) +1 ;
	 Supplier<String> s2= ()-> { String str="hi acorn" ; return str ; } ;

	 System.out.println( s.get());
	 System.out.println( s2.get());
	}
}
```