

Threads and concurrency

- Change the class name to 'implements runnable' - Implement the method for parallelising (e.g. isPrime) - Implement sync method. For any code executed in sync method, threads are paused until the last call is complete. - Implement the actual parallelising in @override run - Use a while loop to execute: `while((n=getNextInteger())<N_INTS){ results[n]=isPrime(n); }`

Arrays and arraylists

Array to Arraylist	<code>Arrays.stream(row_array).boxed().collect(Collectors.toList());</code>
ArrayList to Array (Int to int)	<code>int[] new = old.stream().mapToInt(i->i).toArray();</code>
Get Max	<code>cats.values().stream().mapToInt(x->x).max().getAsInt()</code>
Copy array	<code>var included= Arrays.copyOfRange(in,0,end-values)</code> ; OR USE FORLOOP.
Reverse list	<code>Collections.reverse(names);</code>
Convert integer to int	<code>int x.intValue()</code>

Classes

NoClassDefFoundError	Make sure when you create the class, it's in the right folder
Initialise non-static inner class from psvm	Switch the 'new' position: <code>Cost cost=new Cost(); Cost.Item item1=cost.new Item("",50);</code>
Compare class of objects	Compare classes <code>animals[x][y].getClass().equals(World.Fox.class)</code>

Misc imperative programming

Case switch syntax	<code>return switch(group){ case LEGUME -> Group.BRASSICA; default -> null; } }</code>
Deep copy	<code>List.copyOf, Arrays.copyOf</code> or <code>Object.clone()</code>
End loop of a scanner	<code>while(sc.hasNextDouble()){ (ctrl+d to end)</code>
ifelse shortened	<code>var last = used.isEmpty() ? null : used.get(-used.size() - 1);</code>
generate random numbers	<code>var rand=newRandom(); var list=rand.ints(10,0,500);</code>
try catch	<code>try{} catch(Exception E){}</code>

Recursion

Keep track of	<code>full_set, used, last, current, expected_next</code>
Ending recursion	<code>if(used.size() == x) -> all_sets.add(List.copyOf(used)); return;</code>
recursing	<code>used.add(crop); getFixedRotation(crops,seasons,used,-rotations); used.remove(crop);</code>

Hashing and HashCodes

Integer hash by division	<code>Integer.toUnsignedLong(value) % nBuckets;</code>
Integer hash by division	<code>Math.floor(nBuckets ((Integer.toUnsignedLong(value) 1.618) % 1));</code>
String hash	iterate over each char: for i, int hashchar = hashfn(value.charAt(i),buckets); hash = (hash + hash_char) % buckets;

Sets

Add items to a set	<code>Set<Food>foods=newHashSet<>(); foods.addAll(List.of(cake,Vegetable_Soup,Waffles,Potato_Gratin));</code>
Sort a Set	1. Create arraylist 2.add items of the set to the arraylist 3. use arraylist.sort method. <code>ArrayList<Item>sorted_items=newArrayList<Item>(); sorted_items.addAll(this.items.values()); sorted_items.sort((a,b)->a.compareTo(b));</code>

Generics

class header	<code>public class Q6DelayedRelease<T></code>
generic array	<code>E[] arr = (E[])new Object[INITIAL_ARRAY_LENGTH];</code>
ClassCastException (Add AFTER declaration of T)	<code>this.items = (Item[]) Array.newInstance(new Item(delay,value).getClass(), slots);</code>
inner class T	try add <T> qualifier to inner and outer class



