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

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: Cost cost=new Cost(); Cost.Item item1=cost.new Item("",50);
Compare class of objects	Compare classes animals[x][y].getClass().equals(World.Fox.class)

Misc imperative programming

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

Recursion

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

Hashing and HashCodes

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

Sets

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

Generics

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