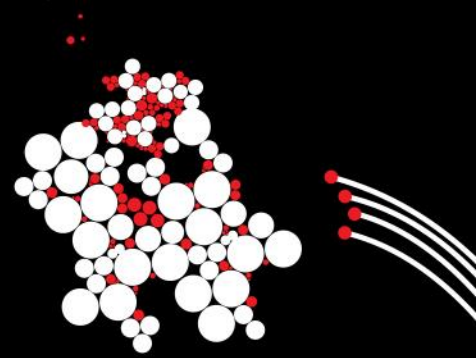


UNIVERSITY OF TWENTE.

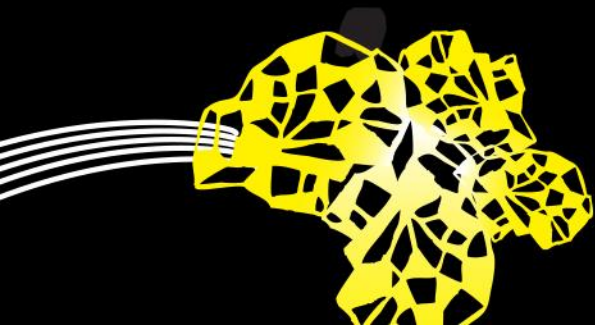


P4.4: Q & A

TOM VAN DIJK

201700117-1B MODULE 2: SOFTWARE SYSTEMS

5 DECEMBER 2019



ASKED QUESTIONS

- Interfaces
 - List and ArrayList (and when to use ArrayList vs other implementations)
 - Mergesort
 - Variable hiding and method overriding
 - Subtyping
-
- After this: whatever you want to talk about

INTERFACES

- Interfaces tell a class that implements them what they should implement
- Interfaces are the contract, the specification
- Example: List and its many implementations
- Often interfaces are written first, then people can work together implementing different interfaces, working on different parts of the software, writing tests and implementations independently
- Often other classes “talk to” the interfaces, not the implementation
- Example: often use the type List as the variable type, and only use LinkedList or ArrayList when constructing the List

METHOD HIDING AND OVERRIDING

- Variables are **hidden** when
 - some class A has a variable “someName”
 - a **subclass** B extending A also has a variable “someName”
 - then in a method of B, if you use “someName”, it refers to the variable declared in the B class
 - you cannot access the someName variable in A anymore from methods of B
 - **unless you use super.someName**
- Variables are **shadowed** when
 - a method of a class A has a local variable “someName”
 - now you must use **this.someName** to access the class variable instead of the local variable
- Methods are **overridden** when
 - a subclass B implements a method with the same name and parameters
 - this method **replaces** the method of class A for objects of subtype B

SUBTYPING

- Dynamic type
 - Important when **type casting**
 - Important when using **instanceof**
- Static type
 - Important when **assigning** some object to some variable
 - Important when **invoking** some object's methods or accessing its variables

MERGESORT

- Just google it
- **Implement it!!!**
- **Learn how to debug your code**