ICT167 ANS2

Object oriented design properties:

- Supports abstraction in particular data abstraction. Data abstraction allows the data in the program to be organised by putting related data together and having a simple name for the big lump of data
- Supports re-use, allowing programmers to reuse their own and other people's code
- Encourages the design of code to be appropriate for a problem domain rather than a particular task

Object vs Class: *Every object is an instance of a class*

- A class is a description of a kind of object or plan for all possible object. An object represents an identity that can be distinctly identified.
- *Instantiation* is the creation of an object from a class. We can then only use the object

- Object inherit the methods of the class and the Instance variable

Components of an object:

- Identity- identify what object you are dealing with
- State- the attributes object may have and it might change
- Behaviour- what the object performs and what things can be done to the object

Primitive type variable vs Class type variable:

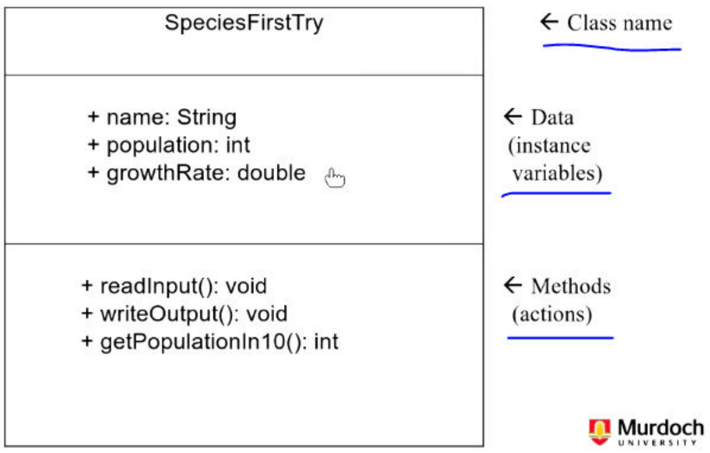
- Each variable must be declared to be a particular type. The two types may include- primitive type (characters) or class type (array and string)
- A primitive type variable stores a value in a memory location assigned to the variable. A class type variable stores the memory address of where the object is located.

- A variable of type: class are a data type class they refer to a particular object belonging to a class. This type of variable contains data and methods for the class
- The primitive types are passed to other methods through call-by-value while class type variable are passed using call-by-reference

(refer to ICT167Ans3 to learn how to pass an object/class type variable to a method)

UML class diagram:

- Outlines the definition of a class diagrammatically
- + public instance variable or method: (refer to future lecture) Can be used by client w/o any restrictions
- - private instance variable or method: (refer to future lecture)



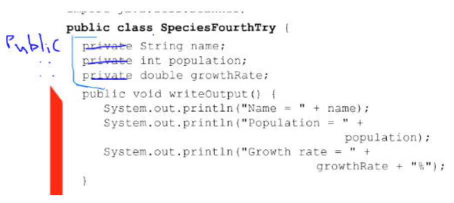
http://pages.cs.wisc.edu/~hasti/cs302/examples/UMLdiagram.html

Instance variable vs local variable vs static variable: [Will be in EXAM. Refer to lecture 4 32mins]

- <u>*Instance variables*</u> are the data that are <u>*attached*</u> to the <u>class</u> itself. They are declared in a class and can be <u>used</u> by <u>all</u> the <u>methods</u> in the class.
    - EG SpeciesOfMonth = 15;
- A <u>local variable</u> are <u>defined</u> within a <u>method</u>. The variable is only <u>used</u> <u>within</u> the <u>method</u> that declared the variable.
- A <u>static variable</u> is a variable that <u>stores</u> the <u>same value</u> for <u>ALL objects</u> created and can be accessed by all objects for that class. Changes to the static variable changes the variable for all object <u>unless</u> there is keyword <u>FINAL</u>

Public Instance variables:

- Means that any other class/program can directly access/change the instance variable. So client can access instance variable from their client program
- Attached to the class itself not inside methods
    - EG: public String name, public int population
- To use: <u>[class variable].variable = …</u> ← Must have instantiated

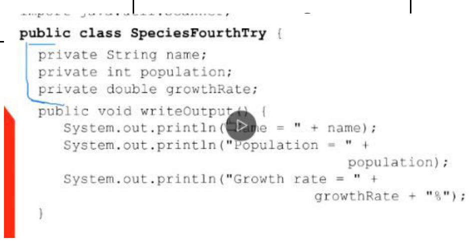


```
public class SpeciesFourthTry {
    ~~private~~ String name;
    ~~private~~ int population;
    private double growthRate;
    public void writeOutput() {
        System.out.println("Name = " + name);
        System.out.println("Population = " +
                                      population);
        System.out.println("Growth rate = " +
                                  growthRate + "%");
    }
```

Private instance variable

- Client can't access outside class itself unless getter/change instance variable from their client program
    - EG for like bank balance variable in ATM
- Attached to class itself not inside methods or method itself is attached to class
    - EG: private String name, private int population
- To use: Create *both* accessor method and mutator method (Refer to lecture 3)

OBJECT	METHOD	DESCRIPTION	RETURN? +



```
public class SpeciesFourthTry {
    private String name;
    private int population;
    private double growthRate;
    public void writeOutput() {
        System.out.println("Name = " + name);
        System.out.println("Population = " +
                                        population);
        System.out.println("Growth rate = " +
                                      growthRate + "%");
    }
```

STRING (class) methods:

```
import java.lang;

String [class variable] = new String("Hello");
```

[class variable]	.length()	length of the string	R → Integer P → No IMPORTANT
[class variable]	.trim()	Returns a copy of the string, with leading and trailing spaces removed.	R → String P → No
[class variable]	.charAt([parameter])	Return the character for the specified index	R → Character P → Yes number/integer variable
[class variable]	.substring([parameter1, parameter 2])	IMPORTANT	
[class variable]	.equals([class variable2])	Compares two strings based on the content of the string. **NOTE == doesn't check this it checks whether string are in common memory location**	R →Boolean P → Yes IMPORTANT
	.equalsIgnoreCase()	Compares two strings based on the content of the string irrespective of case of the string (ie capital letters)	
	.compareTo()	Compares two strings lexicographically	IMPORTANT
	.compareToIgnoreCase()		
	indexOf()		

Character Input:

```
Char InputCharacter = ‘[character]’;
```

Must have ‘’