COMP90041
Programming and Software Development

UML

UML

- Graphical representation systems for program design have been used
  - *Flowcharts* and *structure diagrams* for example
- *Unified Modeling Language* (*UML*) is yet another graphical representation formalism
  - UML is designed to reflect and be used with the OOP philosophy

History of UML

- In 1996, Brady Booch, Ivar Jacobson, and James Rumbaugh released an early version of UML
  - Its purpose was to produce a standardized graphical representation language for object-oriented design and documentation
- Since then, UML has been developed and revised in response to feedback from the OOP community
  - Today, the UML standard is maintained and certified by the Object Management Group (OMG)

UML Diagrams

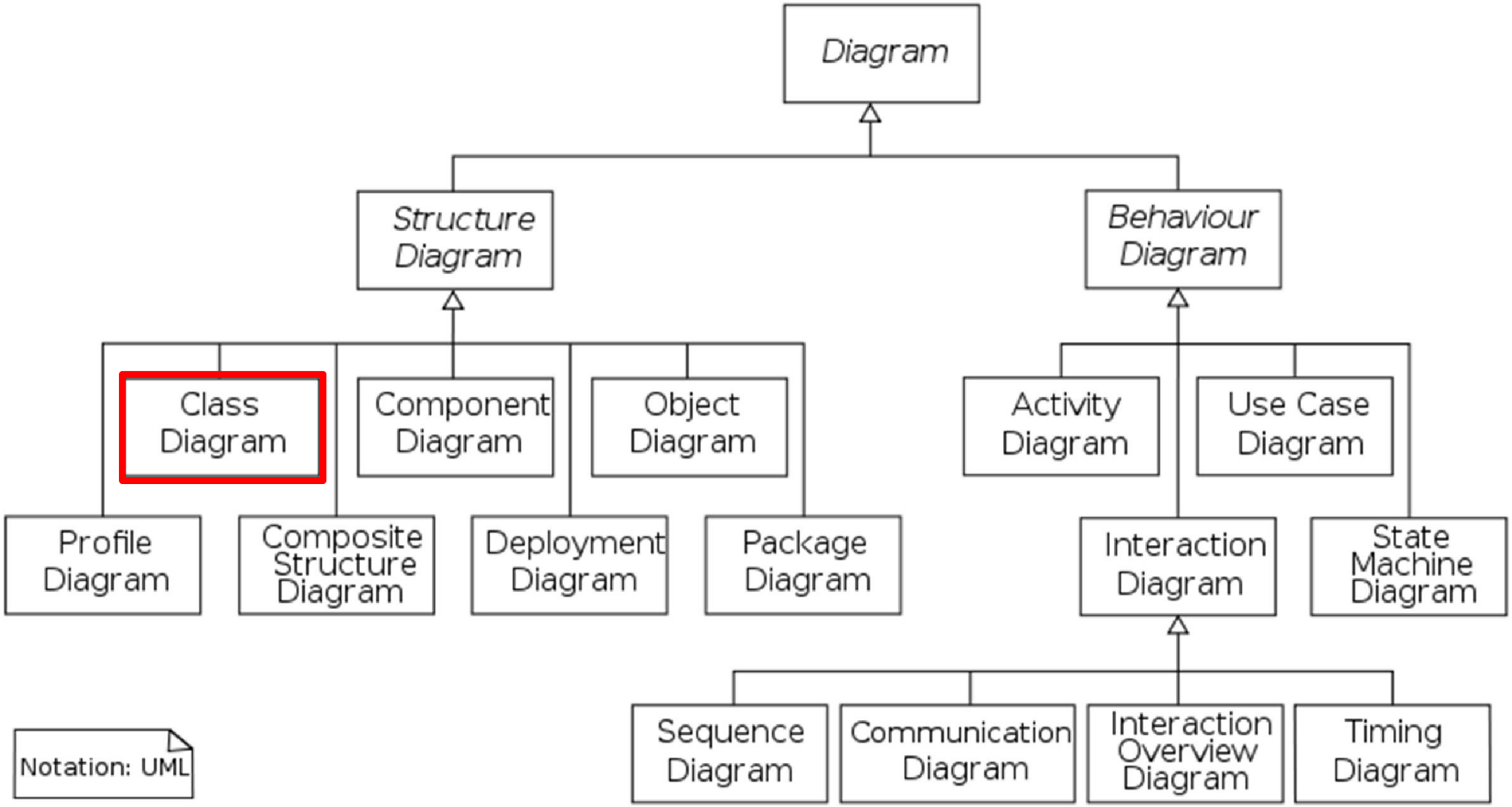


Image from wikipedia

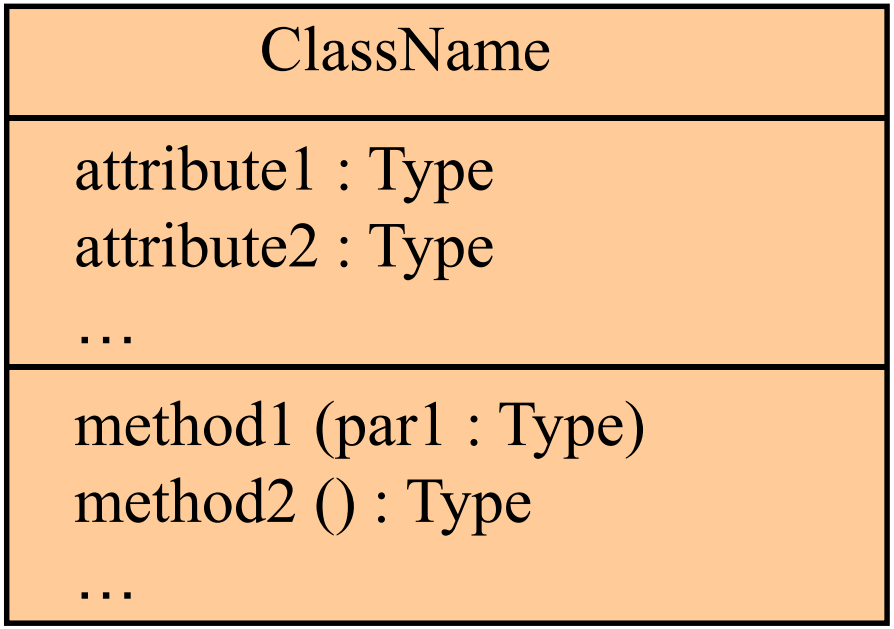
UML Class Diagrams

- A type of structure diagram
- Describes the structure of a system by showing the system's:
  - Classes
  - Their attributes (and the accessibility)
  - The relationships among the classes

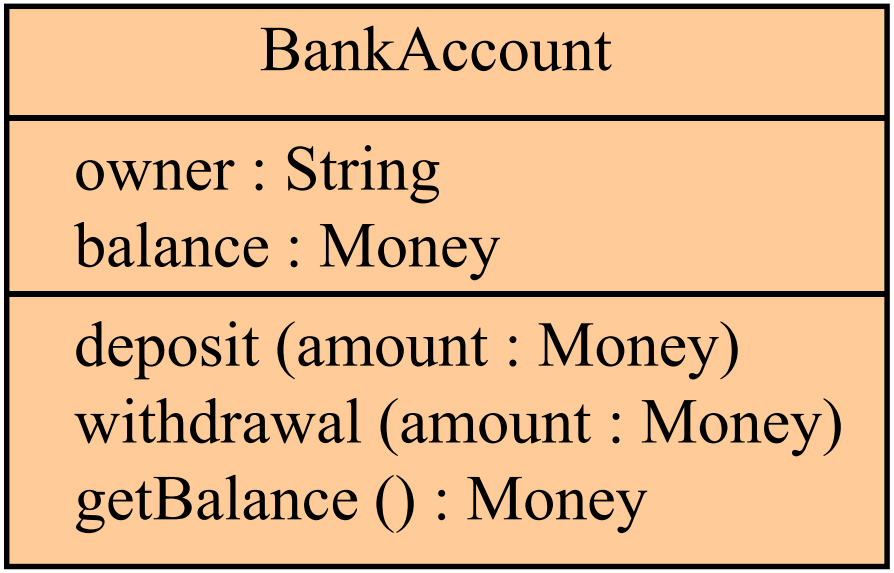
UML Class Diagrams

- Classes are central to OOP, and the ***class diagram*** is the easiest of the UML graphical representations to understand and use
- A class diagram is divided up into three sections
  - The top section contains the class name
  - The middle section contains the data specification for the class
  - The bottom section contains the actions or methods of the class

UML Class Diagrams



A UML Class


BankAccount
owner : String
balance : Money
deposit (amount : Money)
withdrawal (amount : Money)
getBalance () : Money

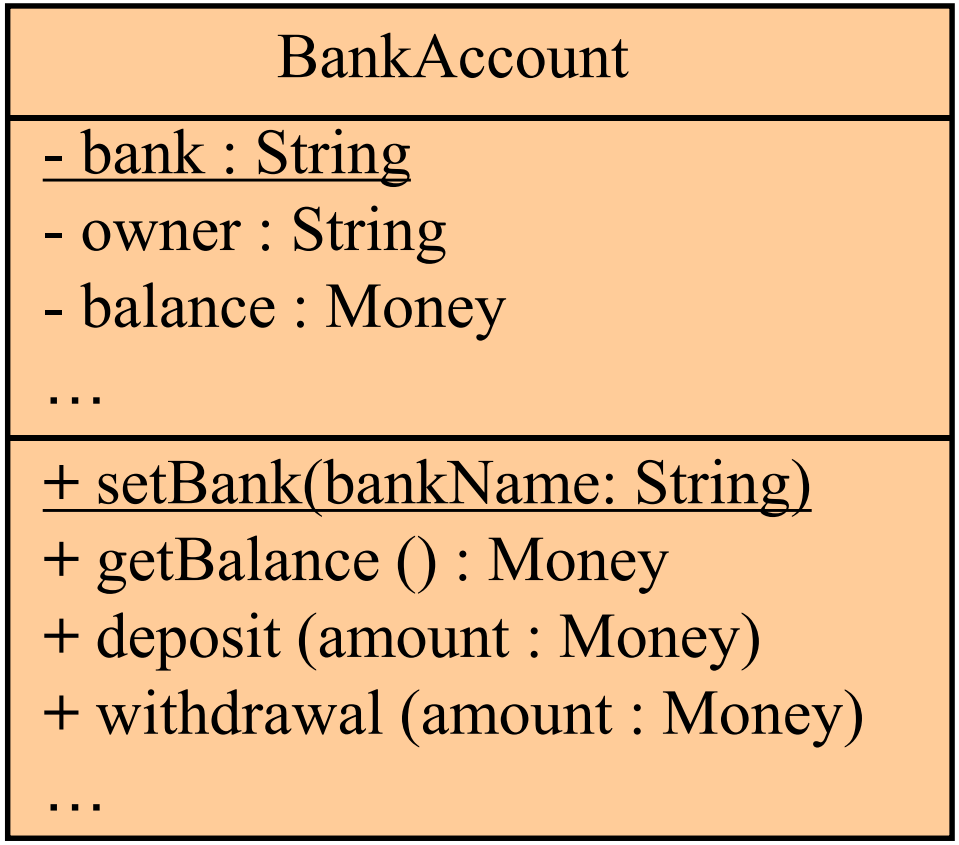
UML Class Diagrams

- The data specification for each piece of data in a UML diagram consists of its name, followed by a colon, followed by its type
- Each name is preceded by a character that specifies its access type:
  - A minus sign (-) indicates private access
  - A plus sign (+) indicates public access
  - A sharp (#) indicates protected access
  - A tilde (~) indicates package access

UML Class Diagrams

- A class diagram need not give a complete description of the class
  - If a given analysis does not require that all the class members be represented, then those members are not listed in the class diagram
  - Missing members are indicated with an ellipsis (three dots)

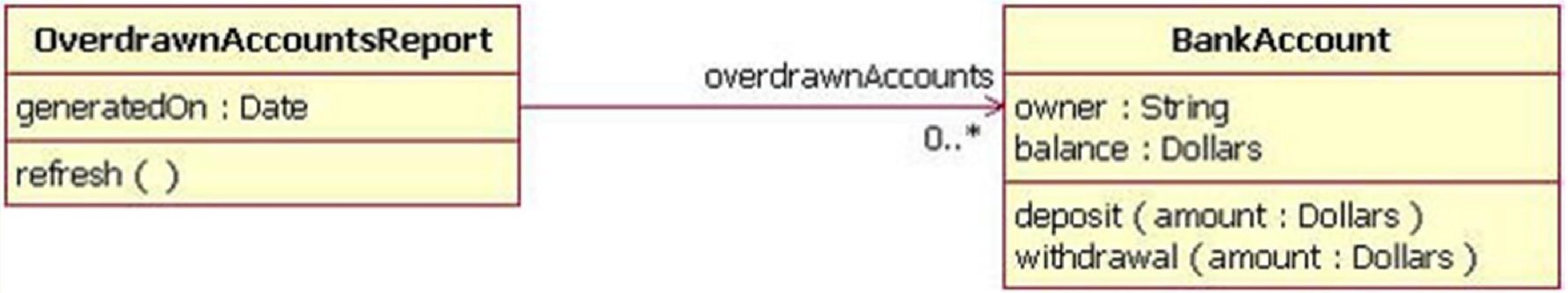
A UML Class


BankAccount
- bank : String
- owner : String
- balance : Money
…
+ setBank(bankName: String)
+ getBalance () : Money
+ deposit (amount : Money)
+ withdrawal (amount : Money)
…

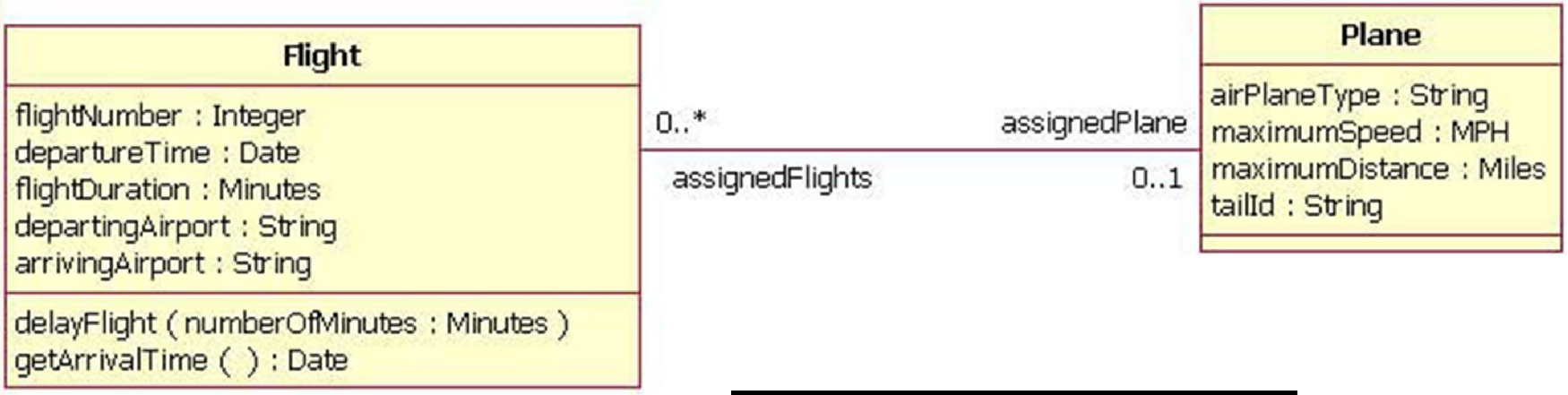
Class Interactions

- Rather than show just the interface of a class, class diagrams are primarily designed to show the interactions among classes
- UML has various ways to indicate the information flow from one class object to another using different sorts of annotated arrows
- UML has annotations for class groupings into packages, for inheritance, and for other interactions
- In addition to these established annotations, UML is extensible

Associations: Unidirectional



Associations: Bidirectional



Multiplicity Values	
Indicator	Meaning
n	exactly n
*	zero or many
0..n	zero to n
m..n	m to n

UML Packages