UML Class Diagram

The Basics of Class Diagrams
for a single class

Unified Modeling Language

- A standard notation for describing *software models and code*
- Unifies the notation of Booch, OMT (Rumbaugh et al), and OOSE (Jacobson et al)

Many Kinds of UML Diagrams

UML has 20+ different kinds of diagrams.

Each diagram shows a different kind of information (or different *view*) of application.

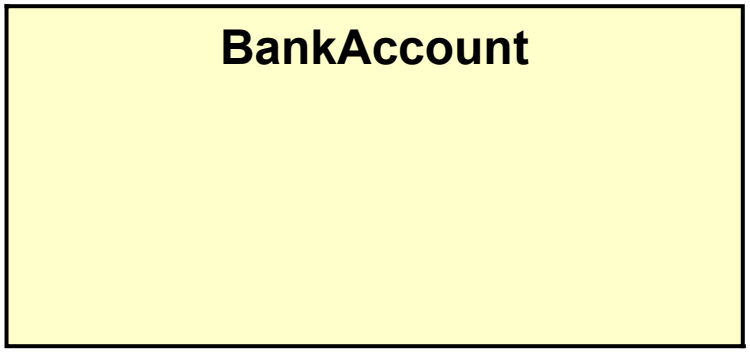
- Class diagram
- Sequence diagram
- State Machine diagram (*aka State Chart Diagram*)
- Object diagram
- Interaction diagram
- Activity diagram
- Package Diagram
- many others!

These 3 are the most common and most important to know.

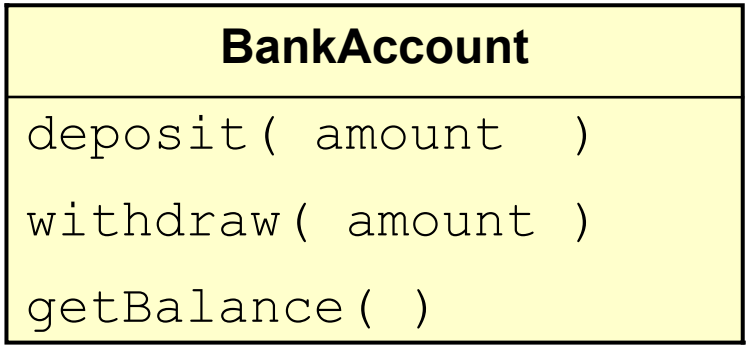
Class Diagram

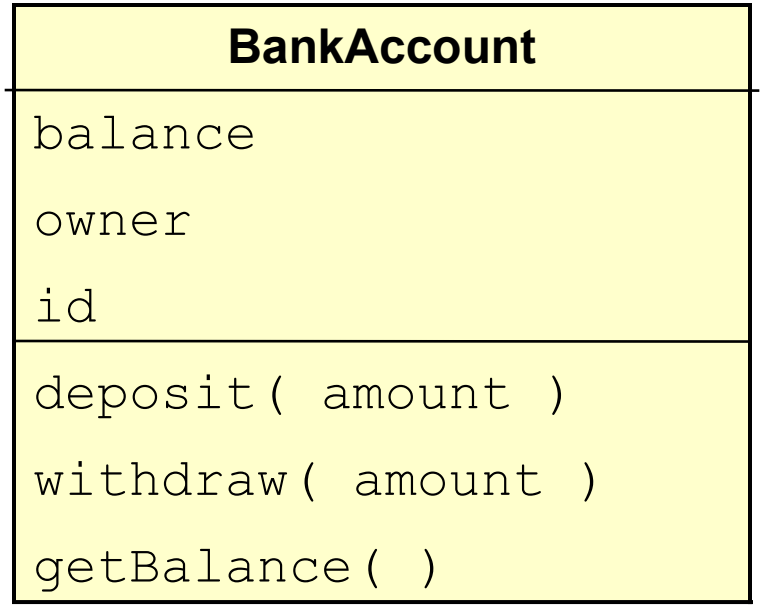
- A class diagram shows the structure of a class
- It can also show relationships between classes

Here is the *simplest possible class diagram*:



Class Diagrams methods & attributes





Class Diagram with data types

- Class diagram can show data types & visibility
- <u>Not</u> Java notation ("double balance")

BankAccount
`-balance: double`

`+deposit(amt: double): void`
`+withdraw(amt: double): boolean`
`+getBalance( ): double`

Visibility of Members

BankAccount
- balance: double
+ deposit(amount: double)
+ withdraw(amount: double)
+ getBalance( ): double

- balance is private (visible only within **BankAccount** objects)
- deposit, withdraw, getBalance are public

Visibility Prefixes

+ means **public**
  - Visible everywhere
– means **private**
  - Visible only in the class in which it is defined
means **protected**
  - Visible either within the class in which it is defined or within subclasses of that class
~ means **package** visibility
  - visible to other classes in the same package

Constructors

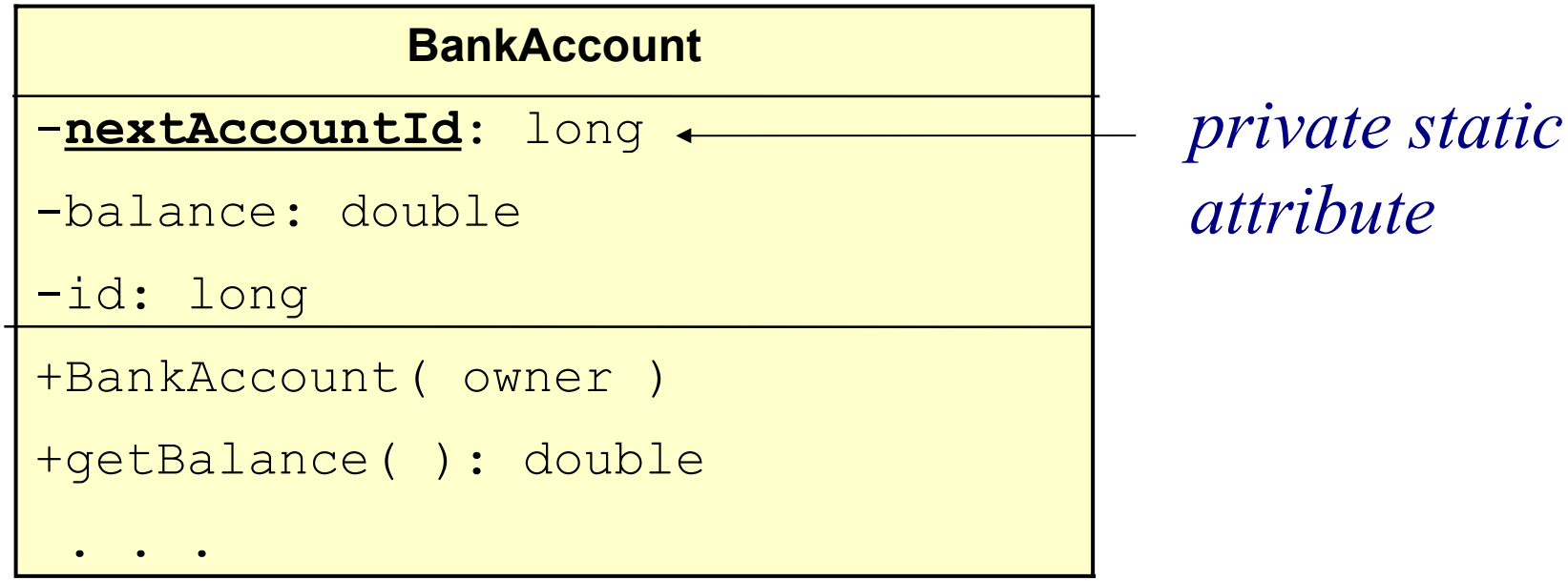
BankAccount
-balance: double
<<constructor>> **+BankAccount(owner)** +deposit(amount) . . .

BankAccount
-balance: double
+BankAccount(owner) +deposit(amount) . . .

Static Members

- Use <u>underscore</u> to show static attributes or methods.

Example: BankAccount class has a static **`nextAccountId`** attribute.



Practice: Draw UML of class

Showing Multiplicity in UML

A Course has zero or more students.

```
public class Course {
  private Student[]
  students;
```

Course
-students: Student[*]

A deck of cards has *exactly* 52 cards.

```
public class CardDeck {
  private Card[] cards =
      new Card[52];
```

CardDeck
-cards: Card[52]

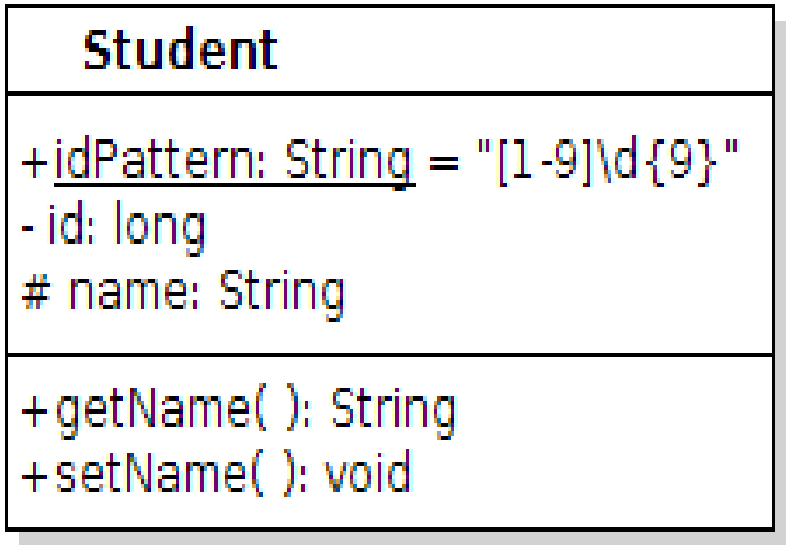
A Single Class

Draw a UML class diagram of this class.

```
public class Student {
    public static String idPattern = "[1-9]\\d{9}";
    private long id;
    protected String name;



    public String getName( ) { . . . }

    public void setName(String aname) { . . . }
```

A Single Class

Draw a UML class diagram of this class.



Reference

UML Distilled, 3rd Edition. Chapter 3 & 5 cover UML class diagrams.