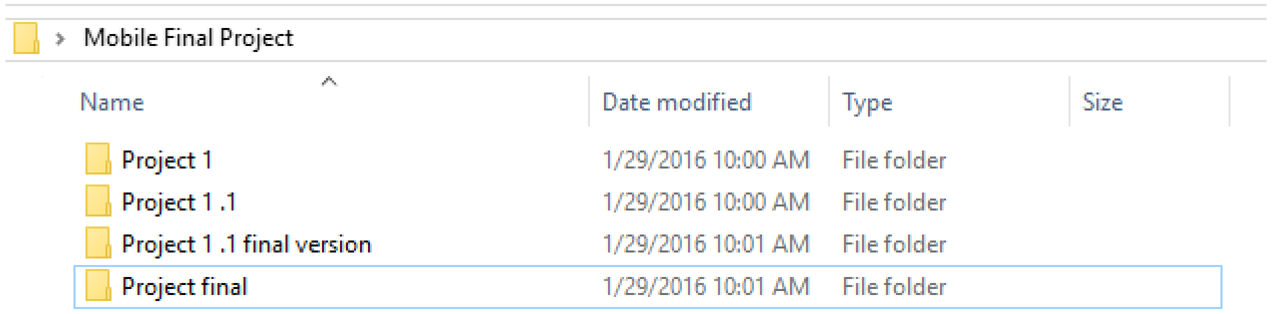
Mobile computation and mobile agents

Juan Camilo Rada

Agenda

- Control version systems
- Object Oriented programing
- Java

How do you store and control your files?



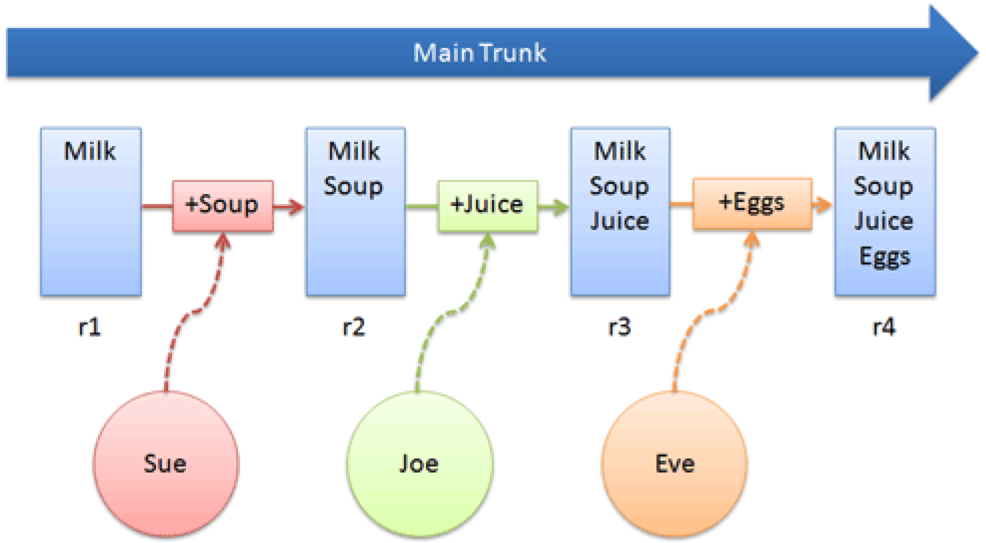
Mobile Final Project

Name	Date modified	Type	Size
Project 1	1/29/2016 10:00 AM	File folder	
Project 1 .1	1/29/2016 10:00 AM	File folder	
Project 1 .1 final version	1/29/2016 10:01 AM	File folder	
Project final	1/29/2016 10:01 AM	File folder	

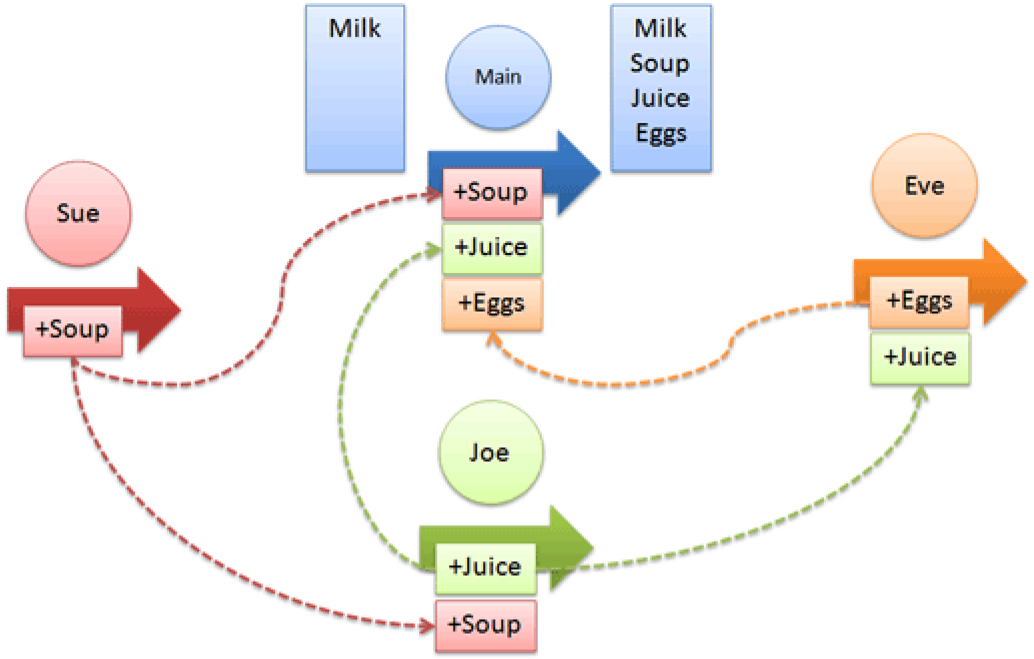
Wikipedia

- https://en.wikipedia.org/wiki/Columbian_mammoth

Centralized VCS



Distributed VCS



Centralized/Distributed

Centralized	Distributed
"first come, first served"	Can make changes to their copy without connecting to any centralized server,
No offline changes support	Offline changes
Single point of failure (Server)	Each dev machine is a repository server
Focus on files	Focus on set of changes
Recording/Downloading and applying a change are single step	Recording/Downloading and applying a change are separated step

Git

“Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.”

Getting started Git

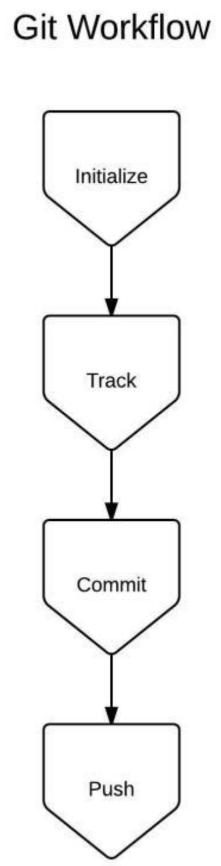
- Download from official web site
- Install
- Configure
- Initialize

Configure Git

```
git config --global user.name “juan.rada”

git config --global user.email juan.rada@javerianacali.edu.co

git config –list

git init
```

Git Workflow

1. **Initialization**
   1. **Track**
   2. **Stage**
   3. **Commit**
   4. **Push**

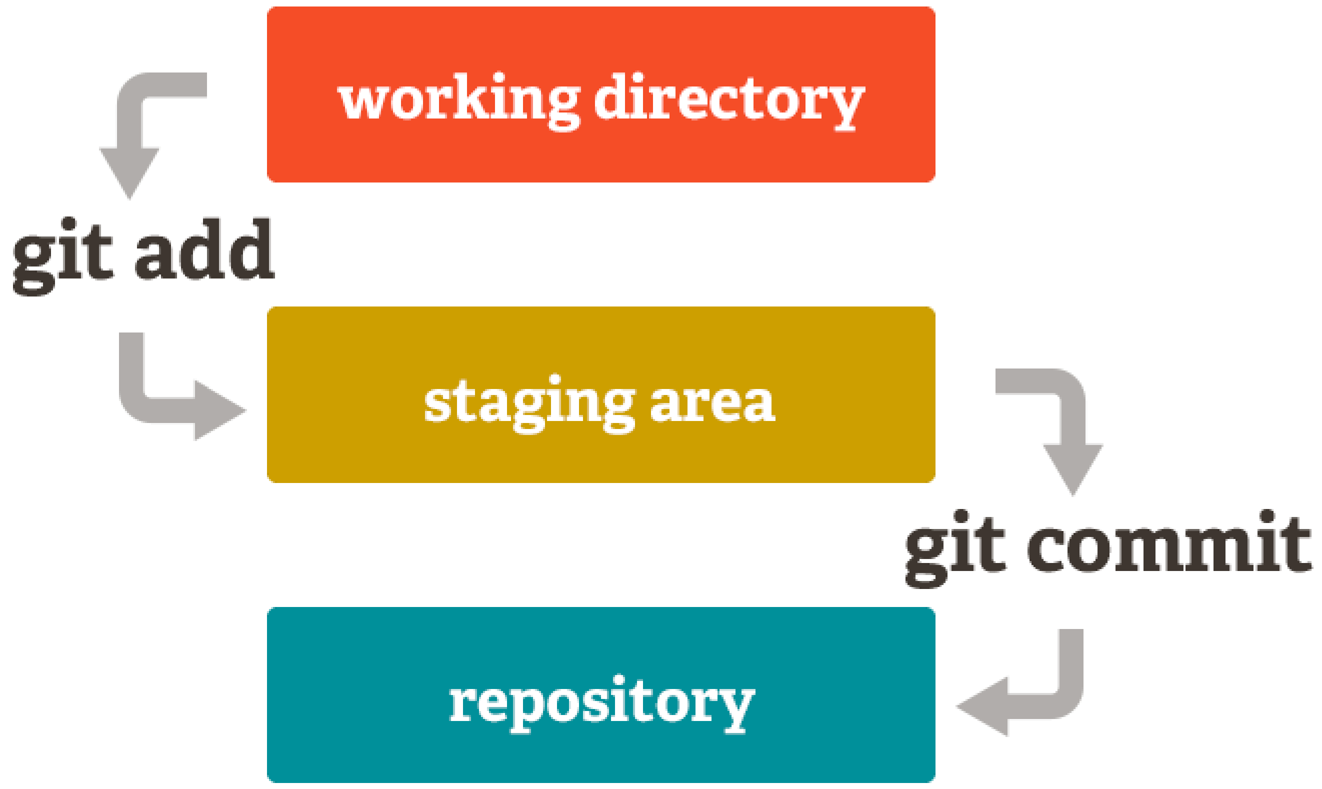


Track

- git status
- git add ***file/folder***
- git rm --cached ***file/folder***

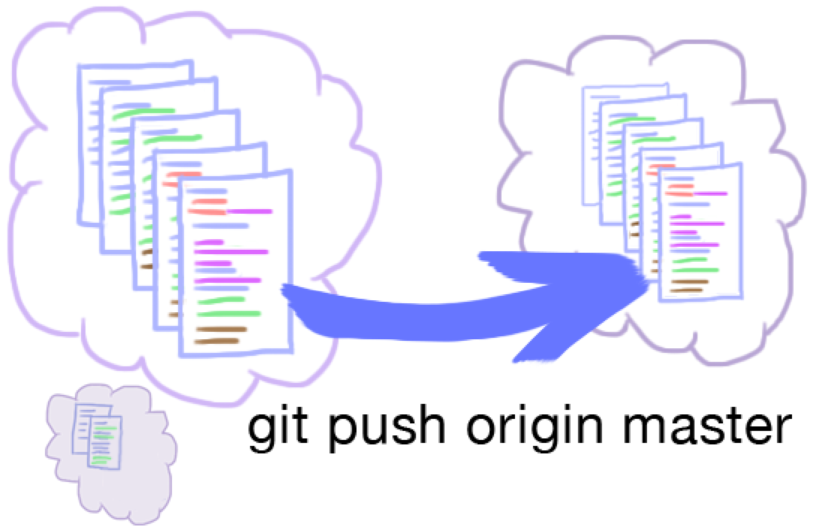
Commit

- git commit –m "Commit message"



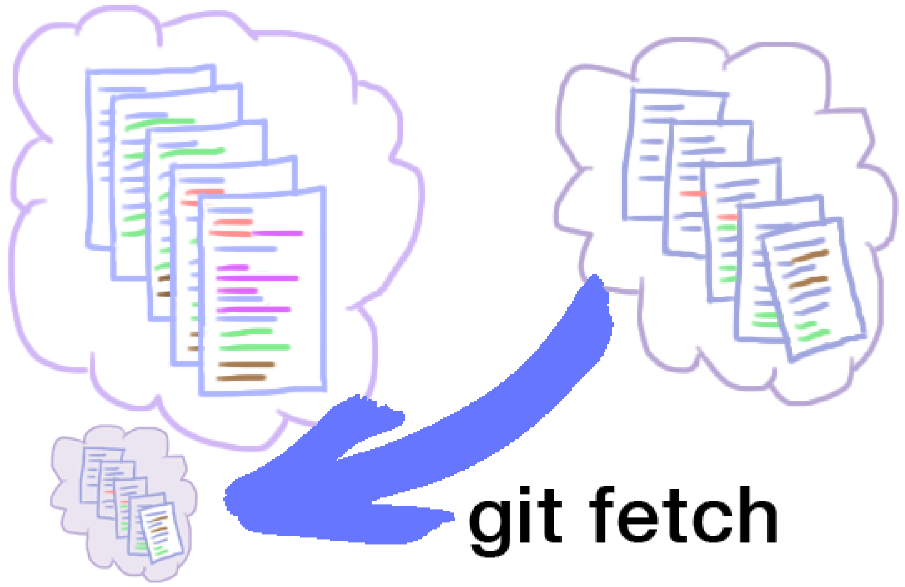
Push

- git remote add origin https://github.com/my_repository.git
- git push origin master



Getting changes

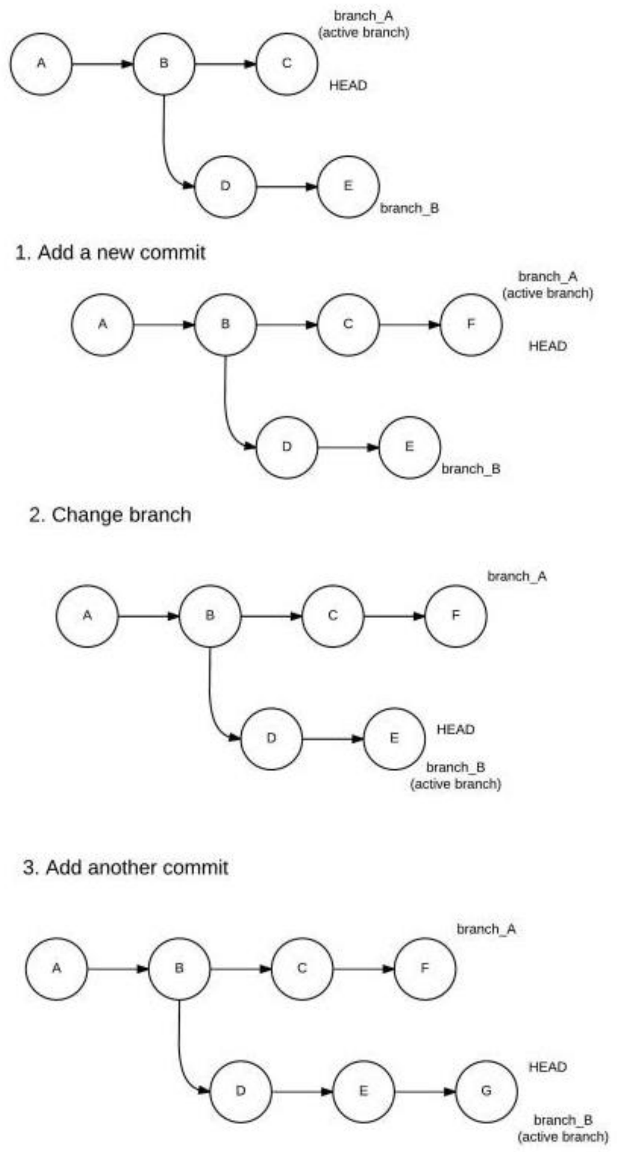
- git pull (fetch and merge)
- git fetch (fetch)



Branches

Branches

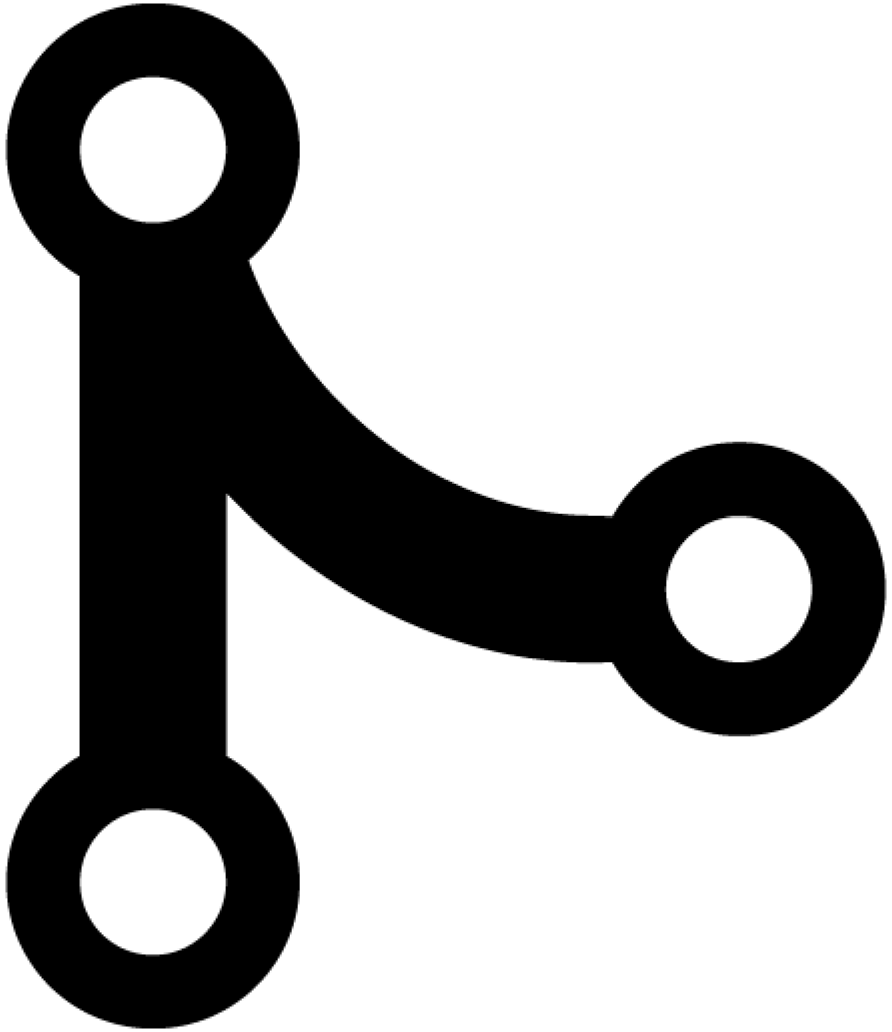
- git branch my_branch
- git checkout my_branch
- git checkout -b my_branch



Merge

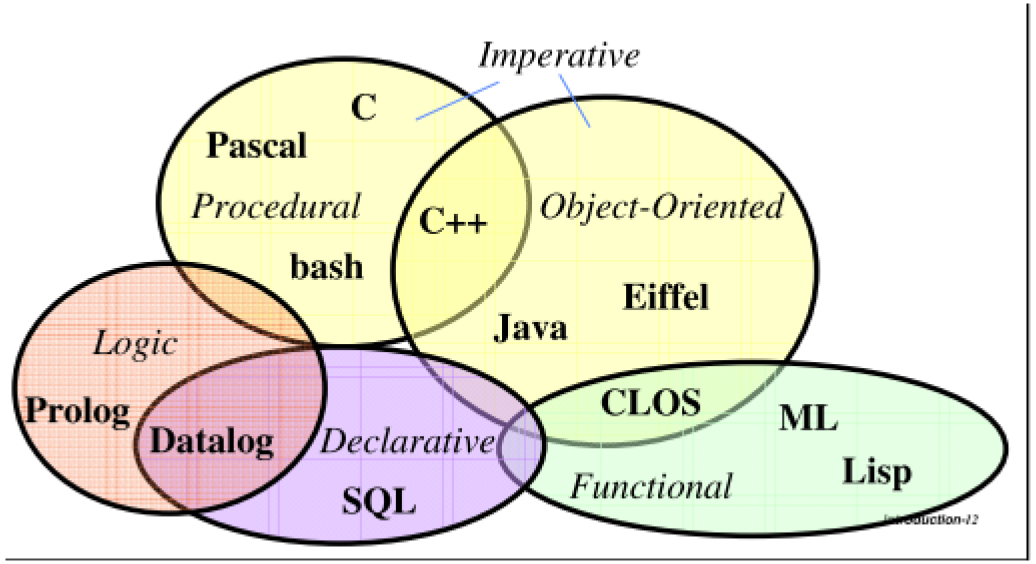
- git merge topic

```
      A---B---C topic
     /
D---E---F---G master

      A---B---C topic
     /         \
D---E---F---G---H master
```

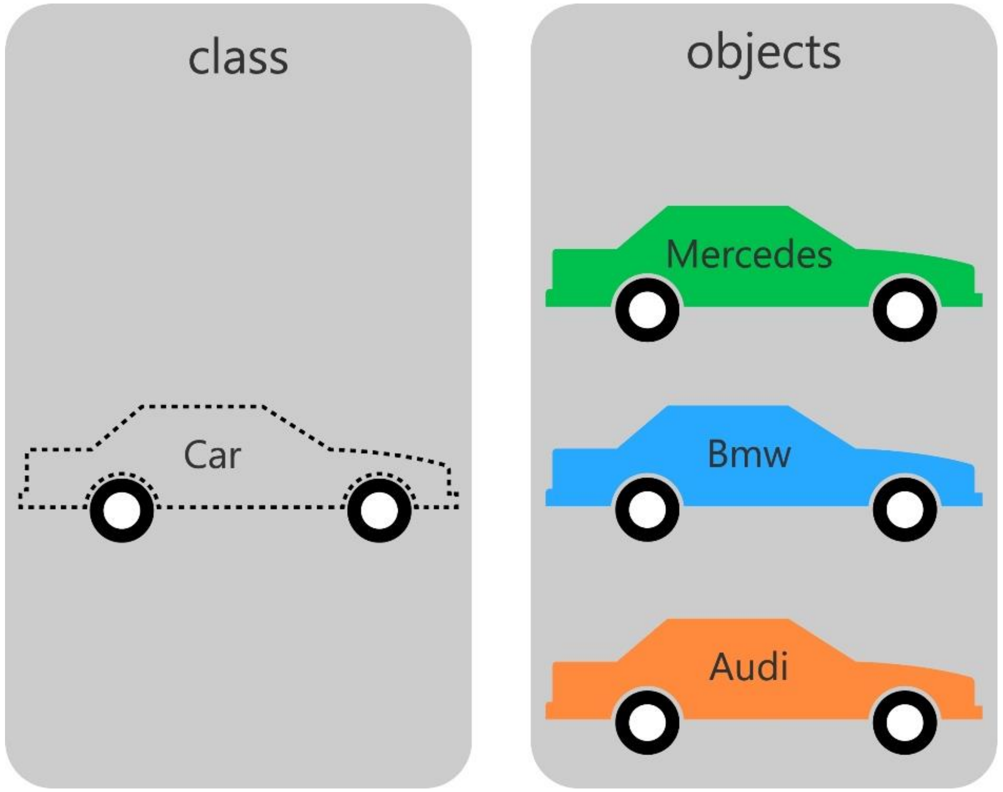


Questions?

OBJECT-ORIENTED PROGRAMMING



OBJECTS AND CLASSES



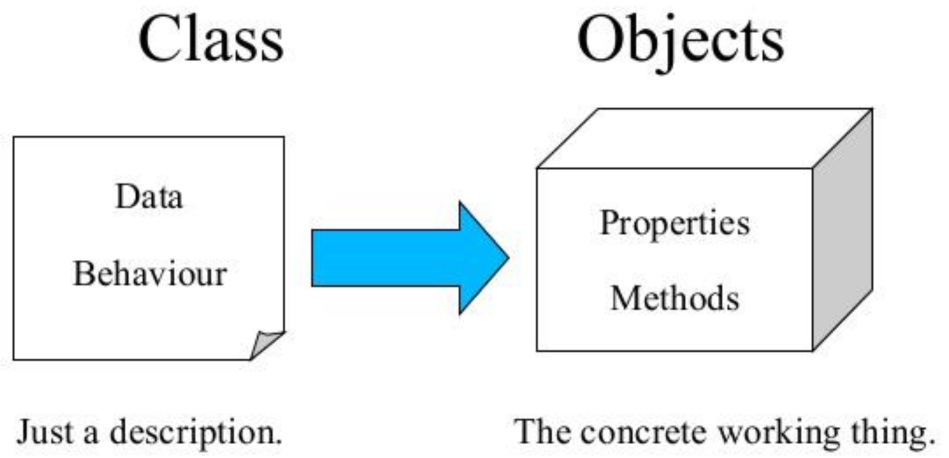
BEHAVIOR

```
public void increaseSpeed(int speedIncrease)
  {

  }  {

  }
```

ATTRIBUTES

```
private String name();
```

OBJECTS AND CLASSES



OBJECTS AND CLASSES

- Create the class ***DigitalWatch*** with chronometer functionality.
  - Hours.
  - Minutes.
  - Date.
  - Start Chronometer.
  - Clear Chronometer.
  - Get Chronometer elapsed time.

Encapsulation

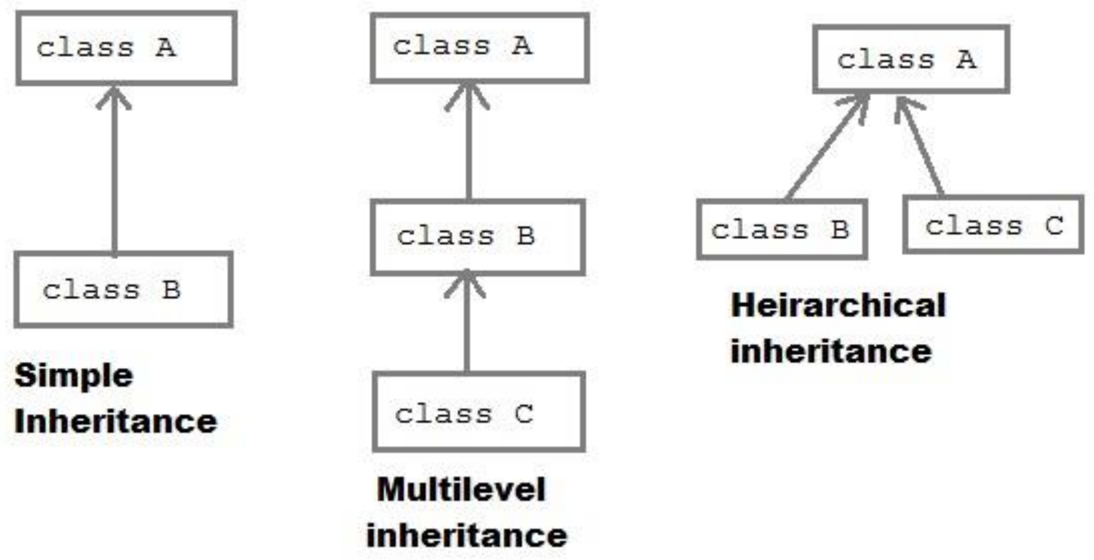
Access Modifiers

- Private
- Public
- Default (Package private)
- Protected

Non Access Modifiers

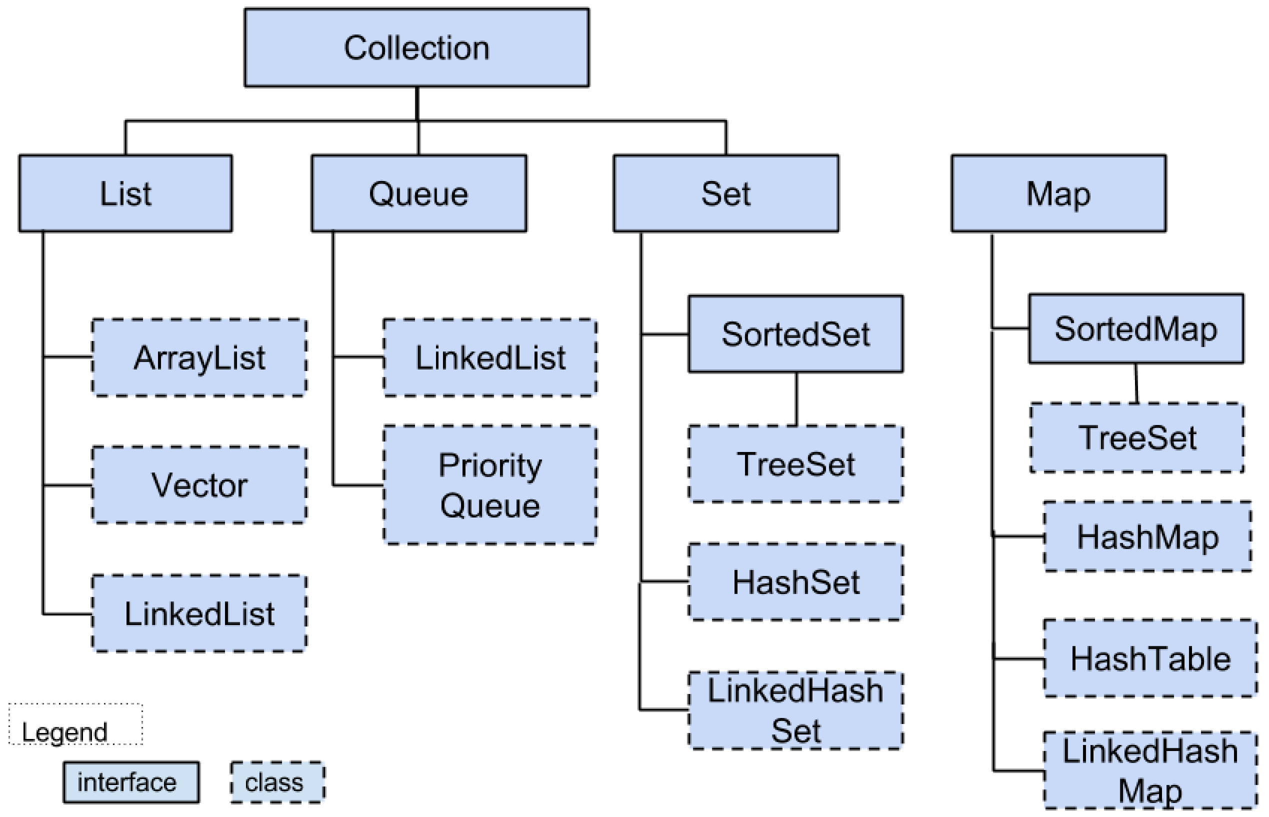
- Static
- Final
- Abstract

Inheritance (EXTENDS)



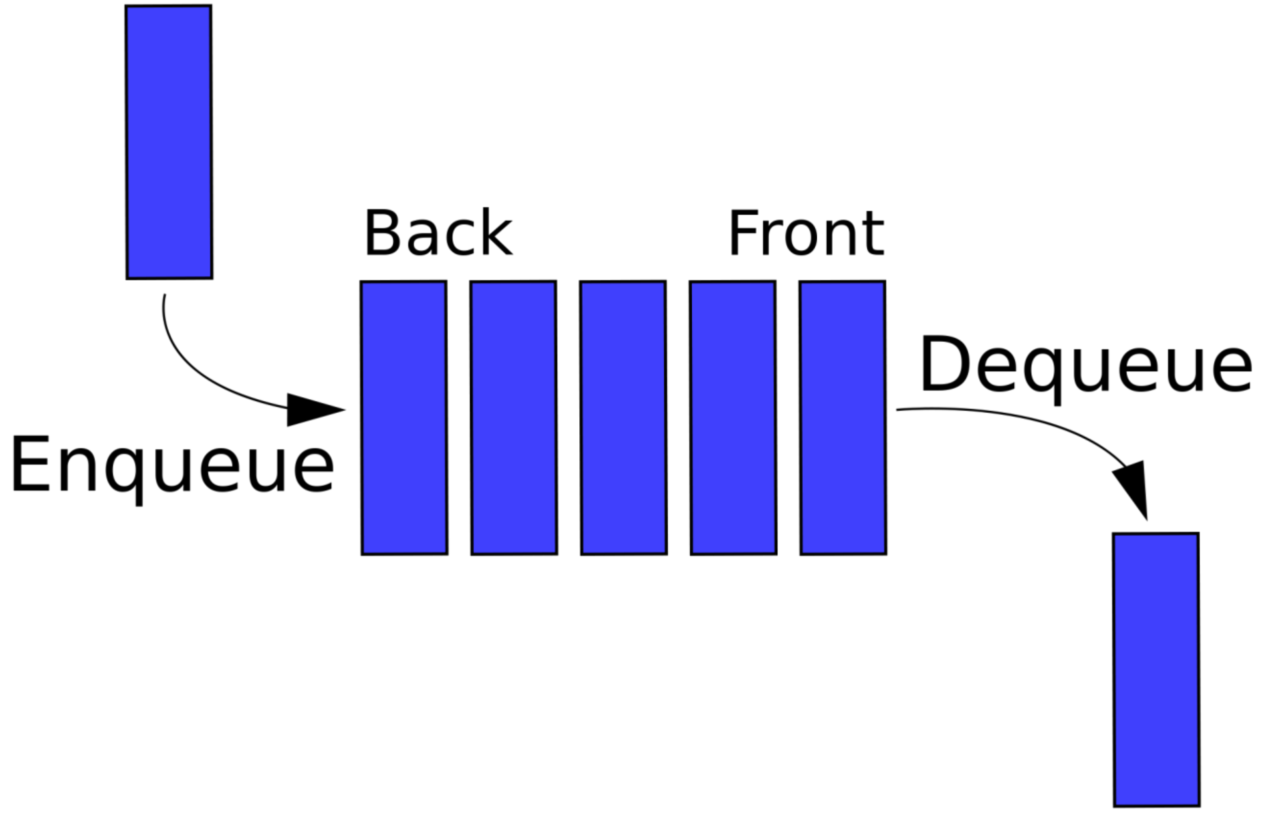
Interfaces (implements)

Data Structures

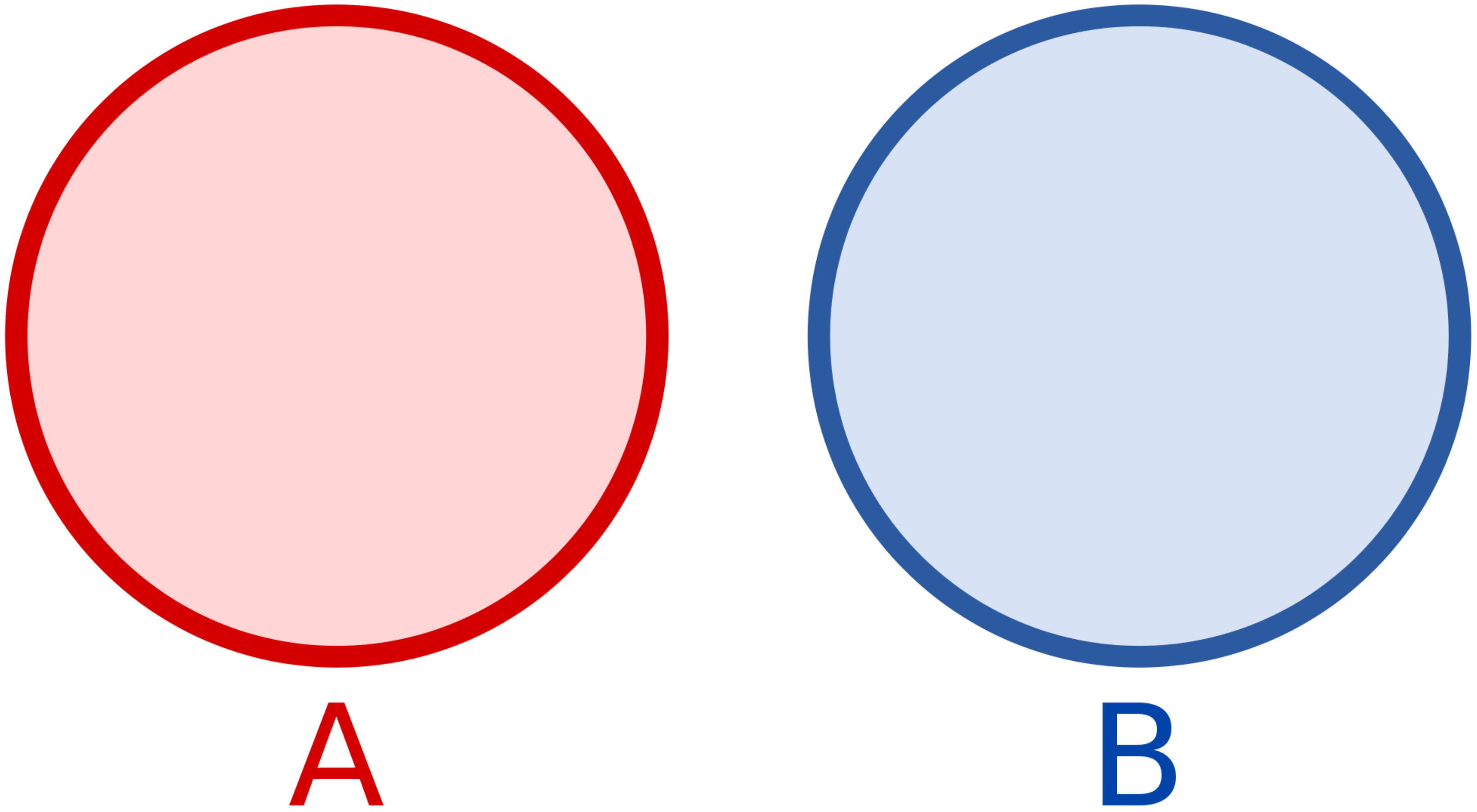


List

Queue (FIFO)



Set



Map

Key	Value
Chronicle of a Death Foretold	Book@asd123
Saving Private Ryan	Book@nas674
The metamorphosis	Book@nas456

Assignment