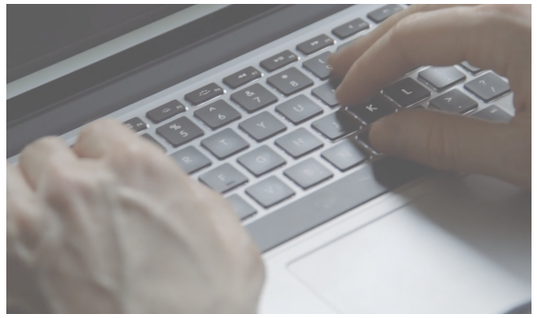
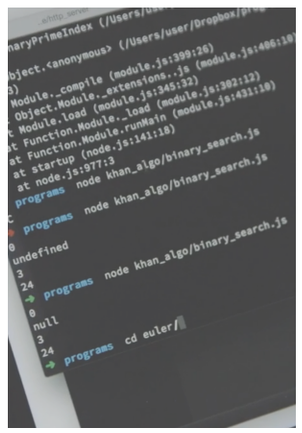

Planning a
Software
Development
Project





Learning Objectives

Aim: To help in the preparation and planning of a software development project.

- Know a range of techniques that help plan a project.
- Understand the importance of good planning.
- Be able to produce an effective plan to assist in the development of a software development product.

The PDA is equivalent to

Work involved is very simple. It is the work you do everyday. At certain times I will share with you the work that needs to be submitted and by what week.

Where to start?

Build a hub for modern interpretations of classics, e.g. noughts and crosses, tic tac toe. Your app will allow users to register an account to track their interactions. Allow them to start new games which they can play against a friend or the computer. They will be able to see a list of the previous games they have played, and a running score of wins/losses. Maybe even a leaderboard of all players across the site, or the facility to start knockout tournaments.

“Gather and prioritise requirements”

- Build a hub for modern interpretations of classics, e.g. noughts and crosses, tic tac toe.
- Your app will allow users to register an account to track their interactions.
- Allow them to start new games which they can play against a friend or the computer.
- They will be able to see a list of the previous games they have played, and a running score of wins/losses.
- Maybe even a leaderboard of all players across the site, or the facility to start knockout tournaments.

- Users
  - register account
  - track interactions
- Game
  - start new game
  - list of previous games
  - running score of losses/wins
- Extra functionality
  - Leaderboard - all players
  - Knockout tournaments

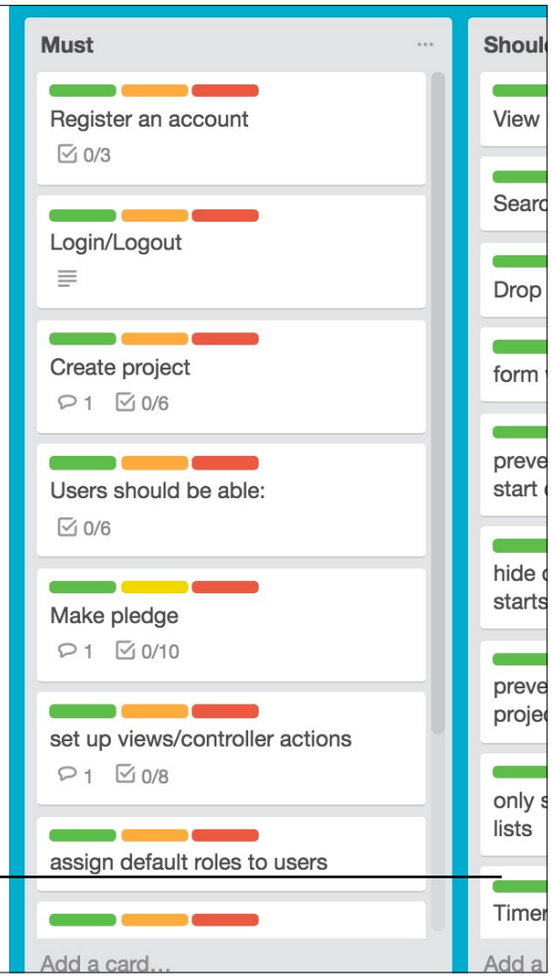
MVP, MoSCoW & Trello

A Minimum Viable Product (MVP) is a product with enough features to satisfy early customers and provide feedback for future development.

Must Should Could Won't have

Record information: Labels, checklists, record diagrams, multi-user access

https://trello.com/



Illustrate the method with Trello board: https://trello.com/b/PvqPDnJv/moscow-illustration

Take a break

Work on intranet stuff

Diagrams — why?

Easy planning, feature scoping and constraint analysis

Quick overview of system architecture

Universal language (UML) and portability

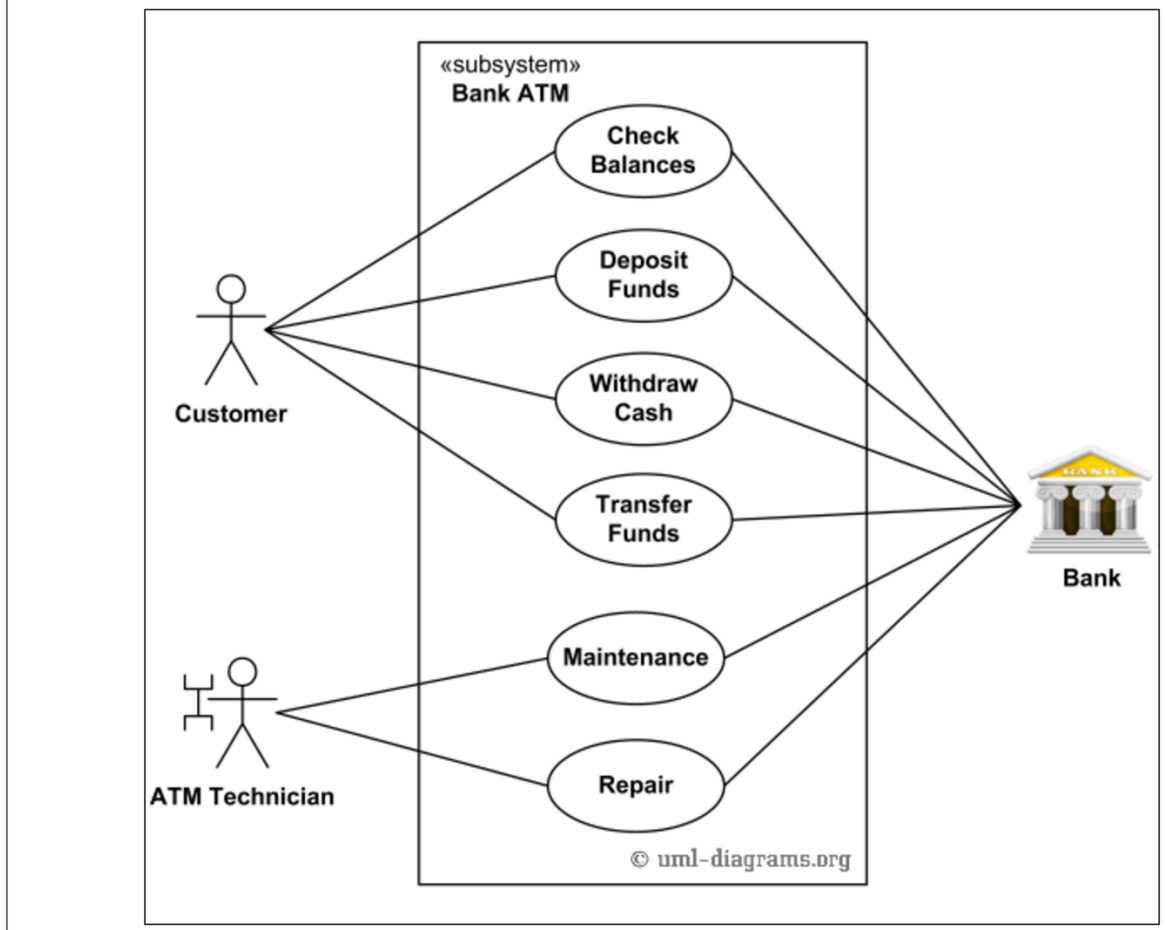
A note on why diagrams are good.

Use Case Diagrams

Use case diagrams give an overview of the usage requirements for a system.

Use case diagrams depict:

- Use cases (sequence of actions)
- Actors
- Associations
- System boundary boxes (optional)



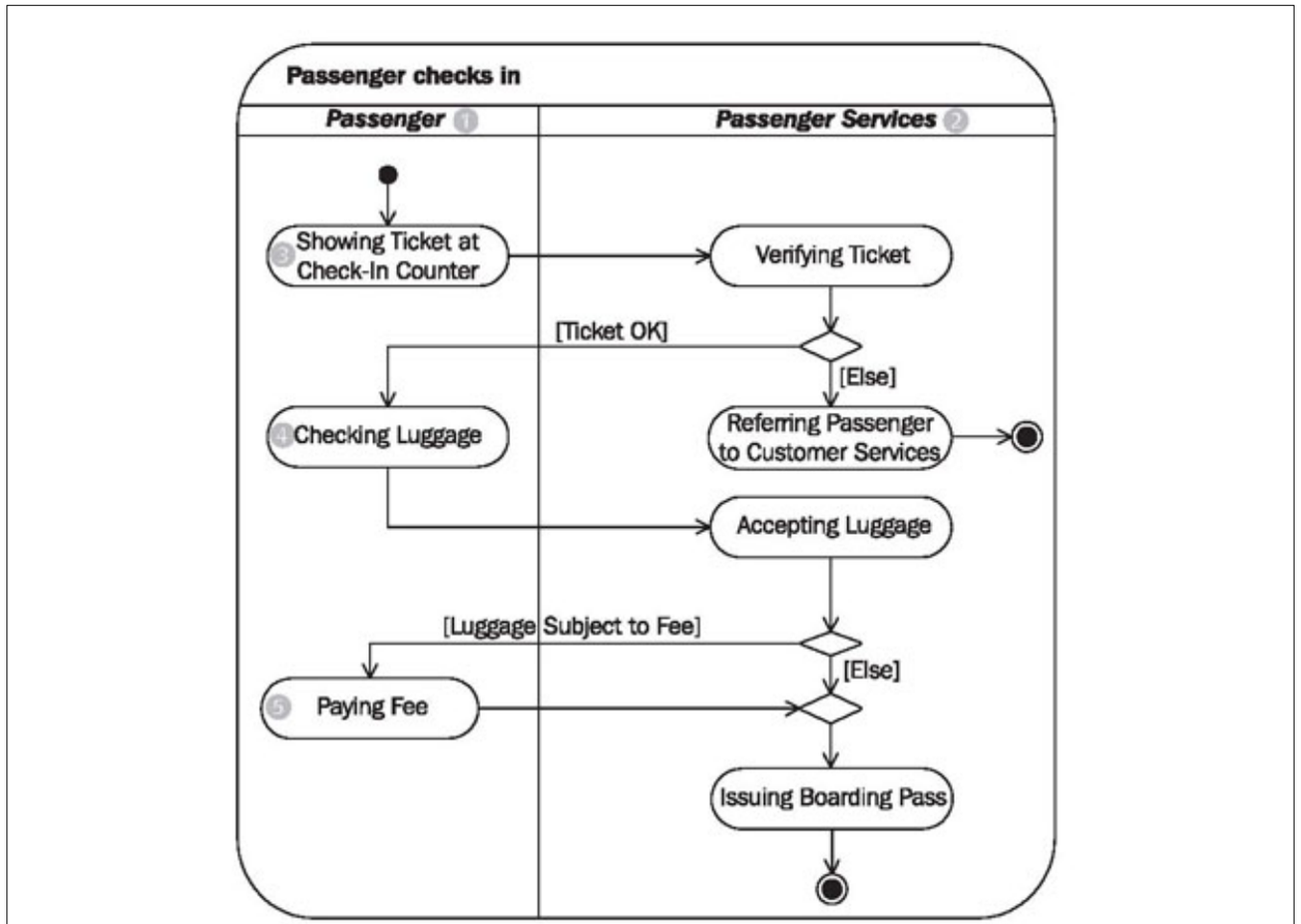
An example of use case diagram

Activity Diagrams

An Activity Diagram is a flow chart to represent the flow from one activity to another activity.

Activity diagrams should include:

- lines with arrows
- rounded rectangles (actions)
- diamonds (decisions)
- bars (start (split) or end (join) of concurrent activities)
- black circle (start of the workflow)
- encircled black circle (the end)



This is an activity diagram.

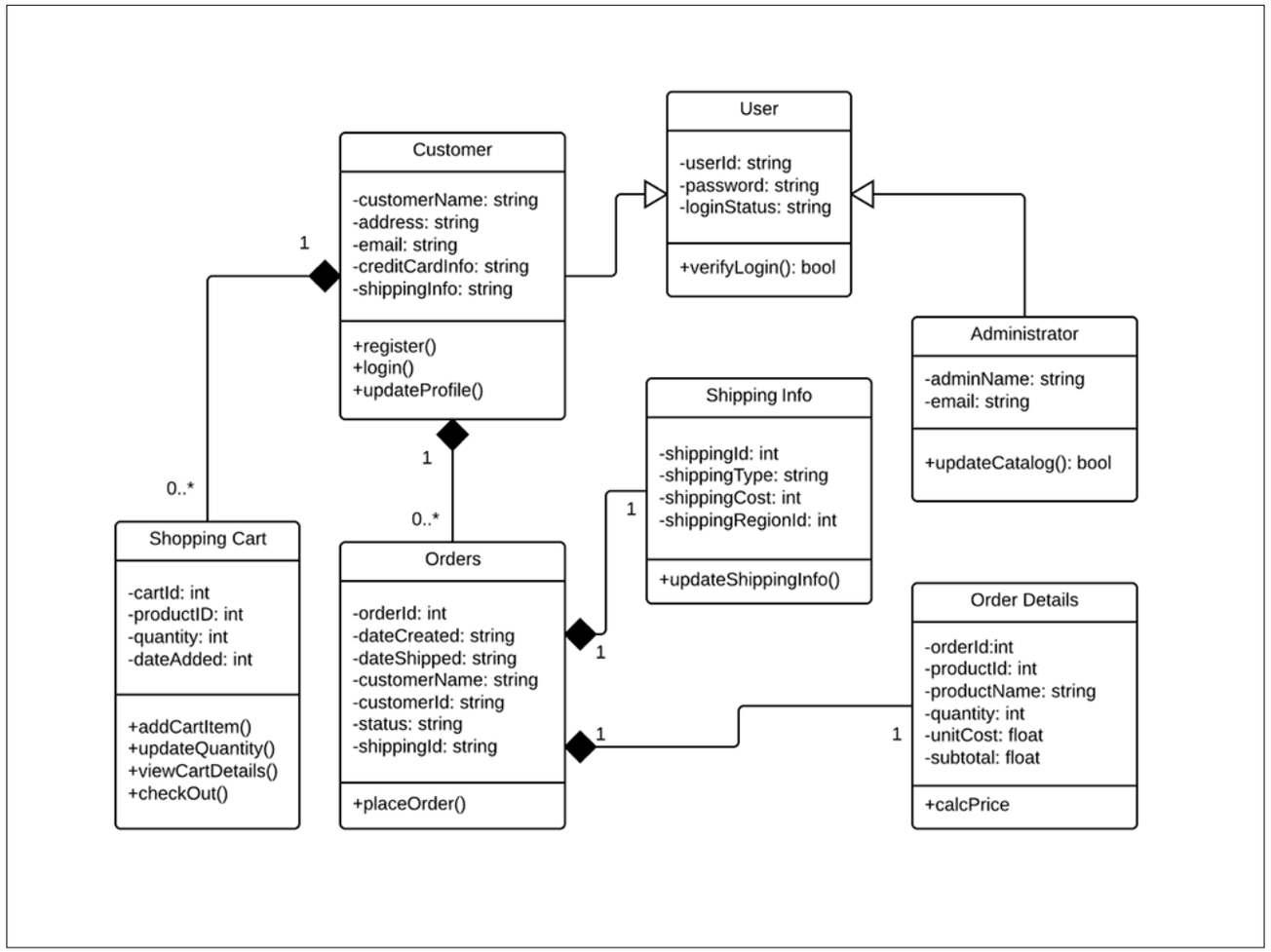
Class Diagrams

Class diagrams describe the structure of a system by showing the system's classes, their attributes, methods and the relationships.

Class diagrams should include:

- Name of each class
- Attributes of each class
- Type of each attribute
- Methods
- Relationships

You can learn more about class diagrams: http://www.agilemodeling.com/artifacts/classDiagram.htm (and many other sources)



An example of a class diagram

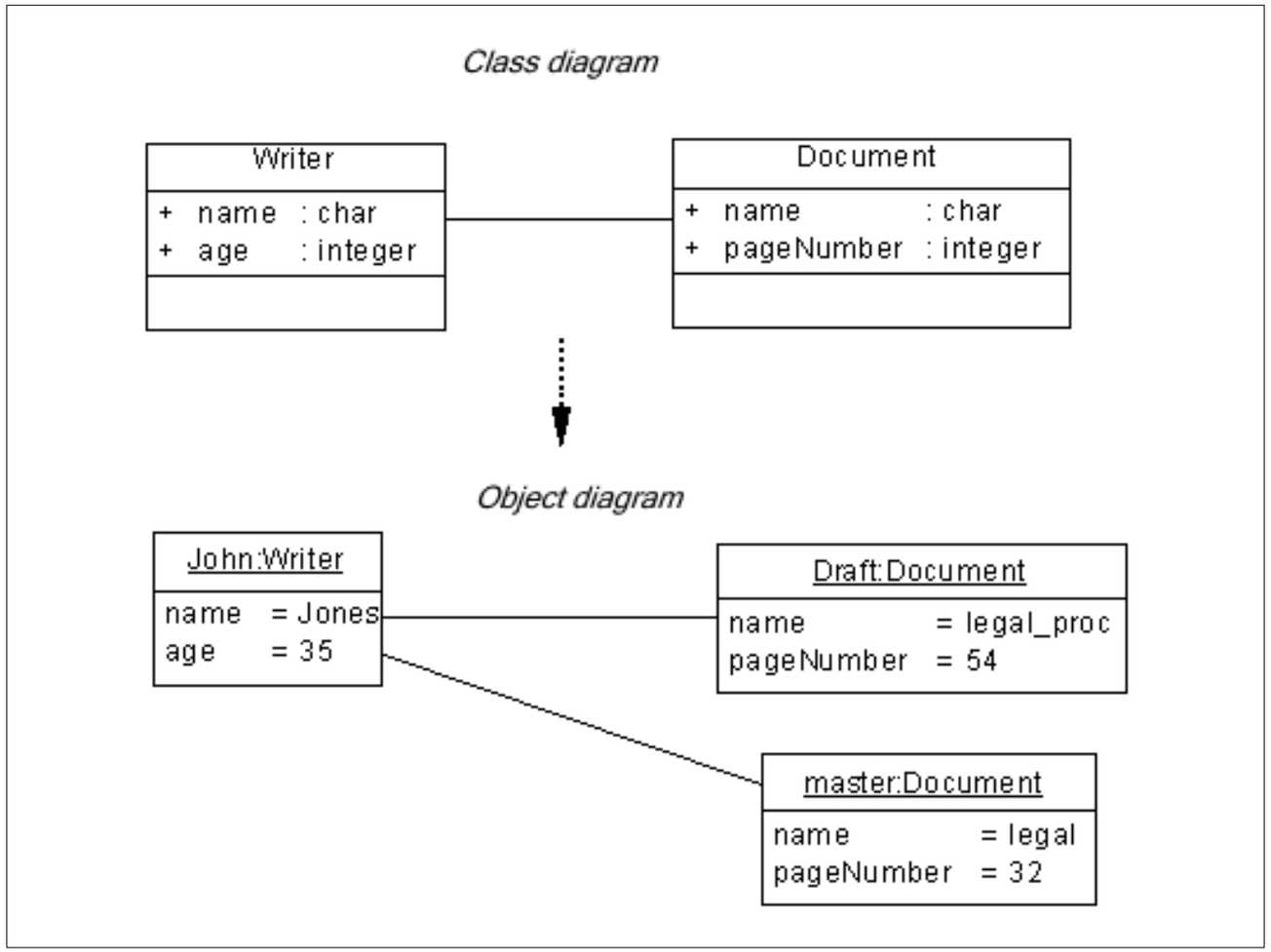
Numbers - multiplicity of classes. Indicates the number of instances. 0..* - zero or more, 1 - exactly one, etc.

Object Diagrams

Object diagrams provide examples or act as test cases for class diagrams.

Object diagrams should include:

- Name of the class
- Instance of that class
- Attributes of each class
- Type of each attribute replaced with an example
- Relationships between the classes



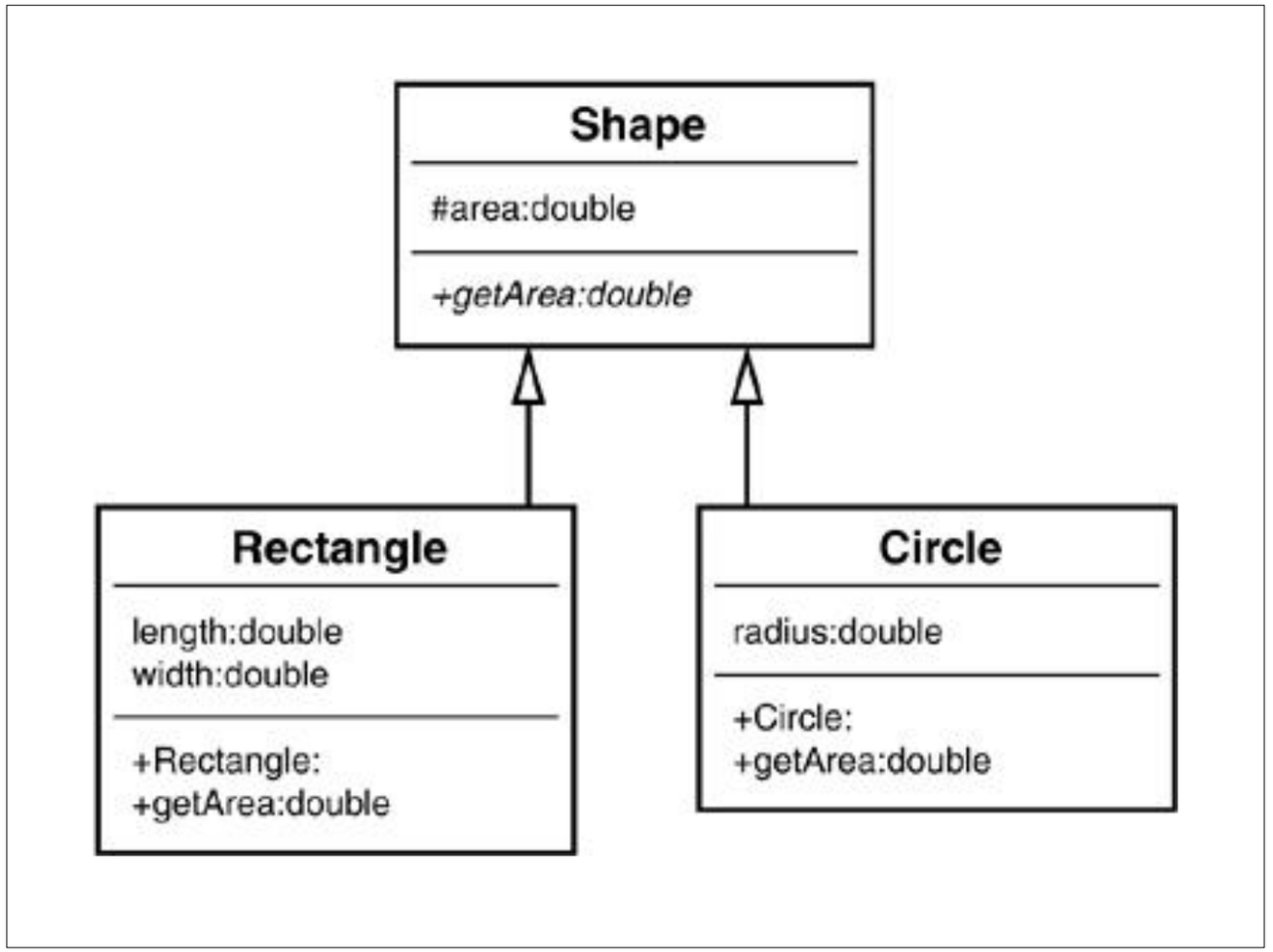
This is an example of an object diagram — as you can see, there are three objects, instances of two classes.

Inheritance Diagrams

An inheritance diagram demonstrates when a child object assumes characteristics of its parent object.

Inheritance diagrams should include:

- Relationships between the classes (using arrows)
- Focus is on the flow of information between classes
- Name of the class
- Attributes of each class
- Type of each attribute


Shape
#area:double
+getArea:double
Rectangle
length:double
width:double
+Rectangle:
+getArea:double
Circle
radius:double
+Circle:
+getArea:double

Implementation Constraints Plan

Implementation constraints are things that might **constrain** the project and stop it from reaching its full potential.

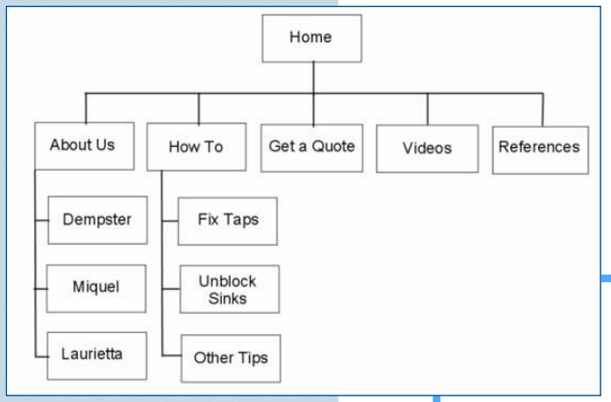
Topic	Possible Effect of Constraint on Product	Solution
Hardware and software platforms	**What could be a constraint on the product. How it could be a constraint of the product Why is it a problem?**	How the constraint will be avoided or how it is not a consideration.
Performance requirements		
Persistant storage and transactions		
Usability		
Budgets		
Time limitations		

Pseudocode

Plan functionality before you even write a line of code. It also helps you to design your unit testing.

```
it('should damage the hero with an attack', function(){
  // for each attack it should use a weapon, villain and hero
  // an example of a weapon to test
  // an example of a villain to test
  // an example of a hero to test
  // the villain should attack the selected hero
  // the canBlock function decides the level of damage inflicted on the hero by the villain
  // the result of the attack should the reduction of the hero's health
}
```

Sitemaps



A sitemaps is a list of pages of a web site accessible to users.

It can be either a diagram used as a planning tool for Web design, or a Web page that lists the pages on a website, typically organised in a hierarchical fashion.

Plan within your plan

Find what works for you.

20 min cycles + 10 mins problem-solving

Division of labour & time management

Planning your project

To help you plan your project:

Project requirements

Trello and MoSCoW

Day to day planning and Time Management

For your PDA:

Object Diagrams

Class Diagrams

Inheritance Diagrams

Use Case Diagrams

Implementation Constraints Plan