

Software Engineering

Functions

- Project?

- how's it going? Any Questions?

Admin

- Midterm discussion
 - Quizzes coming
- Wednesday classes don't meet on Feb 21
 - Monday classes will meet instead.
 - Actually not many classes before midterm.

Function/Method Length

- Many have heard me opine on this before
 - How long should a function/method be?

Function/Method Length

- Many have heard me opine on this before
 - How long should a function/method be?
 - My rule of thumb was half a screen full
 - Of course that was with a reasonable font
 - 20-30 lines at most
 - Someone reading your method/function should be able to easily tell what your method is doing
 - Remember Miller's magic number 7.
- Your book goes further
 - Functions should be <20 lines with no more than 2 levels of indent.

Function Length

- Of course lines need to be reasonable too
- Bad single line:
 - `return level4 != null ? GetResources().Where(r => (r.Level2 == (int)level2) && (r.Level3 == (int)level3) && (r.Level4 == (int)level4)).ToList() : level3 != null ? GetResources().Where(r => (r.Level2 == (int)level2) && (r.Level3 == (int)level3)).ToList() : level2 != null ? GetResources().Where(r => (r.Level2 == (int)level2)).ToList() : GetAllResourceList();`
 - Thanks to stackoverflow user 26507 for this example

Functions

- A function or a method
 - Should do how many things?

Functions

- A function or a method
 - Should do how many things?
 - One of course
 - Do they?

How about this method?

- `void printOwing() {`
- `Enumeration e = _orders.elements();`
- `double outstanding = 0.0;`
- `// print banner`
- `System.out.println ("*****");`
- `System.out.println ("***** Customer Owes *****");`
- `System.out.println ("*****");`
- `// calculate outstanding`
- `while (e.hasMoreElements()) {`
- `Order each = (Order) e.nextElement();`
- `outstanding += each.getAmount(); }`
- `//print details`
- `System.out.println ("name:" + _name);`
- `System.out.println ("amount" + outstanding); } cite: sourcemaking.com`

Ordering methods in a class

- How should you order functions/methods in a class/module?

Ordering methods in a class

- How should you order functions/methods in a class/module?
- In the old days
 - Alphabetically was popular
 - You would be able to find the method you were looking for
 - Today not needed

Ordering methods in a class

- How should you order functions/methods in a class/module?
 - Book suggests top down
 - High levels of abstraction at top of file/class
 - Then next level of abstraction
 - Till finally the functions that deal with the nitty gritty are at the bottom.

Arguments/parameters

- What have you heard about parameters?
 - How many should you have?

Arguments/parameters

- What have you heard about parameters?
 - How many should you have?
 - As few as possible
- Parameters should be input only
 - With OOP this is normal anyway
 - Working with older code is where you are likely to have issues
- Flag arguments
 - Maybe you should have two methods?

Arguments/Parameters

- When more than one param
 - Make it clear from method name which param does what
 - Consider creating an object for wrapping multiple related data items
 - Though modern ides make this easier.

Side Effects

- What do we mean by side effects in a method/function?

Side Effects

- What do we mean by side effects in a method/function?
 - Old style definition: anything that isn't a result calculated by a function
 - Pop up a window.
 - More current:
 - Anything that isn't clear from method definition that method is doing.
 - Lets look at listing 3-6 on page 44 of book.

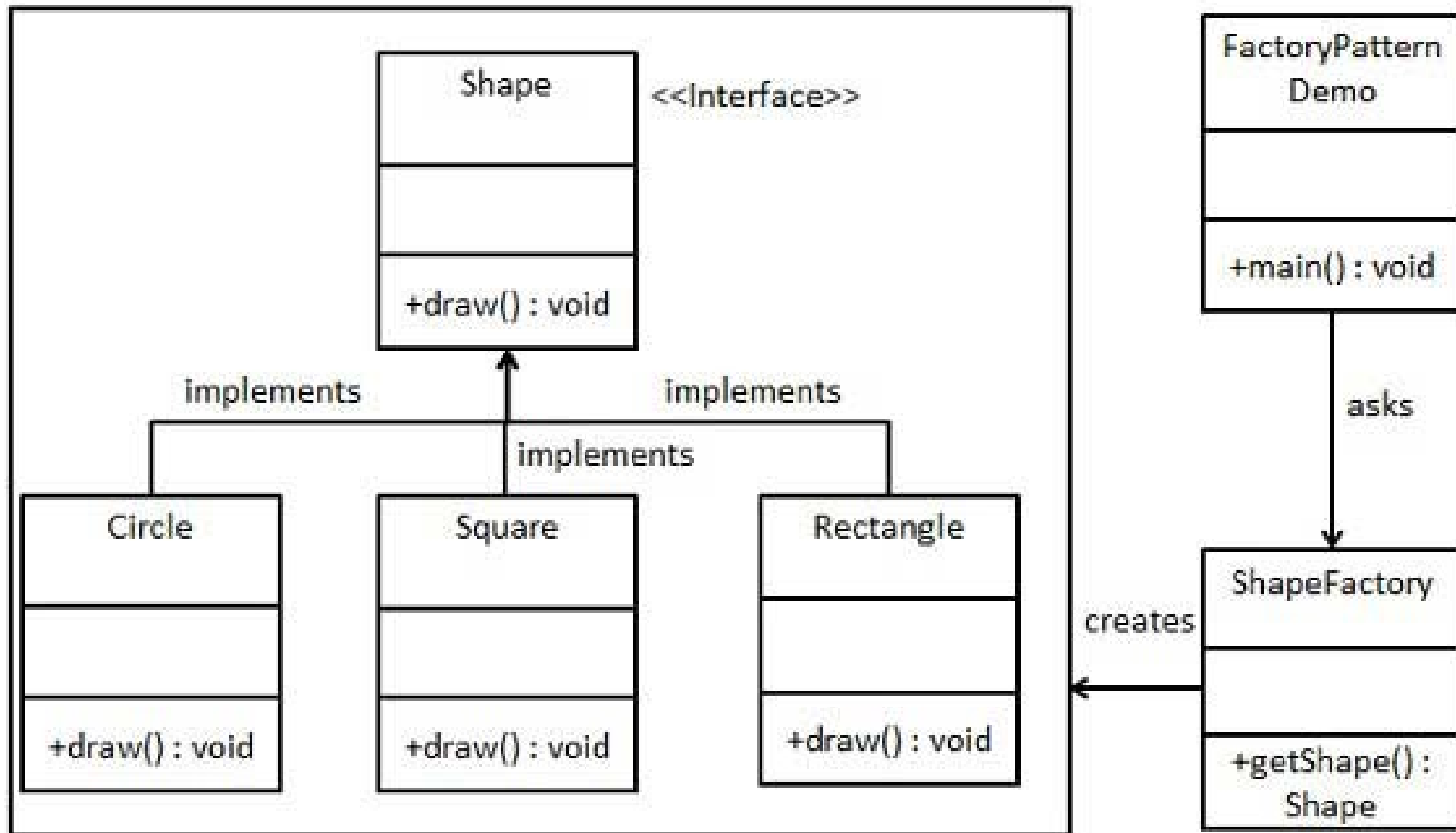
Exceptions

- An OOP standard:
 - Prefer exceptions to error codes
 - Using C? Too bad
 - Using a language from the last 30 years?
 - Use exceptions and not error codes.
- Also consider:
 - Each method should do one thing:
 - Error handling is one thing. One method for error handling, the other for doing something that might cause an error.

Factory pattern

- Do on board with employee and subclasses
 - Salaried
 - Commissioned
 - Hourly
 - Factory interface and class

UML for factory pattern



The code and UML examples come from
https://www.tutorialspoint.com/design_pattern/factory_pattern.htm

The code (in java) – part 1

- public interface Shape {
- void draw();
- }
- public class Rectangle implements Shape {
- @Override
- public void draw() {
- System.out.println("Inside Rectangle::draw() method.");
- }}
- public class Square implements Shape {
- @Override
- public void draw() {
- System.out.println("Inside Square::draw() method.");
- }}

The code (in java) – part 1

- public class Circle implements Shape {
- @Override
- public void draw() {
- System.out.println("Inside Circle::draw()
method.");
- }
- }

- public class ShapeFactory {
- //use getShape method to get object of type shape
- //sometimes this is a static method
- public Shape getShape(String shapeType){
- if(shapeType == null){
- return null;}
- if(shapeType.equalsIgnoreCase("CIRCLE")){
- return new Circle();
- } else if(shapeType.equalsIgnoreCase("RECTANGLE")){
- return new Rectangle();
- } else if(shapeType.equalsIgnoreCase("SQUARE")){
- return new Square();
- }
- return null;
- }}

- public class FactoryPatternDemo {
- public static void main(String[] args) {
- ShapeFactory shapeFactory = new ShapeFactory();
- //get an object of Circle and call its draw method.
- Shape shape1 = shapeFactory.getShape("CIRCLE");
- //call draw method of Circle
- shape1.draw();
- //get an object of Rectangle and call its draw method.
- Shape shape2 = shapeFactory.getShape("RECTANGLE");
- //call draw method of Rectangle
- shape2.draw();
- //get an object of Square and call its draw method.
- Shape shape3 = shapeFactory.getShape("SQUARE");
- //call draw method of circle
- shape3.draw();
- }

Reading

- Read chapter 3 in clean code