OOP Design

Design

- So far, we’ve been covering the *mechanics* of how to work with OOP
- Things like, How do I define a class? How do I define a method? How do I make one class extend from another class? How do I make objects from these classes?
- As important as the mechanics are, *How do I define good object-oriented programs?* is every bit as important
  - That is, how do you identify the right things that should be classes; how do you model the relationships between them, and how do you decide what each class should do
- Good OOP design makes the system much easier to extend and maintain; bad design will bewilder other developers

Design: Identifying Classes

- Typically, you don’t just create a design out of thin air
- Instead, you typically start with some *problem description*
- The problem description describes what your code is supposed to do; what real-world situation your code is going to model, and how that situation is supposed to behave
- The starting point in an OO design is usually to identify “What things from this problem need to be classes?”
- So, how do you do that?

Design: Identifying Classes

- *Nouns* in the problem description you’re trying to model are good candidates for *classes* in your implementation
- Let’s take a look at how this might work with an example

Design: Public Works

- Suppose you’ve been hired by the City of Baltimore to handle designing and building a piece of software to track their Public Works department
    - Public works basically being city infrastructure
- *The Baltimore Public Works (“BPW”) needs to keep track of all of the infrastructure that is under their supervision. Common pieces of infrastructure include roads (and their associated accessories, such as traffic lights, stop signs, guard rails, etc), water treatment (both fresh water purification & wastewater) plants, the Baltimore AMTRAK station, parks, the electrical grid, and the Baltimore Schools Department. Each piece of infrastructure needs to track its name, type, location, annual budget, and a list of high-priority and low-priority improvements for it. It should be possible to mark an improvement as completed when it is done; completed improvements should still be tracked. BPW also needs a way to track employees. Each employee should have their name, ID number, and specialty (f.ex Civil Engineer, Electrician). BPW should have a way to let the public request a new piece of infrastructure, and to request maintenance on an existing piece of infrastructure. It should be possible to find all BPW infrastructure that needs high-priority improvements, as well as all employees currently assigned to (or currently not assigned to) a job.*
- That’s a lot to do! How do we start?

Public Works: Class Identification

Definitely Classes

The Baltimore Public Works ("BPW") needs to keep track of all of the infrastructure that is under their supervision. Common pieces of infrastructure include roads (and their associated accessories, such as traffic lights, stop signs, guard rails, etc), water treatment (both fresh water purification & wastewater) plants, the Baltimore AMTRAK station, parks, the electrical grid, and the Baltimore Schools Department. Each piece of infrastructure needs to track its name, type, location, annual budget, and a list of high-priority and low-priority improvements for it. It should be possible to mark an improvement as completed when it is done; completed improvements should still be tracked. Each improvement should have an estimated cost, a list of expected benefits, and the employee or contractor who will complete it. BPW also needs a way to track employees. Each employee should have their name, ID number, and specialty (f.ex Civil Engineer, Electrician). BPW should have a way to let the public request a new piece of infrastructure, and to request maintenance on an existing piece of infrastructure. It should be possible to find all BPW infrastructure that needs high-priority improvements, as well as all employees currently assigned to (or currently not assigned to) a job.

Public Works: Class Identification

Definitely Classes

Maybe Classes

The Baltimore Public Works ("BPW") needs to keep track of all of the infrastructure that is under their supervision. Common pieces of infrastructure include roads (and their associated accessories, such as traffic lights, stop signs, guard rails, etc), water treatment (both fresh water purification & wastewater) plants, the Baltimore AMTRAK station, parks, the electrical grid, and the Baltimore Schools Department. Each piece of infrastructure needs to track its name, type, location, annual budget, and a list of high-priority and low-priority improvements for it. It should be possible to mark an improvement as completed when it is done; completed improvements should still be tracked. Each improvement should have an estimated cost, a list of expected benefits, and the employee or contractor who will complete it. BPW also needs a way to track employees. Each employee should have their name, ID number, and specialty (f.ex Civil Engineer, Electrician). BPW should have a way to let the public request a new piece of infrastructure, and to request maintenance on an existing piece of infrastructure. It should be possible to find all BPW infrastructure that needs high-priority improvements, as well as all employees currently assigned to (or currently not assigned to) a job.

Public Works: Class Identification

Definitely Classes

Maybe Classes

Probably Not Classes

The Baltimore Public Works ("BPW") needs to keep track of all of the infrastructure that is under their supervision. Common pieces of infrastructure include roads (and their associated accessories, such as traffic lights, stop signs, guard rails, etc), water treatment (both fresh water purification & wastewater) plants, the Baltimore AMTRAK station, parks, the electrical grid, and the Baltimore Schools Department. Each piece of infrastructure needs to track its name, type, location, annual budget, and a list of high-priority and low-priority improvements for it. It should be possible to mark an improvement as completed when it is done; completed improvements should still be tracked. Each improvement should have an estimated cost, a list of expected benefits, and the employee or contractor who will complete it. BPW also needs a way to track employees. Each employee should have their name, ID number, and specialty (f.ex Civil Engineer, Electrician). BPW should have a way to let the public request a new piece of infrastructure, and to request maintenance on an existing piece of infrastructure. It should be possible to find all BPW infrastructure that needs high-priority improvements, as well as all employees currently assigned to (or currently not assigned to) a job.

Public Works: Not A Class

- When designing an OOP system, it can be easy to go overboard and make *everything* a class
- It's important, though, to figure out what *shouldn't* be a class
- Anything that only stores only one or two pieces of data, and doesn't have any specialised behaviour, probably shouldn't be a class
- For example, the name and type of an infrastructure project seem reasonable to store as strings
- Likewise with the name, id number, and specialty of employees
- Location is a bit iffier
    - Plausibly this could be stored as a tuple (latitude, longitude)
    - Might also make sense to store a geofence with many (latitude, longitude) pairs, and has functionality f.ex for calculating area
        - In this case, making it a full class as opposed to just using a tuple would make more sense

Public Works: Describing Classes

- **BPW** tracks **Infrastructure**, **Employees**
- **Infrastructure** tracks name, type, location, budget, **improvements** (high-priority, low-priority, completed), assigned employees
- **Improvement** tracks cost, benefits, assigned employee or contractor
- **Employee** tracks name, ID number, specialty, assigned jobs

Public Works: Identifying Inheritance

- We just identified the main classes that are useful in designing this system
- Are there any opportunities to use inheritance to improve code reuse?
- We might want to introduce subclasses for (at least some of) the different types of infrastructure
  - For example, Road has the notion of "accessories" that isn't present in other types of infrastructure
  - Given a more complex scenario, it *might* make sense for all infrastructure to be their own classes
- Depending on whether "contractor" represents a contracting *company* or individuals, it may make sense to introduce a superclass that stores the shared information & behaviour of Employee and Independent Contractor
  - Given the requirements as stated, probably not at this stage, though

Design: Identifying Behaviour

- To identify promising candidates classes, we looked for nouns
- Similarly, to identify promising methods, we can look for *verbs* or *actions* that are necessary
- Let’s take a look at how we can do this

Public Works: Method Identification

Definitely Classes

Maybe Classes

Probably Not Classes

Probably Methods

The Baltimore Public Works (“BPW”) needs to keep track of all of the infrastructure that is under their supervision. Common pieces of infrastructure include roads (and their associated accessories, such as traffic lights, stop signs, guard rails, etc), water treatment (both fresh water purification & wastewater) plants, the Baltimore AMTRAK station, parks, the electrical grid, and the Baltimore Schools Department. Each piece of infrastructure needs to track its name, type, location, annual budget, and a list of high-priority and low-priority improvements for it. It should be possible to mark an improvement as completed when it is done; completed improvements should still be tracked. Each improvement should have an estimated cost, a list of expected benefits, and the employee or contractor who will complete it. BPW also needs a way to track employees. Each employee should have their name, ID number, and specialty (f.ex Civil Engineer, Electrician). BPW should have a way to let the public request a new piece of infrastructure, and to request maintenance on an existing piece of infrastructure. It should be possible to find all BPW infrastructure that needs high-priority improvements, as well as all employees currently assigned to (or currently not assigned to) a job.

Method Identification

- Plausible methods:
  - `Infrastructure.complete_improvement(which_improvement)`
    - Find the improvement, mark as completed
  - `BPW.request_infrastructure(infrastructure_details)`
    - *Delegates* to the Infrastructure class – searches list of infrastructure to make sure not a duplicate, then creates a new Infrastructure object & adds to list
  - `BPW.request_maintenance(infrastructure, requested_maintenance)`
    - *Delegates* to Infrastructure class – searches list of Infrastructure; creates a new Improvement proposal & adds to list of found object
  - `BPW.find_needing_maintenance()`
    - Loops through list of Infrastructure; for each one that has a list of Improvements, adds it to a new list
  - `BPW.find_employees(assigned?)`
    - Loops through list of employees, finding ones either assigned, or not, depending on argument

Implicit Behaviour

- Not (explicitly) mentioned in the requirements was saving all infrastructure (so that if the program is closed, data isn’t lost) or loading in a text file (or other persistent data store) of infrastructure, employees, etc
- Behaviour like this is *probably* expected – confirm w. your customer to make sure

Pulling it Together

- We now have a plan – we've identified probable classes from a scenario, decided what *shouldn't* be a class, and identified relationships between classes
- We've also identified what each class *should do* – the behaviour necessary for the system to function as desired
- Some sort of analysis like this should *always* be your starting point
    - Think first about what needs to be done before jumping directly into code