

Assignment 1 Plans

This document outlines plans for *some* of the methods in the suggested class diagram.

class Building

readMode() / readName() / readDestination() / readPersonId() / readChoice()

Each of these methods uses the read pattern.

use()

The menu pattern.

addPerson()

This handles the “add person” menu item.

- read the person id
- do a lookup on that person id
- if an existing person was found, print “ID already exists”
- otherwise add a new person with that ID and a name read from the user starting at the building entrance level.

removePerson()

This handles the “remove person” menu item.

- read the person id
- do a lookup on that person id
- if a person was found, remove that person
- otherwise, print “No such person”

showPeople()

This handles the “show people” menu item.

- for each person
 - print the person

showLifts()

This handles the “show lifts” menu item.

- for each lift
 - print the lift

call()

This handles the “call lift” menu item. As written in the assignment guide, a first incremental goal will be to get the call feature working by assuming that the first lift will always be called. i.e. always `lifts.get(0)` will be used.

- read the person id
- do a lookup on that person id
- if the person was not found, print “No such ID”
- otherwise:
 - read the destination
 - get the first lift
 - set the person’s destination (push it right!)
 - add the person to the lift’s queue (push it right!)

If you are not familiar with the “push it right” design rule, it was introduced for the first time in Study 4, and has been repeated in every example since then, especially in Study 5 and Study 6.

Advice for the remaining goals

If you have been struggling with coding the goals above, does it help now to have plans? Hopefully you can now have a better appreciation of the value of plans.

From here on, you need to come up with your own plans. If you try to just write code directly without planning first, your code will be messy, difficult to write, and most likely incorrect, and you will waste a lot of time trying to debug and fix your code’s logic.

Write the plan first in English (or your native language). Get the structure and logic right first, and make sure it corresponds to what the specification says. THEN, write the code according to the plan.