

```

//
// ProcessData.swift
// SimulatedPageAllocation
//
// Created by Brandon Baars on 4/3/18.
// Copyright © 2018 Brandon Baars. All rights reserved.
//

import Foundation

class ProcessData {

    var numOfDataPages: Int
    var numOfCodePages: Int

    var codePageTable = [Int:Frame]()
    var dataPageTable = [Int:Frame]()

    var isTerminated: Bool
    var processNumber: Int

    var codeLength: Int
    var dataLength: Int

    // return our data in a readable string format
    public func toString() -> String {
        return "Loading \(processNumber) into RAM: code=\(codeLength) (\
            numOfCodePages) pages) data=\(dataLength) (\(numOfDataPages) pages)"
    }

    public func addFrameToPageTable(withFrame frame: Frame) {

        // adds the associated frame to our Page Table when it's loaded into
        ram
        if frame.type == .code {
            codePageTable[frame.index] = frame
            print("Loading Code frame from index: ", frame.index)
        } else {
            dataPageTable[frame.index] = frame
            print("Loading Data frame from index: ", frame.index)
        }
    }

    public func removeFromRAM() {
        // Removes the Page Tables of our Code/Data
        self.codePageTable.removeAll()
        self.dataPageTable.removeAll()
    }

    public func queueProcessForRAM(withFrameSize size: Int) {

```

```
// Queue process for loading into RAM by finding the number of pages
// they will need
let codePages = Int(ceil(Double(self.codeLength)/Double(size)))
let dataPages = Int(ceil(Double(self.dataLength)/Double(size)))
```

```
self.numOfCodePages = codePages
self.numOfDataPages = dataPages
```

```
}
```

```
init(numOfDataPages: Int, numOfCodePages: Int, codePageTable: [Int:Frame],
    dataPageTable: [Int:Frame], isTerminated: Bool, processNumber: Int,
    codeLength: Int, dataLength: Int) {
```

```
    self.numOfDataPages = numOfDataPages
    self.numOfCodePages = numOfCodePages
    self.codePageTable = codePageTable
    self.dataPageTable = dataPageTable
    self.isTerminated = isTerminated
    self.processNumber = processNumber
    self.codeLength = codeLength
    self.dataLength = dataLength
```

```
}
```

```
}
```