

PH502: Scientific Programming Concepts

Irish Centre for High End Computing (ICHEC)

September 23, 2020

- Carrying on with our example.
- We need to manage a list of people:
 1. our employees,
 2. and our customers.

- Whatever we store into it, a list is a basic data structure: it is a sequence of items, with a first and a last item.
 - ▶ When pushing a new item in a list, we can either push the item at the front or at the back.
 - ▶ When 'popping' an item out of the list, we can take the first or the last.
- Unfortunately, our Employee list and Customer list don't push/pop in the same ways. Plus we're dealing with two distinct types so we may need two distinct types of lists, one to store each type.

- We want an EmployeeList structure capable of doing the following:

```
createEmptyEmployeeList()  
pushEmployeeInList(employeeelist, employee)  
popEmployeeFromList(employeeelist)  
getEmployeeByName(employeeelist, name)  
getTotalOfSalaries(employeeelist)
```

- 'push' will put the new employee at the back of the list.
- 'pop' will take out the last employee in the list.

- We want a CustomerList structure capable of doing the following:

```
createEmptyCustomerList()
```

```
pushCustomerInList(customerlist, customer)
```

```
popCustomerFromList(customerlist)
```

```
getCustomerByName(customerlist, name)
```

- 'push' will put the new customer at the back of the list.
- 'pop' will take out the first customer from the list.

```
elist = createEmptyEmployeeList()
clist = createEmptyCustomerList()
e = createEmployee('john doe', 'j@d.ie', 30000)
pushEmployeeInList(elist, e)
e = createEmployee('sean lee', 's@l.ie', 29000)
pushEmployeeInList(elist, e)
e = createEmployee('mary kyneeeeeee', 'm@a.ie', 31000)
pushEmployeeInList(elist, e)
c = createCustomer('bob wheatleyy', 'b@w.net')
pushCustomerInList(clist, c)
c = createCustomer('alice martins', 'a@m.com')
pushCustomerInList(clist, c)
```

```
e = getEmployeeByName(elist, "mary kyneeeeeee")
modifyEmployeeName(e, "mary kyne")
c = getCustomerByName(clist, "bob wheatley")
modifyCustomerName(c, "bob wheatley")
c = popCustomerFromList(clist)
helpCustomer(c)
deleteCustomer(c)
e = popEmployeeFromList(elist)
sendP45(e)
deleteEmployee(e)
```

- Note how all of a person's details are neatly passed as a single variable.
- Still, that's a lot of different functions to do almost the same thing...

A new type of person

- Let's imagine we want to add a special type of employee: the 'Executive'.
- An Executive has a name, contact details and a salary.
- An Executive also has a formal title, and a bonus entitlement.
- If an Executive is fired, he must be paid a higher redundancy than regular Employees.

- How do we manage executives in our system?
 - ▶ Add title/bonus to all Employees and let the fields empty for non-execs?
 - ▶ Wasteful in terms of resources. Also, the 'sendP45' function would have to manage two different cases.
 - ▶ Create a new data structure for execs?
 - ▶ 50% more functions with similar names. Problems with inserting an Executive in an EmployeeList.
- How do these designs scale shall we get another 2/3/4/... kinds of employees?
- What of the next programmer whose job it will be to debug/improve this system?
- Object-Oriented design is one possible approach for such a problem.