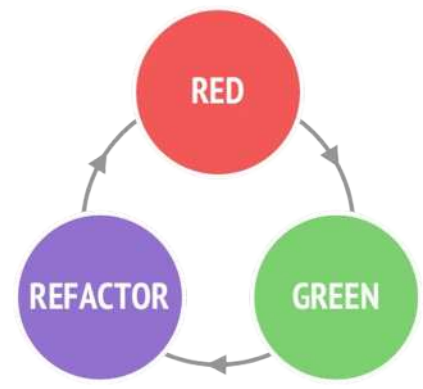
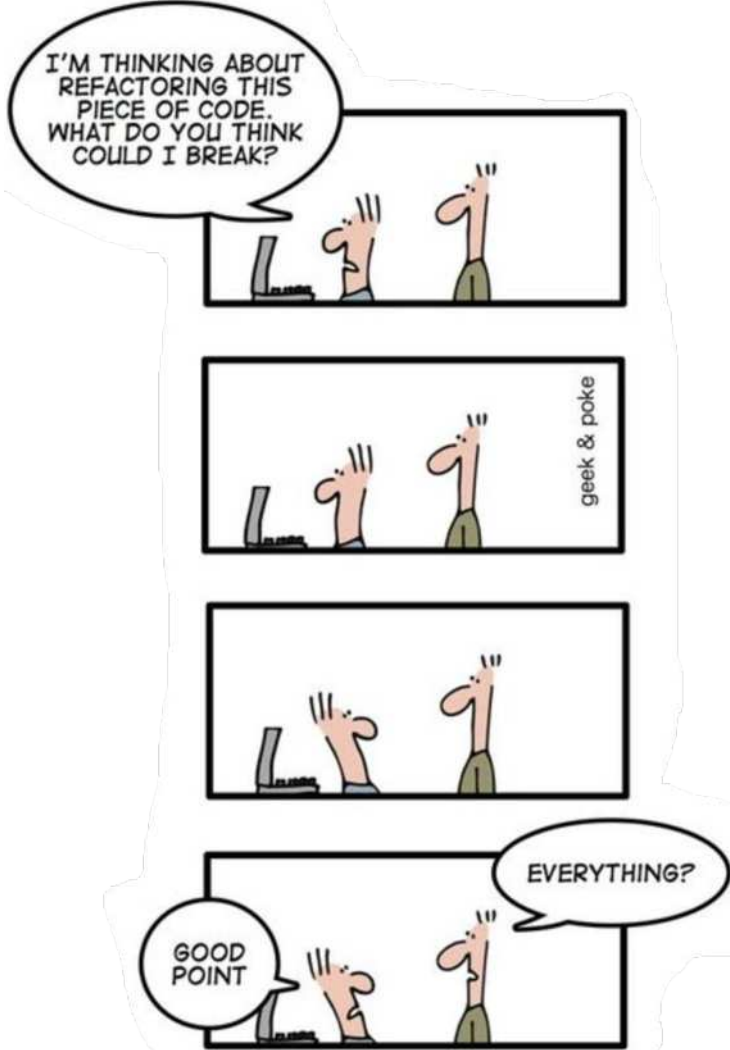


Object Calisthenics

Kholoud Mohamed





Rules

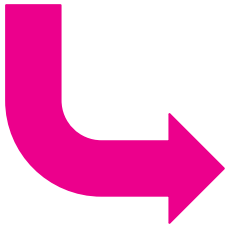
1. **Only One Level Of Indentation Per Method**
2. **Don't Use The ELSE Keyword**
3. **Wrap All Primitives And Strings**
4. **First Class Collections**
5. **One Dot Per Line**
6. **Don't Abbreviate**
7. **Keep All Entities Small**
8. **No Classes With More Than Two Instance Variables**
9. **No Getters/Setters/Properties**



1. Only **One** Level Of Indentation Per Method



```
async selectCasesWithArig(...arigList) {
  for (let arig of arigList) {
    let nextPageAvailable;
    const checkboxSelectorOfArig = `${this.checkboxSelectOneCase}[arigv2='${arig}']`;
    do {
      const arigAvailableInCurrentPage = await I.grabNumberOfVisibleElements(checkboxSelectorOfArig);
      if (arigAvailableInCurrentPage) {
        I.click(checkboxSelectorOfArig);
      }
      nextPageAvailable = await I.grabNumberOfVisibleElements(this.anchorNextPage);
      if (nextPageAvailable) {
        I.click(this.anchorNextPage);
      }
    } while (nextPageAvailable);
  }
}
```



```
async selectMultipleCasesWithArig(...arigList) {
  for (let arig of arigList) {
    await this.selectOneCaseWithArig(arig);
  }
}

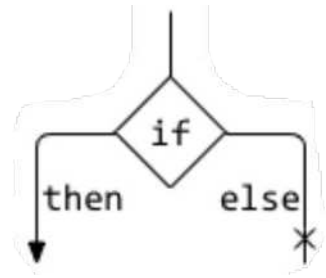
async selectOneCaseWithArig(arig) {
  let nextPageAvailable;
  do {
    await this.clickOnArigIfAvailableInCurrentPage(arig);
    nextPageAvailable = await I.grabNumberOfVisibleElements(this.anchorNextPage);
    nextPageAvailable && I.click(this.anchorNextPage);
  } while (nextPageAvailable);
}

async clickOnArigIfAvailableInCurrentPage(arig) {
  const checkboxSelectorOfArig = `${this.checkboxSelectOneCase}[arigv2='${arig}']`;
  const arigAvailableInCurrentPage = await I.grabNumberOfVisibleElements(checkboxSelectorOfArig);
  arigAvailableInCurrentPage && I.click(checkboxSelectorOfArig);
}
```

2. Don't use the **ELSE** keyword



```
async clickOnCaseWithArig(arig) {
  const arigv1Selector = `tbody>tr[arigv1='${arig}']`;
  const arigv2Selector = `tbody>tr[arigv2='${arig}']`;
  const isArigV1 = await tryTo(() => I.waitForVisible(arigv1Selector));
  const isArigV2 = await tryTo(() => I.waitForVisible(arigv2Selector));
  assert.ok( value: isArigV1 || isArigV2, message: `Arig ${arig} was not found in the list`);
  if (isArigV1) {
    I.click( locator: `${arigv1Selector} a`);
  } else {
    I.click( locator: `${arigv2Selector} a`);
  }
}
```

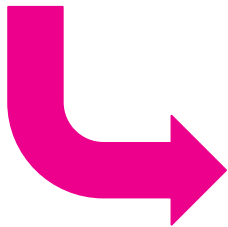
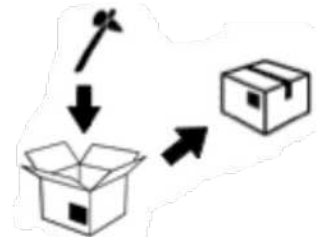


```
if (isArigV1) {
  return I.click( locator: `${arigv1Selector} a`);
}
return I.click( locator: `${arigv2Selector} a`);
}
```

3. Wrap All Primitives And Strings



```
delegateCaseToSpecific(role, person, comment) {  
  I.click(this.buttonDelegate);  
  I.click( locator: `[data-test='delegate__dropdown-delegate--${role}']`);  
  modal.selectUser(person);  
  modal.addComment(comment);  
  modal.clickDelegatеButton();  
}
```

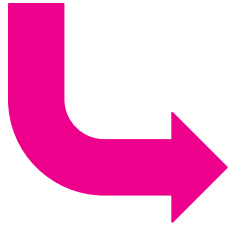


```
delegateCaseToSpecific(role, person, comment) {  
  I.click(this.buttonDelegate);  
  const delegateRoleSelector = `[data-test='delegate__dropdown-delegate--${role}']`;  
  I.click(delegateRoleSelector);  
  modal.selectUser(person);  
  modal.addComment(comment);  
  modal.clickDelegatеButton();  
}
```

5. One Dot Per Line



```
if (valueOrUrn.includes('urn')) {  
  // This to extract the search text from the urn, fourth string after ':' and first before '-' (if any dashes)  
  searchText = valueOrUrn.split(':')[3].split(separator: '-')[0].substring(0, 4);  
} else {  
  searchText = valueOrUrn.substring(0, 4);  
}
```



```
if (valueOrUrn.includes('urn')) {  
  const searchTextWithoutUrn=valueOrUrn.split(':')[3];  
  const searchTextBeforeDash = searchTextWithoutUrn.split('-')[0];  
  searchText = searchTextBeforeDash.substring(0, 4);  
} else {  
  searchText = valueOrUrn.substring(0, 4);  
}
```



6. Don't Abbreviate



```
async getAppLoc() {  
  I.waitForVisible(this.spanSubmissionLocation);  
  return await I.grabTextFrom(this.spanSubmissionLocation);  
}
```



```
async waitForVisibleAndGetApplicationIntakeLocationFromHeader() {  
  I.waitForVisible(this.spanSubmissionLocation);  
  return await I.grabTextFrom(this.spanSubmissionLocation);  
}
```


4. First Class Collections

7. Keep All Entities Small



MAX
= 50 =



380

AllCasesPage

PageHeader
ButtonSearch
...

ClickOnCaseWithID()
ApproveCase()
BulkDelegateCase()
...

158

MyCasesPage

tdAssigneeColumn
ButtonFetch
...

GetTravelDateOf()
GetNoOfCasesAssignedToMe()
...

AllCasesPageElements

MyCasesPageElements

CasesListPageElements

AllCasesPage

MyCasesPage

CasesListPage





CONCLUSION

The advantages of having a small set of rules is that you are free to apply only those you find relevant or only one at the time, to ease your learning curve.

- **Readable**
- **Reusable**
- **Testable**
- **Maintainable**



Thank you!

QUESTIONS?
