|  |  |  |
| --- | --- | --- |
| **Test** | **Description** | **Command** |
| /01-math | Simple example of writing a test  Data-driven test | npm run test:math |
| /02-age-verify | Simple example writing a test  stryker | npm run test:age  npm run stryker |
|  |  |  |
| 00a-the-first-test | Write test  Fails BowlingGame not created yet | npm run test:00a |
| 00b-the-first-test | Make test pass  Create BowlingGame minimum to make pass | npm run test:00b |
| 00c-the-first-test | Update to test to roll 20 gutter balls  Fails roll not written | npm run test:00c |
| 00d-the-first-test | Write roll method to pass  minimum code to make it pass | npm run test:00d |
| 00e-the-first-test | Add expected to test  Fails | npm run test:00e |
| 00f-the-first-test | Update source to score returns -1  Fails -1 instead of 0 | npm run test:00f |
| 00g-the-first-test | Update source to return 0  Success instead of 0 | npm run test:00f |
|  |  |  |
| 01a-the-second-test | Write test handle all 1 pin rolls  Fails returns 0 expected 20 | npm run test:01a |
| 01b-the-second-test | Write source to pass tests | npm run test:01a |
| 01c-the-second-test | REFACTOR  Remove duplicate test code | npm run test:01a |
|  |  |  |
| 02a-the-third-test | Write test handle 1 spare  Contains ugly comment  Fails 13 expected 16 (doesn’t include bonus pins for spare) | npm run test:02a |
| 02b-the-third-test | Code design  Wrong design | npm run test:02b |
| 02c-the-third-test | Refactor code for design.  Comment new test and make sure existing test pass | npm run test:02c |
| 02d-the-third-test | Refactor code for design.  Comment new test and make sure existing test pass with new design | npm run test:02d |
| 02e-the-third-test | Refactor  Remove unused code \_score attribute | npm run test:02e |
| 02f-the-third-test | Uncomment test and make sure it fails | npm run test:02f |
| 02g-the-third-test | Refactor  Design still not correct  Need to refer to frames and not rolls  Need to walk through array of two balls (one frame) at a time | npm run test:02g |
| 02h-the-third-test | Uncomment test and verify fails | npm run test:02h |
| 02i-the-third-test | Write production code to pass test  Ugly comment // spare  i is a bad variable name | npm run test:02i |
| 02j-the-third-test | Refactor  Better name for i  Tests still pass | npm run test:02j |
| 02k-the-third-test | Refactor  Remove ugly comment // spare  Tests still pass | npm run test:02k |
| 02l-the-third-test | Refactor  Remove ugly comment in test  Write rollSpare method  Tests still pass | npm run test:02l |
|  |  |  |
| 03a-the-fourth-test | Write Test make sure it fails  Ugly comment // strike | npm run test:03a |
| 03b-the-fourth-test | Make the test pass  Ugly comment in condition// strike  Ugly expressions strike && spare | npm run test:03b |
| 03c-the-fourth-test | Refactor  Ugly expressions strike && spare  Make sure tests pass | npm run test:03c |
| 03d-the-fourth-test | Refactor  Ugly comment in condition// strike  Make sure tests pass | npm run test:03d |
| 03e-the-fourth-test | Refactor  Ugly comment in test // strike | npm run test:03e |
| 04a-the-fourth-test | Add test perfect game  This test should pass if everything is done right. | npm run test:04a |
| Mocks | http requests in source  returns random value  spy to return known value  test error, via mocks/spies, (sad paths) which may be impossible on live server | npm run mocks |

**snapshot-testing-angular**

npm test jest

|  |  |
| --- | --- |
| Description | Command |
| Install nodejs |  |
| Install angular cli globally | npm install -g @angular/cli |
| Create angular project | ng new snapshot-testing-ng |
|  | cd snapshot-testing-ng |
| Start to verify app | ng serve |
| install dependencies | npm install --save-dev jest jest-preset-angular @types/jest |
| Remove jasmine |  |
| Update npm scripts package.json | "test": "jest",  "test:watch": "jest --watch",  "test:ci": "jest --runInBand",  "test:coverage": "jest --coverage",  "test:clean": "jest --clearCache", |
| Update package.json | "jest": {  "preset": "jest-preset-angular",  "roots": [  "<rootDir>/src/"  ],  "setupTestFrameworkScriptFile": "<rootDir>/src/setupJest.ts",  "moduleNameMapper": {  "@lib/(.\*)": "<rootDir>/src/lib/$1"  }  } |
| Create setupJest.ts |  |
| Create jestGlobalMocks.ts |  |
| App.component.spec.ts | import {CUSTOM\_ELEMENTS\_SCHEMA, NO\_ERRORS\_SCHEMA} from "@angular/core"; |