

Helsingin Yliopisto
Tietojenkäsittelytieteen laitos
Ohjelmistotuotantoprojekti

Esimerkkituoteperhe

Testausraportti

Ryhmä 6

Juha Andersson

Jarmo Kielosto

Leo Linnanmaa

Jan Tilles

Joose Vettenranta

Sisältö

1 Johdanto.....	4
2 Testaus.....	4
2.1 Toteutus.....	4
2.2 Testikattavuudet.....	4
3 Yksikkötestaus.....	6
3.1 CommonTests.....	6
3.1.1 TestApplicationManager.....	6
3.1.2 TestComponentLoader.....	6
3.1.3 TestComponentReference.....	6
3.1.4 TestCore.....	6
3.1.5 TestDataContainer.....	6
3.1.6 TestFileHandling.....	7
3.1.7 TestGameLogic.....	7
3.1.8 TestGameState.....	7
3.1.9 TestGameValidator.....	7
3.1.10 TestHighScore.....	7
3.1.11 TestPlayer.....	7
3.1.12 TestPlayerManager.....	8
3.1.13 TestAboutDialog.....	8
3.1.14 TestPlayerAmountDialog.....	8
3.1.15 GameEventComponent-rajapinnan toteuttavien käyttöliittymäkomponenttien testit.	8
3.2 BowlingTests.....	9
3.2.1 TestBowlingState.....	9
3.2.2 TestBowlingValidator.....	9
3.3 SnookerTests.....	9
3.3.1 TestSnookerHelper.....	9
3.4 DartsAndSnookerTests.....	9
3.4.1 TestSimpleHelper.....	9
3.4.2 TestSimpleState.....	9
3.4.3 TestSimpleValidator.....	10
4 Integrintitestaus.....	11
4.1 CommonTests.....	11
4.1.1 TestPlayerManagement.....	11
testCreateNewPlayer.....	11
testCreateNewPlayerString.....	11
testRemovePlayer.....	11
testRemoveAllPlayers.....	11
4.1.2 TestScoreCounterAndResultHolder.....	11
testAddAndGeetPoints.....	11
testChangePoints.....	12
4.1.3 TestReportManagement.....	12
testGetReportManager.....	12
testGetFileHandling.....	12

testReportManagerSetAndGetNameOfTheGame.....	12
testResultListingToString.....	12
testHighScore.....	12
4.1.4 TestGameManagement.....	12
testGameStartResetAndStop.....	13
testGetAndSetPlayerValue.....	13
testGetPlayerName.....	13
testGameEventHandlerCreatePlayerEvent.....	13
4.1.5 TestGUI.....	13
testGUI.....	13
4.2 BowlingTests.....	13
4.2.1 TestBowlingGameManagement.....	13
testBowlingGameEventHandler.....	13
4.2.2 TestBowlingLogic.....	14
testBowlingStateEvents.....	14
4.3 DartsTests.....	14
4.3.1 TestDartsGameManagement.....	14
testDartsGameEventHandler.....	14
4.4 SnookerTests.....	14
4.4.1 TestSnookerGameManagement.....	14
testSnookerGameEventHandler.....	14
4.4.2 TestSnookerReport.....	14
testHighScore.....	14
4.5 DartsAndSnookerTests.....	15
4.5.1 TestSimpleLogic.....	15
testSimpleValidatorValidateInput.....	15
testSimpleStateGetAndSetState.....	15
testSimpleStateGiveEvent.....	15
5 Järjestelmätestaus.....	16

1 Johdanto

Tässä dokumentissa kuvataan Ohtur6-ryhmän ohjelmistotuotantoprojektin toteuttaman esimerkkituoteperheen testaus. Luvussa kaksi kuvataan yleisesti testauksen toteutus ja tulokset. Luvussa kolme kuvataan lyhyesti kaikki yksikkötestit ja niiden tulokset. Luku neljä kuvaa vastaavasti integrointitestauksen. Viidennessä luvussa esitellään järjestelmätestauksen tulokset.

2 Testaus

Tässä luvussa kuvataan yleisellä tasolla tuoteperheen testaus.

2.1 Toteutus

Testaus suoritettiin testaussuunnitelman mukaisesti Helsingin yliopiston tietojenkäsittelytieteen laitoksen Linux-ympäristössä. Testit toteutettiin JUnit-testeinä siten, että ne ovat suoritettavissa asiakkaan kehittämässä RITA-työkalussa.

Testaussuunnitelman mukaisesti päävastuussa testauksesta oli Jarmo Kielosto. Juha Andersson auttoi yksikkötestauksessa ja testasi erityisesti ne osat jotka Jarmo koodasi, jotta testaaaja oli aina eri henkilö kuin koodaaja. Testaussuunnitelmasta poiketen integrointitestauksen suoritti kokonaan Jarmo ja järjestelmätestauksen Joose Vettenranta.

Löytyneiden virheiden korjaus sovittiin tehtäväksi siten, että pienemmät triviaalisti korjattavat virheet korjasi testaaaja. Isommat virheet raportoitiin luokan koodauksesta vastaavalla. Kaikki testauksessa löydetty virheet korjattiin.

2.2 Testikattavuudet

Testikattavuudet laskettiin RITA-työkalulla. Asiakkaan vaatimus tuotteen ja testien yhteensopivuudesta RITAn kanssa tuli näin samalla testattua.

Testisuunnitelmassa tavoitteeksi asetettua täydellistä lausekattavuutta ei saavutettu. Testikattavuutta laski erityisesti käyttöliittymän vaikea testattavuus. Toisaalta käyttöliittymää testattiin myös järjestelmätestauksessa.

Taulukossa 1 on esitetty RITAn laskemat testikattavuuden tuoteperheen yksittäisille tuotteille ja koko tuoteperheelle. Tuoteperheen kattavuuksissa on mukana vain kaikille tuotteille yhteiset testit, tuotekohtaisissa kaikki kyseiselle tuotteelle ajettavissa olevat testit.

	<i>Keilailu</i>	<i>Mökkitikka</i>	<i>Snooker</i>	<i>Tuoteperhe</i> <i>(kaikille yhteiset testit)</i>
Total statement coverage	60.41	63.70	68.10	47.69
Total decision coverage	43.85	51.06	49.69	32.06
Avg. statement coverage	78.37	80.63	81.26	67.96
Avg. hook statement coverage	32.43	42.74	34.73	20.24
Call coverage	34.64	51.71	44.40	51.69
Total template method coverage	30.00	44.44	40.00	18.68

Taulukko 1: RITAn laskemat testikattavuudet.

3 Yksikkötestaus

Yksikkötestit on jaettu neljään testijoukkoon sen mukaan mitä tuotteita ne testaavat.

3.1 *CommonTests*

Testijoukkoon on koottu testit, jotka ovat ajettavissa kaikille tuotteille.

3.1.1 TestApplicationManager

Testaa luokan ApplicationManager-konstruktoria.

Testin tulos: ei virheitä

3.1.2 TestComponentLoader

Testaa ComponentLoader-luokan toimintaa virhetilanteessa, jossa konstruktorille annetaan parametrina tiedosto, jota ei ole olemassa.

Testin tulos: ei virheitä

3.1.3 TestComponentReference

Testaa ComponentReference-luokan metodeja, joilla asetetaan ja haetaan sovelluskohtaisten luokkien ilmentymät. Get-alkuiset metodit testataan kutsumalla metodia kaksi kertaa ja varmistamalla, että palautetut oliot ovat sama olio. New-alkuiset metodit testataan samoin, mutta varmistetaan että palautetut oliot ovat eri olioita.

Testien tulos: ei virheitä

3.1.4 TestCore

Testaa Core-luokan konstruktoria.

Testin tulos: ei virheitä

3.1.5 TestDataContainer

Testaa DataContainer-luokan toimintaa varmistamalla, että getData()-metodin palauttama olio on sama kuin setData(Object)-metodilla asetettu.

Testin tulos: ei virheitä

3.1.6 TestFileHandling

Testaa FileHandling-luokkaa tallentamalla ensin merkkijonon ja tarkistamalla että sama merkkijono palautetaan ladattaessa.

Testin tulos: ei virheitä

3.1.7 TestGameLogic

Testaa GameLogic-luokkaa tarkistamalla että getNumberOfPlayers() ja getRound()-metodit palauttavat samat arvot kuin setNumberOfPlayers(int) ja setRound(int) -metodeilla on asetettu asetettu.

Testin tulos: ei virheitä

3.1.8 TestGameState

Testaa GameState-luokkaa tarkistamalla että getCurrentPlayer() ja getState()-metodit palauttavat samat arvot kuin setCurrentPlayer(int) ja setState(int) -metodeilla on asetettu asetettu.

Testin tulos: ei virheitä

3.1.9 TestGameValidator

Testaa GameValidator-luokkaa tarkistamalla, että validateNumberOfPlayers(int)-metodi palauttaa true kun parametri on 1-10 ja validatePlayerName(String)-metodi palauttaa true, kun parametrin pituus on korkeintaan 1-50 merkkiä.

Testin tulos: ei virheitä

3.1.10 TestHighScore

Testaa HighScore-luokkaa tarkistamalla että getHighScore()-metodi palauttaa saman arvon kuin setHighScore(String)-metodilla tai luokan konstruktorilla on asetettu.

Testin tulos: ei virheitä

3.1.11 TestPlayer

Testaa Player-luokkaa tarkistamalla että getName(), getDataHolder(), getPlayerValue(String), getIndex() ja toString() -metodit palauttavat samat arvot kuin setName(String), getDataHolder

(DataHolder), getPlayerValue(String, Object) ja setIndex(int) -metodeilla on asetettu.

Testin tulos: ei virheitä

3.1.12 TestPlayerManager

Testaa PlayerManager-luokan virhetilanteita, yrittämällä hakea ja poistaa olematonta pelaajaa, sekä tarkistaa että getNumberOfPlayers()-metodi palauttaa nolla.

Testin tulos: ei virheitä

3.1.13 TestAboutDialog

Testaa AboutDialog-luokkaa luomalla AboutDialog-olion ja kutsumalla sen actionPerformed (ActionEvent)-metodia null parametrilla.

Testin tulos: ei virheitä

3.1.14 TestPlayerAmountDialog

Testaa PlayerAmountDialog-luokkaa luomalla PlayerAmountDialog-olion ja kutsumalla sen actionPerformed(ActionEvent)-metodia null parametrilla.

Testin tulos: ei virheitä

3.1.15 GameEventComponent-rajapinnan toteuttavien käyttöliittymäkomponenttien testit

Testiluokat:

- **TestGameButton,**
- **TestGameComboBox,**
- **TestGameDialog,**
- **TestGameLabel,**
- **TestGameMenuBar,**
- **TestGameMenuItem,**
- **TestGameMenu,**
- **TestGameTextField ja**
- **TestSnookerButton**

sisältävät kaikki samat testitapaukset, koska ne testaavat saman GameEventComponent-rajapinnan toteuttavia luokkia. Testit testaavat että getValue(), getPlayer, getEventValue() ja getFormSlotNumber() -metodit palauttavat samat arvot kuin setValue(String), setPlayer(String),

setEventValue(int) ja setFromSlotNumber(int) -metodeilla on asetettu.

Testien tulos: ei virheitä

3.2 BowlingTests

Testijoukkoon on koottu testit, jotka ovat ajettavissa vain keilailusovellukselle.

3.2.1 TestBowlingState

Testaa BowlingState-luokkaa tarkistamalla että getState()-metodi palauttaa aluksi nolla.

Testin tulos: ei virheitä

3.2.2 TestBowlingValidator

Testaa BowlingValidator-luokkaa tarkistamalla että validateInput(String, String)-metodi validoi oikein keilailun sovelluskohtaiset syötteen.

Testin tulos: ei virheitä

3.3 SnookerTests

Testijoukkoon on koottu testit, jotka ovat ajettavissa vain snookersovellukselle.

3.3.1 TestSnookerHelper

Testaa SnookerHelper-luokkaa tarkistamalla että getMaxPlayerCount()-metodi palauttaa arvon kaksi ja validateInput(String, String)-metodi validoi oikein snookerin sovelluskohtaiset syötteen.

Testin tulos: ei virheitä

3.4 DartsAndSnookerTests

Testijoukkoon on koottu testit, jotka ovat ajettavissa sekä snookerille että mökkitikalle, mutta ei keilailulle.

3.4.1 TestSimpleHelper

Testaa SimpleHelper-luokkaa tarkistamalla että getMaxPlayerCount()-metodi palauttaa arvon 10 ja validateInput(String, String)-metodi validoi oikein mökkitikan sovelluskohtaiset syötteen.

Testin tulos: ei virheitä

3.4.2 TestSimpleState

Testaa SimpleState-luokkaa tarkistamalla että getState()-metodi palauttaa aluksi nolla.

Testin tulos: ei virheitä

3.4.3 TestSimpleValidator

Testaa SimpleValidator-luokan konstruktoreita.

Testin tulos: ei virheitä

4 Integrointitestausta

Integrointitestit on jaettu viiteen testijoukkoon sen mukaan mitä tuotteita ne testaavat. Lisäksi testaussuunnitelmasta poiketen käyttöliittymää testaavat testit jaettiin erilliseen testijoukkoon.

4.1 *CommonTests*

CommonTests-testijoukkoon on koottu testit, jotka ovat ajettavissa kaikille tuotteille.

4.1.1 TestPlayerManagement

Luokan testit testaavat PlayerManagement-komponentin luokkien, PlayerManagerin ja Playerin, yhteistoimintaa.

testCreateNewPlayer

Testaa createNewPlayer()-metodia kutsumalla sitä kaksi kertaa ja varmistamalla getPlayerCount()-metodi palauttaa arvon kaksi ja getPlayer(int) metodin palauttamat oliot ovat eri olioita.

Testin tulos: ei virheitä

testCreateNewPlayerString

Kuten edellinen testi, mutta antaa metodille parametrina pelaajan nimen.

Testin tulos: ei virheitä

testRemovePlayer

Testaa removePlayer(int)-metodia luomalla ja poistamalla pelaajia.

Testin tulos: ei virheitä

testRemoveAllPlayers

Testaa pelaajien poistamista removeAllPlayers()-metodilla, sekä poistetun tai olemattoman pelaajan hakemisen aiheuttamaa poikkeusta getPlayer(int)-metodilla.

Testin tulos: ei virheitä

4.1.2 TestScoreCounterAndResultHolder

Testit testaavat pisteiden hallinnasta vastaavien luokkien, ScoreCounter ja ResultHolder, yhteistoimintaa.

testAddAndGeetPoints

Testaa addPoints(Object), getSubResults() ja getPoints() -metodeja lisäämällä pisteitä ja tarkistamalla, että väli- ja lopputulokset palautetaan oikein.

Testin tulos: ei virheitä

testChangePoints

Testaa jo syötettyjen pisteiden muuttamista `changePoints(int,int)`-metodilla. Testi tarkistaa että vain oikeat pisteet muuttuvat ja tulokset lasketaan oikein.

Testin tulos: ei virheitä

4.1.3 TestReportManagement

Luokan testit testaavat ReportManagement-komponentin kaikille sovelluksille yhteisiä ominaisuuksia.

testGetReportManager

Testaa Core-komponentin ComponentReference-luokan `getReportManager()`-metodia, jota ei testattu yksikkötestauksessa, koska `ReportManager()`-konstruktorin kutsu aiheuttaa kutsuja muihin komponentteihin.

Testin tulos: ei virheitä

testGetFileHandling

Kuten edellinen testi, mutta testaa `getFileHandling()`-metodia.

Testin tulos: ei virheitä

testReportManagerSetAndGetNameOfTheGame

Testaa ReportManager-luokan metodeja, jotka asettavat ja palauttavat sovelluksen nimen, jota käytetään tulosteluksissa.

Testin tulos: ei virheitä

testResultListingToString

Testaa ResultListing-luokan `toString()`-metodia.

Testin tulos: ei virheitä

testHighScore

Testaa HighScore-luokan perustoteutusta, ei mahdollista sovelluskohtaista toteutusta., asettamalla pelaajille tuloksia ja tarkistamalla, että tulokset järjestetään oikein.

Testin tulos: ei virheitä

4.1.4 TestGameManagement

Luokan testit testaavat GameManagement-komponenttia ja sen erikoistavia sovelluskohtaisia komponentteja ja niiden yhteistoimintaa muiden komponenttien kanssa.

testGameStartResetAndStop

Testaa GameManager-luokan GameStart(), GameReset() ja GameStop() -metodeja. Tarkistaa, että GameReset()-metodin kutsumisen jälkeen pelaajien pisteet on nollattu, mutta pelaajatiedot ovat ennallaan.

Testin tulos: ei virheitä

testGetAndSetPlayerValue

Testaa GameManager-luokan setPlayerValue() ja getPlayerValue() -metodeja asettamalla pelaajalle arvon set-metodilla ja tarkistamalla että get-metodi palauttaa oikean arvon.

Testin tulos: ei virheitä

testGetPlayerName

Testaa GameManager-luokan getPlayerName(int)-metodia varmistamalla, että se palauttaa saman nimen kuin pelaajaa luotaessa on annettu.

Testin tulos: ei virheitä

testGameEventHandlerCreatePlayerEvent

Testaa haveEvent-metodia, jonka kaikki sovellukset erikoistavat.

Testin tulos: ei virheitä

4.2 BowlingTests

Testijoukkoon on koottu testit, jotka ovat ajettavissa vain keilailulle.

4.2.1 TestBowlingGameManagement

testBowlingGameEventHandler

Testaa BowlingGameManagement-komponentin erikoistamaa haveEvent(int, GameEventComponent)-metodia sovelluskohtaisilla tapahtumilla.

Testin tulos: ei virheitä

4.2.2 TestBowlingLogic

testBowlingStateEvents

Testaa BowlingState-luokan giveEvent(int)-metodia.

Testin tulos: ei virheitä

4.3 DartsTests

Testijoukkoon on koottu testit, jotka ovat ajettavissa vain mökkitikalle.

4.3.1 TestDartsGameManagement

Testaa DartsGameManagement-komponentin erikoistamaa GameEventHandler-luokkaa ja sen yhteistoimintaa muiden komponenttien kanssa. Kaikille sovelluksille yhteiset eventit, kuten pelaajan luominen, on testattu GameManagement-luokan testeissä.

testDartsGameEventHandler

Testaa DartsGameEventHandlerin GameEventHandler-luokasta erikoistamaa haveEvent(int,GameEventComponent)-metodia. Testi yrittää syöttää pelaajalle virheellisen pistemäärän, sekä käy läpi kahden pelaajan pelin varmisten että peli loppuu ja pisteet ovat oikein, kun molemmille pelaajille on syötetty 10 pistettä.

Testin tulos: ei virheitä

4.4 SnookerTests

Testijoukkoon on koottu testit, jotka ovat ajettavissa vain snookerille.

4.4.1 TestSnookerGameManagement

testSnookerGameEventHandler

Testaa SnookerGameManagement-komponentin erikoistamaa haveEvent(int,GameEventComponent)-metodia sovelluskohtaisilla tapahtumilla.

Testin tulos: ei virheitä

4.4.2 TestSnookerReport

Testaa SnookerReport-komponentin erikoistamaa HighScore-luokkaa ja sen yhteistoimintaa PlayerManagement-komponentin kanssa.

testHighScore

Testaa SnookerHighScoren HighScore-luokasta erikoistamia metodeja asettamalla pelaajille tuloksia ja tarkistamalla, että tulokset järjestetään oikein.

Testin tulos: Heittää NullPointerException-poikkeuksen kun testi ajetaan RITAlla. Erillään ajettaessa ei virheitä.

4.5 DartsAndSnookerTests

Testijoukkoon on koottu testit, jotka ovat ajettavissa sekä snookerille että mökkitikalle.

4.5.1 TestSimpleLogic

Luokan testit testaavat SimpleLogic-komponentin luokkia, sekä snookerille ajettaessa SnookerLogic-komponentin erikoistamia luokkia.

testSimpleValidatorValidateInput

Testaa SimpleValidator-luokan erikoistaman validateInput(String,Object)-metodin toimintaa.

Testin tulos: ei virheitä

testSimpleStateGetAndSetState

Testaa SimpleState-luokan erikoistamien setState(int) ja getState()-metodien toimintaa.

Testin tulos: ei virheitä

testSimpleStateGiveEvent

Testaa SimpleState-luokan erikoistaman giveEvent(int)-metodin toimintaa.

Testin tulos: ei virheitä

4.6 GuiTests

4.6.1 TestGUI

testGUI

Pyytää ComponentReferenceltä tuotekohtaisen-luokan ja GUI kutsuu sen createUI()-metodia, joka alustaa sovelluskohtaisen käyttöliittymän.

Testin tulos: ei virheitä

4.6.2 TestGameGuiManager

testGameGuiManager

Testaa GameGuiManager-luokkaa ja sen erikoistavia sovelluskohtaisia luokkia hakemalla

ComponentReferenceltä sovelluskohtaisen luokan ja kutsumalla sen metodeja
createApplicationUI(), showPlayerAmountDialog(int, GameGuiManager),
showPlayerNameDialog(int, GameGuiManager), showAbout() ja alert(String), confirm(String).

Testin tulos: ei virheitä

5 Järjestelmätestaus

Järjestelmätestauksessa testattiin sovellusten käyttöliittymää ja toiminnallisuutta testaussuunnitelmassa määriteltyjen testitapausten perusteella. Testauksen tulokset on koottu taulukkoon 2.

	<i>Mökkitikka</i>	<i>Keilailu</i>	<i>Snooker</i>
JT-1	Ok	Ok	-
JT-2	Jos syöttää väärän tiedon pelaajanimen kentään, niin se kyllä valittaa että tiedot olivat vääräiä kuten pitää, mutta ei anna jatkaa tietojen syöttämistä sen jälkeen.	Jos syöttää väärän tiedon pelaajanimen kentään, niin se kyllä valittaa että tiedot olivat vääräiä kuten pitää, mutta ei anna jatkaa tietojen syöttämistä sen jälkeen.	Jos syöttää väärän tiedon pelaajanimen kentään, niin se kyllä valittaa että tiedot olivat vääräiä kuten pitää, mutta ei anna jatkaa tietojen syöttämistä sen jälkeen.
JT-3	Ok	Ok	Ok
JT-4	Ok	Ok	Ok
JT-5	Ok	-	-
JT-6	Toiminnallisuutta ei toteutettu	-	-
JT-7	-	Ok	-
JT-8	-	-	Pisteet lisääntyy oikein, mutta kentät ei disabloidu ollenkaan, koska toiminnallisuutta ei toteutettu
JT-9	Ok	Ok	Ok
JT-10	Ok	Ok	Ok
JT-11	Ok	Ok	Ok
JT-12	-	Ok	Toiminnallisuutta ei toteutettu
JT-13	-	Ok	Toiminnallisuutta ei toteutettu

Taulukko 2: Järjestelmätestauksen tulokset.