Leveltestplan – Komponententests

Testentwurfsspezifikationen

Alle Tests sind Blackboxtests und werden automatisiert durchgeführt. Als Framework wird dafür innerhalb der Javaapplikation JUnit verwendet. Es werden Klassen korrelierend zu den zu testenden Komponenten geschaffen. In diesen werden Testfälle definiert und automatisiert durchgeführt. Die Ergebnisse werden in Protokollen festgehalten.

clothingSizeCalculatorModel

Äquivalenzklassen:

A.01. Der User gibt eine Höhe kleiner als 92 cm ein.

A.02. Der User gibt eine Höhe von genau 92 cm ein.

A.03. Der User gibt eine Höhe zwischen größer 92 cm und 98 cm ein.

A.04. Der User gibt eine Höhe zwischen größer 98 cm und 104 cm ein.

A.05. Der User gibt eine Höhe zwischen größer 104 cm und 110 cm ein.

A.06. Der User gibt eine Höhe zwischen größer 110 cm und 116 cm ein.

A.07. Der User gibt eine Höhe zwischen größer 116 cm und 122 cm ein.

A.08. Der User gibt eine Höhe zwischen größer 122 cm und 128 cm ein.

A.09. Der User gibt eine Höhe zwischen größer 128 cm und 134 cm ein.

A.10. Der User gibt eine Höhe zwischen größer 134 cm und 140 cm ein.

A.11. Der User gibt eine Höhe zwischen größer 140 cm und 146 cm ein.

A.12. Der User gibt eine Höhe zwischen größer 146 cm und 152 cm ein.

A.13. Der User gibt eine Höhe zwischen größer 152 cm und 158 cm ein.

A.14. Der User gibt eine Höhe zwischen größer 158 cm und 164 cm ein.

A.15. Der User gibt eine Höhe zwischen größer 164 cm und 170 cm ein.

A.16. Der User gibt eine Höhe zwischen größer 170 cm und 176 cm ein.

A.17. Der User gibt eine Höhe größer als 176 cm ein.

Grenzwerte:

A.18. Der User gibt eine Höhe um den Grenzwert 92 cm ein.

A.19. Der User gibt eine Höhe um den Grenzwert 98 cm ein.

A.20. Der User gibt eine Höhe um den Grenzwert 104 cm ein.

A.21. Der User gibt eine Höhe um den Grenzwert 110 cm ein.

A.22. Der User gibt eine Höhe um den Grenzwert 116 cm ein.

A.23. Der User gibt eine Höhe um den Grenzwert 122 cm ein.

A.24. Der User gibt eine Höhe um den Grenzwert 128 cm ein.

A.25. Der User gibt eine Höhe um den Grenzwert 134 cm ein.

A.26. Der User gibt eine Höhe um den Grenzwert 140 cm ein.

A.27. Der User gibt eine Höhe um den Grenzwert 146 cm ein.

A.28. Der User gibt eine Höhe um den Grenzwert 152 cm ein.

A.29. Der User gibt eine Höhe um den Grenzwert 158 cm ein.

A.30. Der User gibt eine Höhe um den Grenzwert 164 cm ein.

A.31. Der User gibt eine Höhe um den Grenzwert 170 cm ein.

A.32. Der User gibt eine Höhe um den Grenzwert 176 cm ein.

growthCoefficientCalculatorModel

Äquivalenzklassen:

B.01. Der User gibt eine Höhe kleiner als 92 cm ein.

B.02. Der User gibt eine Höhe größer als 176 cm ein.

B.03. Der User gibt einen Hüftumfang kleiner als 51 cm ein.

B.04. Der User gibt einen Hüftumfang größer als 70 cm ein.

B.05. Der User gibt einen Brustumfang kleiner als 54 cm ein.

B.06. Der User gibt einen Brustumfang größer als 88 cm ein.

Grenzwerte:

B.07. Der User gibt eine Höhe um den Grenzwert 92 cm ein.

B.08. Der User gibt eine Höhe um den Grenzwert 176 cm ein.

B.09. Der User gibt einen Hüftumfang um den Grenzwert 51 cm ein.

B.10. Der User gibt einen Hüftumfang um den Grenzwert 70 cm ein.

B.11. Der User einen Brustumfang um den Grenzwert 54 cm ein.

B.12. Der User einen Brustumfang um den Grenzwert 88 cm ein.

checkValuesService

Äquivalenzklassen:

B.01. Der User gibt eine Höhe kleiner als 92 cm ein.

B.02. Der User gibt eine Höhe größer als 176 cm ein.

B.03. Der User gibt einen Hüftumfang kleiner als 51 cm ein.

B.04. Der User gibt einen Hüftumfang größer als 70 cm ein.

B.05. Der User gibt einen Brustumfang kleiner als 54 cm ein.

B.06. Der User gibt einen Brustumfang größer als 88 cm ein.

E.01. Der User gibt eine Zahl kleiner oder gleich 0 ein.

Negativtest:

E.02. Dem Scanner wird ein String statt ein float übergeben.

messagesService

Das Interface wird in die Testklasse implementiert. Der Streamoutput wird manipuliert (Bytestream/Printstream) um eine automatisierte Überprüfung zu ermöglichen. Der Testklasse werden valide Werte übergeben – der Output wird überprüft.

consoleView

Der Streamoutput wird manipuliert (Bytestream/Printstream) um eine automatisierte Überprüfung zu ermöglichen. Der Testklasse werden valide Werte übergeben – der Output wird überprüft.

Testfallspezifikationen

[Testfallspezifikationen Äquivalenzklassen.docx](file:///C:\DATEN\Text\Studium\Drittes_Semester\SoftwareTestStrategien\Übung\childrenClothingSizeCalculator\Diagrams%20and%20Strategies\Software%20Test%20Documentation\Testfallspezifikationen%20Äquivalenzklassen.docx)

[Testfallspezifikationen Grenzwertanalyse.docx](Testfallspezifikationen%20Grenzwertanalyse.docx)

[Testfallspezifikationen Streamoutput.docx](Testfallspezifikationen%20Streamoutput.docx)

[Testfallspezifikationen Negativtest.docx](Testfallspezifikationen%20Negativtest.docx)

Protokollierung

[Protokoll clothingSizeCalulatorModel.docx](file:///C:\DATEN\Text\Studium\Drittes_Semester\SoftwareTestStrategien\Übung\childrenClothingSizeCalculator\Diagrams%20and%20Strategies\Software%20Test%20Documentation\Protokoll%20clothingSizeCalulatorModel.docx)

[Protokoll growthCoefficientCalculatorModel.docx](file:///C:\DATEN\Text\Studium\Drittes_Semester\SoftwareTestStrategien\Übung\childrenClothingSizeCalculator\Diagrams%20and%20Strategies\Software%20Test%20Documentation\Protokoll%20growthCoefficientCalculatorModel.docx)

[Protokoll checkValuesService.docx](Protokoll%20checkValuesService.docx)

[Protokoll messagesService.docx](file:///C:\DATEN\Text\Studium\Drittes_Semester\SoftwareTestStrategien\Übung\childrenClothingSizeCalculator\Diagrams%20and%20Strategies\Software%20Test%20Documentation\Protokoll%20messagesService.docx)

[Protokoll consoleView.docx](file:///C:\DATEN\Text\Studium\Drittes_Semester\SoftwareTestStrategien\Übung\childrenClothingSizeCalculator\Diagrams%20and%20Strategies\Software%20Test%20Documentation\Protokoll%20consoleView.docx)