

Softwareentwicklung 2 Repitorium - Block 1

Michael Strassberger

Juli 2019

Welche Merkmale von Softwaresystemen zeigen deren Komplexität?

- viele Anforderungen
 - Zuständigkeiten des Systems
- Randbedingungen
 - Gesetze
 - Unternehmensstruktur
- viele Beteiligte
 - Viele Köche verderben den Brei
- Interdisziplinarität
 - Security
 - Datenbanken
 - UI, Ergonomie
 - Administration, Deployment
- Interoperabilität zwischen externen Systemen und der Umgebung
 - Maschinensteuerung
 - Warenbeschaffung
 - Mensch Computer Interface

Welche Merkmale von Softwaresystemen zeigen deren Komplexität?

- viele Anforderungen
 - Zuständigkeiten des Systems
- Randbedingungen
 - Gesetze
 - Unternehmensstruktur
- viele Beteiligte
 - Viele Köche verderben den Brei
- Interdisziplinarität
 - Security
 - Datenbanken
 - UI, Ergonomie
 - Administration, Deployment
- Interoperabilität zwischen externen Systemen und der Umgebung
 - Maschinensteuerung
 - Warenbeschaffung
 - Mensch Computer Interface

Welche Merkmale von Softwaresystemen zeigen deren Komplexität?

- viele Anforderungen
 - Zuständigkeiten des Systems
- Randbedingungen
 - Gesetze
 - Unternehmensstruktur
- viele Beteiligte
 - Viele Köche verderben den Brei
- Interdisziplinarität
 - Security
 - Datenbanken
 - UI, Ergonomie
 - Administration, Deployment
- Interoperabilität zwischen externen Systemen und der Umgebung
 - Maschinensteuerung
 - Warenbeschaffung
 - Mensch Computer Interface

Welche Merkmale von Softwaresystemen zeigen deren Komplexität?

- viele Anforderungen
 - Zuständigkeiten des Systems
- Randbedingungen
 - Gesetze
 - Unternehmensstruktur
- viele Beteiligte
 - Viele Köche verderben den Brei
- Interdisziplinarität
 - Security
 - Datenbanken
 - UI, Ergonomie
 - Administration, Deployment
- Interoperabilität zwischen externen Systemen und der Umgebung
 - Maschinensteuerung
 - Warenbeschaffung
 - Mensch Computer Interface

Welche Merkmale von Softwaresystemen zeigen deren Komplexität?

- viele Anforderungen
 - Zuständigkeiten des Systems
- Randbedingungen
 - Gesetze
 - Unternehmensstruktur
- viele Beteiligte
 - Viele Köche verderben den Brei
- Interdisziplinarität
 - Security
 - Datenbanken
 - UI, Ergonomie
 - Administration, Deployment
- Interoperabilität zwischen externen Systemen und der Umgebung
 - Maschinensteuerung
 - Warenbeschaffung
 - Mensch Computer Interface

Welche Merkmale von Softwaresystemen zeigen deren Komplexität?

- viele Anforderungen
 - Zuständigkeiten des Systems
- Randbedingungen
 - Gesetze
 - Unternehmensstruktur
- viele Beteiligte
 - Viele Köche verderben den Brei
- Interdisziplinarität
 - Security
 - Datenbanken
 - UI, Ergonomie
 - Administration, Deployment
- Interoperabilität zwischen externen Systemen und der Umgebung
 - Maschinensteuerung
 - Warenbeschaffung
 - Mensch Computer Interface

Nennen Sie wichtige Quellen von Veränderungen mit Auswirkungen auf Softwaresysteme!

- Optimierung von Workflows
- Anpassung an den Markt
 - Online Handel
 - Individual Lösungen für Kunden
- Randbedingungen
 - neue Gesetze
 - Fusion mit anderen Unternehmen
- Technische weiterentwicklungen
 - Java 11
 - C++20

Nennen Sie wichtige Quellen von Veränderungen mit Auswirkungen auf Softwaresysteme!

- Optimierung von Workflows
- Anpassung an den Markt
 - Online Handel
 - Individual Lösungen für Kunden
- Randbedingungen
 - neue Gesetze
 - Fusion mit anderen Unternehmen
- Technische weiterentwicklungen
 - Java 11
 - C++20

Nennen Sie wichtige Quellen von Veränderungen mit Auswirkungen auf Softwaresysteme!

- Optimierung von Workflows
- Anpassung an den Markt
 - Online Handel
 - Individual Lösungen für Kunden
- Randbedingungen
 - neue Gesetze
 - Fusion mit anderen Unternehmen
- Technische weiterentwicklungen
 - Java 11
 - C++20

Nennen Sie wichtige Quellen von Veränderungen mit Auswirkungen auf Softwaresysteme!

- Optimierung von Workflows
- Anpassung an den Markt
 - Online Handel
 - Individual Lösungen für Kunden
- Randbedingungen
 - neue Gesetze
 - Fusion mit anderen Unternehmen
- Technische weiterentwicklungen
 - Java 11
 - C++20

Nennen Sie wichtige Quellen von Veränderungen mit Auswirkungen auf Softwaresysteme!

- Optimierung von Workflows
- Anpassung an den Markt
 - Online Handel
 - Individual Lösungen für Kunden
- Randbedingungen
 - neue Gesetze
 - Fusion mit anderen Unternehmen
- Technische weiterentwicklungen
 - Java 11
 - C++20

Nennen Sie drei grundsätzliche Ansätze, Komplexität für Menschen beherrschbar zu machen!

(1) Aufteilen

- Module
- Information Hiding (privater Zustand)
- Separation of concerns

(2) Hierarchiebildung

- Abstraktion, Dekomposition, Generalisierung
- Zerlegung komplexer Strukturen → WAM

(3) Pattern

- Verwendung von bereits etablierten Software Bibliotheken
- Beobachtermuster
- Frameworks für Datenbanken
- Architekturstil → WAM

Nennen Sie drei grundsätzliche Ansätze, Komplexität für Menschen beherrschbar zu machen!

(1) Aufteilen

- Module
- Information Hiding (privater Zustand)
- Separation of concerns

(2) Hierarchiebildung

- Abstraktion, Dekomposition, Generalisierung
- Zerlegung komplexer Strukturen → WAM

(3) Pattern

- Verwendung von bereits etablierten Software Bibliotheken
- Beobachtermuster
- Frameworks für Datenbanken
- Architekturstil → WAM

Nennen Sie drei grundsätzliche Ansätze, Komplexität für Menschen beherrschbar zu machen!

(1) Aufteilen

- Module
- Information Hiding (privater Zustand)
- Separation of concerns

(2) Hierarchiebildung

- Abstraktion, Dekomposition, Generalisierung
- Zerlegung komplexer Strukturen → WAM

(3) Pattern

- Verwendung von bereits etablierten Software Bibliotheken
- Beobachtermuster
- Frameworks für Datenbanken
- Architekturstil → WAM

Erläutern Sie verschiedene Arten von Abstraktion!

- Reduzierung auf Schlüsseigenschaften (Abstraktion)
 - Wie soll sich ein Konto verhalten (Saldo speichern, abheben, einzahlen, überweisen)
- Gemeinsame Eigenschaften fokussieren (Generalisierung)
 - Klasse der Insekten, Mensch, Land ...
- Fachliche Fakten → Geschäftsobjekt Kunde

Erläutern Sie verschiedene Arten von Abstraktion!

- Reduzierung auf Schlüsseleigenschaften (Abstraktion)
 - Wie soll sich ein Konto verhalten (Saldo speichern, abheben, einzahlen, überweisen)
- Gemeinsame Eigenschaften fokussieren (Generalisierung)
 - Klasse der Insekten, Mensch, Land ...
- Fachliche Fakten → Geschäftsobjekt Kunde

Erläutern Sie verschiedene Arten von Abstraktion!

- Reduzierung auf Schlüsseigenschaften (Abstraktion)
 - Wie soll sich ein Konto verhalten (Saldo speichern, abheben, einzahlen, überweisen)
- Gemeinsame Eigenschaften fokussieren (Generalisierung)
 - Klasse der Insekten, Mensch, Land ...
- Fachliche Fakten → Geschäftsobjekt Kunde

Erläutern Sie verschiedene Arten von Abstraktion!

- Reduzierung auf Schlüsseigenschaften (Abstraktion)
 - Wie soll sich ein Konto verhalten (Saldo speichern, abheben, einzahlen, überweisen)
- Gemeinsame Eigenschaften fokussieren (Generalisierung)
 - Klasse der Insekten, Mensch, Land ...
- Fachliche Fakten → Geschäftsobjekt Kunde

Worin besteht der Unterschied der Fehler-Begriffe error, failure, fault, mistake?

Menschliche Fehlhandlungen sogenannte _ 1 _ können zu Fehlern im Quelltext der Software als _ 2 _ führen. Ein _ 3 _ beschreibt das Versagen eines System aufgrund eines oder mehrerer _ 2 _. Wenn eine Software nicht den Forderungen entspricht sprechen wir von _ 4 _.

- (1) Mistake
- (2) Fault (s)
- (3) Failure
- (4) Error

Worin besteht der Unterschied der Fehler-Begriffe error, failure, fault, mistake?

Menschliche Fehlhandlungen sogenannte _ 1 _ können zu Fehlern im Quelltext der Software als _ 2 _ führen. Ein _ 3 _ beschreibt das Versagen eines System aufgrund eines oder mehrerer _ 2 _. Wenn eine Software nicht den Forderungen entspricht sprechen wir von _ 4 _.

- (1) Mistake
- (2) Fault (s)
- (3) Failure
- (4) Error

Worin besteht der Unterschied der Fehler-Begriffe error, failure, fault, mistake?

Menschliche Fehlhandlungen sogenannte 1 können zu Fehlern im Quelltext der Software als 2 führen. Ein 3 beschreibt das Versagen eines System aufgrund eines oder mehrerer 2 . Wenn eine Software nicht den Forderungen entspricht sprechen wir von 4 .

- (1) Mistake
- (2) Fault (s)
- (3) Failure
- (4) Error

Worin besteht der Unterschied der Fehler-Begriffe error, failure, fault, mistake?

Menschliche Fehlhandlungen sogenannte 1 können zu Fehlern im Quelltext der Software als 2 führen. Ein 3 beschreibt das Versagen eines System aufgrund eines oder mehrerer 2 . Wenn eine Software nicht den Forderungen entspricht sprechen wir von 4 .

- (1) Mistake
- (2) Fault (s)
- (3) Failure
- (4) Error

Worin besteht der Unterschied der Fehler-Begriffe error, failure, fault, mistake?

Menschliche Fehlhandlungen sogenannte _ 1 _ können zu Fehlern im Quelltext der Software als _ 2 _ führen. Ein _ 3 _ beschreibt das Versagen eines System aufgrund eines oder mehrerer _ 2 _. Wenn eine Software nicht den Forderungen entspricht sprechen wir von _ 4 _.

- (1) Mistake
- (2) Fault (s)
- (3) Failure
- (4) Error

Worin besteht defensiver Programmierstil?

- Risiken vermeiden
- Debugger benutzen
- Anwesenheit von Fehlern erwarten
- Fehlerreaktion auf minimale Auswirkungen richten
- Korrekte Rechtschreibung
- Kaffee trinken, wacher schreibt man weniger Fehler
- Auswirkungen von Fehlern verringern im besten fall ganz vermeiden

Erläutern Sie das Prinzip Design by Contract!

Das Vertragsmodell beschreibt ein Vertragsverhältnis bei 1 zwischen Klassen.

Dabei wird beschrieben welche 2 der Klient gegenüber dem Dienstleister erbringen muss.

Im gegenzug kann der Dienstleister gegenüber dem Klienten 3 garantieren.

Diese Vertragsbedingungen werden durch 4 beschrieben.

Zusätzlich zu 2 und 3 gibt es noch Bedingungen die immer gelten sollen sogenannte 5.

- (1) Benutzt-Beziehung
- (2) Vorleistung / Vorbedingung
- (3) Zusicherung / Nachbedingung
- (4) boolesche Ausdrücke (Prädikate)
- (5) Klassen-Invarianten

Erläutern Sie das Prinzip Design by Contract!

Das Vertragsmodell beschreibt ein Vertragsverhältnis bei 1 zwischen Klassen.

Dabei wird beschrieben welche 2 der Klient gegenüber dem Dienstleister erbringen muss.

Im gegenzug kann der Dienstleister gegenüber dem Klienten 3 garantieren.

Diese Vertragsbedingungen werden durch 4 beschrieben.

Zusätzlich zu 2 und 3 gibt es noch Bedingungen die immer gelten sollen sogenannte 5.

- (1) Benutzt-Beziehung
- (2) Vorleistung / Vorbedingung
- (3) Zusicherung / Nachbedingung
- (4) boolesche Ausdrücke (Prädikate)
- (5) Klassen-Invarianten

Erläutern Sie das Prinzip Design by Contract!

Das Vertragsmodell beschreibt ein Vertragsverhältnis bei 1 zwischen Klassen.

Dabei wird beschrieben welche 2 der Klient gegenüber dem Dienstleister erbringen muss.

Im gegenzug kann der Dienstleister gegenüber dem Klienten 3 garantieren.

Diese Vertragsbedingungen werden durch 4 beschrieben.

Zusätzlich zu 2 und 3 gibt es noch Bedingungen die immer gelten sollen sogenannte 5.

- (1) Benutzt-Beziehung
- (2) Vorleistung / Vorbedingung
- (3) Zusicherung / Nachbedingung
- (4) boolesche Ausdrücke (Prädikate)
- (5) Klassen-Invarianten

Erläutern Sie das Prinzip Design by Contract!

Das Vertragsmodell beschreibt ein Vertragsverhältnis bei 1 zwischen Klassen.

Dabei wird beschrieben welche 2 der Klient gegenüber dem Dienstleister erbringen muss.

Im gegenzug kann der Dienstleister gegenüber dem Klienten 3 garantieren.

Diese Vertragsbedingungen werden durch 4 beschrieben.

Zusätzlich zu 2 und 3 gibt es noch Bedingungen die immer gelten sollen sogenannte 5.

- (1) Benutzt-Beziehung
- (2) Vorleistung / Vorbedingung
- (3) Zusicherung / Nachbedingung
- (4) boolesche Ausdrücke (Prädikate)
- (5) Klassen-Invarianten

Erläutern Sie das Prinzip Design by Contract!

Das Vertragsmodell beschreibt ein Vertragsverhältnis bei 1 zwischen Klassen.

Dabei wird beschrieben welche 2 der Klient gegenüber dem Dienstleister erbringen muss.

Im gegenzug kann der Dienstleister gegenüber dem Klienten 3 garantieren.

Diese Vertragsbedingungen werden durch 4 beschrieben.

Zusätzlich zu 2 und 3 gibt es noch Bedingungen die immer gelten sollen sogenannte 5.

- (1) Benutzt-Beziehung
- (2) Vorleistung / Vorbedingung
- (3) Zusicherung / Nachbedingung
- (4) boolesche Ausdrücke (Prädikate)
- (5) Klassen-Invarianten

Erläutern Sie das Prinzip Design by Contract!

Das Vertragsmodell beschreibt ein Vertragsverhältnis bei 1 zwischen Klassen.

Dabei wird beschrieben welche 2 der Klient gegenüber dem Dienstleister erbringen muss.

Im gegenzug kann der Dienstleister gegenüber dem Klienten 3 garantieren.

Diese Vertragsbedingungen werden durch 4 beschrieben.

Zusätzlich zu 2 und 3 gibt es noch Bedingungen die immer gelten sollen sogenannte 5.

- (1) Benutzt-Beziehung
- (2) Vorleistung / Vorbedingung
- (3) Zusicherung / Nachbedingung
- (4) boolesche Ausdrücke (Prädikate)
- (5) Klassen-Invarianten

Wie sollte das Vertragsmodell in Java umgesetzt werden?

Vorbedingungen können in Java mithilfe des `assert` keywords geprüft werden. *wichtig* hierbei ist, dass diese Überprüfung eingeschaltet sein muss beim Aufruf des Programmes

Nachbedingungen können mithilfe von Unit-Test überprüft werden.

Wie kann das Vertragsmodell bei der Erstellung von Tests helfen?

- Vorbedingungen geben gültige Parameter an (Positivtest)
- Vorbedingungen geben eventuelle Grenzfälle an (erster gültiger wert, letzter ...)
- Nachbedingungen können mithilfe von Vergleichoperationen nachgeprüft werden im test

Wie sollte das Vertragsmodell in Java umgesetzt werden?

Vorbedingungen können in Java mithilfe des `assert` keywords geprüft werden. *wichtig* hierbei ist, dass diese Überprüfung eingeschaltet sein muss beim Aufruf des Programmes

Nachbedingungen können mithilfe von Unit-Test überprüft werden.

Wie kann das Vertragsmodell bei der Erstellung von Tests helfen?

- Vorbedingungen geben gültige Parameter an (Positivtest)
- Vorbedingungen geben eventuelle Grenzfälle an (erster gültiger wert, letzter ...)
- Nachbedingungen können mithilfe von Vergleichoperationen nachgeprüft werden im test

Wie sollte das Vertragsmodell in Java umgesetzt werden?

Vorbedingungen können in Java mithilfe des `assert` keywords geprüft werden. *wichtig* hierbei ist, dass diese Überprüfung eingeschaltet sein muss beim Aufruf des Programmes

Nachbedingungen können mithilfe von Unit-Test überprüft werden.

Wie kann das Vertragsmodell bei der Erstellung von Tests helfen?

- Vorbedingungen geben gültige Parameter an (Positivtest)
- Vorbedingungen geben eventuelle Grenzfälle an (erster gültiger wert, letzter ...)
- Nachbedingungen können mithilfe von Vergleichoperationen nachgeprüft werden im test

Wie sollte das Vertragsmodell in Java umgesetzt werden?

Vorbedingungen können in Java mithilfe des `assert` keywords geprüft werden. *wichtig* hierbei ist, dass diese Überprüfung eingeschaltet sein muss beim Aufruf des Programmes

Nachbedingungen können mithilfe von Unit-Test überprüft werden.

Wie kann das Vertragsmodell bei der Erstellung von Tests helfen?

- Vorbedingungen geben gültige Parameter an (Positivtest)
- Vorbedingungen geben eventuelle Grenzfälle an (erster gültiger wert, letzter ...)
- Nachbedingungen können mithilfe von Vergleichoperationen nachgeprüft werden im test

Exception Hierarchie

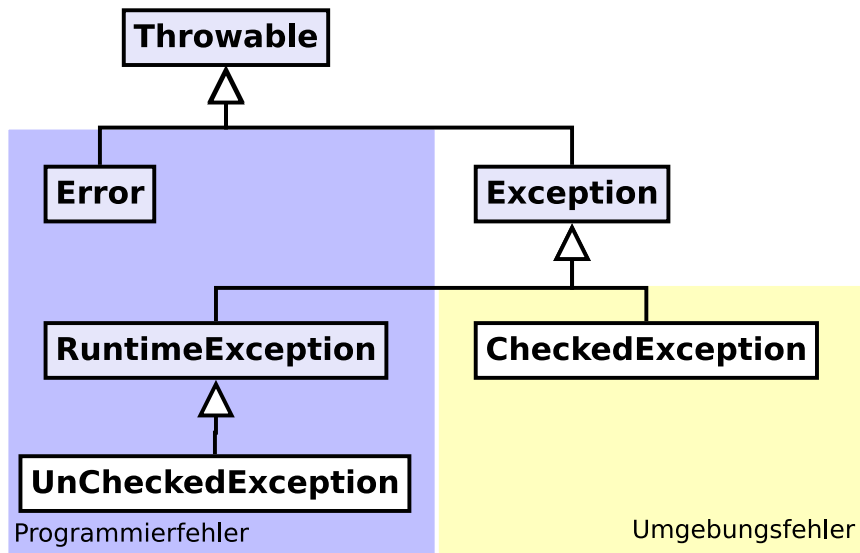


Figure 1: Java Exceptionklassen

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U

angelehnt aus: SE2-VL05-FehlerModule.pdf Folie 49

Ordnen sie den Fehler beschreibungen zu ob es sich um einen Umgebungsfehler (U) oder Programmierfehler (P) handelt

- Implementation erfüllt nicht die Anforderungen der Spezifikation P
- Bei schreibendem Zugriff auf Datei hat ein Programm keine Schreibrechte U
- Bei Anforderung einer Dienstleistung hält sich Klient nicht an Vorbedingungen P
- Für Zugriff auf Webseite hat Browser keinen Zugang zum Internet. U
- Bei Array-Zugriff wird ein ungültiger Index benutzt P
- Durch konkurrierenden (gleichzeitigen) Zugriff zweier Prozesse auf ein Objekt gerät dieses in einen inkonsistenten Zustand P
- Ein Benutzer tippt in ein Eingabefeld einer grafischen Oberfläche Daten im falschen Format. U