Curriculum für

Certified Professional for Software Architecture (CPSA)®
Advanced Level

Modul
FM

Formale Methoden

2024.1-RC1 - DE-20240918

Inhaltsverzeichnis

Die Abkürzung "e. V." ist Teil des offiziellen Namens des iSAQB und steht für "eingetragener Verein", der seinen Status als juristische Person nach deutschem Recht beschreibt. Der Einfachheit halber wird iSAQB e. V. im Folgenden ohne die Verwendung dieser Abkürzung als iSAQB bezeichnet.

Verzeichnis der Lernziele

Einführung: Allgemeines zum iSAQB Advanced Level

Was vermittelt ein Advanced Level Modul?

Das Modul kann unabhängig von einer CPSA-F-Zertifizierung besucht werden.

- Der iSAQB Advanced Level bietet eine modulare Ausbildung in drei Kompetenzbereichen mit flexibel gestaltbaren Ausbildungswegen. Er berücksichtigt individuelle Neigungen und Schwerpunkte.
- Die Zertifizierung erfolgt als Hausarbeit. Die Bewertung und mündliche Prüfung wird durch vom iSAQB benannte Expert:innen vorgenommen.

Was können Absolventen des Advanced Level (CPSA-A)?

CPSA-A-Absolventen können:

- eigenständig und methodisch fundiert mittlere bis große IT-Systeme entwerfen
- in IT-Systemen mittlerer bis hoher Kritikalität technische und inhaltliche Verantwortung übernehmen
- Maßnahmen zur Erreichung von Qualitätsanforderungen konzeptionieren, entwerfen und dokumentieren sowie Entwicklungsteams bei der Umsetzung dieser Maßnahmen begleiten
- architekturrelevante Kommunikation in mittleren bis großen Entwicklungsteams steuern und durchführen

Voraussetzungen zur CPSA-A-Zertifizierung

- erfolgreiche Ausbildung und Zertifizierung zum Certified Professional for Software Architecture, Foundation Level® (CPSA-F)
- mindestens drei Jahre Vollzeit-Berufserfahrung in der IT-Branche; dabei Mitarbeit an Entwurf und Entwicklung von mindestens zwei unterschiedlichen IT-Systemen
    - Ausnahmen sind auf Antrag zulässig (etwa: Mitarbeit in Open-Source-Projekten)
- Aus- und Weiterbildung im Rahmen von iSAQB-Advanced-Level-Schulungen im Umfang von mindestens 70 Credit Points aus mindestens drei unterschiedlichen Kompetenzbereichen
- erfolgreiche Bearbeitung der CPSA-A-Zertifizierungsprüfung

Grundlegendes

Was vermittelt das Modul „FM“?

Das Modul präsentiert den Teilnehmerinnen und Teilnehmern Formale Methoden als … Am Ende des Moduls kennen die Teilnehmerinnen und Teilnehmer … und können …

Struktur des Lehrplans und empfohlene zeitliche Aufteilung

Inhalt	Empfohlene Mindestdauer (min)
1. Thema mit Einleitung	180
2. Thema über xz	150
3. Thema mit viel Theorie	120
4. Thema mit xy und Beispiel	180
5. Thema mit abc und d	210
6. Thema mit Abschlussbeispiel	120
Summe	960 (16h)

Dauer, Didaktik und weitere Details

Die unten genannten Zeiten sind Empfehlungen. Die Dauer einer Schulung zum Modul FM sollte mindestens **3** Tage betragen, kann aber länger sein. Anbieter können sich durch Dauer, Didaktik, Art und Aufbau der Übungen sowie der detaillierten Kursgliederung voneinander unterscheiden. Insbesondere die Art der Beispiele und Übungen lässt der Lehrplan komplett offen.

Lizenzierte Schulungen zu FM tragen zur Zulassung zur abschließenden Advanced-Level-Zertifizierungsprüfung folgende Credit Points) bei:

Methodische Kompetenz:	**10** Punkte
Technische Kompetenz:	**10** Punkte
Kommunikative Kompetenz:	**10** Punkte

Voraussetzungen

Teilnehmerinnen und Teilnehmer **sollten** folgende Kenntnisse und/oder Erfahrung mitbringen:

- Voraussetzung 1
- Voraussetzung 2, etc.

Hilfreich für das Verständnis einiger Konzepte sind darüber hinaus:

- Kenntnisgruppe 1:
  - Kenntnis 1
  - Erfahrung 2
  - Kenntnis 3

- Erfahrung 4
- Wissen 5

Gliederung des Lehrplans

Die einzelnen Abschnitte des Lehrplans sind gemäß folgender Gliederung beschrieben:

- **Begriffe/Konzepte**: Wesentliche Kernbegriffe dieses Themas.
- **Unterrichts-/Übungszeit**: Legt die Unterrichts- und Übungszeit fest, die für dieses Thema bzw. dessen Übung in einer akkreditierten Schulung mindestens aufgewendet werden muss.
- **Lernziele**: Beschreibt die zu vermittelnden Inhalte inklusive ihrer Kernbegriffe und -konzepte.

Dieser Abschnitt skizziert damit auch die zu erwerbenden Kenntnisse in entsprechenden Schulungen.

Ergänzende Informationen, Begriffe, Übersetzungen

Soweit für das Verständnis des Lehrplans erforderlich, haben wir Fachbegriffe ins iSAQB-Glossar aufgenommen, definiert und bei Bedarf durch die Übersetzungen der Originalliteratur ergänzt.

1. Dies ist der Titel des ersten Moduls

Dauer: XXX Min.	Übungszeit: XXX Min.

1.1. Begriffe und Konzepte

Begriff 1, Begriff 2, Begriff 3

1.2. Lernziele

LZ 1-1: Dies ist das erste Lernziel, in Kategorie xy

Hier wird beschrieben, was Teilnehmer:innen in diesem Lernziel lernen sollen. Das kann in Prosa-Text in ganzen Sätzen oder in Aufzählungen mit Unterpunkten erfolgen. Dabei kann auch unterschieden werden, wie wichtig einzelne Aspekte des Lernziels sind. Es kann hier bereits auf Literatur verwiesen werden.

- Erstes Teilziel
- Zweites Unterthema
- Dritter Aspekt

LG 1-5: Intuitionistic vs. Classical Logic

Understand the difference between intuitionistic and classical logic:

- Constructive vs. non-constructive proofs
- Axioms and inferences only admissible in classical logic (LEM, double negation elimination)
- Correspondence of intuitionistic logic to programmign and type systems

1.3. Referenzen

[Schöning 2008], [Troelstra and Schwichtenberg 2012], [Harrison 2009], [Fitting 1996], [Enderton 2001], [Ebbinghaus et al. 2021], [Gallier 2015]

2. Hier steht der Titel der zweiten Lerneinheit

Dauer: XXX Min.	Übungszeit: XXX Min.

2.1. Begriffe und Konzepte

Begriff 1, Begriff 2, Begriff 3

2.2. Lernziele

LZ 2-1: Lorem ipsum dolor sit amet, consectetur adipiscing elit

Hier wird beschrieben, was Teilnehmer:innen in diesem Lernziel lernen sollen. Das kann in Prosa-Text in ganzen Sätzen oder in Aufzählungen mit Unterpunkten erfolgen. Dabei kann auch unterschieden werden, wie wichtig einzelne Aspekte des Lernziels sind. Es kann hier bereits auf Literatur verwiesen werden.

- Erstes Teilziel
- Zweites Unterthema
- Dritter Aspekt

LZ 2-2: Hier ist ein zweites Lernziel in diesem Kapitel

tbd.

2.3. Referenzen

[ISO 24765], [Knuth 1997], [Milner 1973], [Nipkow 2014], [Paulson 1993]

3. Der Titel des dritten Moduls

Dauer: XXX Min.	Übungszeit: XXX Min.

3.1. Begriffe und Konzepte

Begriff 1, Begriff 2, Begriff 3

3.2. Lernziele

LZ 3-1: Dies ist das erste Lernziel in Kapitel 3, das mit xyz

Hier wird beschrieben, was Teilnehmer:innen in diesem Lernziel lernen sollen. Das kann in Prosa-Text in ganzen Sätzen oder in Aufzählungen mit Unterpunkten erfolgen. Dabei kann auch unterschieden werden, wie wichtig einzelne Aspekte des Lernziels sind. Es kann hier bereits auf Literatur verwiesen werden.

- Erstes Teilziel
- Zweites Unterthema
- Dritter Aspekt

LZ 3-2: Hier ist ein zweites Lernziel in diesem Kapitel

tbd.

3.3. Referenzen

[Aniculaesei et al. 2018], [Aniculaesei et al. 2021], [Ball 2000], [Boca et al. 2009], [Brinkmann et al. 2018], [Drechsler 2018], [Gnesi et al. 2013], [Klein et al. 2009], [Kuper 2017 a], [Kuper 2017 b], [Lehman 1980], [Leroy 2009], [Merz et al. 2008]

4. Viertes Modul, das ist sein Titel

Dauer: XXX Min.	Übungszeit: XXX Min.

4.1. Begriffe und Konzepte

Begriff 1, Begriff 2, Begriff 3

4.2. Lernziele

LZ 4-1: Dies ist das erste Lernziel in Kapitel 4, das mit xyz

Hier wird beschrieben, was Teilnehmer:innen in diesem Lernziel lernen sollen. Das kann in Prosa-Text in ganzen Sätzen oder in Aufzählungen mit Unterpunkten erfolgen. Dabei kann auch unterschieden werden, wie wichtig einzelne Aspekte des Lernziels sind. Es kann hier bereits auf Literatur verwiesen werden.

- Erstes Teilziel
- Zweites Unterthema
- Dritter Aspekt

LZ 4-2: Hier ist ein zweites Lernziel in diesem Kapitel

tbd.

4.3. Referenzen

[Claessen and Hughes 2002], [Chlipala 2022], [Nipkow 2014], [Kaufmann et al. 2000], [Stump 2016], [Bove et al. 2009], [Cousot 2021]

5. Beispiele

Dauer: XXX Min.	Übungszeit: XXX Min.

Dieser Abschnitt ist nicht prüfungsrelevant.

5.1. Begriffe und Konzepte

Innerhalb jeder lizenzierten Schulung muss mindestens ein Beispiel für FM vorgestellt werden.

Art und Ausprägung der vorgestellten Beispiele können von der Schulung bzw. den Interessen der Teilnehmer abhängen und werden seitens iSAQB nicht vorgegeben.

Referenzen

A

- [Aniculaesei et al. 2021] Aniculaesei Adina, Vorwald, Andreas, Zhang, Meng, and Rausch, Andreas. Architecture-based hybrid approach to verify safety-critical automotive system functions by combining data-driven and formal methods. In Cyrille Artho and Rudolf Ramler, editors, 2021 IEEE International Conference on Software Architecture Companion (ICSA-C), pages 139–148. IEEE, 2021.
- [Aniculaesei et al. 2018] Aniculaesei Adina, Howar, Falk, Denecke, Peer, and Rausch Andreas. Automated generation of requirements-based test cases for an adaptive cruise control system. In Cyrille Artho and Rudolf Ramler, editors, 2018 IEEE Workshop on Validation, Analysis and Evolution of Software Tests (VST@SANER), pages 11–15. IEEE, 2018

B

- [Baader et al. 2012] Baader F., Nipkow T. (2012). Term Rewriting and All That. Cambridge University Press. https://doi.org/10.1017/CBO9781139172752
- [Ball 2000] Ball Thomas and Rajamani Sriram: Checking Temporal Properties of Software with Boolean Programs, Workshop on Advances in Verification, 2000.
- [Barrett et al. 2018] Barrett, C., Tinelli, C. (2018). Satisfiability Modulo Theories. In: Clarke, E., Henzinger, T., Veith, H., Bloem, R. (eds) Handbook of Model Checking. Springer, Cham. https://doi.org/10.1007/978-3-319-10575-8_11
- [Berard et al. 2001] Bérard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., Schnoebelen, P., McKenzie, P. (2001): Systems and Software Verification - Model-Checking Techniques and Tools. Springer.
- [Biere et al. 2021] Biere, A., Heule, M., Van Maaren, H., Walsh, T. (2021). Handbook of Satisfiability (Second Edition): Volume 336 Frontiers in Artificial Intelligence and Applications. IOS Press.
- [Boca et al. 2009] Boca P., Bowen J. P., Siddiqi J. (2009). Formal Methods: State of the Art and New Directions. Springer London. https://doi.org/10.1007/978-1-84882-736-3
- [Bove et al. 2009] Ana Bove, Peter Dybjer and Ulf Norell (2009) . A Brief Overview of Agda - A Functional Language with Dependent Types. International Conference on Theorem Proving in Higher Order Logics.
- [Brinkmann et al. 2018] Brinkmann, R., Kelf, D. (2018). Formal Verification—The Industrial Perspective. In: Drechsler, R. (eds) Formal System Verification. Springer, Cham. https://doi.org/10.1007/978-3-319-57685-5_5

C

- [Chlipala 2022] Adam Chlipala (2022). Certified Programming with Dependent Types. MIT Press, 2022.
- [Clarke et al. 2018] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, Roderick Bloem (2018). Handbook of Model Checking. Springer, Cham. https://doi.org/10.1007/978-3-319-10575-8
- [Cousot 2021] Patrick Cousot. Principles of Abstract Interpretation. MIT Press, 2021.

D

- [Drechsler 2018] Drechsler, R. (2018). Formal System Verification - State-of the-Art and Future Trends. Springer Cham. https://doi.org/10.1007/978-3-319-57685-5

E

- [Ebbinghaus et al. 2021] Ebbinghaus Heinz-Dieter, Flum Jörg, Thomas Wolfgang (2021). Mathematical Logic. Springer. https://doi.org/10.1007/978-3-030-73839-6
- [Enderton 2001] Enderton, Herbert B. (2001). A Mathematical Introduction to Logic. Academic Press.

F

- [Fitting 1996] Fitting, Melvin (1996). First-Order Logic and Automated Theorem Proving. Springer. https://doi.org/10.1007/978-1-4612-2360-3

G

- [Gallier 2015] Gallier Jean H. (2015). Logic for Computer Science. Dover Publications.
- [Gnesi et al. 2013] Gnesi, S. Margaria, T. (2013). Formal Methods for Industrial Critical Systems: A Survey of Applications. IEEE. https://doi.org/10.1002/9781118459898
- [Grumberg et al. 2012] Grumberg, O., Nipkow, T., Hauptmann, B. (2012). Software Safety and Security: Tools for Analysis and Verification: Volume 33 of NATO Science for Peace and Security Series - D: Information and Communication Security. IOS Press.

H

- [Harrison 2009] Harrison, John (2009). Handbook of Practical Logic and Automated Reasoning. Cambridge University Press. https://doi.org/10.1017/CBO9780511576430

I

- [ISO 13568] ISO/IEC 13568:2002 Information technology — Z formal specification notation — Syntax, type system and semantics.
- [ISO 24765] ISO/IEC/IEEE 24765:2017. Systems and software engineering.

H

- [Claessen and Hughes 2002] Koen Claessen, John Hughes (2000). QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs. Proceedings of the ACM SIGPLAN International Conference on Functional Programming (ICFP).

K

- [Kaufmann et al. 2000] Matt Kaufmann, Panagiotis Manolios, and J Strother Moore (2000). Computer-Aided Reasoning: An Approach. Kluwer Academic Publishers. https://www.cs.utexas.edu/users/moore/publications/acl2-books/car/index.html
- [Klein et al. 2009] Klein, Gerwin, Elphinstone, Kevin, Heiser, Gernot, Andronick, June, Cock, David, Derrin, Philip, Elkaduwe, Dhammika, Engelhardt, Kai, Kolanski, Rafal, Norrish, Michael, Sewell, Thomas, Tuch, Harvey and Winwood, Simon. seL4: formal verification of an OS kernel. SOSP '09: Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles. October 2009, pages 207–220.
- [Knuth 1997] Knuth, D. (1997) The Art of Computer Programming, Volume 1: Fundamental Algorithms. 3rd edition. Addison-Wesley. https://www-cs-faculty.stanford.edu/~knuth/taocp.html

- [Kroening et al. 2017] Kroening D., Strichman O. (2017): Decision Procedures - An Algorithmic Point of View. Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-50497-0
- [Kuper 2017 a] Kuper, L. (2017): Proving that safety-critical neural networks do what they're supposed to: where we are, where we're going (part 1 of 2). https://decomposition.al/blog/2017/05/30/proving-that-safety-critical-neural-networks-do-what-theyre-supposed-to-where-we-are-where-were-going-part-1-of-2/
- [Kuper 2017 b] Kuper, L. (2017): Proving that safety-critical neural networks do what they're supposed to: where we are, where we're going (part 2 of 2). https://decomposition.al/blog/2017/05/31/proving-that-safety-critical-neural-networks-do-what-theyre-supposed-to-where-we-are-where-were-going-part-2-of-2/

L

- [Lamport 2022] Leslie Lamport (2022). Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley.
- [Lehman 1980] Meir Lehman (1980). Programs, Life Cycles, and Laws of Software Evoluation. Proceedings of the IEEE, Volume: 68, Issue: 9, September 1980.
- [Leroy 2009] Xavier Leroy: Formal verification of a realistic compiler, Communications of the ACM 52(7) 2009, pages 107—115.

M

- [Marques-Silva et al. 2018] Marques-Silva J., Malik S. (2018). Propositional SAT Solving. In: Clarke, E., Henzinger, T., Veith, H., Bloem, R. (eds) Handbook of Model Checking. Springer, Cham. https://doi.org/10.1007/978-3-319-10575-8_9
- [Merz et al. 2008] Merz S., Navet N. (2008). Modeling and Verification of Real-Time Systems: Formalisms and Software Tools. ISTE Ltd. https://doi.org/10.1002/9780470611012
- [Milner 1973] Milner, R. (1973) Models of LCF. Stanford Artifical Intelligence Laboratory Memo AIM-186.

N

- [Nipkow 2014] Nipkow, T., Klein, G. (2014) Concrete Semantics with Isabelle/HOL. Springer. http://www.concrete-semantics.org/

P

- [Paulson 1993] Paulson, Lawrence C. (1993). Isabelle: The Next 700 Theorem Provers. CoRR, cs.LO/9301106. https://arxiv.org/abs/cs/9301106

S

- [Schöning 2008] Schöning, Uwe (2008). Logic for Computer Scientists. Birkhäuser Boston. https://doi.org/10.1007/978-0-8176-4763-6
- [Stump 2016] Aaron Stump (2016). Verified Functional Programming in Agda. ACM.

T

- [Troelstra and Schwichtenberg 2012] Troelstra A. S., Schwichtenberg H. (2012). Basic Proof Theory.

Cambridge University Press. https://doi.org/10.1017/CBO9781139168717

W

- [Wayne 2018] Hillel Wayne (2018). Practical TLA+: Planning Driven Development. Apress.