

A1 Schneller starten

GraalVM for JDK 22

Die Oracle GraalVM ist eine Java-Laufzeitumgebung für Anwendungen, die in JavaScript, Python, Ruby, R oder mit den JVM-basierten Programmiersprachen Java, Scala, Kotlin, Clojure und den LLVM-basierten Sprachen C/C++ geschrieben wurden. Das ab März 2024 verfügbare Release GraalVM for JDK 22 bietet Performance-Verbesserungen und polyglotte Unterstützung für verschiedene Programmiersprachen, womit sie die Interoperabilität in einer gemeinsamen Laufzeitumgebung ermöglicht. Mit der Ahead-of-Time-Fähigkeit von GraalVM kompilierte Native Images haben eine optimierte Start-up-Zeit und verringern den Speicherverbrauch von JVM-basierten Applikationen. Die GraalVM kann wahlweise als Open Source Community Edition (CE) oder als optimierte Oracle GraalVM mit GFTC-Lizenz innerhalb der Java SE Universal Subscription verwendet werden.

In meinem Vortrag stelle ich die GraalVM-Architektur im Java-Ökosystem dar und erläutere die Einsatzgebiete, beispielsweise den in Java entwickelten C2 JIT Compiler oder die Verwendung von GraalVM Native Images. Beim GraalVM Native-Image-Build biete ich konkrete Empfehlungen zur Durchsatzverbesserung (Profile-Guided Optimizations) an und schlage sowohl eine maximale Heap-Größe mit vorbestimmter Speichernutzung als auch eine optimierte Prozessorarchitekturunterstützung vor. GraalVM kann für Spring-Boot- und Microservices-Frameworks verwendet werden.

A2 Den Pfad selbst beschreiten

Eine eigene Path-Implementierung in Java wagen

Es war einmal vor langer Zeit auf einer weit, weit entfernten Campus-Veranstaltung, dass eine rudimentäre Implementierung eines RAM-Filesystems skizziert wurde – und das noch in der Early-Access-Phase. Nach Jahren des mehr oder weniger intensiven Darauf-herum-Brütens hat es mich dazu getrieben, eigene Implementierungen des Filesystems zu wagen.

Dieser Vortrag zeigt, was konkret daraus wurde, wozu das überhaupt sinnvoll ist, welche lustigen Effekte es haben kann und wie sich die Ergebnisse in der produktiven Anwendung benehmen.

A3 Kotlin, oh so sweet

Syntaktischer Zucker in Kotlin

In diesem Vortrag geht es darum, wie Kotlin die Arbeit von Java-Entwicklern erleichtern kann. Ein besonderer Fokus wird dabei auf den syntaktischen Zucker in Kotlin gelegt, der den Code lesbarer und verständlicher macht. Im Vergleich zu Java bietet Kotlin viele nützliche Funktionen wie Null-Safety, automatische Typinferenz und die Möglichkeit der String-Interpolation. Diese Features sorgen dafür, dass der Code lesbarer und weniger überladen wird, wodurch er weniger fehleranfällig ist und die Entwickler produktiver arbeiten können.

Der Vortrag richtet sich primär an Java-Entwickler, die bisher wenig oder keine Erfahrung mit Kotlin haben, und zeigt auf, wie sie von diesem leistungsfähigen Sprach-Upgrade profitieren können.

A4 Alles im Fluss

Eine Einführung in Kotlin-Flows

Strukturierte Nebenläufigkeit und Coroutinen sind Begriffe, auf die man stößt, wenn man sich in Kotlin mit dem Thema Nebenläufigkeit auseinandersetzt. Aber es geht auch eine Stufe komfortabler. Aufsetzend auf den genannten Konzepten bietet Kotlin mit den asynchronen Flows eine einfache, leicht zu erlernende API, mit der ein Strom asynchron erzeugter Werte bereitgestellt bzw. verarbeitet werden kann. Auch wenn das einfache API-Design bei Kotlin-Flows klar im Vordergrund steht, sind sie konzeptionell reaktive Streams und lassen sich mit den entsprechenden Bibliotheken auch in die Welt der Reactive Streams oder RxJava integrieren.

Dieser Vortrag gibt – am konkreten Beispiel – eine Einführung in das Erzeugen und Verwenden von Kotlin-Flows. Er geht auf Themen wie Fehlerbehandlung und Flow-Abschluss ein und zeigt, wie hierbei sowohl imperative als auch deklarative Herangehensweisen unterstützt werden. Schließlich wird auch die Möglichkeit zur Integration in gängige reaktive Ansätze wie Reactive Streams oder RxJava diskutiert.

A5 Typen-Tango: Mit Eleganz und Präzision gegen Fehler tanzen

Type-Level-Programming

Typen sind ein Werkzeug zur Fehlerverhinderung, zur Beschreibung von Annahmen und zur Kommunikation mit anderen Entwicklern. Sie sind wie Tests, aber während Tests nur das Vorhandensein von Fehlern nachweisen, beweisen Typen tatsächlich die Abwesenheit von Fehlern. Machen wir einen Schritt aus unseren herkömmlichen Alltagssprachen wie Java/C# heraus und erkunden, was wir mit stärkeren Typsystemen erreichen könnten, in denen es möglich ist, Berechnungen rein auf der Typenebene zu beschreiben. Wie könnte das überhaupt aussehen? Welche Art von Problemen könnten wir lösen? Welche Kategorie von Fehlern könnten wir verhindern?

Dieser Vortrag zeigt anhand von ein wenig Geschichte, Beispielen und Live-Coding, wie das Type-Level-Programming entstanden ist, was es ist und wie wir es zu unserem Vorteil nutzen.

A6 Steinchenstapeln für Erwachsene

Mit Apache Jexl und Googles Blockly schnell und einfach mächtige No-Code-Geschäftsanwendungen erstellen

Low-Code und No-Code – nichts weniger als ein Paradigmenwechsel, dessen Bedeutung in den nächsten Jahren dramatisch wachsen wird. Höchste Zeit, sich mit dem Thema zu beschäftigen! Glücklicherweise findet sich im großen Java-Ökosystem Unterstützung dafür.

In meinem Vortrag demonstriere ich eine einfache Möglichkeit, wie mit Apache Jexl und Googles Blockly No-Code-Entwicklung in eine Anwendung integriert werden kann, die den Ansprüchen von Geschäftsanwendungen genügt. Backendseitig erstelle ich eine Spring-Boot-Anwendung, in der Apache Jexl als Laufzeitumgebung für den Output der No-Code-Entwicklung verwendet wird. Mit der Integration von Spring Beans in Jexl können wir den Funktionsumfang

beliebig erweitern. Als Frontend kommt eine Angular-Anwendung mit Integration von Blockly zum Einsatz. Damit können Programmbausteine einfach zusammengepuzzelt werden. Für die Abbildung domänenspezifischer Anforderungen zeige ich, wie Blockly individuell erweitert werden kann, sowohl fachlich-inhaltlich als auch optisch. Am Ende steht eine Anwendung, die wohl noch nicht produktiv genutzt werden wird, über ein „Hallo Welt“-Beispiel aber deutlich hinausgehen soll.

A7 Strippen ziehen

Nebenläufigkeit in modernen Programmiersprachen

Kotlin kennt Coroutinen, C# und Java kennen (virtuelle) Threads. Dart und JavaScript sind single-threaded, ermöglichen aber trotzdem eine asynchrone Programmausführung. Der Wunsch, Dinge gleichzeitig tun zu können, ist praktisch so alt wie das Programmieren selbst. Sobald es die Hardware hergegeben hat, wurden entsprechende Konzepte erdacht und umgesetzt. Und nein, Coroutinen wurden nicht für Kotlin erfunden.

In diesem Vortrag sehen wir uns moderne Umsetzungen an. Dabei geht es nicht um die Frage, welche Sprache es besser macht, sondern zum einen um Begriffserklärungen (was bedeuten Nebenläufigkeit und Asynchronität eigentlich?) und zum anderen um die Nutzung im Code: Was gehört zur Programmiersprache, was zur (Klassen-)Bibliothek?

A8 Aalgola vs. Java

Ein hinkender Vergleich

Während Java inzwischen alle halbe Jahre die coolsten Features ergänzt, will Aalgola einfach nicht fertig werden. Deshalb stellt sich unweigerlich die Frage: Lohnt sich das Warten auf Aalgola überhaupt?

In diesem Vortrag wird Aalgola mit Java verglichen, um zu zeigen, wo die Stärken von Aalgola liegen. Dabei werden auch die letzten Neuerungen – insbesondere die von Java 22 – berücksichtigt, um herauszustellen, wer eigentlich hinter wem herhinkt.

B1 Unlocking Potential

Using AI and generative AI for personalized learning

Can AI and generative AI improve knowledge work, knowledge management, and learning in organizations? In this talk, we will emphasize that the applications for knowledge workers, many executives, and learning and development professionals are indeed enormous.

Based on concrete use cases and applications we will illustrate how future solutions could look like – e. g. privacy compliant information processing and knowledge transfer according to the EU AI Act, while using the AI and generative AI capabilities of LLMs (Large Language Models) and providing customized learning materials by using a Smart Content Crawler. On the one hand, our talk will highlight the technical aspects of using and fine-tuning LLMs, combined with a small demonstration, and on the other hand, we will discuss the challenges that organizations face today and in the coming years – e. g. the urgent need for knowledge transfer in the face of retirement waves and the increasing individualization of learning needs. It is only through a balanced combination of cultural and technical aspects that the potential of AI for learning and knowledge in organizations can really unfold.

B2 Willkommen im Winterwonderland der Datenverarbeitung!

Snowflake leicht gemacht

Begleiten Sie uns auf einer Reise durch die Welt von Snowflake und erleben Sie, wie unkompliziert es sein kann, Daten in den unterschiedlichsten Formaten effektiv, effizient und skalierbar zu verarbeiten. Vom Import relationaler und semi-strukturierter Daten über die Aufbereitung bis hin zur Visualisierung mithilfe einer Streamlit-App zeigen wir Schritt für Schritt die Umsetzung in Snowflake. Lassen Sie sich von den Funktionalitäten inspirieren, die wir gemeinsam unterwegs entdecken werden!

B3 Chat with your data

Ein praxisorientierter Ansatz für interaktive Datenbankabfragen mithilfe von LLMs und PromptFlow

Dieser Vortrag führt durch den Entwicklungsprozess einer Anwendung, die LLMs (Large Language Models) und PromptFlow nutzt, um Benutzerfragen mithilfe von Datenbankabfragen zu beantworten. Mit PromptFlow – einem Python-Package mit Visual-Studio-Extension – können ausführbare Flows erstellt werden, die LLMs, Prompts und Python-Code verbinden. An diesem praxisorientierten Beispiel soll der gesamte Entwicklungszyklus von der Idee bis zur Umsetzung beleuchtet werden, inklusive der obligatorischen Irrwege beim Prompt Engineering.

B4 Sprung in den Vulkan

Was steckt hinter moderner Computergrafik?

OpenGL und Vulkan: zwei leistungsstarke APIs, die für die Darstellung von Computergrafik weit verbreitet sind. OpenGL bietet eine plattformunabhängige Lösung für die Entwicklung von 2D- und 3D-Grafikanwendungen, die breiten Einsatz u. a. in Spielen findet. Vulkan ist eine modernere Alternative zu OpenGL. Es wurde entwickelt, um die Leistung und Skalierbarkeit von Grafikanwendungen auf verschiedenen Plattformen zu verbessern. Welche API entspricht Ihren Anforderungen am besten?

Nach diesem Vortrag werden Sie es wissen!

B5 MAUI vs Flutter battle

Let's explore two powerful frameworks by creating identical to-do apps!

Microsoft's MAUI and Google's Flutter have emerged as leading contenders in the quest for efficient and seamless cross-platform development.

Through this session, we aim to provide a comprehensive understanding of the strength, weaknesses, and unique features of each framework, empowering you to make informed decisions when choosing the right tool for your projects. Key highlights and comparisons are: architecture and design principles; language choices (C# and Dart); app development step-by-step; API communication; UX and responsiveness; and last but not least deployment and maintenance.

B6 Log-Management mit Graylog

Eine Reise durch unsere Erfahrungen im Kundeneinsatz

Dieser Vortrag bietet einen umfassenden Einblick in unsere mehrjährige Nutzung von Graylog als zentrales Log-Management-Tool. Wir starten mit einer kurzen Einführung in Graylog und erläutern, warum wir uns für dieses Open-Source-System entschieden haben. Anhand konkreter An-

wendungsfälle beleuchten wir die verschiedenen Funktionen von Graylog, wie Log-Sammlung, -Suche und -Visualisierung. Anschließend skizzieren wir den Wandel unseres Setups über die Jahre, inklusive Herausforderungen, Lösungen und Infrastrukturveränderungen, und teilen dabei unsere Erfahrungen und Best Practices, um Sie bei der Implementierung und Nutzung von Graylog zu unterstützen. Zum Schluss gibt es noch einen Sneak Peek auf zukünftige Pläne und Entwicklungen.

B7 Endlich vereint

Gemeinsame Dokumentation für TypeScript und Vue.js erzeugen

Wie wichtig die Dokumentation in einem Projekt ist, zeigt der Umstand, dass Programmiersprachen und Bibliotheken oft ihre eigenen Dokumentationswerkzeuge mitbringen. So bieten beispielsweise sowohl TypeScript als auch Vue.js eigene Tools, mit denen sich die Quellen bzw. Komponenten dokumentieren lassen. Da deren Dokumentationen aber unabhängig voneinander erstellt werden, ist ein einheitlicher Zugriff nicht möglich. Dies ist aber für eine Projektdokumentation eher unbefriedigend. Sphinx wäre eine gute Lösung gewesen, wenn es TypeScript unterstützen würde; aber das funktioniert leider nur für Python.

Dieser Vortrag stellt drei Werkzeuge vor, mit denen sich aus verschiedenen Dokumentationen eine gemeinsame erstellen lässt.

B8 Aus einem Guss

Einheitliches Look-and-feel für heterogene Frontends

Wie kann man Frontends unterschiedlicher Anwendungen – und sogar mit unterschiedlichen Technologien entwickelt – so gestalten, dass sie wie aus einer Hand aussehen? Vor dieser Herausforderung stand mein Team vor einigen Jahren.

In diesem Vortrag zeige ich, wie wir ein Designsystem und eine Web-Component-Library aufgebaut haben, welche Tools und Technologien wir dafür einsetzen und was die Vor- und Nachteile dieses universellen Ansatzes sind. Dabei betrachten wir den gesamten Entwicklungsprozess vom Design einer Webkomponente über die Umsetzung in TypeScript, HTML und CSS bis hin zur Dokumentation in Storybook und die Einbindung in verschiedene Anwendungen.

C1 Immer schön sauber bleiben

Clean Architecture als Unterstützung für evolutionäre Architektur

Softwaresysteme werden immer komplexer und die Anforderungen an sie wandeln sich ständig. Evolutionäre Softwarearchitektur versucht, sich kontinuierlich und effizient an sich verändernde Rahmenbedingungen, Technologien und Umgebungen anzupassen. Die Ideen einer Clean Architecture sind wichtige Grundlagen für evolutionäre Architektur. Denn durch die klare Trennung von Verantwortlichkeiten, die Unabhängigkeit von Bibliotheken/Frameworks, die erhöhte Testbarkeit und Flexibilität bietet Clean Architecture eine robuste Struktur, um die Weiterentwicklung einfacher zu machen.

In diesem Vortrag diskutieren wir, wie diese beiden Ansätze den Weg für eine nachhaltige Softwareentwicklung ebnen, insbesondere wie sie Risiken bei Änderungen reduzieren, die Portabilität fördern und die Zusammenarbeit im Entwicklerteam verbessern.

C2 Cool, da weiß ich ja genau, was ich machen muss!

Software-Engineering mit Spezifikationen

Würde man Konzerte ohne Partitur komponieren oder Häuser ohne Bauzeichnung bauen? Und würde man sich für ein Haus, an dem man ständig an- und umbaut, eher weniger oder eher mehr Zeichnungen wünschen? Ausgerechnet iterative Projekte profitieren von präzisen Funktionsbeschreibungen. Wenn der User den Button „Kündigen“ drückt, dann macht das System ... Und diese drei Punkte sind in kontinuierlich wachsenden Systemen irgendwann nichts Triviales mehr, was jemand mal eben im Kopf hat, sondern etwas, das sich nur mühevoll aus Code oder alten Jira-Tickets ablesen lässt. Wenn das nirgends steht, wird jeder Change-Request zu einem Glücksspiel und einem unnötigen Stressfaktor.

Dieser Vortrag zeigt an kleinen Experimenten, welche Eigenschaften Spezifikationen für agile Umfelder haben müssen, z.B. in Hinblick auf Änderungsfreundlichkeit und Vokabular, und wo die Denkfallen liegen, wenn man sie einfach weglässt, um vermeintlich Aufwand zu sparen und schneller zu liefern.

C3 Mondgesteine und UIs

Mehr Mensch, weniger Maschine

Kennen Sie Projekte, die viele Stakeholder haben und noch mehr unterschiedliche Anforderungen? Ja? Und wissen Sie auch, was Mondgesteine mit hübschen Oberflächen gemein haben? Nein? Macht nichts, das werden Sie bei uns erfahren. Was wir Ihnen jetzt schon verraten: Für uns steht der Mensch im Mittelpunkt unserer Bemühungen. Das erfordert ein Umdenken von der Leitung bis zum Entwickelnden. Bei über 100 eigenen IT-Anwendungen und einer der größten IT-Landschaften Deutschlands eine echte Herausforderung.

Wir erklären, wie manche Projektanforderungen den Menschen aus dem Fokus nehmen. Wird dadurch der Projekterfolg zu einer Frage der Perspektive? Für uns ist eine UI für Menschen mit unterschiedlichen Bedürfnissen die Antwort. Vielleicht sind solche Benutzeroberflächen auch für Sie eine realistische Lösung und nicht nur sehr teures Mondgestein ...

C4 Eine für viele

Multi-Tenancy-Software

Mehrmandantenfähige Software erlaubt es einzelnen Mandanten, dieselbe Software zu benutzen, aber dabei trotzdem keinen Zugriff auf die Daten und Ressourcen eines anderen Mandanten zu erlangen. Oft sind Mandanten verschiedene Unternehmen. Sie sind nicht mit Benutzern zu verwechseln, die ebenfalls geschützte Ressourcen besitzen können; Mandanten haben aber oft mehrere Benutzer. Software-as-a-Service-Anwendungen werden meist als mehrmandantenfähige Software realisiert. Doch warum überhaupt? Welche Vor- und Nachteile ergeben sich daraus?

Wir schauen uns an, wie mehrmandantenfähige Anwendungen realisiert, wie Mandanten unterschieden und welche Strategien der Datenhaltung gewählt werden können. Am Ende dieses Vortrags wissen Sie auch, welche Herausforderungen möglicherweise auftreten und mit welchen Lösungsstrategien sie gemeistert werden können.

C5 Pedal to the Metal

Software unter der Haube

Ein Hoch auf moderne Softwareentwicklung mit all ihren Abstraktionen, Frameworks, Runtimes, Garbage-Collectors und vielen anderen Annehmlichkeiten, die uns das tägliche Leben leichter machen. Doch was machen wir, wenn auf den vielen Ebenen zwischen unserem Quellcode und der darunterliegenden Hardware doch mal irgendwo etwas lahm? Wie können wir überhaupt herausfinden, wie unser Quellcode tatsächlich ausgeführt wird?

Wir starten eine Reise hoch oben im Software-Stack und arbeiten uns Schicht für Schicht bis zur Hardware hinunter. Entlang des Weges schauen wir uns Werkzeuge und Hilfsmittel an, mit denen wir unsere Software auch unter der Haube besser verstehen und damit im Alltag schneller entwickeln, testen und warten können.

C6 Stirb langsam

Migration eines Legacy-Systems

Arbeiten Sie an einem Legacy-System, das schwer zu warten ist? Wollen Sie es modernisieren, aber Sie wissen nicht, wie? Gehen wir es gemeinsam an!

In meinem Vortrag gibt es keine Folien – stattdessen migrieren Sie ein Legacy-System als praxisnahes Beispiel. Sie setzen die neuen Anforderungen zusammen mit den anderen Teilnehmern spielerisch um. Dabei liegt der Fokus auf der Mikroarchitektur eines Teilsystems.

C7 Interpreting numbers using number interpreters

Eine Zeitreise durch programmierbare Einheiten vom Großen bis zum Kleinen

Dieser Vortrag liefert anhand von (vielen) Zahlen einen Überblick über die erstaunlichen Entwicklungen im Bereich der Mikroprozessoren, Speicher und insbesondere Mikrocontroller. Dabei geht es um Größenordnungen von Millimeterbruchteilen bis zu Tausenden von Kilometern.

Der Vortragende blickt zurück auf über 45 Jahre Erfahrung mit programmierbaren Einheiten, nimmt aber auch insbesondere die Entwicklung von Mikrocontrollern in den jüngst vergangenen Jahren in den Blick. Brandneue Eindrücke von der embedded Messe 2024 in Nürnberg werden in letzter Minute in den Vortrag eingearbeitet.

C8 No RISC, no fun

Entwerfen wir unsere eigene CPU!

Als Softwareentwickler sind wir vertraut mit dem Schreiben, Warten und Debuggen von Programmen. Doch haben wir uns schon einmal Gedanken darüber gemacht, wie das Gehirn des Computers – die CPU – den Code ausführt? Lassen Sie uns dazu in die Welt der Prozessorarchitekturen eintauchen!

Um die theorielastige Materie zu veranschaulichen, betrachten wir einen einfach gehaltenen Softcore. Das ist ein Prozessor, dessen Schaltung nicht „in Silizium gegossen“, sondern in einer Hardwarebeschreibungssprache wie z. B. VHDL modelliert ist. Das ermöglicht uns, die CPU auf einem FPGA, also einem Chip, in den man logische Schaltungen programmieren kann, in Betrieb zu nehmen. Nach und nach werden die einzelnen Komponenten des Cores besprochen und durch kleinere Erweiterungen ergänzt. Auf dem FPGA testen wir unseren Prozessor dann gegen kleine C++-Programme. Abschließend widmen wir uns leistungssteigern-

den Methoden, die auch in modernen CPUs zum Einsatz kommen. Hilfreich, jedoch keine Voraussetzung, sind grobe Vorkenntnisse in digitalen Schaltungen, VHDL, Assembler (RISC-V) und C++.

D1 2024: A Cloud Odyssey

Eventbasiertes Arbeiten in der Azure Cloud

Wie funktioniert eventbasiertes Arbeiten in der Azure Cloud? Auf welche Weise verbessern Events, Azure Service Bus und Azure Functions die Technologielandschaft? Welche Möglichkeiten bietet die Cloud-Eventverarbeitung und auf welche Weise fördert sie den Fortschritt in verschiedenen Anwendungsbereichen?

Die Antworten gibt's in unserem Vortrag.

D2 Events im Rampenlicht

Einführung von eventbasierter Architektur in heterogenen Landschaften

In einer Ära, in der Systeme zunehmend miteinander verschmelzen, wird der Fokus auf Skalierbarkeit, Erweiterbarkeit und Resilienz zur grundlegenden Melodie. Eventbasierte Architekturen, lange als bewährtes Architekturmodell bekannt, treten hier auf den Plan und erleichtern nicht nur die Integration von Systemen, sondern gestalten auch den Informationsaustausch harmonischer.

Dieser Vortrag stellt anhand eines Praxisbeispiels dar, welche Herausforderungen bei der Einführung von eventbasierten Architekturen in heterogenen IT-Landschaften auftreten und wie sie bewältigt werden können. Dabei geht es nicht nur um die Entkopplung von Systemen, sondern auch darum, bestehende Integrationen elegant in eine eventbasierte Architektur zu überführen und die Grenzen dieses Ansatzes zu beleuchten. Grundlegende Kenntnisse über Event-Broker-Systeme sowie über Software-Integration sind von Vorteil.

D3 Revitalizing Legacy Code: Approval Testing Unleashed

Erfahrungen aus der Legacy-App-Portierung und Einblicke in die Welt von Verify

Dieser Vortrag gewährt Einblicke in die erfolgreiche Migration einer Legacy-App von PHP nach C#, wobei die Anwendung von Approval-Tests eine Schlüsselrolle spielte. Im Fokus steht die nahtlose Integration bestehender Funktionalitäten. Neben einer klaren Definition und Anwendung von Approval-Tests als bedeutende Ergänzung zu Unit-Tests und Teststrategien wie TDD wird auch die Verify-Bibliothek als aktuelle Standardresource in diesem Kontext vertieft. Diese Bibliothek eröffnet nicht nur Lösungswege für Herausforderungen im aktuellen Projekt, sondern bietet auch vielfältige Möglichkeiten in kommenden Projekten.

Begleiten Sie mich auf dieser Reise durch die Code-Revitalisierung und lassen Sie sich vom Potenzial der Approval-Tests und der Verify-Bibliothek inspirieren.

D4 Master refactoring in C# .NET

How to elevate your codebase with proven strategies and expert techniques

Embark on a captivating exploration of C# .NET as we delve into the art of refactoring during this insightful talk. Elevate your coding skills with proven strategies and advanced techniques, ensuring your codebase becomes a masterpiece of maintainability, readability, and scalability. Uncover time-tes-

ted approaches to identifying and addressing code smells, optimizing performance, and embracing clean coding principles. Discover the collaborative power of refactoring tools and methodologies, fostering teamwork among development peers. Whether you're a seasoned developer or just beginning your coding journey, this speech equips you with knowledge to deliver robust, efficient, and elegant solutions. Refactor with confidence, embrace industry standards, and join me on a journey towards code excellence.

D5 Built it! Ansible im Projekteinsatz

In DevOps-Projekten deckt ein Team alle Aufgaben von der Programmierung der Funktionalität bis zum Betrieb ab. Ansible ist ein Infrastructure-as-Code-Werkzeug, mit dem man den Systemaufbau modelliert und dann automatisiert aufbaut. So kann auch ein kleines Team eine Systemlandschaft mit vielen Servern, Services und heterogenen Technologien warten und häufige Änderungen ausrollen. Wie andere Werkzeuge dieser Art schlägt auch Ansible eine Brücke zwischen Entwickler und Betrieb.

Wir beginnen im Vortrag mit den Grundelementen von Ansible wie Tasks, Playbooks, dem Inventory mit Host- und Gruppenvariablen und der Ausführung. Anschließend integrieren wir weitere Elemente, die für typische Aufgaben in der Praxis benötigt werden, wie Templating, Aufruf von APIs und Skripten, Integration mit Cloud-Provisionierung und Einbindung von Herstellertasks. Zum Abrunden besprechen wir die Erfahrung aus drei Projekten mit Ansible und die Auswirkung auf die Rollenverteilung im Team.

D6 Wenn's läuft, wo's nicht laufen soll Cloud und Application Drain

Anhand der Sicherheitslücke „ownCloud/Nextcloud Unprotected Data Directory“ geben wir Einblicke in die Risiken und Herausforderungen, denen Unternehmen in der heutigen Zeit bezüglich der Cybersicherheit gegenüberstehen. Im Rahmen einer Studie wurden über 921 Millionen Domains gescannt und Tausende verwundbare Instanzen identifiziert.

Wir präsentieren deren Ergebnisse, das Responsible-Disclosure-Verfahren, die technischen Hintergründe, die Umsetzung sowie das Deployment von Servern und Scannern. Auch Individualsoftware hat es nicht leichter bzw. macht es oft nicht besser. Wo und warum es trotz bekannter OWASP Top 10 für Web- und API-Security oft Schwachstellen gibt, stellen wir im zweiten Teil unseres Vortrags dar.

D7 Vernetzt und zugenäht Vom Siegeszug der Graphentechnologien

Spielen die Beziehungen zwischen Entitäten eine große Rolle, eignen sich zur Speicherung nicht immer klassische, sprich relationale Datenbanken. Hier kommen Graphdatenbanken ins Spiel. 2025 werden laut Prognosen Graphentechnologien in 80 Prozent der Daten- und Analyseinnovationen verwendet, im Vergleich zu 10 Prozent im Jahr 2021. Typische Anwendungsbeispiele sind etwa die Darstellung von Beziehungen in sozialen Netzwerken oder die Analyse von IT-Netzwerken.

In diesem Vortrag stelle ich die Zugriffsmöglichkeiten mittels Apache TinkerPop und dessen Sprache Gremlin auf die Azure Cosmos DB praktisch vor. Beispiele zeige ich sowohl in Java als auch in C#.

D8 Das führt doch alles zu nix! Ein Einblick in reproduzierbare Umgebungen und die aktuelle Begeisterung um nix

Wer in den letzten Jahren sein Ohr an den Gleisen der (digitalen) Bastler hatte, wird sehr wahrscheinlich zumindest einmal von nix oder nixos gehört haben. Obwohl es bereits über 20 Jahre alt ist, wächst das Interesse an nix aktuell merklich an, mit Anwendungsgebieten von Servermanagement über Programmkonfigurationen bis hin zu wiederverwendbaren Entwicklungsumgebungen.

In diesem Talk schauen wir uns die grundlegenden Ideen von nix an und warum man es überhaupt interessant finden sollte und nutzen möchte. Es wird minimales Vorwissen vorausgesetzt: Man sollte schon einmal etwas JSON gesehen und ein git-Repository geklont haben.

E1 Die letzten 50 Ähren einer Ära Nachruf auf eine geniale Erfindung

Vor 400 Jahren wurde der Rechenschieber erfunden, um das Rechnen mit Logarithmen zu vereinfachen. Seit dieser Zeit – und in immer stärkerem Maß seit Beginn des 19. Jahrhunderts – erstellten Ingenieure Konstruktionen mithilfe von Rechenschiebern. Hochhäuser, Brücken, Maschinen, Zeppeline, Fahrzeuge und Flugzeuge wurden damit berechnet. Einstein hat ihn benutzt, Konrad Zuse und Alan Turing wohl auch. Noch bei den Apollo-Mondflügen waren Rechenschieber mit an Bord. Nicht lange danach, Anfang der 1970er-Jahre, wurden diese wunderbaren Werkzeuge von elektronischen Rechenhilfsmitteln verdrängt.

In meinem Vortrag erkläre und zeige ich anhand von Originalen aus meiner Sammlung, wie man damit rechnete und wie die Skalen immer weiter spezialisiert und verfeinert wurden.

E2 Strategie statt Lotterie Wie man Innovation systematisch herbeiführt

Innovation wird in einem mystischen kreativen Moment geboren. Dieses Bild prägt unser Denken, wenn es um Erfindungen geht. Mit diesem Vorurteil möchten wir in unserem Vortrag brechen.

Wir zeigen, wie man als Unternehmen systematisch und effizient zu erfolgreichen Innovationen kommt – und wie man die gängigen Praktiken im Bereich Innovation auch als Software-Dienstleistungsunternehmen einsetzen kann. Wir teilen Erfolgsgeschichten, aber auch ehrlich und ungeschönt, wo es manchmal hakt und Innovation einem Glücksspiel gleicht.

E3 Leadership im Keller Von einem, der einzog, die Remote-Arbeit zu lernen

Erinnern Sie sich noch daran, morgens aufstehen und zur Arbeit fahren zu müssen? Ich auch nicht. Moment, es ist schon 13 Uhr? Zeit, mich umzuziehen: Pyjama aus, Polo-Shirt und Jogginghose an. Ja, ich arbeite zu 100 Prozent remote. 2021 habe ich mich dafür entschieden, eine Leadership-Rolle in einer nahezu 100-prozentigen Remote-Firma zu übernehmen. Selber 100 Prozent remote zu sein ist eine Herausforderung. Teil des Remote-Leaderships zu sein ist eine Hausnummer. In einer nahezu 100-prozentigen Remote-Firma zu arbeiten ist eine Achterbahnfahrt.

In diesem Vortrag bringe ich Ihnen zweieinhalb Jahre an Experimenten, Fehlversuchen, Erfolgen und Erfahrungen näher. Wir sprechen über Kellerbüros, Wohnzimmerbüros,

Schlafzimmerbüros, Terrassenbüros und Spazierbüros, aber auch über Distanz, Kommunikation, soziale Nähe, individuelle Bedürfnisse, Familienleben, Burn-out, Self-Discipline, Self-Care und ... noch viel mehr!

E4 Nicht von dieser Welt, sondern von meiner Worldbuilding oder wie man eine neue Welt erfindet

Wie erfindet man eine Welt? Wie benennt man einen Ort? Wo entstehen Städte? Was bringt Monarchien zu Fall? Wie versteckt man Magie vor Muggeln? Wie gestaltet man Aliens?

Vom Worldbuilding hängt es mitunter stark ab, ob wir uns in Büchern und Filmen zu Hause fühlen, ob wir in deren Welt eintauchen können – ob diese Welt vorstellbar ist. Möchte man selbst eine Welt erschaffen, kann man ihr eine Geschichte geben, Flora und Fauna, Zivilisationen erfinden, sie zu Fall bringen, Karten zeichnen und Legenden knüpfen. Sich für oder gegen Magie entscheiden. In die Zukunft oder die Vergangenheit reisen oder an Orte, an die noch kein Mensch je gedacht hat. Die eigene Fantasie ist die einzige Grenze.

Einige Konzepte des Worldbuildings und wie sie in Literatur, Fernsehen und Spielen angewendet werden, soll dieser Vortrag zeigen. Und das Publikum vielleicht sogar in fremde Welten entführen ...

E5 Wohnst du noch oder automatisierst du schon? Einführung in Heimautomatisierung und Homelabbing

Die fortschreitende Digitalisierung hat Heimautomatisierung zu einem zentralen Aspekt unseres täglichen Lebens gemacht.

Ich biete einen praxisorientierten Einstieg in die Welt der Heimautomatisierung unter Verwendung von Home Assistant, einer Open-Source-Plattform für Smart-Home-Lösungen, in Verbindung mit dem kostengünstigen Raspberry Pi. Mit diesem kleinen Computer kann man sein Haus automatisieren, um das Licht zu steuern, Heizpläne zu erstellen oder eine kostengünstige Alarmanlage zu bauen. Zusätzlich stelle ich verschiedene Dienste und Erweiterungen vor, die nahtlos in Home Assistant integriert werden können.

E6 Fokuspokus – concentratio expeditur! Konzentriertes Arbeiten ist keine Hexerei

In einer Zeit der ständigen Erreichb... oh, Moment, Anruf, muss ich kurz ran. So, wo waren wir? Ja richtig, unsere Aufmerksamkeit. Och nee, jetzt will die Chefin auch noch was. Ich brauch jetzt erst mal einen Kaffee.

So ähnlich kennen wir das alle: Meetings und Anfragen zerhacken den Arbeitstag und ständig wechselnde Prioritäten bestimmen die Agenda. Gefühlt findet Arbeiten nur noch zwischen zwei Unterbrechungen statt.

In dieser Session beleuchten wir, warum fokussiertes Arbeiten so wichtig ist, welche Umgebung wir dafür brauchen und was Sie und Ihre Führungskräfte dafür tun können. Nach der Session werden Sie wissen, was Fokus fördert und stört, warum unser Gehirn für fokussiertes Arbeiten geschaffen ist, wie sich mangelnder Fokus auf Wohlbefinden und Arbeitsergebnisse auswirkt, auf welchen Ebenen Störfaktoren auftreten und wie Sie für sich und Ihr Team ein besseres Umfeld für fokussiertes Arbeiten schaffen können.

E7 Kinder, es ist kompliziert ... Warum manche Dinge (in der IT) so sind, wie sie sind

Warum kann man unter Windows keinen Ordner mit dem Namen „con“ anlegen? Was ist der Unterschied zwischen CR, LF, CR+LF?

Diesen und weiteren unterhaltsamen Fragen für Jung und Alt geht der Vortrag nach. Und Achtung: Ältere Menschen kennen die Antwort möglicherweise oder haben sie längst verdrängt ...

E8 The Art of Caring Eine Untersuchung des Zusammenhangs zwischen sozialer Unterstützung und Gesundheit

Wir Menschen sind durch und durch soziale Wesen. Tausende von Studien bestätigen den Zusammenhang zwischen sozialer Unterstützung und der körperlichen wie psychischen Gesundheit von Erwachsenen und Kindern. Sollten wir also die gegenseitige Unterstützung nicht stets im Fokus haben? So leicht ist es aber nicht immer. Manchmal versuchen wir zu helfen, aber es scheint uns einfach nicht zu gelingen, das Richtige zu sagen oder zu tun. Warum ist das so?

In diesem Vortrag werden wir untersuchen, welche verschiedenen Arten sozialer Unterstützung existieren, wie sie sich sowohl auf den Unterstützer als auch auf den Empfänger auswirken kann und welche Ansätze es gibt, soziale Unterstützung zu fördern und zu verbessern.

M1 Kaffeeklatsch Neuigkeiten und Wissenswertes rund um die Softwareentwicklung

In dieser Keynote wird es – wie immer – im engeren und weiteren Sinne um das Thema Softwareentwicklung gehen, mit Klatsch und Tratsch über alles, was es verdient, in diesem Rahmen Erwähnung zu finden. Angereichert wird das Ganze mit einigen überraschenden Lightning-Talks.

M2 Doch nicht ganz so clever Publikumsdiskussion

Obwohl das vergangene Jahr einige bemerkenswerte Errungenschaften der künstlichen Intelligenz hervorbrachte, hat der Hype um die KI einen empfindlichen Dämpfer erhalten. Ein Large-Language-Model entpuppt sich jetzt doch eher als das, was es ist, nämlich „nur“ ein Sprachmodell. Es verzählt sich schon mal, lügt unwissentlich (dass sich die Balken biegen) und verkauft Programmierern Code, der gut aussieht, aber leider nicht das macht, was er machen soll.

Diese Publikumsdiskussion dreht sich darum, welche Rolle die KI beim Programmieren übernehmen soll, welchen Mehrwert ihr Einsatz bei der Softwareentwicklung bieten kann und was dies für Programmieranfänger bedeutet, – und beantwortet vielleicht auch die Frage, ob die KI uns arbeitslos macht.

M3 Campus Champions Cup Kickertunier

Wenn alles gut geht, kann dieses Jahr der Campus Champions Cup wieder mit drei Kickertischen stattfinden. Damit werden die Wartezeiten verkürzt, das Spielvergnügen gesteigert und die Spannung vervielfacht. Dem Kampf um die begehrten Campus-Trophäen in gewohnter Vielfalt steht damit nichts mehr im Wege. Möge das beste Team gewinnen!

M4 ... wo noch niemand zuvor gewesen ist

Fremde Welten im Film

Wenn man in der Milchstraße oder in fernen Galaxien unterwegs ist, stellt sich das Problem, dass meist noch niemand dort gewesen ist. Wie sieht es also in unserem Sonnensystem, in unserer Galaxis und an wirklich weit entfernten Orten aus? Und wie kommen wir da überhaupt zu Lebzeiten hin?

Dieser Filmvortrag beschäftigt sich damit, wie Filmmacher sich fremde Welten und das Reisen dorthin vorstellen. Dazu werfen wir zuerst einen kurzen Blick in unser Sonnen-

system, bevor wir uns in weiter entfernte Gefilde aufmachen, um uns einen Überblick über die dortige Flora und Fauna sowie fremde Gepflogenheiten zu verschaffen. Darüber hinaus machen wir uns ein paar Gedanken darüber, welche Hürden überwunden werden müssen, um möglichst schnell zu einem weit entfernten Ort zu kommen, und welche Überraschungen einen dabei möglicherweise erwarten. Aber: Keine Panik! Wir kehren auf jeden Fall wieder zur Erde zurück.

WORKSHOPS

W1 Ktor oder was?

REST-Services in Kotlin entwickeln

Kotlin ist als Programmiersprache immer häufiger das Mittel der Wahl. Doch wenn es darum geht, Kotlin im Backend einzusetzen – etwa um REST-APIs bereitzustellen –, greifen Entwickler häufig auf Spring bzw. Spring Boot als ergänzendes Framework zurück. Kotlin und Spring sind sicher ein gutes Team, aber es geht eben auch ohne diese zusätzliche Unterstützung.

In diesem Workshop lernen Sie anhand praktischer, am eigenen Rechner nachvollziehbarer Beispiele, wie Sie mit Ktor rein Kotlin-basierte HTTP-Webservices implementieren können. Dabei gehen wir sowohl auf grundlegende Themen wie Routing und Content-Negotiation als auch auf die Absicherung der Services, Unit-Testing und die Entwicklung eigener Ktor-Erweiterungen ein. Und auch das Thema Deployment kommt dabei nicht zu kurz.

W2 Lazy aufm Sofa

Daten mobiler Endgeräte mit Couchbase-Cluster syncen

Zeitversetztes Arbeiten an unterschiedlichen Orten auf verschiedenen Geräten – in Ihrem Unternehmen schon Realität?

In diesem Workshop stellen wir eine Möglichkeit vor, all dies entspannt umzusetzen: Couchbase überzeugt mit seinem flexiblen und freien Datenmodell, seiner horizontalen Skalierbarkeit in verteilten Umgebungen und seiner brillanten Performance. Wir binden gemeinsam eine Android-App an einen Couchbase-Cluster an und richten eine Synchronisierung ein. Ist das Mobilgerät offline, werden die Daten in einer lokalen Couchbase Lite abgelegt; besteht wieder eine Netzwerkverbindung, werden sie automatisch mit dem Cluster synchronisiert. Sammeln Sie praktische Erfahrungen mit dieser leistungsstarken NoSQL-Datenbank, die Datenhaltung und Synchronisation in der mobilen App-Entwicklung revolutioniert.

W3 Dürfen wir auch mitspielen?!

Deep Learning für Java-Entwickler

Schon zweimal in der nun gut 70-jährigen Geschichte der künstlichen Intelligenz gab es euphorische und erfolgversprechende Phasen, wovon jedoch jede in eine Art Resignation, ja gar einen „KI-Winter“ mündete. Ist es heute anders?

Einiges spricht dafür, dass es diesmal ein Durchbruch wird, der von Dauer ist: gigantische Rechenpower, verbunden über ein weltumspannendes Internet, und riesige Datenmengen von Myriaden Sensoren und anderen Quellen – all dies ist erst jetzt zum ersten Mal und gleichzeitig vorhanden.

Damit Java-Entwickler in der von Python, R, Tensorflow und Co. dominierten KI-Welt auch mitspielen und mitreden können, schauen wir uns DeepLearning4j an, das mit seiner Meilensteinversion 1.0.0-M2.1 kurz vor einer stabilen 1.0 steht und sich in Sachen Performance nicht zu verstecken braucht. In diesem Workshop lernen wir die Bibliothek kennen, schauen uns ein paar Anwendungen aus der realen Welt an, spielen mit den Konfigurationsparametern herum und überlegen uns schließlich neue Anwendungsmöglichkeiten.

W4 Leise rieselt der Schnee

Vom Datenimport bis zur Visualisierung in Snowflake

Willkommen im Winterwonderland der Datenverarbeitung! Wir ziehen uns symbolisch gesprochen die Fausthandschuhe an und entdecken in diesem Workshop hands-on, wie unkompliziert es sein kann, Daten in den unterschiedlichsten Formaten effektiv, effizient und skalierbar zu verarbeiten. Vom Import relationaler und semistrukturierter Daten über die Aufbereitung bis hin zur Visualisierung mithilfe einer Streamlit-App – Schritt für Schritt schreiten wir die Umsetzung in Snowflake ab. Sie werden überrascht sein von den Funktionalitäten, denen wir auf unserem Winterspaziergang begegnen!

W5 Ich hab Durst

Bier brauen

Wir brauen gemeinsam in kleinen Gruppen obergäriges Bier in der Gaststätte Thalmühle, unweit vom Erlanger Bahnhof. Neben der handwerklichen Tätigkeit werfen wir auch einen Blick auf die Theorie des Brauprozesses sowie auf die rechtlichen Aspekte. Wir starten mit einem Weißwurstfrühstück, arbeiten uns zu einer Brotzeit zu Mittag vor, um dann à la carte in der Thalmühle gemeinsam den Abend ausklingen zu lassen.

Mitzubringen sind festes, wasserfestes Schuhwerk – am besten Sicherheitsschuhe – und ein Gefäß bzw. Kanister mit einem Fassungsvermögen von ca. zehn Litern, um die Gärung und Reifung des Bieres daheim zu vollenden.



ZEIT	A	B	C	D	E
9:30 – 11:00	M1 Kaffeeklatsch Neuigkeiten und Wissenswertes rund um die Softwareentwicklung				
11:30 – 12:45	A1 Schneller starten GraalVM for JDK 22	B1 Unlocking Potential Using AI and generative AI for personalized learning	C1 Immer schön sauber bleiben Clean Architecture als Unterstützung für evolutionäre Architektur	D1 2024: A Cloud Odyssey Eventbasiertes Arbeiten in der Azure Cloud	E1 Die letzten 50 Ähren einer Ära Nachruf auf eine geniale Erfindung
14:00 – 15:15	A2 Den Pfad selbst beschreiten Eine eigene Path-Implementierung in Java wagen	B2 Willkommen im Winterwonderland der Datenverarbeitung! Snowflake leicht gemacht	C2 Cool, da weiß ich ja genau, was ich machen muss! Software-Engineering mit Spezifikationen	D2 Events im Rampenlicht Einführung von eventbasierter Architektur in heterogenen Landschaften	E2 Strategie statt Lotterrie Wie man Innovation systematisch herbeiführt
15:45 – 17:00	A3 Kotlin, oh so sweet Syntaktischer Zucker in Kotlin	B3 Chat with your data Ein praxisorientierter Ansatz für interaktive Datenbankabfragen mithilfe von LLMs und Prompt-Flow	C3 Mondgesteine und Uis Mehr Mensch, weniger Maschine	D3 Revitalizing Legacy Code: Approval Testing Unleashed Erfahrungen aus der Legacy-App-Portierung und Einblicke in die Welt von Verify	E3 Leadership im Keller Von einem, der einzog, die Remote-Arbeit zu lernen
17:30 – 18:45	A4 Alles im Fluss Eine Einführung in Kotlin-Flows	B4 Sprung in den Vulkan Was steckt hinter moderner Computergrafik?	C4 Eine für viele Multi-Tenancy-Software	D4 Master refactoring in C#.NET How to elevate your codebase with proven strategies and expert techniques	E4 Nicht von dieser Welt, sondern von meiner Worldbuilding oder wie man eine neue Welt erfindet
MITTAGESSEN					
20:15 – 21:30	M2 Doch nicht ganz so clever Publikumsdiskussion	M3 Campus Champions Cup Kickertunier	ABENDESSEN		
21:45 – 24:00	M4 ... wo noch niemand zuvor gewesen ist Fremde Welten im Film				



ZEIT	A	B	C	D	E
9:30 – 10:45	A5 Typen-Tango: Mit Eleganz und Präzision gegen Fehler tanzen Type-Level-Programming	B5 MAUI vs Flutter battle Let's explore two powerful frameworks by creating identical to-do apps!	C5 Pedal to the Metal Software unter der Haube	D5 Built it! Ansible im Projekteinsatz	E5 Wohnst du noch oder automatisierst du schon? Einführung in Heimauto- matisierung und Homelabbing
11:15 – 12:30	A6 Steinchenstapeln für Erwachsene Mit Apache Jexl und Googles Blockly mächtige No-Code- Geschäftsanwendungen erstellen	B6 Log-Management mit Graylog Eine Reise durch unsere Erfahrungen im Kundeneinsatz	C6 Stirb langsam Migration eines Legacy-Systems	D6 Wenn's läuft, wo's nicht laufen soll Cloud und Application Drain	E6 Fokuspokus – concentratio expeditur! Konzentriertes Arbeiten ist keine Hexerei
MITTAGESSEN					
13:45 – 15:00	A7 Strippen ziehen Nebenläufigkeit in modernen Programmiersprachen	B7 Endlich vereint Gemeinsame Dokumentation für TypeScript und Vue.js erzeugen	C7 Interpreting numbers using number interpreters Eine Zeitreise durch program- mierbare Einheiten vom Großen bis zum Kleinen	D7 Vernetzt und zugenäht Vom Siegeszug der Graphen- technologien	E7 Kinder, es ist kompliziert ... Warum manche Dinge (in der IT) so sind, wie sie sind
15:30 – 16:45	A8 Aalgola vs. Java Ein hinkender Vergleich	B8 Aus einem Guss Einheitliches Look-and-feel für heterogene Frontends	C8 No RISC, no fun Entwerfen wir unsere eigene CPU!	D8 Das führt doch alles zu nix! Ein Einblick in reproduzierbare Umgebungen und die aktuelle Begeisterung um nix	E8 The Art of Caring Eine Untersuchung des Zu- sammenhangs zwischen sozialer Unterstützung und Gesundheit