

Companion Proceedings of the 9th International Conference on the Art, Science, and Engineering of Programming

Programming 2025, June 2–6, 2025, Prague 1, Czechia

Edited by

Jonathan Edwards

Roly Perera

Tomas Petricek



Editors

Jonathan Edwards 

Independent, Boston, USA
jonathanmedwards@gmail.com

Roly Perera 

University of Cambridge, UK
University of Bristol, UK
roly.perera@dynamicspects.org

Tomas Petricek 

Charles University, Prague, Czechia
tomas@tomasp.net

ACM Classification 2012

Software and its engineering → Software notations and tools

ISBN 978-3-95977-382-9

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <https://www.dagstuhl.de/dagpub/978-3-95977-382-9>.

Publication date

September, 2025

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists all publications of this volume in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <https://portal.dnb.de>.

License

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0): <https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/OASlcs.Programming.2025.0

ISBN 978-3-95977-382-9

ISSN 1868-8969

<https://www.dagstuhl.de/oasics>

OASlcs – OpenAccess Series in Informatics

OASlcs is a series of high-quality conference proceedings across all fields in informatics. OASlcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Editorial Board

- Daniel Cremers (TU München, Germany)
- Barbara Hammer (Universität Bielefeld, Germany)
- Marc Langheinrich (Università della Svizzera Italiana – Lugano, Switzerland)
- Dorothea Wagner (*Editor-in-Chief*, Karlsruher Institut für Technologie, Germany)

ISSN 1868-8969

<https://www.dagstuhl.de/oasics>

■ Contents

Preface

Jonathan Edwards, Roly Perera, and Tomas Petricek 0:ix

9th International Workshop on Modern Language Runtimes, Ecosystems, and VMs (More VMs 2025)

Filip Křikava and Stefan Marr 0:xi–0:xii

11th International Workshop on Programming Experience (PX/25)

Luke Church, Richard P. Gabriel, Hidehiko Masuhara, and Robert Hirschfeld 0:xiii–0:xv

3rd Quantum Programming workshop (QP 2025)

Shinobu Saito and Jose García-Alonso 0:xvii–0:xviii

4th Workshop on Value Independence in Modern Programming Languages (VIMPL 2025)

Nguyen Pham and Dimi Racordon 0:xix–0:xx

9th Symposium on Computer Languages, Implementations and Tools (SCLIT 2025)

Gordana Rakić and Nicolás Cardozo 0:xxi–0:xxii

9th International Workshop on Modern Language Runtimes, Ecosystems, and VMs (More VMs 2025)

Comparing R Bytecode Compilers Written in R, Java, and Rust

Pierre Donat-Bouillud, Filip Křikava, Jakob Hain, Adam Plodek, and Jan Vitek . 1:1–1:6

Towards a Java Virtual Machine for Processing-In-Memory

Kazuki Ichinose, Shigeyuki Sato, and Tomoharu Ugawa 2:1–2:5

A Pragmatic Approach to Replay Compilation

Andrej Pečimúth, David Leopoldseder, and Petr Tůma 3:1–3:4

Debugging a Smalltalk VM Assisted by Large Automated Reasoning

Boris Shingarov and Jan Vraný 4:1–4:6

11th International Workshop on Programming Experience (PX/25)

It's OK to Want to Have a Good Time

Luke Church and Mariana Marasoiu 5:1–5:4

Exploration and Complexity Management in Graph-Based Programming Environments

Max Boksem and L. Thomas van Binsbergen 6:1–6:18

In-Situ Visual Programming

Ulrich Brandstätter and Bernhard Schenkenfelder 7:1–7:11

Fuzzing as Editor Feedback

Marcel Garus, Jens Lincke, and Robert Hirschfeld 8:1–8:15

Companion Proceedings of the 9th International Conference on the Art, Science, and Engineering of Programming (Programming 2025).

Editors: Jonathan Edwards, Roly Perera, and Tomas Petricek

OpenAccess Series in Informatics



ASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

COP Layer Encapsulating Non-Functional Requirements for Physical Systems on Hakoniwa Environment <i>Yudai Yamada, Nobuhiko Ogura, Kenji Hisazumi, and Harumi Watanabe</i>	9:1–9:10
Shica – Improving the Programming Experience for Agent-Based, Distributed, Physical Computing Systems <i>Hiroto Shikada and Ian Piumarta</i>	10:1–10:22
Renkon-Pad: A Live and Self-Sustaining Programming Environment Based on Functional Reactive Programming <i>Yoshiki Ohshima, Adam Bouhenguel, and Matthew Good</i>	11:1–11:20
Im-C – A Memory-Safe C Interpreter Providing a Better Learning, Testing, and Debugging Experience <i>Masaki Kunii and Ian Piumarta</i>	12:1–12:18
Dimensions of Examples: Toward a Framework for Qualifying Examples in Programming <i>Toni Mattis, Lukas Böhme, Stefan Ramson, Tom Beckmann, Martin C. Rinard, and Robert Hirschfeld</i>	13:1–13:11
PShapeTrace: Linking Drawing Instructions with Visual Outcomes in Processing <i>Takashi Ishio and Yuta Yamasaki</i>	14:1–14:12
Encouraging Experimentation Through Programming by Proximity <i>Tom Beckmann, Leonard Geier, Stefan Ramson, Marcel Taeumel, and Robert Hirschfeld</i>	15:1–15:15

3rd Quantum Programming workshop (QP 2025)

Exploring a Quantum Programming Language with Concurrency <i>Manisha Jain, Vitor Fernandes, Alexandre Madeira, and Luís S. Barbosa</i>	16:1–16:9
Locus: A Proposal for Quantum Software Composition <i>Javier Zayas Gallardo, Francisco Chicano, Carlos Canal, and Juan Manuel Murillo</i>	17:1–17:10
Quantum Table: A Tangible Quantum Circuit Demonstrator <i>Stefan Hillmich, Raphael Zefferer, Mathias Gartner, Bernhard Schenkenfelder, Sonja Bruckner, and Ulrich Brandstätter</i>	18:1–18:3

9th Symposium on Computer Languages, Implementations and Tools (SCLIT 2025)

Is There Hypothesis for Attribute Grammars? <i>Emanuel Rodrigues, José Nuno Macedo, and João Saraiva</i>	19:1–19:15
On the Effectiveness of Interpreter-Guided Compiler Testing <i>Federico Lochbaum and Guillermo Polito</i>	20:1–20:15
A Comparison of Three Program Query Languages to Detect Python Programming Misconceptions <i>Quentin Colla, Kim Mens, and Julien Liénard</i>	21:1–21:15

Identifying Security Issues in Elixir Web Applications <i>Smiljana Knežev, István Bozó, and Melinda Tóth</i>	22:1–22:15
The Pyttern Program Query Language <i>Julien Liénard, Kim Mens, and Siegfried Nijssen</i>	23:1–23:15
4th Workshop on Value Independence in Modern Programming Languages (VIMPL 2025)	
Toward a Typed Intermediate Language for R <i>Mickaël Laurent, Jakob Hain, Filip Krikava, Sebastián Krynski, and Jan Vitek</i> ...	24:1–24:4
Who Owns the Contents of a Doubly-Linked List? <i>Dimi Racordon</i>	25:1–25:10
Mutable Value Semantics Through a Runtime-Enforced Framework in Scala <i>Hamza Remmal</i>	26:1–26:3

Preface

Welcome to the Companion Proceedings for the 9th International Conference on the Art, Science, and Engineering of Programming, «Programming» 2025, a conference focused on programming topics. «Programming» seeks papers that advance our knowledge of programming on any relevant topic, including programming practice and the experience of programming. This ninth edition was held in Prague, Czech Republic, 2-6 June 2025.

To help build a community and foster an environment where participants can exchange ideas and experiences related to practical software development, «Programming» hosts several co-located events. In the 2025 edition, there were five workshops, one symposium and one tutorial. The workshops included the 9th International Workshop on Modern Language Runtimes, Ecosystems, and VMs (MoreVMs 2025), the 11th edition of the Programming Experience Workshop (PX/25), the 3rd edition of the Quantum Programming workshop (QP 2025), the 1st edition of a workshop on Software Substrates (Substrates 2025), and the 4th Workshop on Value Independence in Modern Programming Languages (VIMPL 2025). The symposium was the 9th Symposium on Computer Languages, Implementations and Tools (SCLIT 2025). The topic of the tutorial, Effekt, was Lexical Effect Handlers In Action.

This companion gathers all the proceedings for the workshops and the symposium. Substrates 2025 and Effekt had no formal proceedings, and so are not represented in this companion. We are grateful to the OASICS team, especially Michael Wagner and Michael Didas, for their diligence and assistance, to the «Programming» 2025 Organising and Steering Committees for their hard work organising the conference and satellite events. Finally we would like to express our gratitude to the authors and reviewers of all contributed and invited papers and talks, and of course to everyone who physically or virtually participated in the conference.

Jonathan Edwards, Roly Perera and Tomas Petricek
Conference Companion Editors

August 2025

■ 9th International Workshop on Modern Language Runtimes, Ecosystems, and VMs (More VMs 2025)

Welcome to the proceedings of the 9th Workshop on Modern Language Runtimes, Ecosystems, and VMs (MoreVMs 2025), which was co-located with «Programming» 2025. It took place on June 2, 2025 in Prague, Czechia. MoreVMs aims to bring together industrial and academic programmers to discuss the design, implementation, and usage of modern languages and runtimes. This includes all aspects of language runtimes, modular implementation, compilation strategies, and the interactions with operating systems and modern hardware architectures.

This year's workshop continued to encourage informal and diverse discussions on how languages and runtimes are currently being utilized, and where they need to improve further. Presentation abstracts discussing experiences, work-in-progress, as well as future visions, from either an academic or industrial perspective were welcomed. They could be either in the form of short papers, or blog posts, or brief talk abstracts. Each submission was reviewed by three members of the program and organizing committees. The Program Committee was selected to have an equal number of academic and industrial language implementers. We received 8 submissions, which were all accepted for presentation.

In addition to our reviewed contributions, MoreVMs 2025 had two invited talks. Karel Zikmund was “Diving into .NET internals” with us and Björn Franke presented his ongoing work on “JIT Compilers – Addressing Bottlenecks in Throughput and Latency”, where he explored how we can improve startup and avoid redundant compilation in the cloud.

We would like to thank the authors and presenters for their submissions and presentations, the invited speakers for sharing their work and insights with us, and of course our program committee for their constructive reviewing.

We hope that you will find this workshop companion thought-provoking, and that you will consider discussing the work with its authors and join us at MoreVMs 2026 next year in Munich.

August 2025

Filip Křikava and Stefan Marr
MoreVMs 2025 Program Co-Chairs

MoreVMs 2025 Invited Talks

Diving into .NET internals

Karel Zikmund, Microsoft, CZ

JIT Compilers - Addressing Bottlenecks in Throughput and Latency

Björn Franke, University of Edinburgh, UK

MoreVMs 2025 Accepted Submissions

Adding garbage collection to our Rust-based interpreters with MMTk

Octave Larose

Companion Proceedings of the 9th International Conference on the Art, Science, and Engineering of Programming (Programming 2025).

Editors: Jonathan Edwards, Roly Perera, and Tomas Petricek

OpenAccess Series in Informatics



ASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A Pragmatic Approach to Replay Compilation

Andrej Pečimúth, David Leopoldseder, Petr Tuma

BYOPL: Building a Bytecode-Interpreted Language on Truffle in a University Setting

Jens Lincke, Tim Felgentreff, Fabio Niephaus, Robert Hirschfeld

Comparing R bytecode compilers written in R, Java, and Rust

Pierre Donat-Bouillud, Filip Křikava, Jakob Hain, Adam Plodek, Jan Vitek

Debugging a Smalltalk VM assisted by large automated reasoning

Boris Shingarov, Jan Vraný

The V8 Sandbox

Samuel Groß, Michael Lippautz, Hannes Payer, Toon Verwaest, Igor Sheludko

Towards a Java Virtual Machine for Processing-in-Memory

Kazuki Ichinose, Shigeyuki Sato, Tomoharu Ugawa

Writing Tiny Programming Systems

Tomas Petricek

MoreVMs 2025 Program Committee

Saam Barati, Epic Games, USA

Daniele Bonetta, VU Amsterdam, Netherlands

Elias Castegren, Uppsala University, Sweden

Matthew Gaudet, Mozilla, Canada

Jens Lincke, Hasso Plattner Institute, Germany

Michael Lippautz, Google, Germany

Mark Stoodley, IBM, Canada

Ben L. Titzer, Carnegie Mellon University, USA

Adam Welc, Mysten Labs, USA

Maria Xekalaki, University of Manchester, UK

MoreVMs 2025 Co-Chairs

Filip Křikava, Czech Technical University in Prague, Czechia

Stefan Marr, University of Kent, UK

■ 11th International Workshop on Programming Experience (PX/25)

Some programming feels fun, other programming feels annoying. Why?

For a while now the study of programming has forced improvements to be described through the Fordist lens of usability and productivity, where the thing that matters is how much software can get built, how quickly.

But along the way, something has gone missing. What makes programmers feel the way they do when they're programming? It's not usually fun to spend an age doing something that could have been done easily, so efficiency and usability still matter, but they're not the end of the story.

Some environments, activities, contexts, languages, infrastructures make programming feel alive, others feel like working in a bureaucracy. This is not purely technologically determined, writing Lisp to do your taxes probably still isn't fun, but it's also not technologically neutral, writing XML to produce performance art is still likely to be `<bureaucratic></bureaucratic>`.

Whilst we can probably mostly agree about what isn't fun, what remains more personal and without a space within the academy to describe it.

PX set its focus on questions like: Do programmers create text that is transformed into running behavior (the old way), or do they operate on behavior directly ("liveness"); are they exploring the live domain to understand the true nature of the requirements; are they like authors creating new worlds; does visualization matter; is the experience immediate, immersive, vivid and continuous; do fluency, literacy, and learning matter; do they build tools, meta-tools; are they creating languages to express new concepts quickly and easily; and curiously, is joy relevant to the experience?

PX also covers the experience that programmers have. What makes it and what breaks it? For whom? What can we build to share the joy of programming with others?

PX/25 was the eleventh edition of PX. Participants met onsite in Prague or joined remotely, authors presented their work in sessions following the Writers' Workshop structure, and everyone engaged in lively discussions.

We would like to thank our program committee, all workshop attendees, and most importantly our authors for their contributions, constructive criticism, hard work, and willingness to share their ideas.

– Luke Church, Richard P. Gabriel, Hidehiko Masuhara, and Robert Hirschfeld

Contributions

Papers and Talks

COP Layer Encapsulating Non-Functional Requirements for Physical Systems on Hakoniwa Environment

Yudai Yamada, Nobuhiko Ogura, Kenji Hisazumi, and Harumi Watanabe

Dimensions of Examples: Toward a Framework for Qualifying Examples in Programming

Toni Mattis, Lukas Böhme, Stefan Ramson, Tom Beckmann, Martin C. Rinard, and Robert Hirschfeld

Companion Proceedings of the 9th International Conference on the Art, Science, and Engineering of Programming (Programming 2025).

Editors: Jonathan Edwards, Roly Perera, and Tomas Petricek

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Encouraging Experimentation Through Programming by Proximity

Tom Beckmann, Leonard Geier, Stefan Ramson, Marcel Taeumel, and Robert Hirschfeld

Exploration and Complexity Management in Graph-Based Programming Environments

Max Boksem and L. Thomas van Binsbergen

Fuzzing as Editor Feedback

Marcel Garus, Jens Lincke, and Robert Hirschfeld

Im-C – A Memory-Safe C Interpreter Providing a Better Learning, Testing, and Debugging Experience

Masaki Kunii and Ian Piumarta

In-Situ Visual Programming

Ulrich Brandstätter and Bernhard Schenkenfelder

It's OK to Want to Have a Good Time

Luke Church and Mariana Marasoiu

PShapeTrace: Linking Drawing Instructions with Visual Outcomes in Processing

Takashi Ishio and Yuta Yamasaki

Renkon-pad: A Live and Self-Sustaining Programming Environment Based on Functional Reactive Programming

Yoshiki Ohshima, Adam Bouhenguel, and Matthew Good

Shica – Improving the Programming Experience for Agent-Based, Distributed, Physical Computing Systems

Hiroto Shikada and Ian Piumarta

Presentations and Demos*Sharing Test Suites Across Programming Languages*

Patrick Rein, Peter Retzlaff, Jens Lincke, and Robert Hirschfeld

Programming with Rectangles: The Hidden Power of Tabular Thinking

David Thomas

Program Committee

- Luke Church, University of Cambridge and Candela Inc, United Kingdom
- Tao Dong, Google, United States
- Jonathan Edwards, Independent, United States
- Richard P. Gabriel, Poet, U.S. Outlying Islands
- Robert Hirschfeld, Hasso Plattner Institute and University of Potsdam, Germany
- Takashi Ishio, Future University Hakodate, Japan
- Eva Krebs, Hasso Plattner Institute and University of Potsdam, Germany
- Raula Gaikovina Kula, The University of Osaka, Japan
- Jens Lincke, Hasso Plattner Institute and University of Potsdam, Germany

- Mariana Marasoiu, University of Cambridge, United Kingdom
- Hidehiko Masuhara, Institute of Science Tokyo, Japan
- Leonel Merino, Pontificia Universidad Católica de Chile, Chile
- Yoshiki Ohshima Independent Contractor, Shizuoka University United States
- Michael Perscheid, SAP Innovation Center Potsdam, Germany
- Ian Piumarta, Kyoto University of Advanced Science, Japan
- David C. Shepherd, Louisiana State University, United States
- Emma Söderberg, Lund University, Sweden
- Marcel Taeumel, Hasso Plattner Institute and University of Potsdam, Germany
- Steven Tanimoto, University of Washington, Seattle, United States
- Dave Thomas, Bedarra Corp, Canada
- Allen Wirfs-Brock, Wirfs-Brock Associates, United States

Organizers

- Luke Church, University of Cambridge and Candela Inc, United Kingdom
- Richard P. Gabriel, Poet, U.S. Outlying Islands
- Robert Hirschfeld, Hasso Plattner Institute and University of Potsdam, Germany
- Hidehiko Masuhara, Institute of Science Tokyo, Japan

Web

- <http://programming-experience.org/px25/>
- <https://2025.programming-conference.org/home/px-2025>

■ 3rd Quantum Programming workshop (QP 2025)

Quantum programming is advancing rapidly, but it also raises questions: What makes a good programming experience in the quantum domain? How do we design languages, tools, and interfaces that are powerful and easy to use?

At QP 2025, we saw these questions addressed from a variety of perspectives. Contributions to the workshop ranged from proposals for new programming abstractions and language features, to explorations of software composition and modularity, to innovative interfaces and tools aimed at making quantum computing more accessible. Together, they highlighted both the theoretical underpinnings and practical challenges of designing environments in which quantum programming can be expressive, maintainable, and engaging.

Our invited talk, “Toward Practical Quantum Computer Systems: A Computer Architect’s Perspective”, by Professor Ilkwon Byun of Kyushu University (Japan) highlighted the architectural challenges in the practical application of quantum systems and the importance of integrating hardware, runtime, and programming abstractions.

Not only technical advances, but also common themes emerged from these contributions. Concurrency and coordination, abstraction and composition, as well as accessibility and education, emerged as key concerns for the future of quantum programming.

In the discussion and closing session, participants reflected on how to bring these threads together. There was broad consensus that quantum programming research must continue to strike a balance between rigor and utility, developing accurate semantics and system models, while creating environments that are easy to learn, engaging, and productive for programmers. The exchange was lively and participants imagined how languages, tools and educational platforms might evolve as quantum computing matures.

QP 2025 demonstrated that while the technology is still developing, the community is beginning to chart the terrain: what makes quantum programming effective, what makes it difficult, and what could make it rewarding.

We would like to thank our program committee, all workshop attendees and, most importantly, our authors for their contributions, constructive criticism, hard work, and willingness to share their ideas.

QP 2025 Invited Talk

Toward Practical Quantum Computer Systems: A Computer Architect’s Perspective

Ilkwon Byun, Kyushu University (Japan)

QP 2025 Accepted Submissions

Exploring a quantum programming language with concurrency

Manisha Jain, Vitor Fernandes, Alexandre Madeira, Luís Barbosa

Locus: A Proposal for Quantum Software Composition

Javier Zayas Gallardo, Francisco Chicano, Carlos Canal, Juan Manuel Murillo

Quantum Table: A Tangible Quantum Circuit Demonstrator

Stefan Hillmich, Raphael Zefferer, Mathias Gartner, Bernhard Schenkenfelder, Sonja Bruckner,

Companion Proceedings of the 9th International Conference on the Art, Science, and Engineering of Programming (Programming 2025).

Editors: Jonathan Edwards, Roly Perera, and Tomas Petricek

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Ulrich Brandstätter

QP 2025 Program Committee

- Rui Abreu, Faculty of Engineering of the University of Porto (Portugal)
- Shaukat Ali, Simula Research Laboratory and Oslo Metropolitan University (Norway)
- Paolo Arcaini, National Institute of Informatics (Japan)
- Johanna Barzen, University of Stuttgart (Germany)
- Carmelo R. Cartiere, University of Oxford (United Kingdom)
- Sebastian Feld, Delft University of Technology (Netherlands)
- Fuyuki Ishikawa, National Institute of Informatics (Japan)
- Wolfgang Mauerer, OTH Regensburg / Siemens AG (Germany)
- Andriy Miranskyy, Toronto Metropolitan University (Canada)
- Jaroslaw Adam Miszczak, Institute of Theoretical and Applied Informatics, Polish Academy of Sciences (Poland)
- Juan Manuel Murillo, University of Extremadura (Spain)
- Hausi Müller, University of Victoria (Canada)
- Mario Piattini, University of Castilla-La Mancha (Spain)
- Luís Soares Barbosa, Universidade do Minho (Portugal)
- Robert Wille, Technical University of Munich & SCCH GmbH (Germany)
- Manuel Wimmer, JKU Linz (Austria)
- Nengkun Yu, Stony Brook University, (USA)
- Tao Yue, Beihang University (China)
- Lei Zhang, University of Maryland Baltimore County (USA)
- Jianjun Zhao, Kyushu University (Japan)

QP 2025 Organizing Committee

- Jose Garcia-Alonso, University of Extremadura (Spain)
- Shinobu Saito, NTT (Japan)

■ 4th Workshop on Value Independence in Modern Programming Languages (VIMPL 2025)

Welcome to the 4th edition of International Workshop on Value Independence in Modern Programming Languages (VIMPL 2025), held on June 3rd at Charles University, Czechia, co-located with «Programming» 2025. Value independence upholds the ability to reason locally about semantics of variables, without sacrificing the performance benefits of direct mutation. With the recent re-emergence of research on safe mutation, especially in a concurrency context, the workshop provides a forum for researchers and practitioners to discuss value independence as a theme in the user model of programming languages, its use in software applications, its use in compilers and interpreters for optimization, and the challenges related to its interaction with other modern programming language features.

In this year’s workshop, we received three submissions in the form of one full paper and two extended abstracts, with a strong focus on progress and challenges as well as techniques to bring reasoning about value independence to both existing and emerging languages. All were accepted for publication and presentation.

In addition to our reviewed contributions, we were delighted to have one invited keynote from Tobias Wrigstad on “Fearless Concurrency in Python”. Through various development from the Verona programming language to the Lungfish dynamically-check ownership model, the talk demonstrated the progress and challenges to bring performant-yet-idiomatic multi-threading to the Python programming language.

We would like to thank the authors and presenters for their submissions and presentations, Professor Wrigstad for sharing with us an in-depth and insightful look at his work in our keynote, and our program committee for the informative reviews.

We hope to see you in the next edition of VIMPL.

August 2025

Nguyen Pham and Dimi Racordon
VIMPL 2025 Program Co-Chairs

VIMPL 2025 Invited Talks

Fearless Concurrency in Python
Tobias Wrigstad

VIMPL 2025 Extended Abstracts

Toward a Typed Intermediate Language for R
Mickaël Laurent, Jakob Hain, Filip Krikava, Sebastián Krynski, and Jan Vitek

Mutable Value Semantics through a Runtime-Enforced Framework in Scala
Hamza Remmal

VIMPL 2025 Full Papers

Who Owns the Contents of a Doubly-Linked List?

Dimi Racordon

VIMPL 2025 Program Committee

- Ralf Jung, ETH Zurich (Switzerland)
- Jaakko Järvi, University of Turku (Finland)
- Yichen Xu, EPFL (Switzerland)

VIMPL 2025 Co-chairs

- Nguyen Pham, EPFL (Switzerland)
- Dimi Racordon, EPFL (Switzerland)

■ 9th Symposium on Computer Languages, Implementations and Tools (SCLIT 2025)

Research on programming languages is one of the core disciplines of computer science. Today, the term *computer languages* usually encompasses not only programming languages, but also all types of artificial languages for different purposes whose ‘sentences’ can be processed by a computer.

The SCLIT series aims to provide a forum for the dissemination of research achievements in areas that include all aspects of computer languages: theory, implementation, and processing and analysis tools. Following current trends in software development, SCLIT pays special attention to supporting multilingual software development and dedicated cross-language implementations, techniques, and tools.

The scope of the workshop is wide and versatile within the theme of computer languages. It includes, but is not limited to, the following topics:

- **Languages.** Theoretical aspects of programming languages, programming paradigms, script languages, modelling languages, .domain-specific languages, graphical languages, markup languages, specification languages, transformation languages, formal languages, intermediate languages, . . .
- **Implementations.** Compilers, debuggers, interpreters, virtual machines, transformation systems, translators, transpilers, intermediate representations, . . .
- **Tools and techniques.** software metrics, static analysers, clone detectors, abstract interpreters, visualizers, . . .

The 9th Symposium on Computer Languages, Implementations, and Tools, SCLIT 2025, was organised within «Programming» 2025. We received 6 submissions and finally accepted 5 of them for presentation at the symposium and publication in this companion proceedings.

We are grateful to all the authors who submitted the work to SCLIT, and all PC members for submitting careful and timely opinions on the submissions. We extend special thanks to the «Programming» 2025 Organising Committee for impeccable organisation of «Programming» 2025 as well as to the team at the Faculty of Mathematics and Physics, Charles University in Prague, for hosting this great event.

SCLIT 2025 Papers and Presentations

A Comparison of Three Program Query Languages to Detect Python Programming Misconceptions

Quentin Colla, Kim Mens, Julien Liénard

Are there Hypothesis for Attribute Grammars?

Emanuel Rodrigues, José Nuno Macedo, João Saraiva

Identifying security issues in Elixir web applications

Smiljana Knežev, István Bozó, Melinda Tóth

On the Effectiveness of Interpreter-Guided Compiler Testing

Federico Lochbaum, Guillermo Polito

Companion Proceedings of the 9th International Conference on the Art, Science, and Engineering of Programming (Programming 2025).

Editors: Jonathan Edwards, Roly Perera, and Tomas Petricek

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The Pyttern Program Query Language

Julien Liénard, Kim Mens, Siegfried Nijssen

SCLIT 2025 Program Committee

Mirjana Ivanović, University of Novi Sad, Serbia

Julien Liénard, Université catholique de Louvain, Belgium

Kim Mens, Université catholique de Louvain, Belgium

Csaba Nagy, Università della Svizzera Italiana, Switzerland

Guillermo Polito, Inria, Université de Lille, France

Zoltan Porkolab, Eotvos Lorond University, Hungary

Filip Prentović, University of Novi Sad, Serbia

Mateo Sanabria Ardila, Universidad de los Andes, Colombia

João Saraiva, University of Minho, Portugal

William Steingartner, Technical university of Kosice, Slovakia

Melinda Tóth, Eotvos Lorand University, Hungary

Balša Šarenac, University of Novi Sad, Serbia

SCLIT 2025 Co-Chairs

Gordana Rakić, University of Novi Sad, Serbia

Nicolás Cardozo, Universidad de los Andes, Colombia