1.	Record Nr.	UNINA9910483571003321
	Titolo	ECOOP 2010 -- Object-Oriented Programming : 24th European Conference, Maribor, Slovenia, June 21-25, 2010, Proceedings / / edited by Theo D'Hondt
	Pubbl/distr/stampa	Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 2010
	ISBN	1-280-38767-X 9786613565594 3-642-14107-2
	Edizione	[1st ed. 2010.]
	Descrizione fisica	1 online resource (XIII, 599 p. 196 illus.)
	Collana	Programming and Software Engineering, , 2945-9168 ; ; 6183
	Altri autori (Persone)	D'HondtTheo
	Disciplina	005.1/17
	Soggetti	Software engineering Computer science Compilers (Computer programs) Computer networks Immunospecificity Software Engineering Computer Science Logic and Foundations of Programming Compilers and Interpreters Computer Communication Networks Adaptive Immunity
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	Bibliographic Level Mode of Issuance: Monograph
	Nota di bibliografia	Includes bibliographical references and index.
	Nota di contenuto	Keynote 1 -- Code Roots -- Programming Environments and Tools -- Detecting Missing Method Calls in Object-Oriented Software -- Debugging Model-Transformation Failures Using Dynamic Tainting -- Automatically Extracting Class Diagrams from Spreadsheets -- Theoretical Foundations of Programming Languages -- Adding Dynamic Types to C -- Essential AOP: The A Calculus -- The Essence of JavaScript -- Formal Methods -- Verifying Executable Object-Oriented Specifications with Separation Logic -- Verifying Generics and Delegates -- Recency Types for Analyzing Scripting Languages --

Concurrency Models in Java -- Correct Refactoring of Concurrent Java Code -- Programming Coordinated Behavior in Java -- JCoBox: Generalizing Active Objects to Concurrent Components -- Special Session: Empirical Methods -- Doubts about the Positive Impact of Static Type Systems on Programming Tasks in Single Developer Projects - An Empirical Study -- Type Systems -- A Type System for Data-Centric Synchronization -- Type-Safe Eventful Sessions in Java -- Capabilities for Uniqueness and Borrowing -- Keynote 2 -- Secret Valley -- Language Design and Implementation -- Embedding Languages without Breaking Tools -- Modules as Objects in Newspeak -- Inline Caching Meets Quickening -- Concurrency Abstractions -- Self-Replicating Objects for Multicore Platforms -- Reasoning about the Implementation of Concurrency Abstractions on x86-TSO -- Concurrent Abstract Predicates -- Experiences -- The Use of Overloading in Java Programs -- Falling Back on Executable Specifications -- Contract-Based Data Structure Repair Using Alloy.

Sommario/riassunto

It is an honor and a pleasure to present the proceedings of the 24th European Conference on Object-Oriented Programming (ECOOP 2010) held in Maribor, Slovenia. As always, the conference catered to a broad audience of members of industry and academia and consequently o'ered a mix of results in both applied andtheoreticalresearchinobjectorientation. Inall,24technicalpapers, 12wo- shops,6tutorials, adoctoralsymposiumandaspecialsessiononempiricalme- ods were organized. Additionally, a seriesof seven summer school sessionsran in parallel with the technical paper sessions. A junior and a senior Dahl-Nygaard award were selected by AITO leading to two keynote talks, complemented by an invited keynote on the subject of empirical methods as applied to software practice. The banquet speech was given by Oscar Nierstrasz. The ECOOP P- gram Committee also selected a best paper award. These proceedings consist of 24 papers and one extended abstract selected from 108 submissions. Each paper received3 and as many as 6 reviewsfrom a ProgramCommittee consisting of 30 internationally reputed researchers. A long and intensive virtual discussion via CyberChairPROled to an author response period and to another week of online deliberation. This wasroundedo'by a ProgramCommittee meeting late Feb- aryinBrussels. Itwasagainshownthatnothingmatchesaphysicalmeeting,and that it in several instances is even indispensable, in spite of the huge e'ort that is required to organize it. The Program Committee discussions followed Oscar Nierstrasz' Champion pattern: for a paper to be accepted at least one manifest proponent is required. Program Committee members were allowed to submit a paper, but these were subjected to a higher level of scrutiny.