

Gary T Leavens

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/11960864/publications.pdf>

Version: 2024-02-01

66
papers

2,774
citations

331670

21
h-index

223800

46
g-index

73
all docs

73
docs citations

73
times ranked

768
citing authors

#	ARTICLE	IF	CITATIONS
1	Exploring True Test Overfitting in Dynamic Automated Program Repair using Formal Methods. , 2021, , .		18
2	More Reliable Test Suites for Dynamic APR by using Counterexamples. , 2021, , .		8
3	Automated translation of VDM to JML-annotated Java. International Journal on Software Tools for Technology Transfer, 2018, 20, 211-235.	1.9	9
4	Unifying separation logic and region logic to allow interoperability. Formal Aspects of Computing, 2018, 30, 381-441.	1.8	5
5	A Methodology for Invariants, Framing, and Subtyping in JML. , 2018, , 19-39.		2
6	An Illustrated Guide to the Model Theory of Supertype Abstraction and Behavioral Subtyping. Lecture Notes in Computer Science, 2018, , 39-88.	1.3	0
7	Aspect-Oriented Programming Reloaded. , 2017, , .		4
8	Information Hiding in RGB Images Using an Improved Matrix Pattern Approach. , 2017, , .		11
9	Reasoning tradeoffs in languages with enhanced modularity features. , 2016, , .		5
10	Spekl: A Layered System for Specification Authoring, Sharing, and Usage. , 2016, , .		0
11	Behavioral Subtyping, Specification Inheritance, and Modular Reasoning. ACM Transactions on Programming Languages and Systems, 2015, 37, 1-88.	2.1	19
12	Conditional effects in fine-grained region logic. , 2015, , .		6
13	Enforcing information hiding in interface specifications: a client-aware checking approach. , 2015, , .		2
14	AspectJML. , 2014, , .		13
15	Optimizing generated aspect-oriented assertion checking code for JML using program transformations: An empirical study. Science of Computer Programming, 2013, 78, 1137-1156.	1.9	4
16	Client-aware checking and information hiding in interface specifications with JML/ajmlc. , 2013, , .		3
17	Specifying subtypes in Safety Critical Java programs. Concurrency Computation Practice and Experience, 2013, 25, 2290-2306.	2.2	4
18	Behavioral interface specification languages. ACM Computing Surveys, 2012, 44, 1-58.	23.0	94

#	ARTICLE	IF	CITATIONS
19	Specifying subtypes in SCJ programs. , 2011, , .		5
20	temporaljmlc: A JML Runtime Assertion Checker Extension for Specification and Checking of Temporal Properties. , 2010, , .		4
21	Tisa: A Language Design and Modular Verification Technique for Temporal Policies in Web Services. Lecture Notes in Computer Science, 2009, , 333-347.	1.3	8
22	Lessons from the JML Project. Lecture Notes in Computer Science, 2008, , 134-143.	1.3	5
23	Ptolemy: A Language with Quantified, Typed Events. Lecture Notes in Computer Science, 2008, , 155-179.	1.3	88
24	Information Hiding and Visibility in Interface Specifications. , 2007, , .		26
25	Tutorial on JML, the java modeling language. , 2007, , .		9
26	Modular verification of higher-order methods with mandatory calls specified by model programs. ACM SIGPLAN Notices, 2007, 42, 351-368.	0.2	1
27	Specification and verification challenges for sequential object-oriented programs. Formal Aspects of Computing, 2007, 19, 159-189.	1.8	86
28	A JML Tutorial: Modular Specification and Verification of Functional Behavior for Java. , 2007, , 37-37.		7
29	MAO: Ownership and Effects for More Effective Reasoning About Aspects. Lecture Notes in Computer Science, 2007, , 451-475.	1.3	10
30	MiniMAO1 : An imperative core language for studying aspect-oriented reasoning. Science of Computer Programming, 2006, 63, 321-374.	1.9	37
31	Modular invariants for layered object structures. Science of Computer Programming, 2006, 62, 253-286.	1.9	92
32	Multijava. ACM Transactions on Programming Languages and Systems, 2006, 28, 517-575.	2.1	74
33	Roadmap for enhanced languages and methods to aid verification. , 2006, , .		41
34	Preliminary design of JML. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2006, 31, 1-38.	0.7	355
35	How the design of JML accommodates both runtime assertion checking and formal verification. Science of Computer Programming, 2005, 55, 185-208.	1.9	107
36	An overview of JML tools and applications. International Journal on Software Tools for Technology Transfer, 2005, 7, 212-232.	1.9	414

#	ARTICLE	IF	CITATIONS
37	Extending JML for Modular Specification and Verification of Multi-threaded Programs. Lecture Notes in Computer Science, 2005, , 551-576.	1.3	38
38	Modular specification of frame properties in JML. Concurrency Computation Practice and Experience, 2003, 15, 117-154.	2.2	37
39	An overview of JML tools and applications1 1www.jmlspecs.org. Electronic Notes in Theoretical Computer Science, 2003, 80, 75-91.	0.9	68
40	How the Design of JML Accommodates Both Runtime Assertion Checking and Formal Verification. Lecture Notes in Computer Science, 2003, , 262-284.	1.3	41
41	A Simple and Practical Approach to Unit Testing: The JML and JUnit Way. Lecture Notes in Computer Science, 2002, , 231-255.	1.3	88
42	Multijava. ACM SIGPLAN Notices, 2000, 35, 130-145.	0.2	47
43	Safely creating correct subclasses without seeing superclass code. ACM SIGPLAN Notices, 2000, 35, 208-228.	0.2	16
44	Executing Formal Specifications with Concurrent Constraint Programming. Automated Software Engineering, 2000, 7, 315-343.	2.9	14
45	A complete algebraic characterization of behavioral subtyping. Acta Informatica, 2000, 36, 617-663.	0.5	12
46	Safely creating correct subclasses without seeing superclass code. , 2000, , .		15
47	Formal Techniques for Java Programs. , 2000, , 41-54.		9
48	Enhancing the pre- and postcondition technique for more expressive specifications. Lecture Notes in Computer Science, 1999, , 1087-1106.	1.3	30
49	JML: A Notation for Detailed Design. , 1999, , 175-188.		210
50	Protective Interface Specifications. Formal Aspects of Computing, 1998, 10, 59-75.	1.8	13
51	Class-Based and Algebraic Models of Objects. Electronic Notes in Theoretical Computer Science, 1998, 14, 214-244.	0.9	3
52	1997 workshop on foundations of component-based systems. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 1998, 23, 38-41.	0.7	4
53	Multiple dispatch as dispatch on Tuples. ACM SIGPLAN Notices, 1998, 33, 374-387.	0.2	4
54	An Overview of Larch/C++: Behavioral Specifications for C++ Modules. , 1996, , 121-142.		29

#	ARTICLE	IF	CITATIONS
55	On Binary Methods. Theory and Practice of Object Systems, 1995, 1, 221-242.	0.7	108
56	Typechecking and modules for multimethods. ACM Transactions on Programming Languages and Systems, 1995, 17, 805-843.	2.1	63
57	Foundations of Object-Oriented Languages. ACM SIGPLAN Notices, 1995, 30, 5-11.	0.2	5
58	Specification and verification of object-oriented programs using supertype abstraction. Acta Informatica, 1995, 32, 705-778.	0.5	61
59	Specification and verification of object-oriented programs using supertype abstraction. Acta Informatica, 1995, 32, 705-778.	0.5	14
60	The Larch/Smalltalk interface specification language. ACM Transactions on Software Engineering and Methodology, 1994, 3, 221-153.	6.0	19
61	Typechecking and modules for multi-methods. , 1994, , .		21
62	Inheritance of interface specifications (extended abstract). ACM SIGPLAN Notices, 1994, 29, 129-138.	0.2	4
63	Typechecking and modules for multi-methods. ACM SIGPLAN Notices, 1994, 29, 1-15.	0.2	12
64	Typed homomorphic relations extended with subtypes. Lecture Notes in Computer Science, 1992, , 144-167.	1.3	8
65	Introduction to the literature on object-oriented design, programming, and languages. ACM SIGPLAN OOPS Messenger, 1991, 2, 40-53.	0.1	3
66	Reasoning about object-oriented programs that use subtypes. ACM SIGPLAN Notices, 1990, 25, 212-223.	0.2	13