A unified type system for the modern general-purpose programming language focusing on reliability and verifiability

Alexey Kanatov

[alexey.v.kanatov@gmail.com](mailto:alexey.v.kanatov@gmail.com)

Eugene Zouev

[eugene.zueff@gmail.com](mailto:eugene.zueff@gmail.com)

**Abstract**

The paper presents an overview of the type system which supports the convergence of procedural, object-oriented, functional, and concurrent programming paradigms relying on static type checking with smart type inference support and the ability to ensure dynamic type safety as well. The key element of the type system that it is fully based on just 2 basis constants and all other constructions are derived

**Keywords**

Object, constant objects, type, unit, class, module, interface, conformance, compatibility, type conversions, setters, reference and value objects, immutability.

[CCS](https://dl.acm.org/ccs/ccs.cfm?id=0&lid=0&CFID=1009033506&CFTOKEN=16747759) →  [Software and its engineering](https://dl.acm.org/ccs/ccs.cfm?id=10011007&lid=0.10011007&CFID=1009033506&CFTOKEN=16747759) →  [Software notations and tools](https://dl.acm.org/ccs/ccs.cfm?id=10011006&lid=0.10011007.10011006&CFID=1009033506&CFTOKEN=16747759) →  [General programming languages](https://dl.acm.org/ccs/ccs.cfm?id=10011008&lid=0.10011007.10011006.10011008&CFID=1009033506&CFTOKEN=16747759) →  [Language types](https://dl.acm.org/ccs/ccs.cfm?id=10011009&lid=0.10011007.10011006.10011008.10011009&CFID=1009033506&CFTOKEN=16747759) →  Multi-paradigm programming languages