ABSTRACT

High Orthogonality is one desired and very important property of any computing system in various disciplines: Hardware, Database, Artificial Intelligence, Vector Processing, and possibly many others. Programming languages – which play important role in implementing the technology in various disciplines – must be designed with high orthogonality in mind.

Despite this, the facts show how current and recent programming languages move away from this principle. Programming languages nowadays become less and less orthogonal. Consequently, the complexity of popular programming languages have become very high.

Often that complexity is attributed to the necessity or price to pay for three features: Polymorphism, Inheritance, and Encapsulation (abbreviated as PIE). Since these three features are known to be advanced by OOP, the complexity of OOPL is attributed to PIE.

After a thorough investigation, we find that the main problem with current theory and implementation of PIE is the lack of orthogonality for PIE. Orthogonality should be applied to polymorphism, to inheritance, and to encapsulation.
This dissertation seeks into establishing the theoretical foundation of orthogonal polymorphism, orthogonal inheritance, and orthogonal encapsulation. Among the three, orthogonal polymorphism turns out to be the most difficult one to specify. As byproduct of this research, the proposed theory is partially implemented in NUSA programming language.

Keywords: Orthogonal; Encapsulation; Inheritance; Polymorphism; NUSA.