

1. Record Nr.	UNISA996465705703316
Autore	Luckham David C
Titolo	ANNA A Language for Annotating Ada Programs [[electronic resource]] : Reference Manual / / by David C. Luckham, Friedrich W. von Henke, Bernd Krieg-Brueckner, Olaf Owe
Pubbl/distr/stampa	Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 1987
ISBN	3-540-47202-9
Edizione	[1st ed. 1987.]
Descrizione fisica	1 online resource (V, 148 p.)
Collana	Lecture Notes in Computer Science, , 0302-9743 ; ; 260
Disciplina	005.13
Soggetti	Programming languages (Electronic computers) Programming Languages, Compilers, Interpreters
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di contenuto	1. Basic Anna concepts -- 2. Lexical elements -- 3. Annotations of declarations and types -- 4. Names and expressions in annotations -- 5. Statement annotations -- 6. Annotation of subprograms -- 7. Package annotations -- 8. Visibility rules in annotations -- 9. Task annotations -- 10. Program structure -- 11. Exception annotations -- 12. Annotation of generic units -- 13. Annotation of implementation-dependent features.
Sommario/riassunto	This reference manual of ANNA is another volume addressed to the ADA community. ANNA is a language extension of ADA to include facilities for formally specifying the intended behavior of ADA programs. It is designed to meet a perceived need to augment ADA with precise machine-processable annotations so that well established formal methods of specification and documentation can be applied to ADA programs. The current ANNA design includes annotations of all ADA constructs except tasking. Similar extensions for formal specification can be made to other Algol-like languages such as Pascal, PL/1, Concurrent Pascal, and Modula; essentially, these extensions would be subsets of ANNA. The design of ANNA was undertaken from the beginning with four principal considerations: 1. Constructing annotations should be easy for the ADA programmer and should depend as much as possible on notation and concepts of ADA. 2. ANNA

should possess language features that are widely used in the specification and documentation of programs. 3. ANNA should provide a framework within which the various established theories of formally specifying programs may be applied to ADA. 4. Annotations should be equally well suited for different possible applications during the life cycle of a program. Such applications include not only testing, debugging and formal verification of a finished program, but also specification of program parts during the earlier stages of requirements analysis and program design.
