CLAIMS

1. A method of computer-assisted programming, comprising:

maintaining, on a computer memory, a first representation of a program code;

obtaining, via a user interface, a selection of at least one program element and a corresponding insertion location in the program code;

updating the first representation, to include the selected at least one program element at said insertion location;

translating the first representation to produce a second representation of the program code; and

displaying the second representation on a user interface.

2. The method of claim 1, wherein the first representation of the program code is formatted as an intermediary-level program code representation, and wherein the second representation of the program code is formatted as textual, high-level, user intelligible programming language representation.

3. The method according to any one of claims 1 and 2, wherein obtaining the selection of the at least one program element and the corresponding insertion location comprises:

receiving, via the user interface, a selection of a first insertion location in the user-level programming language representation;

identifying a second insertion location, in the intermediary-level program code representation that corresponds to the first insertion location;

presenting, via the user interface, a list of selectable program elements, that are valid for insertion at the second insertion location, according to rules pertaining to a programming language; and

receiving, via the user interface, the selection of the at least one textual program element from the list of selectable valid program elements.

4. The method according to claim 3, further comprises preventing the user from inserting a program element into the stored program code in any way that is devoid of selection of at least one selectable program element from the list of selectable valid program elements.

5. The method according to any one of claims 3-4, wherein producing the presented list of selectable valid program elements comprises:

traversing a list of available program elements;

for one or more program elements of the list of available program elements, traversing over rules of the programming language; and

determining whether the relevant program element complies with said rules and is thus valid for insertion at the location of said insertion point; and

adding the relevant program element determined as valid to the list of selectable valid program elements.

6. The method according to any one of claims 1-5, wherein the selectable program elements are presented to the user as high-level, human intelligible text of a programming language.

7. The method according to any one of claims 2-6, further comprising executing the intermediary-level program code representation on a computing device without requiring compilation or parsing of source code.

8. The method according to any one of claims 2-7, wherein translating the first representation of the intermediary-level program code format to the second the representation of the high-level program code format further comprises creating a location table, associating a user-marked location with corresponding program elements in the first representation of the intermediate-level code format, and wherein identifying the second insertion location corresponding to the first insertion location is done based on the location table.

9. The method according to any one of claims 2-8, wherein intermediate-level program code is structured as a hierarchical structured program code model, representing a hierarchical structure of the program code; and the method further comprising:

determining a context of one or more program elements according to the hierarchical structured program code model; and

determining a scope of one or more symbols of program elements in the program code according to the hierarchical structured program code model.

10. The method of any one of claims 1-9, comprising:

executing the first representation of the program code by a virtual machine.

11. The method of any one of claims 1-10, comprising:

transferring the first representation of the program code to other computing devices to be executed on other computing devices and/or operating systems.

12. The method of any one of claims 1-11, comprising:

deleting an at least one existing program instruction; and

automatically replacing the at least one existing program instruction with a placeholder if required to maintain valid program structure.

13. The method of any one of claims 1-12, comprising:

editing an at least one or more existing program instructions, wherein the editing comprises at least one of:

selecting an existing program instruction from the at least one or more existing program instructions which declare a program symbol and renaming said program symbol while asserting that the newly entered name is valid for said program symbol according to the language syntax;

selecting an existing program element which defines a program value and editing said program value while asserting that newly entered value complies with the requirements of the program;

selecting one or more existing program instructions from the at least one or more existing program instructions and replacing the existing one or more program instructions with another program instruction from a displayed list of other program instructions that are valid for insertion in the same location; and

selecting one or more existing program instructions from the at least one or more existing program instructions and copying and pasting the one or more existing program instructions to another location, wherein the one or more existing program instructions are assimilated in said other location and constitute a valid program.

14. The method of any one of claims 1-13, comprising:

selecting one or more existing program elements from the at least one or more existing program elements and deleting the one or more existing program elements;

validating the deletion of the one or more existing program elements in accordance with the one or more rules of the programming language; and

omitting the one or more existing program elements from the program code based on the validation.

15. The method of any one of claims 1-14, comprising:

moving an at least one indicated program element in the program code;

validating the movement of the at least one indicated program element in accordance with the one or more programming language rules; and

moving the at least one indicated program element in the program code based on said validation.

16. The method according to claims 1-15, further comprises not enabling or facilitating insertion of program elements into program code via methods of typing text directly into program code.

17. A computing device configured to generate a program code comprising a computing chip, wherein the computing chip is configured to execute the method of claims 1-16.

18. A method of computer-assisted programming performed by a computer chip, comprising:

maintaining, on a computer memory a first representation of a program code;

translating the first representation of the program code to produce a second representation of the program code;

displaying the second representation of the program code on a user interface;

producing an at least one list of selectable program elements that are valid for insertion into the program code at a marked location;

receiving, through the user interface, a selection of at least one selectable program element from the list of selectable program elements;

inserting the at least one selectable program element into the first representation of the program code, at the marked location; and

displaying an update of the second representation of the program code on the user interface, wherein the update is to reflect the change in the first representation of the program code.

19. The method of claim 18, wherein the first representation is formatted as an intermediary level program code representation, and the second representation is formatted as a user intelligible programming language representation.

20. The method of any one of claims 18-19, comprising:

receiving, via the user interface, a selection of a first marked location in the user intelligible programming language representation;

identifying a second marked location in the intermediary-level program code representation that corresponds to the first marked location;

presenting via the user interface in the user intelligible programming language representation a list of selectable program elements that are valid for insertion at the second marked location; and

receiving, via the user interface, the selection of the at least one selectable program element from the list of selectable program elements.

21. The method of any one of claims 18-20, comprising:

executing the intermediary-level program code representation on a computing device without requiring compilation of a source code.

22. The method of any one of claims 18-21, comprising:

producing an error-free program code.

23. The method of any one of claims 18-22, comprising:

executing the first representation of the program code by a virtual machine.

24. The method of any one of claims 18-23, comprising:

transferring the first representation of the program code to other computing devices to be executed on the other computing devices.

25. The method of any one of claims 18-24, comprising:

transferring and executing the first representation of the program code on other operating systems.

26. The method of any one of claims 18-25, comprising:

validating a program element in accordance with one or more rules of a programming language; and

inserting the program element, being validated, into the at least one list of selectable program elements.

27. The method of any one of claims 18-26, comprising:

creating a computer program without typing code.

28. The method of any one of claims 18-27, comprising:

deleting an at least one existing program instruction; and

automatically replacing the at least one existing program instruction with a placeholder if required to maintain valid program structure.

29. The method of any one of claims 18-28, comprising:

editing an at least one or more existing program instructions, wherein the editing comprises at least one of:

selecting an existing program instruction from the at least one or more existing program instructions which declare a program symbol; and

renaming said program symbol while asserting that the newly entered name is valid for said program symbol according to the language syntax;

selecting an existing program element which defines a program value and editing said program value while asserting that newly entered value complies with the requirements of the program;

selecting one or more existing program instructions from the at least one or more existing program instructions and replacing the existing one or more program instructions with another program instruction from a displayed list of other program instructions that are valid for insertion in the same location; and

selecting one or more existing program instructions from the at least one or more existing program instructions and copying and pasting the one or more existing program instructions to another location, wherein the one or more existing program instructions are assimilated in said other location and constitute a valid program.

30. The method of any one of claims 18-29, comprising:

selecting one or more existing program elements from the at least one or more existing program elements and deleting the one or more existing program elements;

validating the deletion of the one or more existing program elements in accordance with the one or more rules of the programming language; and

omitting the one or more existing program elements from the program code based on the validation.

31. The method of any one of claims 18-30, comprising:

moving an at least one indicated program element in the program code;

validating the movement of the at least one indicated program element in accordance with the one or more programming language rules; and

moving the at least one indicated program element in the program code based on said validation.

32. The method of any one of claims 18-31, wherein the second representation is formatted as a formal, high-level programming language.

33. A computing device configured to generate a program code comprising a computing chip, wherein the computing chip is configured to execute the method of claims 18-32.