The creation of an audio interface is not only of scientific but also of human interest, particularly in the rehabilitation of the visually impaired (whom we shall call C-users). The mass character of this problem should be noted. In education in Ukraine alone, there are 675 kindergartens and 23 boarding schools with, respectively, 8800 and 4400 children with impaired vision. The Ukrainian Society for the Blind has about 60,000 members, including 25,000 of ages 14 to 35. About a thousand sightless persons hold administrative positions in the educational and industrial enterprises of the Society. Noteworthy is the high creative and scientific potential of the sightless, among whom are many students, teachers, and scholars; it is sufficient to recall such well-known mathematicians as L. S. Pontryagin and A. G. Vitushkin. Hence the importance of computerization of the work and life of persons of this category.

Programming is among the important areas of human activity involving the use of computers. The professional activity of sightless programmers in Ukraine is in an embryonic state due to the limited numbers and, primarily, the lack of the required software and hardware for interacting with them. At the same time, abroad the profession of programmer is very popular among the visually impaired: special personal computers equipped with speech synthesizers, braille printers, and tactile devices and notebook computers for the blind have been developed to create a friendly interface for C-users (without sighted intermediaries). In addition to the above-mentioned devices, the Notaphon portable computer has a clock, a calendar, a cardfile for addresses and telephone numbers, etc. The brochures accompanying the computers point to the high efficiency of C-users and the independence, value, and comfort of their positions in programmers' groups [1, 2].

The work experience of C-user programmers is based on the use of conventional programming languages and the above-mentioned software and hardware. It should be noted that modern programming technology is oriented toward visualization of the program-development process and is thus difficult to master by C-users. Friendly language and programming tools that allow for the specific nature of computer work by C-users are required to facilitate substantially the process of reproducing for C-users the current state of a program and the data being processed in a corresponding computing environment. Such tools must support text design, analysis, and synthesis.

Studies are underway in the Department of Computer-Aided Programming of the V. M. Glushkov Institute of Cybernetics of the Academy of Sciences of Ukraine to create efficient methods for program design and generation. In particular, a method of multilevel structured program design (MSPD) has been under development since 1978; it is based on an algebraic—grammatical formalism that arises from systems of algorithmic algebras (SAA) [3]. An arsenal of support tools — the MULTIPROCESSIST system — has been created for this method [4].

We shall describe the development of tools for voice-generated text by C-users — the ORIENTIR system [5] — which is a further development of the MSPD tools.

C-user orientation determines the following requirements on the system to be developed:

• Closeness of the language for description of text (in particular, program) structure to natural language.
• Restriction of the input-language of the system to a relatively small set of simple constructs.
• Orientation to a wide range of C-users, including nonprogrammers who wish to learn computer programming.
• Simplicity of selection of a subject area and target programming language.
• Automatic compiling and debugging of programs.
• Tools for the accumulation of knowledge and experience in text and program development.

On the whole, the system is designed for voice generation of structured texts in a stylized natural-language form that takes into account the specific nature of the perception of information by C-users.
The article consists of four sections. The first section deals with the interface tools of the system. A simple and convenient language for algorithm and program design by C-users is described in the second section. The possibilities of the language are illustrated by the example of an insertion-sort algorithm [4]. The third and fourth sections are devoted to the processes of construction, editing, and synthesis of texts (in particular, programs) by the tools of the ORIENTIR system.

1. INTERFACE TOOLS OF ORIENTIR SYSTEM

The system interface is supported by hardware, software, and language tools. They include the following:

- A speech-synthesis subsystem for audible checking of keyboard input and voice-message output as a built-in dynamic and a parallel-interface port.
- A speech-recognition subsystem with an ADC/DAC board for voice control of the system operation.
- A language for the representation of text structure that is supported by text-editor, syntactic-analyzer, and an interactive constructor of syntactically correct programs (DSP constructor).
- A knowledge base that stores tests and their descriptions and includes a reference component and tools for text composition according to plans.

In the design of the interface, preference was given to the concept of system-initiated dialog. Support for other forms of interaction with C-users was found to be unjustifiably complicated and did not provide the effect achieved with sighted users. In fact, it is complicated for the C-user to become immediately oriented in conventionally organized multilevel menus, especially when horizontal and vertical windows are combined. Instruction input by the simultaneous pressing of two keys (for example, "Ctrl-K") is difficult, since the fingers of the C-user are oriented toward the conventional keyboard configuration. The direct use of function keys does not speed up operation, since the sightless user enters data from the keyboard according to a protocol: the pressing of any function key is accompanied by an audio reproduction of the name of the instruction to be evoked, after which the system waits for confirmation of the selection.

The following concept of interaction with C users has been adopted.

1. At each step, the user is presented with a list of the instructions that are usually executed in the current mode with the cursor on the most-likely action for the given step, whose name is pronounced. The user can move the cursor through the instruction list only along one coordinate (vertically). To speed up this movement, the two-letter abbreviation of the instruction is articulated (the spacebar is pressed to hear the full name of the instruction).

2. An instruction is executed by pressing the "Enter" key (or corresponding mouse button). The system reports the absence of data and presents a menu for decision making. For example, when an attempt is made to edit a nonexisting text, the system suggests that a new file be opened or that a file be selected from a list of those available.

3. The system makes it possible to organize hierarchical lists of instructions based on the classification of objects and actions.

The "Insert" and "Delete" keys are also used for working with lists of objects. The "Enter" key allows one to move from an object to the instructions associated with it; the "Insert" key, to specific objects of a given type; and the "Delete" key, to the attributes of an object. For a <Text> object, for example, the "Enter" key opens a list of actions text as a whole (read, save, format), the "Insert" key provides access to a list of text file names, and the "Delete" key offers text components (blocks, paragraphs, sentences).

4. When an instruction is executed, all actions not associated with it are disabled and the unused keys are blocked. In block selection, for example, only cursor movement through the text is enabled; the block cannot be edited or moved.

2. INPUT LANGUAGE OF ORIENTIR SYSTEM

The SAA/1 structured-text language, which is based on the SAA apparatus, was developed to provide a friendly interface for C-users; it is a natural-language subset consisting of a fixed set of simple and semantically transparent constructs. Unlike the well-known foreign specification languages, SAA/1 is intended for the design and synthesis of stylized texts. Projects are a level-by-level detailing of any depth of the texts and programs to be designed. Elementary texts, which are at