

consisting of strings of high level database commands. Modules for slightly esoteric applications (such as an accounting system dedicated to stockbrokers) can be constructed without having to write the program from scratch. In view of the limited size of the market, a stockbroker package that runs under dBase II may well be better than one written in BASIC, because the program's author will have been able to concentrate all his efforts on the *operation* of the program, rather than the *writing* of the code. The parts of the program most susceptible to bugs (for example, the file handling) will have been written by the generator's authors and tested in different applications by thousands of users.

But the main difference between a program generator and an application generator is in their user-friendliness. The final program created by the former type of package will consist entirely of artificially-written code, which will probably be in a language such as BASIC. Such code will be inferior, both in efficiency and style, to code generated by humans. With the application generator, however, perhaps as much as 99 per cent of the final program will consist of code written by the software house, and this will probably be in machine code as well. This is the case with Silicon Office, one of the most sophisticated and easy to use application generators available for business microcomputers. The resulting program will be faster and more efficient, incorporate checking procedures to detect operator errors, and produce clearly laid-out menu-driven screen displays.

Furthermore, application generators are not restricted to business programs. Perhaps the best example of a non-business package is the Pinball Construction Set (see page 241), in which the application module is effectively specified by laying out the elements of the required pinball table.

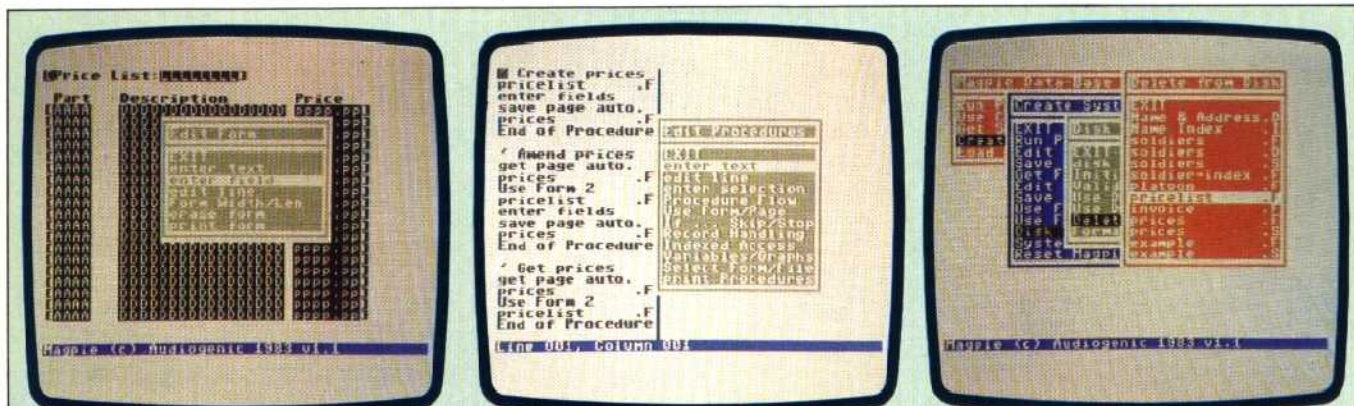
There is, in fact, a great deal of overlap

between this subject and object oriented programming, which we have discussed before (see page 242), but which may be broadly summarised as: encouraging the programmer to implement his applications purely by specifying the objectives required of the program. Even simple spreadsheet programs, available for home computers such as the Sinclair Spectrum, can be regarded as application generators — you simply specify the relationship between the various fields, and the package does all the routine work for you.

Magpie — produced by Audiogenic for the Commodore 64 with one disk drive — is an application generator that is geared towards business or other serious applications. This is another package that makes good use of visual object oriented programming: relationships between items of data in different records are specified when designing the layout of those records.

Although they are not strictly regarded as application generators, an increasing number of packages are now incorporating some of these principles. When first run, such 'parameter-driven' programs will ask the user a whole series of questions and record the answers on disk alongside the program. This information will determine some of the details of the program's operation. An invoicing program, for example, would ask questions relating to the information that the company likes to have included in each invoice, and the standard credit periods that it allows. An arcade game might ask how many aliens, bases and rockets the user would like to start with, or even give him the opportunity to design the invaders.

Increasingly, software is being designed to protect the user from having to learn programming, while at the same time providing a high degree of flexibility in operation. A situation where the software adjusts itself to fit the user's requirements (rather than the user adjusting to the software) is a highly desirable goal.



## Magpie On The Commodore 64

The first stage in creating an application is to specify the layout of all forms, transactions and reports, such as this price list. By filling the columns with letters (A,D,P) the user specifies which fields from the database are to be used

Next, all the calculations and processing are specified in the form of a list of instructions in a high level programming language, to be interpreted by Magpie. Shown here are the routines to amend the prices and to recall (GET) them from disk

Magpie is menu-driven. As any option is selected (e.g. CREATE), another will appear beside it, showing all the CREATE options. This screen displays the result of selecting CREATE, then DISK, then DELETE, then the file to be deleted, in this case PRICE LIST