Packages and Automatic Header Inference

The separate compilation facilities of Java and C# eliminate `.h` files. Java introduces a formal notion of module, called a **package**. Every compilation unit, which may be a file or (in some implementations) a record in a database, belongs to exactly one package, but a package may consists of many compilation units, each of which begins with an indication of the package it belongs to. Unless explicitly declared as `public`, a class in Java is visible in all and only those compilation units that belong to the same package.

As in C++, a compilation unit that needs to use classes from another package can access them using qualified names, or via name-at-time or package-at-a-time import.

When asked to import names from package M, the Java compiler will search for M in a standard set of places, and will recompile it if appropriate. The compiler will then **automatically** extract the information that would have been needed in a C++ `.h` file or an Ada or Modula-3 header. If the compilation of M requires other packages, the compiler will search for them as well, recursively.

To mimic the software engineering practice of early header file construction, a Java or C# design team can create a skeleton version of (the public classes of) its packages or namespaces, which can then be used, concurrently and independently, by the programmers responsible for the full versions.