

The ACM Java Task Force

Executive Summary

In recent years, Java has steadily gained prominence as the language of choice for introductory computer science courses. At the same time, its complexity creates problems for both students and teachers. To address these problems, the ACM Education Board created the ACM Java Task Force in early 2004 with the following general charter:

To review the Java language, APIs, and tools from the perspective of introductory computing education and to develop a stable collection of pedagogical resources that will make it easier to teach Java to first-year computing students without having those students overwhelmed by its complexity.

The Task Force began its review process by seeking to identify the challenges instructors face in teaching Java. An extensive review of the educational literature and a survey of the community revealed the following problems, which were widely regarded as significant:

- Static methods, most notably **main**
- The lack of a simple input mechanism
- The conceptual difficulty of the graphics model
- The tools for creating graphical user interfaces (GUIs) are inappropriate for novices.

The packages that form the JTF collection—**acm.program**, **acm.io**, **acm.graphics**, **acm.gui**, and **acm.util**—offer solutions to each of these problems. These solutions, moreover, have been presented at two successive conferences of the ACM Special Interest Group on Computer Science Education (SIGCSE) so that the design could profit from the input of the computing education community as a whole.

We believe that the following represent the greatest strengths of the JTF packages:

- *A simple object-oriented model for programs.* The **Program** class defined in the **acm.program** package offers an easy-to-use model for writing simple programs. In addition to hiding the static **main** method, the **Program** class and its standard subclasses provide a highly intuitive example of object-oriented class hierarchies.
- *A model for input and output that treats traditional console I/O and dialog I/O symmetrically.* The **acm.io** package defines the classes **IOConsole** and **IODialog** that share a common interface for all input-output operations. This design addresses the lack of a simple input mechanism in a way that emphasizes the value of interface-based design.
- *An extensive library of graphical objects.* The **acm.graphics** package implements a simple but extremely powerful model for creating graphical pictures based on the metaphor of a felt board in which students construct graphical objects of various types and place them on a canvas. This design emphasizes the use of objects and frees the student from having to respond explicitly to repaint requests.
- *A minimal set of new classes to support development of graphical user interfaces.* The **acm.gui** package includes a small set of classes to bring Java's extensive GUI-development resources within reach of novice programmers.
- *Backward compatibility for applets.* Unlike most Java code today, programs developed using the JTF packages can typically be executed as applets even on older web browsers. This flexibility makes them ideal for web-based teaching tools and lecture demonstrations.