

Programmer Thinking B. The Facade Design Pattern for Human Interface

Slow Talking Humans vs Fast Moving Information between Programs

The Problem: it's easy to confuse two very different things, Input / Output to the user and Values going to and from function calls.

Slow Talkers: That's us humans. We work in seconds (maybe milliseconds at the fastest). We read things with our eyes, write and type with our hands. Same when we work with a computer (keyboard and display screen).

Computers: Software talks with other software much faster, close to the instruction speed of the computer itself (Gigahertz speeds = billions of things a second).

Always remember the mismatch! While you press down one key on the keyboard (say at your fastest, 100 ms) the computer does about 500 million things (or in C# statements about 1+ million lines of program code).

One common solution to prevent confusion to the programmer, and to stop the computer from getting bored waiting on us slow talkers: the facade design pattern. Put all the interaction with the user (input and output) in one place, often the main of a simple program. Have the facade call functions to do the real work of the app. These functions can call other functions, that can call other functions, that can call None of the functions do slow talking to humans.

So, how do these functions do their work? They pass parameters and return values among themselves until they are finished doing their work. Only then do they tell the facade the answer to pass back to us (or ask for more information, or see if we are done, etc.) After all their work we won't even notice the time that went by (we are still thinking of reaching for the keyboard!)