Profiling

Class **CB** provides the basic means for profiling your application. These means include static methods to measure execution time of a part of code, as well as the methods allow to get a memory allocation and time of the request to the server.

For example, you can measure the execution time of a code fragment as follows:

```
// The start point of profiling of the code fragment.
CB::pStart('fragment1');
// Some code to profile.
foreach (range(0, 10) as $i)
{
  // The start point of profiling of the second code fragment.
  CB::pStart('fragment2');
  // The second code fragment to profile.
  foreach (range(0, 100000) as $j) $n = $i + $j;
  // Shows execution time of the second code fragment.
  echo 'Fragment2: ' . CB::pStop('fragment2') . '<br />';
}
// Shows execution time of the first code fragment.
echo 'Fragment1: ' . CB::pStop('fragment1');
```

> Method **pStop()** returns FALSE if for the given code fragment method **pStart()** was not invoked.

Execution time and memory allocation can be obtained as follows:

```
// Script execution time in seconds.
echo CB::getExecutionTime();
// Request time to the server.
echo CB::getRequestTime();
// The volume of the script's memory in bytes.
echo CB::getMemoryUsage();
// The maximum volume of allocated memory.
echo CB::getPeakMemoryUsage();
```