

Task 实现报告

介绍

本次作业使用 `ManualResetEvent` 的语义实现了简单的 Task（异步任务），该任务使用泛型，因而支持多种类型的返回值。

接口

```
// 构造函数，TResult 为任务的返回值类型
public MyTask(Func<Object, TResult> task, Object args);

// 任务是否结束
public bool IsFinish();

// 开始执行任务
public void Start();

// 主动阻塞直到任务完成
public void wait();

// MyTask.result 返回执行结果
```

使用

MyTask 使用与 [Task](#) 方法类似，`MyTask.cs/Program.Main()` 中也提供了对应的例子：

```
// Main Method
class Program {
    static int calculate(Object arg) {
        // 执行一个耗时操作
        Thread.Sleep(1000);

        int num = (int) arg;
        int sum = 0;
        for(int i = 0; i < num; i++) sum += i;
        return sum;
    }

    public static void Main() {
        MyTask<int> task = new MyTask<int>(Calculate, 10);

        Console.WriteLine("开始执行任务");
        task.Start();

        Console.WriteLine("调用 start() 后返回，任务还没执行结束\n-----
        -----");

        task.wait();
        Console.WriteLine("任务执行结束，结果为: {0}", task.result);
    }
}
```

执行结果如下：

```
PS > .\MyTask.exe
```

开始执行任务

调用 `Start()` 后返回，任务还没执行结束

任务执行结束，结果为：45