Чигарев Дмитрий 381807-1

# Лабораторная работа 1.8

# «Барьеры»

В многопоточной программе может возникнуть ситуация, когда для корректной работы одного потока, необходим результат работы другого.

Для того чтобы разом оповестить несколько потоков о завершении работы целевого потока, в стандартной библиотеке Threading есть класс ManualResetEvent. Принцип работы с этим классом следующий: создаются экземпляры класса, олицетворяющие зависимости между потоками, зависимые потоки ограждают барьером операции с результатами целевого потока путем вызова WaitOne() на соответствующей зависимости – поток будет находится в блокировке, пока зависимость не разрешится. В свою очередь от целевого потока требуется вызвать метод Set() на соответствующей зависимости, когда она разрешится:

```
static ManualResetEvent dep1 = new ManualResetEvent(false);
static ManualResetEvent dep2 = new ManualResetEvent(false);

void ThreadWork() {
  calculate1();
  // Оповещает все подписанные потоки о том,
  // что зависимость dep1 разрешена
  dep1.Set();
  calculate2();
  // Оповещает все подписанные потоки о том,
  // что зависимость dep2 разрешена
  dep2.Set();
}

void Work1() {
  // Ожидаем разрешение зависимости dep1
  dep1.WaitOne();
  // Попадем сюда только тогда, когда разрешится dep1
  process_dep1_result();
}

void Work2() {
  // Ожидаем разрешение зависимости dep2
  dep2.WaitOne();
  // Попадем сюда только тогда, когда разрешится dep2
  process_dep2_result();
}

void Work3() {
  // Ожидаем разрешение зависимостей dep1 и dep2
  EventWaitHandle.WaitAll(new EventWaitHandle[] { dep1, dep2 });
  // Попадем сюда только тогда, когда разрешится и dep1 и dep2
  process_dep1_2_result();
}
```

В качестве упражнения найдем корни квадратного уравнения весьма необычным образом: разобьем формулу для корней уравнения на атомарные операции и к каждой операции назначим на выполнение свой поток, в итоге у нас получится дерево зависимостей, которое нам поможет разрешить упомянутая выше механика барьеров.

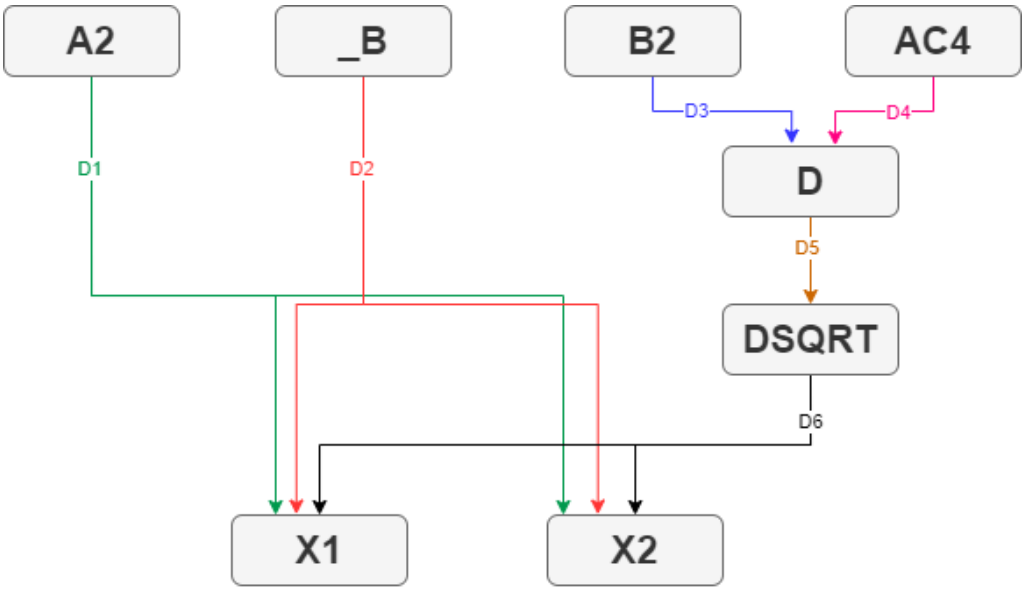
Исходная формула корней квадратного уравнения выглядит следующим образом:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

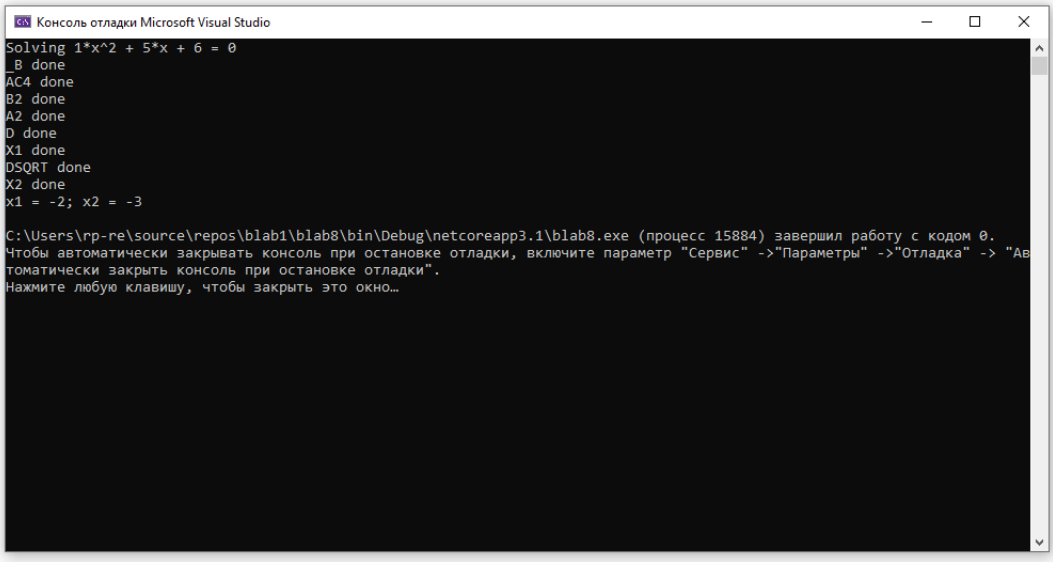
Разобьем её на следующие работы:

$$A2 := 2a$$
$$\_B := -b$$
$$B2 := b^2$$
$$AC4 := 4ac$$
$$D := B2 - AC4$$
$$DSQRT := \sqrt{D}$$
$$X1 := (\_B + DSQRT) / A2$$
$$X2 := (\_B - DSQRT) / A2$$

Естественным образом построим следующее дерево зависимостей:




Теперь у нас есть всё, чтобы написать описанную программу по примеру выше.



# Приложения

1. Исходный код программы: https://github.com/proxodilka/csharp-labs/tree/master/blab8