



(/wiki/Rosetta_Code)

ROSETTACODE.ORG

173.245.53.90 (/wiki/User:173.245.53.90) Talk for this IP address (/wiki/User_talk:173.245.53.90)

Create account (/mw/index.php?title=Special:UserLogin&returnto=Active+object&type=signup)

Log in (/mw/index.php?title=Special:UserLogin&returnto=Active+object)

Search



Page (/wiki/Active_object) Discussion (/wiki/Talk:Active_object)

Edit (/mw/index.php?title=Active_object&action=edit) History (/mw/index.php?title=Active_object&action=history)

Active object

In object-oriented programming (/wiki/Object-oriented_programming) an object is active when its state depends on clock. Usually an active object encapsulates a task (/wiki/Task) that updates the object's state. To the outer world the object looks like a normal object with methods that can be called from outside. Implementation of such methods must have a certain synchronization mechanism with the encapsulated task in order to prevent object's state corruption.

A typical instance of an active object is an animation widget. The widget state changes with the time, while as an object it has all properties of a normal widget.

The task

Implement an active integrator object. The object has an input and output. The input can be set using the method *Input*. The input is a function of time. The output can be queried using the method *Output*. The object integrates its input over the time and the result becomes the object's output. So if the input is $K(t)$ and the output is S , the object state S is changed to $S + (K(t_1) + K(t_0)) * (t_1 - t_0) / 2$, i.e. it integrates K using the trapeze method. Initially K is constant 0 and S is 0.

In order to test the object:

1. set its input to $\sin(2\pi f t)$, where the frequency $f=0.5\text{Hz}$. The phase is irrelevant.
2. wait 2s
3. set the input to constant 0
4. wait 0.5s

Verify that now the object's output is approximately 0 (the sine has the period of 2s). The accuracy of the result will depend on the OS (/wiki/OS) scheduler time slicing and the accuracy of the clock.

Contents

- 1 Ada
- 2 BBC BASIC
- 3 C
- 4 C++
- 5 Clojure
- 6 E
- 7 EchoLisp
- 8 Erlang
- 9 Factor
- 10 F#



(/wiki/Category:Solutions_by_Programming_Task)

Active object

You are encouraged to solve this task

(/wiki/Rosetta_Code:Solve_a_Task) according to the task description, using any language you may know.

```
from time import time, sleep
from threading import Thread

class Integrator(Thread):
    'continuously integrate a function `K`, at each `interval` seconds'
    def __init__(self, K=lambda t:0, interval=1e-4):
        Thread.__init__(self)
        self.interval = interval
        self.K = K
        self.S = 0.0
        self.__run = True
        self.start()

    def run(self):
        "entry point for the thread"
        interval = self.interval
        start = time()
        t0, k0 = 0, self.K(0)
        while self.__run:
            sleep(interval)
            t1 = time() - start
            k1 = self.K(t1)
            self.S += (k1 + k0)*(t1 - t0)/2.0
            t0, k0 = t1, k1

    def join(self):
        self.__run = False
        Thread.join(self)

if __name__ == "__main__":
    from math import sin, pi

    ai = Integrator(lambda t: sin(pi*t))
    sleep(2)
    print ai.S
    ai.K = lambda t: 0
    sleep(0.5)
    print ai.S
```

Racket (/wiki/Category:Racket)