```
R"""An timed counter class                                                         1
```

Author : Anders Andersen
Created On : Thu Apr 22 12:02:06 1999
Last Modified By: Anders Andersen
Last Modified On: Thu Apr 22 14:49:36 1999
Status : Unknown, Use with caution!

This module implements an timed counter class `TimedCounter`. This example tries to illustrate how it works (a closer description is found in the documentation of the class):

```
>>> c = TimedCounter(30, 3)
>>> print c.val
3
>>> c.add(2)
>>> print c.val
5
>>> print c.val     # 30 seconds later
3
```

```
"""
                                                                                   28
# We need to do some timing                                                        29
import time                                                                        30
                                                                                   31
class TimedCounter:                                                                32
    R"""A class for timed counters                                                 33
```

This class implements a special type of counters where added and subtracted values only are valid in a given period. A timed counter has its period set to 30 (seconds) and 1 is added to it. 30 seconds later will 1 be subtracted from the counter. Instances of this class have two (public) attributes `val` and `period`, the value of the counter and the valid period of added and subtracted values respectively. Two methods `add` and `sub` are provided to respectively add a value to the counter and to subtract a value from the counter.

```
    """
                                                                                   47
    def __init__(self, period, val=0):                                            48
        R"""Initialise a timed counter                                             49
```

Set the initial state of a timed counter. This includes its valid period (the `period` argument) and its initial value (the optional `val` argument).

```
        """
        self.__dict__["period"] = period                                          56
        self.__dict__["_val"] = val                                               57
        self.__dict__["_addlist"] = []                                            58
                                                                                   59
    def add(self, v):                                                             60
        R"""Add a value to the counter                                             61
```

Add the value v to the counter. This value will automatically be subtracted from the counter after `period` seconds.

```
        """
        self.__dict__["_val"] = self.val + v                                      68
        self._addlist.append((time.time()+self.period, v))                        69
                                                                                   70
    def sub(self, v):                                                             71
```

R"""Subtract a value to the counter   72

Subtract the value v from the counter. This value will automatically be added to the counter after `period` seconds.

```
"""
self.add(-v)                                                                                             79
                                                                                                         80
```

```python
def __getattr__(self, key):                                                                              81
    R"""Fetching the counter value                                                                       82
```

The counter value is stored in the attribute `val`. The trick is that this attribute doesn't exists in the object and every attempt to read it will end up calling this method. This method returns the value, but it also removes any additions and subtractions that are not valid any more.

```python
    """
    if key == "val":                                                                                     91
        c = time.time(); n = 0                                                                           92
        for (t,v) in self._addlist:                                                                      93
            if t > c:                                                                                    94
                break                                                                                    95
            else:                                                                                        96
                n = n + 1                                                                                97
                self.__dict__["_val"] = self._val - v                                                    98
        self.__dict__["_addlist"] = self._addlist[n:]                                                    99
        return self._val                                                                                 100
    else:                                                                                                101
        raise AttributeError, key                                                                        102
                                                                                                         103
def __setattr__(self, key, v):                                                                           104
    R"""Change attribute value                                                                           105
```

You can change the value of `val` and `period`, but any attempt to change the value of other attributes will generate an `AttributeError`.

```python
    """
    if key == "val":                                                                                     112
        self.__dict__["_val"] = v                                                                        113
    elif key == "period":                                                                                114
        self.__dict__["period"] = v                                                                      115
    else:                                                                                                116
        raise AttributeError, key                                                                        117
```