Write an iterator class that takes in a list and calculates the sum of the list thus far.

Solution to part1:

```
class Accumulator:
     def __init__(self, lst):
          self.lst = lst
          self.index = 0
          self.sum = 0
     def __next__(self):
          if self.index >= len(self.lst):
               raise StopIteration()
          self.sum += self.lst[self.index]
          self.index += 1
          return self.sum
     def __iter__(self):
          return self
```

what we need to change for the extra credit

Solution to part 2:

In the above solution we use facts we know to be true about lists to iterate over one. Now we **replace lst with iterable**. What do we know to be true about an iterable?

1. It has an iter function that returns an iterator.
2. We can call next on an iterator to get the next element.
3. Iterators keep track of when there is nothing to iterate over.

```
class Accumulator:
     def __init__(self, iterable):
          Self.iterator = iter(iterable)
          self.sum = 0
     def __next__(self):
          // no need to call raise StopIteration
          self.sum += next(self.iterator)
          return self.sum
     def __iter__(self):
          return self
```