

From engine overloading to SPL

Filtering with Iterators

Debug Session

Marcus Börger

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```



```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```



```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

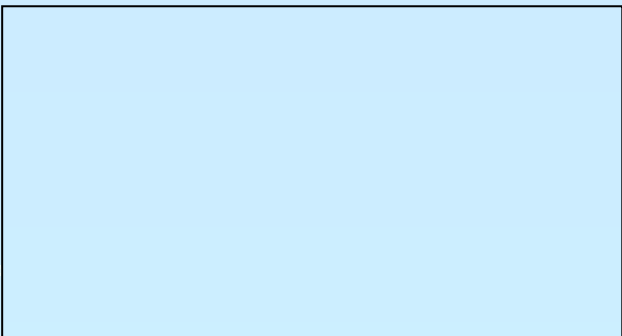
```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```



```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

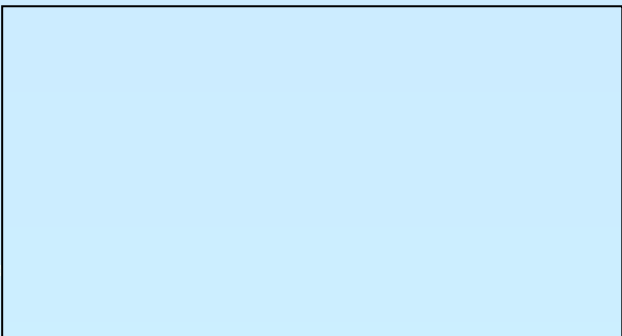


```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```



```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```



```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```



```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
1 => 2
3 => 8
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
3 => 8
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```



```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
3 => 8
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
3 => 8
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val \n";
}
?>
```

```
1 => 2
3 => 8
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```

```
<?php
$a = array(1, 2, 5, 8);
$i = new EvenFilter(new MyIterator($a));
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
1 => 2
3 => 8
```

```
<?php
class EvenFilter extends FilterIterator {
    function __construct(Iterator $i) {
        parent::__construct($i); }
    function accept() {
        return $this->current() % 2 == 0; }
}
class MyIterator implements Iterator {
    function __construct($ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function current() {
        return current($this->ar); }
    function key() {
        return key($this->ar); }
    function next() {
        next($this->ar); }
}
?>
```