Abstrakcyjne struktury danych w PHP

PHP ze względu na swoją specyfikę (brak wskaźników) nie ułatwia tworzenia abstrakcyjnych struktur danych, takich jak drzewa, grafy (możemy opierać się tylko na tablicach). Tworząc aplikacje internetowe relatywnie rzadko istnieje potrzeba użycia bardziej zaawansowanych struktur danych (Ba! Programiści piszący tylko w PHP w znacznej części nie słyszeli nigdy o B-Drzewach, drzewach AVL, grafach i innych tego typu strukturach). Mimo tego warto zaznajomić się z typami struktur jakie oferuje podstawowa biblioteka SPL.

Lista dwukierunkowa

Lista dwukierunkowa jest jedną z najprostszych struktur. Umożliwia przemieszczanie się po liście do przodu oraz do tyłu. Dlaczego do tego nie użyć zwykłych tablic? W PHP tablice są dynamiczne, więc nie ogranicza nas ich stały rozmiar. Jednak, jeżeli chcielibyśmy wstawić element pośrodku, musielibyśmy przesuwać pozostałe elementy tablicy co nie jest zbyt wygodne.

Jeżeli chcemy stworzyć listę dwukierunkową wystarczy zainicjować obiekt klasy ***SplDoublyLinkedList***.

Kolejka

Kolejka rożni się od listy tylko tym, że dane pobierane są z początku kolejki, a zapisywane na końcu (bufor typu **FIFO**, *First In, First Out*; *pierwszy na wejściu, pierwszy na wyjściu*).

Jeżeli chcemy stworzyć kolejkę wystarczy zainicjować obiekt klasy ***SplQueue***.

Kolejka priorytetowa

Kolejka priorytetowa jest szczególnym przypadkiem kolejki. Dodając element podajemy wartość priorytetu, według którego kolejka jest uporządkowana. Na wyjściu mamy element o najwyższym priorytecie.

Jeżeli chcemy stworzyć kolejkę priorytetową wystarczy zainicjować obiekt klasy ***SplPriorityQueue***.

Stos

Stos jest strukturą danych, w której dane dokładane są na wierzch stosu i z wierzchołka stosu są pobierane (bufor typu **LIFO**, Last In, First Out; ostatni na wejściu, pierwszy na wyjściu). Ideę stosu danych można zilustrować jako stos położonych jeden na drugim talerzy – nowy talerz kładzie się na wierzch stosu i z wierzchu stosu zdejmuje się kolejne talerze. Elementy stosu poniżej wierzchołka można wyłącznie obejrzeć, aby je ściągnąć, trzeba najpierw po kolei ściągnąć to, co jest nad nimi.

Jeżeli chcemy stworzyć stos wystarczy zainicjować obiekt klasy ***SplStack***.

Kopiec

Kopiec jest nieco bardziej zaawansowaną strukturą opartą na drzewie. W kopcu wartości potomków węzła są w stałej relacji z wartością rodzica (na przykład wartość rodzica jest nie mniejsza niż wartości jego potomka).

Jeżeli chcemy stworzyć kopiec wystarczy zainicjować obiekt klasy ***SplMaxHeap*** lub ***SplMinHeap***.

SplFixedArray

SplFixedArray rożni się od zwykłych tablic PHP tym, że przy tworzeniu obiektu podajemy rozmiar tablicy. Ponadto indeksami mogą być tylko liczby (nie jest to tablica asocjacyjna). Według zapewnień twórców jest to szybsza implementacja od „tradycyjnych" tablic w PHP.

SplObjectStorage

SplObjectStorage jest strukturą do gromadzenia obiektów.

Przykłady

```
<?php
$list = new  SplDoublyLinkedList(); // tworzy obiekt listy

$list->push(1); // dodaje elementy
$list->push(7);
$list->push(2);
$list->push(3);

$list->rewind(); // ustawia iterator na start
while($tmp=$list->current()) {
    echo $tmp."\n";
    $list->next(); // przesuwa w prawo o jeden
}
?>
```

```
<?php
$queue = new SplPriorityQueue(); // tworzy obiekt kolejki

$queue->insert("Akcja 1", 2); // dodaje elementy
```

```
$queue->insert("Akcja 2", 3);
$queue->insert("Akcja 3", 1);


foreach ($queue as $temp) {
    echo $temp."\n";
}
?>
```

```
<?php
$stack = new SplStack();

$stack[] = 1;
$stack[] = 2;
$stack[] = 3;

foreach($stack as $elem) {
    echo $elem."\n";
}
?>
```

```
<?php
$s = new SplObjectStorage();

$o1 = new StdClass;
$o2 = new StdClass;
$o3 = new StdClass;

$s->attach($o1);
$s->attach($o2);

var_dump($s->contains($o1));
var_dump($s->contains($o2));
var_dump($s->contains($o3));

$s->detach($o2);
?>
```