

```

37 ~ } else {
38 ~     echo "User not found in: " . $filename;
39 ~ }
40 ~ }
41 ~
42 ~ class vdXCoinInfo {
43 ~     public $name;
44 ~     public $ticker;
45 ~     public $start;
46 ~     public $last;
47 ~     public $change;
48 ~     public $changeProcent;
49 ~     public $pointsStart;
50 ~     public $coinWeightingStart;
51 ~     public $pointsLast;
52 ~     public $pointsChange;
53 ~     public $pointsChangeProcent;
54 ~     public $spaceShipOperatorChange;
55 ~     public $lastStatus;
56 ~     public $logo;
57 ~     public $timeStamp;
58 ~     public $currentWeightingProcent;
59 ~     public $short;
60 ~
61 ~     public function __construct($name, $ticker, $start, $lastApi,$pointsStart, $logo, $short = false){
62 ~         $this->name = $name;
63 ~         $this->ticker = $ticker;
64 ~         $this->start = $start;
65 ~         $this->last = $lastApi;
66 ~         $this->change = $this->Change($this->last, $this->start);
67 ~         $this->changeProcent = $this->ChangeProcent($this->last, $this->start);
68 ~         $this->pointsStart = $pointsStart;
69 ~         $this->logo = $logo;
70 ~         $this->short = $short;
71 ~         $this->timeStamp = gmdate(DATE_RFC822);
72 ~         // Check if this is a short position
73 ~         if($this->short) {
74 ~             // For a short position, flip the change
75 ~             $this->change = $this->Change($this->start, $this->last); // Notice the order is switched
76 ~             $this->changeProcent = $this->ChangeProcent($this->start, $this->last); // Notice the order is switched
77 ~         } else {
78 ~             // Regular position
79 ~             $this->change = $this->Change($this->last, $this->start);
80 ~             $this->changeProcent = $this->ChangeProcent($this->last, $this->start);
81 ~         }
82 ~
83 ~         $this->coinWeightingStart = $this->coinWeightingStart($this->pointsStart, $this->start);
84 ~         $this->pointsLast = $this->LastPoints($this->coinWeightingStart, $this->last);
85 ~         $this->pointsChange = $this->ChangePoints($this->pointsLast, $this->pointsStart);
86 ~         $this->pointsChangeProcent = $this->pointsChangeProcent($this->pointsChange, $this->pointsStart);
87 ~         $this->spaceShipOperatorChange = $this->change <=> 0;
88 ~         $this->lastStatus = $this->last ? "true" : "false";
89 ~
90 ~     }
91 ~
92 ~
93 ~     public function Change(float $last, float $start): float {
94 ~         return $last - $start;
95 ~     }
96 ~
97 ~     public function ChangeProcent(float $last, float $start): float {
98 ~         return (($last - $start) / abs($start)) * 100;

```