### Atrybuty

* Każdy z nich posiada:
  + wartość,
  + id: nazwę oraz domenę – nazwę drugy atrybutów do które należy

|  |  |  |
| --- | --- | --- |
| **Atrybut** | **Klasa** | **Kod** |
| **Nazwa** | UnitName | name |
| **Opis** | UnitDescrption | description |
| **Doświadczenie** |  | currentXp nextLvlXp |
| **Poziom** |  | level |
| **Rodzaj** | UnitType | type  W – wojownik, R – strzelec, S – wsparcie, M – Mag,  **uwaga**: nie jest wczytywany z properties,  a wyznaczany na pods. *zasięgu* oraz *liczby celi* |
| **Punkty życia** | UnitHitPoints | hitPoints |
| **Max punkty zycia** | UnitMaxHitPoints | maxHitPoints |
| **~~Odporność pancerza~~** | ~~UnitArmorDurability~~ | ~~armorDurability określa liczbę punktów obrażeń, które może przyjąć pancerz po których jego ochrona obniży się o 1%~~ |
| **Odporność** | UnirResistance**X** | resistance**X**  **X** = Fire, Water, Air, Death, Poison, Mind, Life, Earth, |
| **Atak** |  | source Fire, Water, Air, Death, Poison, Mind, Meele, Projectile, Life, Earth  **uwaga**: MEELE = PROJECTILE, rozróżnienie istnieje tylko dla sprawdzania *pancerza* |
| **Celność** |  | accuracy |
| **Obrażenia** | UnitDamage | damage |
| **Inicjatywa** |  | initiative |
| **Zasięg** |  | range |
| **Cele** |  | targets |
| **Ruch** |  | stamina |
| **„Gniew”** |  | specialPoints |

### Modyfikatory wartości atrybutów:

* jako dane wejściowe są opisywane przez liczby
  + całkowite np. 120 = dodać 120, -50 = odjąć 50
  + zmiennoprzecinkowe: 50.0 = +50%, zaś -33.0 = -33%
    - w programie są przechowywane jako liczby: pomiędzy (0;1)
    - w plikach z danymi są zapisywane w postaci X.0 np. 50.0 = 50%
* obsługuje je klasa *AttrbuteValueChangeFactorImpl*

### Walidatory

* każdy obiekt przechowujący dane powinien implementować interfejs *Validatable*

### Powiązanie atrybutów

* implementują interfejs *LinkedAttributesChange*
* każdy z nich przechowuje mapę: atrybut – zamiana w postaci wpółczynnika *double*
* przy zmniejszanu wartości atrybutów powiązanych następuje przeskalowanie zmiany, która uwzględnia różne skale i wartości atrybutów
  + odbywa się w to w *AttributeValueChangeFactorImpl.getPecentageChangeFactor*
* **proces**
  + zmiany odbywa się w metodach *atrybut.updateReferencedAttributes*
  + przebieg:
    - obliczana jest zmiana wartości atrybutu powiązanego w zależności od jego współczynnika zmiany (*Linked...*) oraz zmiany wartości atrybutu głównego
    - pobierana jest obecna wartość atrybutu powiązanego
    - jego wartość jest aktualizowana przed współczynnik (...*Factor*)