

# Projektowanie i programowanie obiektowe

## Wielokrotne dziedziczenie klas

mgr inż. Krzysztof Rewak

6 listopada 2019

### 1 Wielokrotne dziedziczenie klas w programowaniu obiektowym

Należy zbudować i uruchomić załączony plik `lab06.py` z dodatkowym plikiem `force_users.py`. Proponowane środowisko programistyczne to PyCharm w wersji Community Edition z Pythonem w wersji 3; można ewentualnie uruchomić internetowy sandbox dostępny pod adresem <https://repl.it/>. Należy przeanalizować kod, a następnie odpowiedzieć na pytania:

- jak wygląda hierarchia dziedziczenia w przykładowym programie?
- czym się różni dziedziczenie w Pythonie od dziedziczenia w PHP, Javie czy C#?
- czym jest funkcja `__init__()`?
- ilu różnych form ataku może użyć lord Sithów?
- czym jest funkcja `super()`?
- jakie mogłoby być uzasadnione wykorzystanie przykładowego programu?

### 2 Zadanie programistyczne

Należy rozszerzyć program z pliku `lab06.py` o następujące funkcjonalności:

- program pozwala wygenerować dowolną liczbę uczestników; generowani uczestnicy mogą mieć losowe nazwy lub po prostu numery identyfikacyjne;
- należy dodać element losowości do siły ataku; można założyć poziom szumu na poziomie  $\pm 10\%$ ;
- należy dodać możliwość bronięcia się przed atakiem; proponowane rozwiązanie to utworzenie dodatkowej klasy `Defence` z wartością wyrażoną w procentach oznaczającą możliwość wyjścia bezpiecznie z sytuacji; przy ataku uczestnika A na uczestnika B sprawdzana byłaby możliwość obrony uczestnika B i na jej podstawie atak by się powiódł, częściowo powiódł lub chybił;
- należy stworzyć system zapamiętywania w jakiej kolejności który uczestnik *odpadł* z zawodów; chronologiczna lista powinna wyświetlić się przy zakończeniu programu.

Ambitni studenci mogą spróbować rozszerzyć program w taki sposób, aby konkretny wybrany uczestnik był sterowany przez osobę uruchamiającą program. Takie sterowanie mogłoby polegać na wyborze ataku podczas tury uczestnika.

Plik z programem należy dołączyć do repozytorium Git. Zalecane jest uporządkowanie zadań w odpowiadającym im katalogach.