Заняття №23. ООП. Поліморфізм

Корисні посилання:
https://www.programiz.com/python-programming/polymorphism

Правила здачі:

- Назва файлу має відповідати назві завдання (програми).
- Форматування файлу має відповідати **PEP8**.
- Файли з рішеннями потрібно запушити на ваш git репозиторій з назвою **itstep**, в окрему папку з назвою **lesson23**. Репозиторій має бути публічним, як це перевірити можна дізнатися в офіційній документації вашого git провайдера, або за допомогою Google.
- Програма ні в якому разі не має закінчуватися помилкою та завершитися будь-яким статусом окрім 0.
- якщо ви хочете тестувати код в тому ж файлі, що пишете помістіть виклики в конструкцію
  if __name__ == “__main__”:
  …
- Посилання на репозиторій помістіть в окремий файл під назвою **solution_lesson23_<your_login>** де **<your_login> це ваш студентський логін** та прикріпіть як рішення в mystat. Уважно перевірте, що ви прикріпляєте саме рішення до поточного заняття!
- за будь-яку невідповідність до вищезазначених пунктів оцінка знижується.

Завдання.

1. Напишіть клас **OrcArmy**, який містить такі поля (повинні задаватися при ініціалізації):
   a. warrior amount
   b. damage per warrior
   c. warrior health

   Реалізуйте перегрузку арифметичних операторів для операцій **+** та **-**. При додаванні створюється новий об’єкт в якому warrior amount це сума двох значень кількість бійців кожного об’єкту, які сумуються, damage per orc стає зважене

середнє арифметичне (але вам потрібно враховувати скільки було орків в кожній армії, щоб вирахувати його вірно) та warrior health також середнє арифметичне. В разі віднімання ми віднімаємо кількість воїнів, якщо число рівне або менше нуля то повертається армія без воїнів(warrior amount = 0).

2. Розширте попередній клас та додайте методи, пошкодження та атаки.
   Метод **receive_damage(damage: int)** приймає число скільки одиниць пошкодження на них кидають та повертає число своїх воїнів, що померли, а також оновлює поле з кількістю воїнів в армії (warrior amount). А прорахувати скільки воїнів померло можна за допомогою ділення атаки на параметр warrior health, що вказує скільки одиниць життя має один воїн.
3. Напишіть клас **ElfArmy,** він не успадковується від попереднього класу, це свій незалежний клас. Він має спільні параметри, що і в класі OrcArmy за виключенням додаткового параметру **shield.** Який дає перевагу цій армії перед армію Орків.
   В цьому класі при вирахуванні методу **receive_damage()** перед тим як починати рахувати померлих воїнів, потрібно відняти від атаки параметр shield який вказує на скільки одиниць армія відражає атаку без втрат.
4. Написати клас **Army,** для того щоб оптимізувати код. І від нього спадкувати класи OrcArmy та ElfArmy.
5. Напишіть програму **army_destroy.py** в якій створіть декілька армій кожного типу і об’єднайте їх в один список. Потім дайте користувачу вносити число на яку кількість одиниць атакувати одночасно всі армії. Після чого виводьте детальний звіт по кожній з армій (в разі якщо бійців в армії не залишилось відображати, що цю армію розбито).