

Laboratorium z przedmiotu Programowanie Obiektowe — lista 6

Programowanie obiektowe — podsumowanie

Piotr Lechowicz

1 Cel zajęć

Zastosowanie dziedziczenia i polimorfizmu.

2 Informacje wstępne

Do rozwiązania zadań można wykorzystać zaimplementowane rozwiązanie listy 5.

W celu stworzenia projektu w MS Visual Studio na podstawie plików źródłowych należy wykonać:

- *File / New / Project From Existing Code...*
- W *What type of project would you like to create?* ustawić *Visual C++*
- *Next*
- Podać lokalizację plików i wybrać nazwę projektu
- *Next*
- W *Project Type* ustawić *Console Application Project*
- *Next/Next/Finish*

3 Zadania

1. Stworzyć klasę `Weapon` zawierającą pole prywatne:
`name : std::string.`
2. Stworzyć konstruktor jednoargumentowy:
`Weapon(const std::string &name)`

3. Stworzyć metodę wirtualną z dostępem chronionym:
`int getAdditionalDamage()`.
Metoda w klasach pochodnych będzie określać ile dana broń zadaje obrażeń.
4. Stworzyć i zaimplementować metodę wirtualną z dostępem publicznym:
`int getDamage(int playerStrength, Monster & monster)`
Podczas wyliczania obrażeń należy uwzględnić siłę gracza, obronę potwora oraz dodatkowe obrażenia wynikające z typu broni.
5. Stworzyć 3 klasy dziedziczące po klasie `Weapon`. Np. `Fist` (pięść), `Sword` (miecz), `Bow` (łuk). Klasy powinny definiować metodę `getAdditionalDamage`.

1 punkt

6. W klasie `Player` stworzyć pole:
`weapon : Weapon *`.
Pole określa jaką bronią dysponuje bohater.
7. Stworzyć mutatora (settera) do tego pola. Mutator pozwala zmienić rodzaj wykorzystywanej przez bohatera broni.

2 punkty

8. Stworzyć metodę:
`void Player::attack(Monster &monster)`
W metodzie należy sprawdzić ilość zadanych obrażeń przez posiadaną broń. Jeżeli liczba zadanych obrażeń jest większa od zera, należy odpowiednio zredukować życie potwora.
(W tym momencie jest wykorzystywany polimorfizm — odwołujemy się do instancji klasy pochodnej (np. `Sword`) przez wskaźnik do klasy bazowej (`Weapon`)). Dlatego zostanie wywołana przeciążona metoda z klasy pochodnej, a nie klasy bazowej.
9. W klasie `Monster` stworzyć metodę:
`void Monster::attack(Player &player)`
W metodzie należy sprawdzić ilość zadanych obrażeń (uwzględnić siłę potwora i obronę bohatera). Jeżeli liczba zadanych obrażeń jest większa od zera, należy odpowiednio zredukować życie bohatera.

3 punkty

10. W pliku `main` stworzyć funkcję:
`combatLoop(Player &player, Monster &monster)`
w której wykonuje się walka.

W trakcie walki najpierw gracz zadaje obrażenia potworowi, a następnie potwór zadaje obrażenia graczowi. Walka trwa do momentu aż zginie albo potwór albo bohater. Po każdej iteracji należy sprawdzić stan życia gracza i potwora.

4 punkty

11. Dodać ograniczenia do wykorzystywanej broni. Przykładowo:

- **Miecz** nie zadaje żadnych obrażeń potworom z klasy **Smok**;
- **Łuk** nie zadaje żadnych obrażeń potworom z klasy **Duch**;
- Podczas ataku **Pięścią** istnieje 50% prawdopodobieństwo, że atak się nie powiedzie.

5 punktów