

Laboratorium z przedmiotu Programowanie Obiektowe — lista 5 Dziedziczenie

Piotr Lechowicz

1 Cel zajęć

Poznanie mechanizmu dziedziczenia oraz funkcji wirtualnych.

2 Informacje wstępne

Do rozwiązania zadań można wykorzystać zaimplementowane rozwiązanie listy 4.

3 Zadania

1. Stworzyć klasę `Potion` zawierającą pole prywatne:
`name : std::string`.
2. Zaimplementować konstruktor jednoargumentowy przyjmujący nazwę eliksiru jako argument i konstruktor kopiujący.
3. Dodac destruktor.
4. Stworzenie następujących klas dziedziczących publicznie po klasie `Potion`:
 - **HealthPotion** — klasa ma posiadać zmienną całkowitoliczbową określającą moc działania eliksiru. Klasa powinna posiadać konstruktor przyjmujący jako parametr nazwę i moc działania eliksiru. Należy jawnie wywołać konstruktor jednoargumentowy klasy bazowej. Następnie należy dopisać jawny destruktor.
 - **StrengthPotion** — klasa ma posiadać zmienną całkowitoliczbową określającą czas działania eliksiru. Klasa powinna posiadać konstruktor przyjmujący jako parametr nazwę eliksiru i czas działania eliksiru. Należy jawnie wywołać konstruktor jednoargumentowy klasy bazowej. Następnie należy dopisać jawny destruktor.

5. W każdym konstruktorze i w destruktorze z klasy bazowej i pochodnych wypisać na ekranie konsoli informację o jego wywołaniu — należy wypisać sygnaturę konstruktora (prototyp) / destruktora.
6. W funkcji `main` stworzyć po jednym obiekcie każdej ze zdefiniowanych klas (`Potion`, `HealthPotion`, `StrengthPotion`). Przeanalizować za pomocą wypisów w konsoli kolejność wywołań konstruktorów i destruktorów podczas tworzenia instancji klasy bazowej i klas pochodnych.

1 punkt

7. W każdej klasie (bazowej i pochodnych) zdefiniować metodę `void printPotionType()` wypisującą informację o eliksirze:
 - Funkcja `void printPotionType()` z klasy `Potion` wypisze: "Eliksir nieznany".
 - Funkcja `void printPotionType()` z klasy `HealthPotion` wypisze: "Eliksir życia, moc=%moc%", gdzie "%moc%" odnosi się do wartości zmiennej określającej moc eliksiru.
 - Funkcja `void printPotionType()` z klasy `StrengthPotion` wypisze: "Eliksir siły, czas działania=%czas_działania%", gdzie "%czas_działania%" odnosi się do wartości zmiennej określającej czas działania eliksiru.
8. W funkcji `main` stworzyć po jednym obiekcie każdej ze zdefiniowanych klas (`Potion`, `HealthPotion`, `StrengthPotion`). Następnie należy stworzyć trzy wskaźniki typu `Potion` służące do wskazywania na te obiekty.
9. Wywołać metodę `printPotionType()` na rzecz każdego obiektu (poprzez zmienną) i przeanalizować z której klasy zostaje wywołana metoda.
10. Wywołać metodę `printPotionType()` za pomocą każdego z trzech stworzonych wskaźników i przeanalizować z której klasy zostaje wywołana metoda.

2 punkty

11. Zmienić funkcję `printPotionType()` w klasie `Potion` na wirtualną. (dodać słowo kluczowe `virtual` do sygnatury funkcji). Sprawdzić czy wyniki z punktów 9 i 10 uległy zmianie.

3 punkty

12. Zmienić destruktor klasy bazowej na destruktor wirtualny.
13. Zadeklarować klasę `Potion` jako czysto wirtualną. Wprowadzić odpowiednie zmiany w funkcji `main`.

4 punkty

14. Stworzyć analogiczne dwie klasy dziedziczące po klasie `Item`.

5 punktów