

Lista 10

Programowanie obiektowe II – Scala. Wariantność i polimorfizm ograniczeniowy w językach Scala i Java

Podczas realizacji zadań należy pamiętać o podstawowych zasadach tworzenia kodu obiektowego. Oznacza to **adekwatne** wykorzystywanie modyfikatorów dostępu, getterów oraz seterów, właściwości, klas abstrakcyjnych oraz cech.

Należy wykorzystywać fakt, że **argumenty konstruktora głównego** w Scali mogą od razu stawać się **polami** danej klasy.

Każde zadanie musi posiadać **kompletny zestaw testów**.

Do wykonania zadań należy wykorzystać mechanizmy poznane na wykładach nr 9 i 10.

- 1) Wykorzystując mechanizmy poznane na wykładach 10 i 11 zamodeluj poniższy wycinek rzeczywistości: (Scala)

„W pewnej sekretnej organizacji, w systemie komputerowym przechowywane są sekrety. Dostęp do systemu mają użytkownicy, którzy posiadają dwa rodzaje uprawnień – na odczyt oraz na zapis informacji. Dostęp do systemu odbywa się za pomocą specjalnych terminali komputerowych odpowiedzialnych za przechowywanie poszczególnych sekretów. Analogicznie jak użytkownicy, terminale posiadają wymagania opisujące minimalne uprawnienia do odczytu i zapisu sekretów, przy czym wymagane jest, aby dany terminal posiadał ograniczenia na zapis przynajmniej na tym samym poziomie co odczyt. Obecnie firma posiada trzy poziomy uprawnień: Niskie, Wysokie oraz Superuprawnienia.”

- Zdefiniuj hierarchię cech: *Low* >: *High* >: *Super*, reprezentujących uprawnienia, (5 pkt.)
- Zdefiniuj cechę *User* reprezentującą użytkownika systemu. Cecha ta ma być parametryzowana uprawnieniami na odczyt i uprawnieniami na zapis oraz umożliwiać odpowiednio przekazanie sekretu do terminala oraz „przechwycenia” sekretu z terminala. Proces ten ma odbywać się poprzez „pole” *secret* : *String*, które jednak nie zajmuje w tej cesze pamięci i jest abstrakcyjne. Zdefiniuj parę klas przykładowych użytkowników, (20 pkt.)
- Zdefiniuj klasę *Terminal* reprezentującą terminal systemu. Typ *Terminal* jest parametryzowany dwoma typami minimalnych uprawnień na odczyt/zapis zgodnie z treścią opisu. Podczas tworzenia obiektu terminala przekazywany jest początkowy sekret, który może być modyfikowany przez użytkowników za pośrednictwem metod *read* i *write* klasy *Terminal*. Obie metody mają przyjmować obiekt użytkownika, który będzie dokonywał operacji. Wywołanie metod ma być możliwe jedynie, gdy użytkownik spełnia wymóg adekwatnego poziomu danego uprawnienia. (25 pkt.)