



Nowoczesne języki programowania obiektowego

Wzorzec Singleton

Singleton

- **Ogólnie**

Tylko jedna instancja obiektu danej klasy.

- **Po co?**

- Jedno połączenie z bazą danych
- Zachowanie konfiguracji aplikacji
- Dziennik zdarzeń
- Synchronizacja wątków
- Bufor dla danych

Singleton

```
1  [- public class Singleton() {
2      private static Singleton inst;
3
4  [-     public Singleton() {
5      -     }
6  [- public static Singleton inst() {
7  [-     if(inst == null) {
8      -         inst = new Singleton();
9      -     }
10     return inst;
11     }
12 [- public String przedstawSię() {
13     return "I am the almighty Singleton!";
14     }
15 [- }
```

Singleton

- Z racji tego, że było „static”, to korzystamy tak:

```
Singleton.inst().przedstawSię();
```

Singleton – lepszy kod 😊

```
1  [-] public class Singleton() {  
2      private static Singleton inst;  
3  
4  [-] private Singleton() {  
5      }  
6  [-] public static synchronized Singleton inst() {  
7  [-]     if(inst == null) {  
8          .....  
9             inst = new Singleton();  
10         }  
11         return inst;  
12     }  
13 [-] public String przedstawSię() {  
14     return "I am the almighty Singleton!";  
15 }  
}
```

Zadanie

- Napisz krótki symulator kasyna korzystając ze wzorca Singleton. Przechowywać w nim będziesz stan kasy w kasynie uaktualniany po każdej wygranej/przegranej użytkownika.
- Zaimplementuj Graczy:
GraczWBlackjacka,
GraczWJednorękiegoBandytę