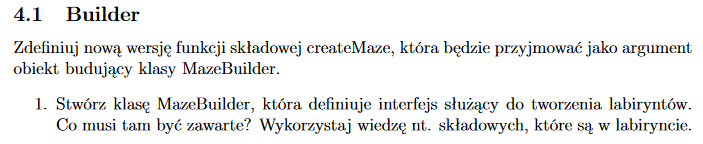
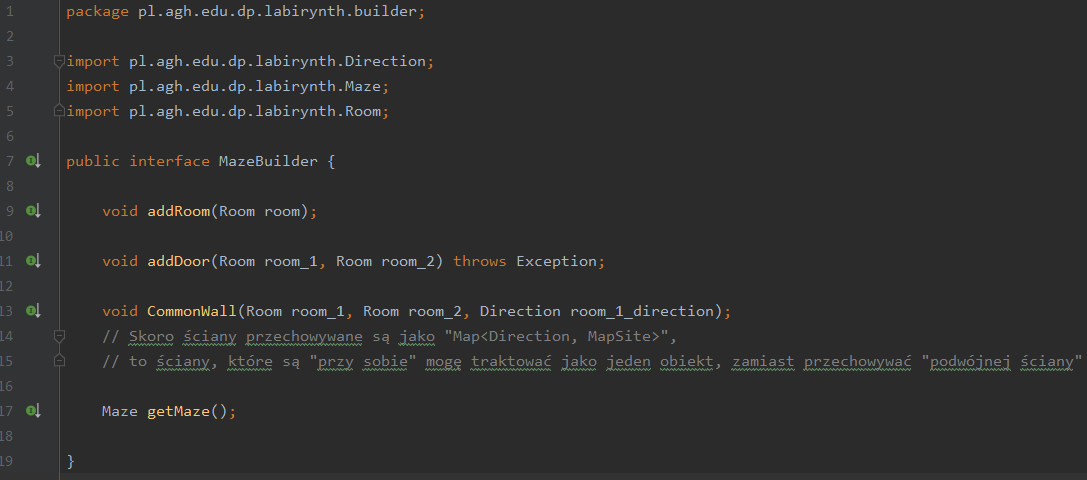
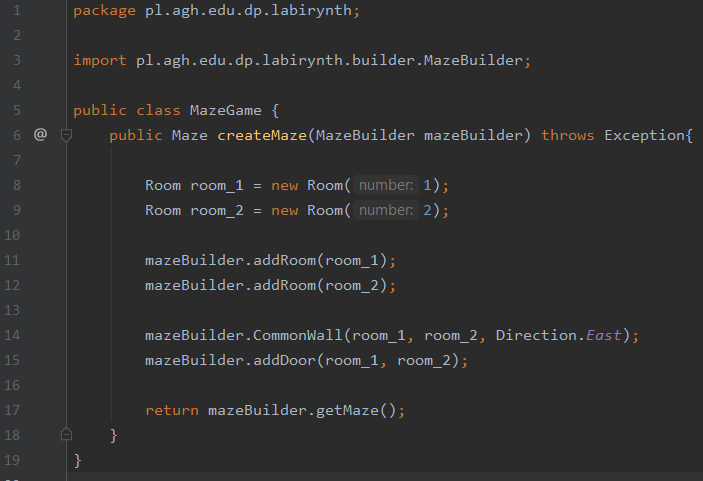
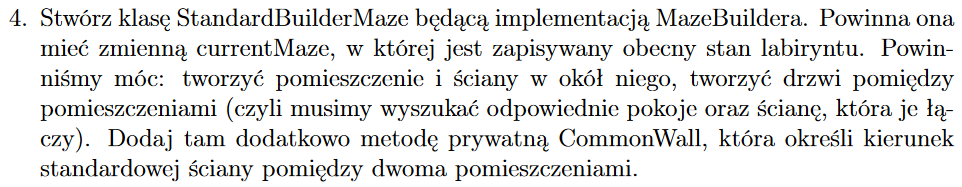
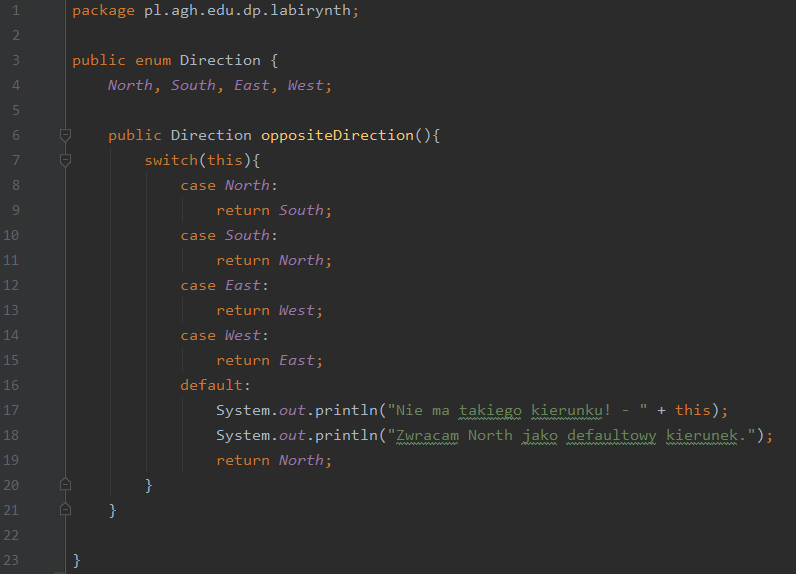
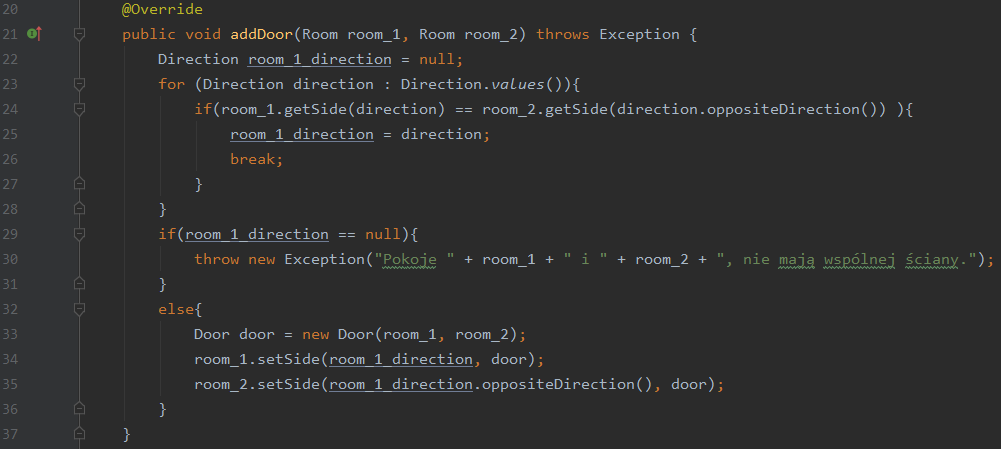
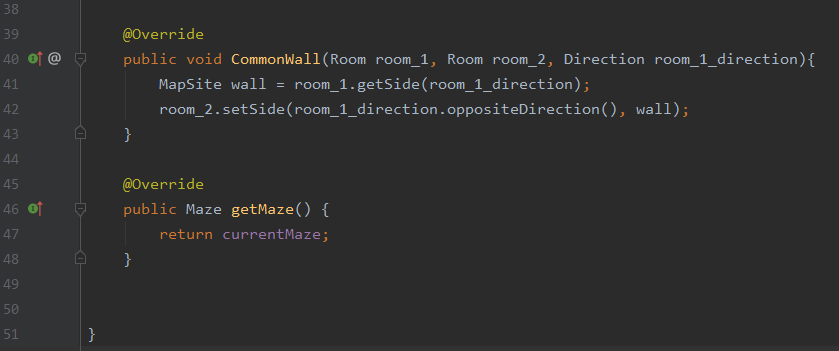
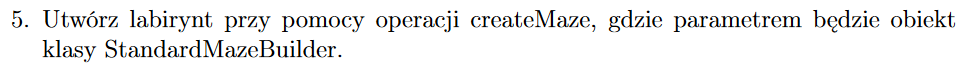
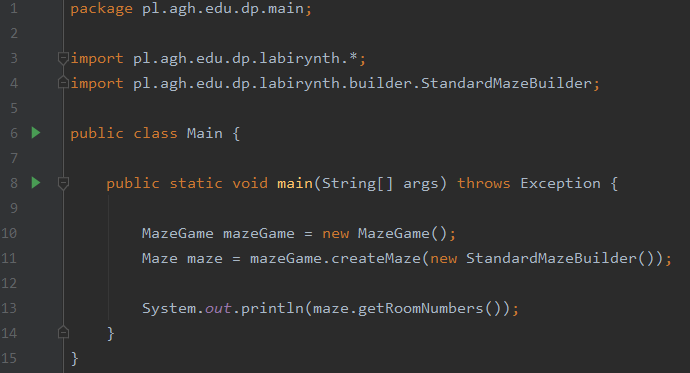
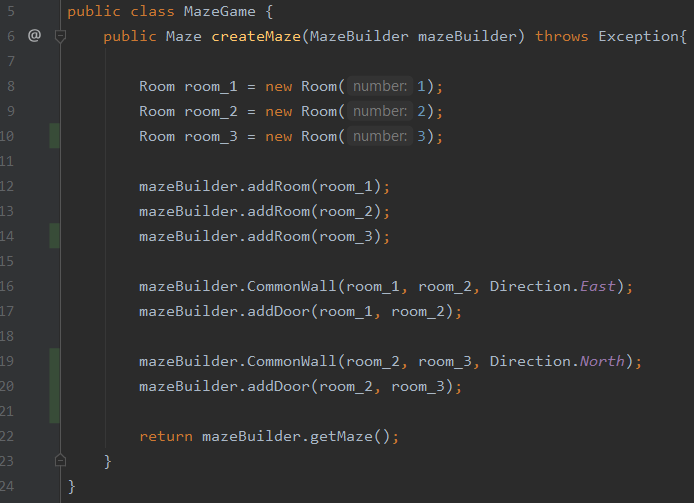
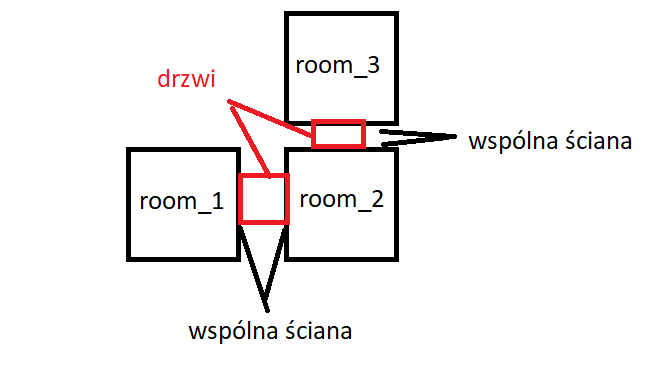
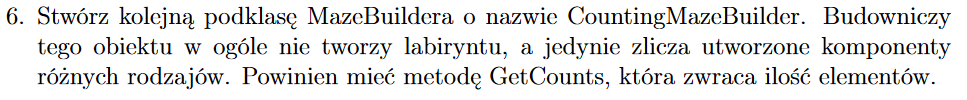
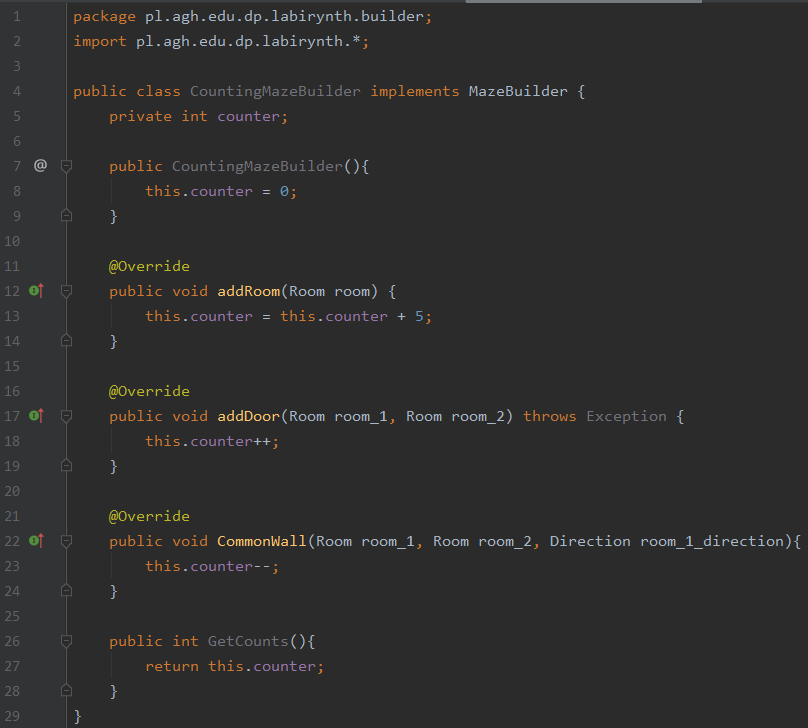
Wojciech Kosztyła  
nr. indeksu 305 338

Projektowanie obiektowe

Laboratorium 3

Wzorce projektowe



1. Stworzyłem interfejs MazeBuilder, a w niej:
   1. dodawanie pokoju
   2. dodawanie drzwi
   3. „uwspółdzielanie” (czynienie wspólną) ściany
      * 1. 
2. 
   * + 1. 
3. 
   1. Wprowadzone zmiany znacznie zwiększyły czytelność kodu.
   2. Zachowanie zasady DRY – Don’t Repeat Yourself
   3. Ułatwienie wprowadzania zmian poprzez „modularyzację”, tworzenie klas implementujących MazeBuilder.
4. 
   1. aby łatwiej implementować te metody, stworzyłem metodę w „Direction”, która zwraca mi przeciwny kierunek
      * 1. 
   2. Następnie wykorzystując metody zdefiniowane w „Room.java” stworzyłem zadane metody.
      1. Nie byłem pewny, co do „dodatkowej metody CommonWall”, która „określi kierunek standardowej ściany pomiędzy dwoma pomieszczeniami”.  
         Pomieszczenia nie są ustawione w macierzy AxB, sam labirynt może być nieskończenie długi i nieregularny, a numery pokojów nie muszą odzwierciedlać ich „sąsiedztwa”.   
         Uznałem, że metoda CommonWall „określi” = „ustawi” wspólną ścianę dla dwóch pomieszczeń.
         1. 
         2. 
         3. 
5. 
   1. Zmieniłem „Main.java”:
      * 1. 
   2. Następnie zmodyfikowałem „MazeGame.java”, aby tworzyła nieco bardziej „skomplikowany” labirynt.
      * 1. 
   3. Szkic labiryntu:
      * 1. 
6. 
   * + 1. 



1. Dodałem