

Wzorce Projektowe, lab. 4

Temat: Wykorzystanie wzorca **Template Method** do tworzenia szablonu algorytmu i rozwijania takiego szablonu w klasie pochodnej za pomocą dziedziczenia.

1. Zaimplementuj klasę **Game** zgodnie z kodem ze slajdu 1 pliku *template_metod.pdf*. Pogrubione nazwy metod powinny być dopisane do klasy, zadeklarowane jako abstrakcyjne i nie zwracać wartości, z wyjątkiem **gameOver()**, która ma zwracać **boolean'a**.
2. Odziedzicz z klasy **Game** klasę **Zgadywanka**, która musi implementować wszystkie wymienione wyżej metody abstrakcyjne. Wzorować się można na slajdzie 2. Klasa **Zgadywanka** ma implementować grę o następujących zasadach:
 - a) użytkownik programu gra z komputerem w zgadywanie liczby całkowitej od 1 do 30;
 - b) program losuje liczbę całkowitą za pomocą obiektu klasy **Random**;
 - c) program pobiera od użytkownika jego „strzał” za pomocą obiektu klasy **Scanner** (metoda **nextInt()**);
 - d) jeżeli strzał jest trafny to użytkownik wygrywa grę;
 - e) jeżeli użytkownik zgadywał nieskutecznie 4 razy to przegrywa grę;
 - f) po każdym nieudanym „strzale” program informuje użytkownika czy szukana liczba jest większa czy mniejsza od podanej przez użytkownika.
3. Wszelkiego rodzaju dane pomocnicze czy komunikacja z użytkownikiem powinny się znaleźć w klasie **Zgadywanka**, a nie **Game**; **Game** stanowi jedynie schemat algorytmu, szczegóły mają być w klasie pochodnej.
4. Metoda **initialize()** powinna się zajmować jednokrotną inicjalizacją, czyli np. tworzyć obiekt klasy **Scanner** czy **Random**, przechowywany następnie w polach klasy **Zgadywanka** (bo są one wykorzystywane wielokrotnie [przynajmniej **Scanner**] i nieefektywnie jest je za każdym razem tworzyć).
5. Metoda **paintScreen()** może być wykorzystana do wyświetlania informacji z punktu **f)** zasad gry, a **makeMoves()** do aktualizacji logiki.
6. Metoda **onEnd()** wyświetla komunikat o wygranej/przegranej oraz robi „końcowe porządki” (jeżeli np. gra wykorzystywałaby pliki czy bazy danych, to sensownie byłoby je tu właśnie pozamykać).

UWAGI:

- Zakładamy, że klasa **Game** jest odgórnie narzuconym szablonem, którego nie możemy modyfikować i dostosowywać do swoich potrzeb. Wszystko, co jest wyspecjalizowane dla przypadku naszej konkretnej gry, ma być zaimplementowane wewnątrz klasy **Zgadywanka**.
- Abstrakcyjne metody z klasy bazowej muszą być zaimplementowane w klasie pochodnej (albo też być zadeklarowane jako abstrakcyjne do implementacji w klasie pochodnej do pochodnej).
- Zadanie nie zostanie zaliczone za same poprawne skompilowanie i wykonanie kodu. Gra ma wyjaśnić użytkownikowi zasady, a następnie na bieżąco komunikować się z nim i informować go, co ma teraz zrobić – ma być grywalna.