

Treść zadania do wykonania – aplikacja konsolowa (Java) i testy jednostkowe

Cel zadania

Wykonać aplikację **konsolową w języku Java** implementującą logikę pojedynczej kości do gry oraz przygotować **testy jednostkowe** sprawdzające poprawność działania rzutu kością.

Organizacja pracy

1. Utworzyć folder nazwany **numerem zdającego**.
 2. W folderze utworzyć podfoldery:
 - o konsolowa
 - o testy
-

Część I – Aplikacja konsolowa

Wymagania ogólne

- Program uruchamiany w konsoli.
- Język programowania: **Java (obiekto)**.
- Kod czytelny, poprawnie sformatowany.
- Znaczące nazwy klas, metod i zmiennych (polskie lub angielskie).

Klasa `Kosc`

Zaprogramować klasę `Kosc`, która realizuje logikę działania pojedynczej kości.

Pola klasy

- Publiczne pole **statyczne** przechowujące liczbę utworzonych instancji klasy `Kosc`.
- Publiczne pola obiekto:
 - o kolekcja nazw plików graficznych: `kosc0.png` – `kosc6.png`,
 - o liczba oczek wyrzuconych na kości (liczba całkowita),
 - o identyfikator pliku graficznego odpowiadającego liczbie oczek,
 - o informacja logiczna określająca, czy kość jest dostępna.

Konstruktory

1. **Konstruktor jednoargumentowy** (argument – liczba oczek):
 - o jeśli argument nie należy do zakresu 1–6, ustawić liczbę oczek na 0,
 - o przypisać wartość liczbie oczek i identyfikatorowi pliku,
 - o ustawić kość jako dostępną,

- zwiększyć licznik instancji klasy.
- 2. **Konstruktor bezargumentowy:**
 - wylosować liczbę z zakresu 1–6,
 - przypisać ją liczbie oczek i identyfikatorowi pliku,
 - ustawić kość jako dostępną,
 - zwiększyć licznik instancji klasy.

Metody klasy

- Metoda publiczna, bezparametrowa, nie zwracająca wartości:
 - wykonująca rzut kością **tylko wtedy, gdy kość jest dostępna**,
 - losująca wartość z zakresu 1–6 i aktualizująca pola obiektu.
- Metoda publiczna, bezparametrowa, nie zwracająca wartości:
 - blokująca kość (ustawiająca ją jako niedostępną).
- Metoda publiczna, bezparametrowa, zwracająca tekst:
 - zwracająca liczbę oczek w postaci słownej (np. 3 → „trzy”).

Program główny

- Utworzyć dwa obiekty klasy `Kosc`:
 - jeden za pomocą konstruktora jednoargumentowego (wartość pobrana z klawiatury),
 - drugi za pomocą konstruktora bezargumentowego.
- Po utworzeniu każdego obiektu wyświetlić:
 - liczbę utworzonych instancji klasy,
 - liczbę oczek (liczbowo i słownie),
 - nazwę pliku graficznego odpowiadającego liczbie oczek.
- Komunikaty muszą być czytelne dla użytkownika.

Dokumentacja

- Wykonać zrzuty ekranu dokumentujące działanie programu.
- Zrzuty powinny obejmować cały ekran.
- Zapisać je w folderze `konsolowa` jako `konsola1.png`, `konsola2.png`, itd.
- W folderze `konsolowa` umieścić archiwum `konsola.zip` zawierające projekt, pliki źródłowe, plik wykonywalny (jeśli istnieje) oraz zrzuty ekranu.

Część II – Testy jednostkowe

Wymagania

- Wykonać testy jednostkowe metody realizującej rzut kością.
- Użyć biblioteki testów jednostkowych dla języka Java.

Przypadki testowe

- Sprawdzenie, czy wylosowana wartość mieści się w zakresie 1–6.

- Sprawdzenie, czy wartość kości nie zmienia się, gdy kość jest niedostępna.

Zasady

- Każdy przypadek testowy zrealizować w osobnej metodzie.
- Nazwy metod testowych muszą jasno określać cel testu.
- Uruchomić wszystkie testy.

Dokumentacja testów

- Wykonać zrzuty ekranu z uruchomionych testów.
- Zapisać je w folderze `testy` jako `test1.png`, `test2.png`, itd.
- Zrzuty muszą obejmować cały ekran.
- W folderze `testy` umieścić archiwum projektu testów wraz z plikami źródłowymi i zrzutami ekranu.