1. **Jaka jest konwencja nazywania klas, metod, stałych i zmiennych w Javie?** 
   1. Klasy – stosujemy litery wielkie i małe, używamy rzeczowników, nazwa powinna być jak prosta i zrozumiała, najlepiej używać pełnych słów. Przykład: obiektyBadan
   2. Metody – stosujemy tutaj czasowniki i tak samo do oddzielenia słów stosujemy wielką literę w środku jak powyższy przykład
   3. Stałe – powinny być zapisywane wielkimi literami, natomiast słowa oddzielamy znakiem „\_”
   4. Zmienne – nazwy zmiennych powinny być krótkie, zrozumiałe dla osoby, która pierwszy raz widzi program, powinny wskazywać na intencje działania. Nie stosujemy na początku nazwy zmiennej znaku „\_”. Stosujemy również wielkie litery do oddzielenia wyrazu.
2. **Jak skompilować i uruchomić program w Javie z linii poleceń?**
   1. Aby skompilować i uruchomić program w Javie z linii poleceń należy zastosować polecenie Java nazwa\_klasy.java
3. **Napisz prosty program, który zapyta użytkownika o imię i nazwisko a następnie połączy je w jedną zmienną, którą następnie wypisze.**

import java.util.Scanner;  
  
public class Ex3 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Wita, podaj swoje imię:");  
 String name = scanner.next();  
 System.*out*.println("Podaj swoje nazwisko:");  
 String surname = scanner.next();  
 System.*out*.println("Nazywasz się:");  
 String namePlusSurname = name + " " + surname;  
 System.*out*.println(namePlusSurname);  
 }  
}

1. **Napisz program, który wczyta od użytkownika dwie liczby, wypisze ich sumę.**

import java.util.Scanner;  
  
public class Ex4 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Wita, podaj pierwszą liczbę:");  
 int x = scanner.nextInt();  
 System.*out*.println("Podaj drugą liczbę:");  
 int y = scanner.nextInt();  
 System.*out*.println("Suma twoich liczb wynosi: ");  
 int suma = x+y;  
 System.*out*.println(suma);  
 }  
}

1. **Napisz program losuje liczbę z pewnego zakresu (np. 1–100) za pomocą Random.  
   Następnie w pętli prosi użytkownika (przy użyciu Scanner) o zgadywanie liczby,  
   podając wskazówki "za dużo" lub "za mało", aż użytkownik trafi. Po odgadnięciu,  
   program wyświetla komunikat gratulacyjny wraz z liczbą prób.  
   (Wskazówka: będziesz potrzebować pętli while lub do-while oraz porównań if-else do sprawdzania odpowiedzi.) (2 punkty)**

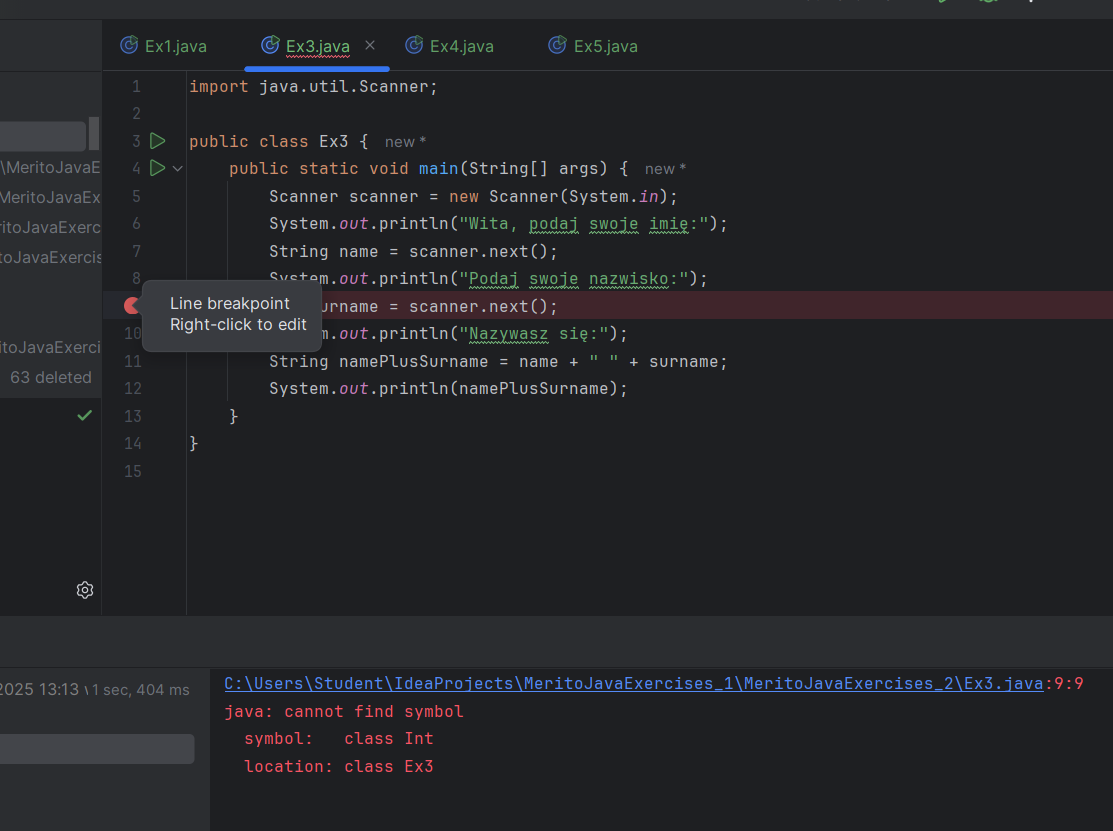
import java.util.Random;  
import java.util.Scanner;  
  
public class Ex5 {  
 public static void main(String[] args){  
 Random r = new Random();  
 int randomNumber = r.nextInt(100) + 1;  
 System.*out*.println("Zgadnij moją liczbę!");  
 boolean y = true;  
 Scanner scanner = new Scanner(System.*in*);  
 while(y){  
 int input = scanner.nextInt();  
 if(input == randomNumber)  
 {  
 System.*out*.println("Zgadłeś liczbę! Liczba to: " + randomNumber);  
 y = false;  
 break;  
 }  
 else if(input < randomNumber) {  
 System.*out*.println("Za mało");  
 } else {  
 System.*out*.println("Za dużo");  
 }  
 }  
 }  
}

1. **Czym jest debugowanie? Po co to się robi? Czym jest debuger?**

Debugowanie to proces analizujący dany kod w celu odnalezienia i identyfikacji zawartych w nim błędów. Proces ten jest wykonywany przez program zwany debugerem.

1. **Czym jest punkt przerwania (także punkt wstrzymania lub pułapka – ang. breakpoint) i jak się go ustawia? Podaj zrzut ekranu.**

Punkt przerwania to specjalne miejsce w kodzie programu, w którym jego działanie zostaje wstrzymane tymczasowo aby programista mógł przeanalizować stan danej linii kodu. Pozwala na sprawdzenie wartości zmiennych oraz wyłapać błędy w kodzie. Ustawiamy go klikając na czerwone kółko z lewej strony podświetlonej na ten sam kolor linii.



1. **Jakie kroki można wykonywać podczas debugowania? (0,5)**

Podczas debugowania możemy przeanalizować zmienne i stan aplikacji zwracając uwagę na podświetlone przez program linie. Możemy naprawić wykryty przez debuger błąd i wznowić proces.

1. **Jak można sprawdzić wartość zmiennych w danym punkcie? Jak sprawdzić zwracaną wartość? Jak sprawdzić wartości wyrażeń?**

Wartość zmiennych można sprawdzić w debugerze, np. w zakładce Variables, najeżdżając kursorem na zmienną lub używając Watch. Zwracaną wartość metody można podejrzeć w Return Value lub wykonać Step Into (F7). Wartości wyrażeń sprawdza się w oknie Evaluate Expression (Alt + F8) lub dodając je do Watch.

1. **Czym jest stos wywołań? (0,5)**

Stos wywołań (Call Stack) to struktura danych przechowująca informacje o aktualnie wykonywanych metodach. Nowe wywołania trafiają na stos, a po zakończeniu są zdejmowane. Można go zobaczyć w debugerze w zakładce Call Stack.