Struktura klasy

# Przed zajęciami

1. Zapoznanie się z polami statycznymi i metodami. Jaka jest różnica między polami statycznymi a polami instancji. Kiedy należy używać metod statycznych.
2. Dowiedz się, od jakiej instrukcji zaczyna się program w Javie. Gdzie znajduje się ta instrukcja.
3. Jaka jest rola konstruktora w programowaniu obiektowym.
4. Tablica to struktura danych składająca się z kolekcji elementów. Jak deklarować i używać tablic w Javie.
5. Zapoznanie się z zastosowaniem iteracji i instrukcji warunkowych w języku Java.
6. Wyjaśnij pojęcie "przeciążanie metod".

# Podczas zajęć

## Statyczni członkowie klasy

1. Zdefiniuj klasę CinemaTicket opisującą bilety do kina. Atrybutami biletu są: nazwa kina, tytuł filmu, rząd, miejsce i cena. Ponieważ nazwa kina jest taka sama dla wszystkich biletów, użyj pola statycznego. Następnie zdefiniuj metodę do wyświetlania danych o biletach. Spróbuj utworzyć dwa bilety i wyświetlić szczegóły biletu.
2. Zdefiniuj klasę SurfaceArea, która zawiera statyczne metody obliczające pole powierzchni dla: koła, prostokąta i trójkąta. Zatwierdzić utworzone metody.

## Punkt wejścia do aplikacji

1. Zdefiniuj klasę Motto. Następnie utwórz program, który wyświetli następujący tekst: "Rób to, co kochasz". Umieść poniższą metodę w zdefiniowanej klasie.

public static void main(String[] args) {  
 System.out.println("Rób to, co kochasz");  
}

Następnie skompiluj i uruchom program.

1. Korzystając z klasy opisującej bilety do kina, napisz program, który tworzy dwa bilety. Wydrukuj utworzone bilety.
2. Korzystając z klasy do obliczania pola powierzchni figur geometrycznych, napisz program tworzący obliczenia i wyświetlający:
   1. pole powierzchni prostokąta o bokach 4 i 5
   2. pole koła o promieniu 3
   3. pole trójkąta o podstawie 3 i wysokości 4

## Konstruktor

1. Zmodyfikuj klasę opisującą bilety do kina. Dodaj konstruktor tworzący bilet na wskazany film, z numerami rzędów i miejsc. Przyjmij, że w dwóch pierwszych rzędach cena biletu wynosi 10 zł, a w pozostałych 25 zł. Następnie napisz program, który w Kinie Gwiazda Poranna tworzy dwa bilety na film "Gladiator", pierwszy w rzędzie 2, a drugi w rzędzie 7. Wydrukuj utworzone bilety.
2. Zdefiniuj klasę Osoba dla rekordu osoby. Klasa powinna zawierać atrybuty: imię i nazwisko, waga (kg) i wzrost (cm). Dodaj konstruktory:
   1. Osoba(imię)
   2. Osoba(imię,waga,wzrost)

Następnie dodaj metody, które reprezentują zachowania obiektu:

* 1. setWeightAndHeight(waga,wysokość)
  2. calculateBMI() - oblicza wskaźnik masy ciała
  3. displayRecord() - wyświetla pełne informacje (imię, waga, wzrost i BMI)

Prawidłowa wartość BMI wynosi od 18,5 do 24,9. Podczas wyświetlania rekordu osoby, jeśli BMI jest zbyt niskie, wyświetl komunikat: "BMI za niskie". Gdy wskaźnik jest zbyt wysoki, wyświetl komunikat: "BMI za wysoki". Napisz program, utwórz trzy osoby i wyświetl ich rekordy.

# Po zajęciach

1. Stwórz bibliotekę metod do konwersji temperatur pomiędzy Celsjuszem, Kelwinem i Fahrenheitem. Możesz nazwać metody np. CelsiusToKelvin(), KelvinToCelsius(), ect. Następnie utwórz program, który oblicza i wyświetla temperaturę:
   1. 25 stopni Celsjusza w Kelwinach i Fahrenheitach
   2. 100 stopni Fahrenheita w Kelwinach i Celsjuszach
   3. 0 stopni Kelwina w skali Celsjusza i Fahrenheita
2. Stworzenie biblioteki metod obliczania podstawowych statystyk:
   1. Liczba elementów w określonym zakresie <x,y>.
   2. Suma liczb z danego zakresu <x,y>
   3. Średnia arytmetyczna liczb z podanego zakresu <x,y>.

Następnie utwórz program, który oblicza i wyświetla podstawowe statystyki dla liczb całkowitych z zakresu <5,10>

1. Istnieje wiele urządzeń, które mogą korzystać z Internetu: komputer osobisty, laptop, tablet, smartfon, radio internetowe czy telewizor. Zdefiniuj klasę InternetDevice, która będzie opisywać takie urządzenia. Klasa powinna zawierać atrybuty: name (nazwa urządzenia), connected (czy urządzenie jest podłączone do Internetu) oraz connectedDevices (liczba podłączonych urządzeń - pole statyczne). Dodaj konstruktor InternetDevice(name) oraz metody do manipulacji urządzeniami: connect(), disconnect(), isConnected(), displayStatus(), displayConnections() (metoda statyczna). Następnie napisz program, który tworzy pięć różnych urządzeń internetowych, z których trzy są podłączone do Internetu. Wyświetl stan każdego urządzenia oraz informację o liczbie urządzeń podłączonych do Internetu.
2. Poniższa definicja klasy StudentGrades pozwala na przechowywanie ocen studenta.

public class StudentGrades {  
  
 String studentName;  
 double[] grades;  
  
 StudentGrades(String name, double[] grades) {  
 this.studentName = nazwa;  
 this.grades = grades;  
 }  
}

Uzupełnij zajęcia o metody, które:

* 1. oblicza najniższy stopień
  2. oblicza się z najwyższej klasy
  3. oblicza liczbę ocen
  4. oblicza średnią ocen
  5. wyświetla kartę studenta z jego nazwiskiem, listą ocen studenta, liczbą ocen, najniższą oceną, najwyższą oceną i średnią ocen

Następnie napisz program, który tworzy oceny dla dwóch uczniów:

* 1. Amanda: 3.5, 4.5, 4.0, 2.0, 5.0, 3.5, 3.5
  2. Jakub: 2.0, 3.0, 2.0, 4.5, 4.5

Wskazówka: przekaż oceny ucznia do konstruktora, jako tablicę, w nawiasach klamrowych, np. StudentGrades({5.0, 3.5}).

1. Dodaj do klasy StudentGrades konstruktor StudentGrades(String name, int numberOfGrades), który pozwala na losowe utworzenie podanej liczby ocen studentów. Wskazówka: użyj generatora liczb losowych, który jest dostępny w klasie Random. Więcej informacji na ten temat znajdziesz w podręczniku.
2. Dodaj konstruktor StudentGrades(String name), który pozwala na wprowadzanie ocen ucznia z klawiatury. Wskazówka: użyj klasy Scanner. Zobacz podręcznik, aby uzyskać więcej informacji.
3. Zdefiniuj klasę Zegar, która reprezentuje zegar cyfrowy. Posiada ona dwa atrybuty: godzinę i minutę. Wartość godziny zawiera się w przedziale od 0 do 23. Wartość minutowa zawiera się w przedziale od 0 do 59. Następnie zdefiniuj następujące konstruktory:
   1. Clock() - inicjalizuje zegar na 00:00
   2. Clock(godzina,minuta) - inicjalizuje zegar na godzinę:minutę

Następnie należy zdefiniować metody:

* 1. setClock(godzina,minuta)
  2. setClock() - resetuje zegar na 00:00
  3. displayTime()
  4. addOneMinute()

Po tym napisz program, który:

* 1. tworzy zegar z czasem początkowym 12:47
  2. wyświetla czas
  3. ustawia zegar na 18:14
  4. wyświetla czas
  5. ustawia zegar na 09:03
  6. wyświetla czas
  7. ustawia zegar na 23:58
  8. wyświetla czas
  9. dodaje jedną minutę
  10. wyświetla czas
  11. dodaje jedną minutę
  12. wyświetla czas

1. Dodaj funkcję alarmu do zegara. Uzupełnij definicję klasy o atrybuty: alarmHour, alarmMinute. Dodaj metody: setAlarm(godzina, minuta) oraz runAlarm(), która wyświetla dźwięk alarmu "beep-beep-beep !!!". Włącz alarm, gdy czas alarmu jest taki sam jak czas zegara(porównaj czasy w metodzie addOneMinute()).
2. Zdefiniuj klasę Room, która reprezentuje pokój hotelowy. Klasa powinna zawierać atrybuty: liczba, łóżka, zajęte oraz guestName. Dodaj konstruktory: Room(numer) (tworzy pokój z dwoma łóżkami) oraz Room(numer,łóżka). Następnie zdefiniuj metody do manipulowania pokojami: checkin(guestName), checkout(), isOccupied() oraz displayStatus(). Następnie napisz program, w którym utwórz trzy pokoje z dwoma łóżkami, dwa pokoje z trzema łóżkami oraz jeden pokój z jednym łóżkiem. Przechowuj wszystkie obiekty pokoi w tablicy:

Room[] rooms = new Room[6];

Następnie wykonaj poniższe kroki:

* 1. Wyświetlenie stanu pojedynczego pokoju (numer pokoju, liczba łóżek, zajętość, nazwisko gościa). Wyświetlanie danych w pojedynczym wierszu.
  2. Wyświetlenie raportu z listą pokoi (numer pokoju, liczba łóżek, zajętość, nazwisko gościa)
  3. Wyświetl raport z listą pokoi, ograniczony do pokoi z określoną liczbą łóżek
  4. Wyświetlanie raportu o tym, ile pomieszczeń jest wolnych, a ile zajętych
  5. Wyświetlanie raportu o liczbie wolnych łóżek

Utwórz osobną metodę statyczną dla każdego raportu. Przekaż listę pomieszczeń (tablicę) do zdefiniowanych metod statycznych.

1. Zmodyfikuj klasę Room. Zdefiniuj metodę toString(), która zwraca informację o stanie pomieszczenia. Zamień metodę displayStatus() na zdefiniowaną metodę toString(). Wskazówka: dowiedz się w instrukcji, jak zdefiniować i używać metody toString().