Programowanie obiektowe - zadania

Elementy języka Java

Zad.1. Napisz program, który sprawdza, czy dana liczba całkowita jest parzysta.

Zad.2. Napisz program, który sumuje dane dwie liczby tylko w przypadku, gdy obie są dodatnie.

Zad.3. Napisz program, który sprawdza, czy dane trzy liczby całkowite tworzą trójkę pitagorejską.

Przykład: liczby 3, 4, 5 stanowią trójkę pitagorejską, ponieważ $3^2 + 4^2 = 5^2$.
Uwaga: W programie należy założyć, że liczby mogą być w dowolnej kolejności, np. $5, 3, 4$.

Zad.4. Napisz program, który daną ocenę wypisuje słownie:

2 - niedostateczny,
3 - dostateczny,
4 - dobry,
5 - bardzo dobry,
6 - celujący.

Wykorzystaj instrukcję *switch*, której działanie jest wyjaśnione na kolejnym slajdzie.

Instrukcja *switch*

```
switch (wyrażenie)
{
   case wartość1  :  instrukcja1;  break;
   case wartość2  :  instrukcja2;  break;
   ...
   default        :  instrukcjaN;  break;
}
```

- Najpierw obliczane jest *wyrażenie* umieszczone w nawiasach po słowie **switch**.
- Jeśli jego wartość odpowiada którejś z wartości podanej w jednej z etykiet **case**, wykonywane będą instrukcje począwszy od tej etykiety.
- Ich wykonywanie kończy się po napotkaniu instrukcji **break**. Działanie instrukcji **switch** zostaje wówczas zakończone.
- Jeśli wartość *wyrażenia* nie zgadza się z żadną z wartości w etykietach **case**, wykonywane będą instrukcje po etykiecie **default**.
- Etykieta **default** jest opcjonalna, tzn. może zostać pominięta. Wówczas, jeśli w zbiorze etykiet **case** nie ma żadnej etykiety równej wartości *wyrażenia*, instrukcja **switch** nie będzie wykonana.
- Instrukcje występujące po etykiecie **case** nie muszą kończyć się instrukcją **break**. Jeśli jej nie umieścimy, to będą wykonywane instrukcje znajdujące się pod następną etykietą **case**.

```
public class Main
{
  public static void main(String[] args)
{
    int x = 100, y = 5;
  char dzialanie = '*';
  switch(dzialanie)
  {
    case '+' :
    System.out.println("Twoj wynik: " + (x + y));
      break;
    case '-' :
    System.out.println("Twoj wynik: " + (x - y));
      break;
    case '*' :
    System.out.println("Twoj wynik: " + (x * y));
      break;
    case '/' :
    System.out.println("Twoj wynik: " + (x / y));
    break;
    default:
    System.out.println("Dzialanie nie jest obslugiwane.");
    break;
  }
  }
}
```

Zad.5. Napisz program obliczający tygodniowe zarobki brutto i netto pracownika, gdy znana jest jego kategoria zaszeregowania i liczba przepracowanych godzin w ciągu tygodnia. Wykorzystaj instrukcję *switch*.

Dla poszczególnych kategorii zaszeregowania obowiązują następujące stawki:

Kategoria zaszeregowania	Stawka [zł/godz.]
A	15
B	25
C	30
D	35

Jeśli pracownik przepracuje więcej niż 40 godzin w ciągu tygodnia, zapłata za każdą nadgodzinę jest dwukrotnie wyższa od stawki pracownika. Podatek od zarobku jest obliczany według następującej tabeli:

Zarobek	Stopa procentowa podatku
≤ 700	15%
$700 - 1200$	20%
> 1200	25%

Zad.6. Napisz program wypisujący wszystkie dzielniki liczby naturalnej n w trzech wersjach:

1. z pętlą *for*,
2. z pętlą *while*,
3. z pętlą *do...while*.

Zad.7. Napisz program obliczający sumę n ułamków postaci $\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \ldots$

Zwróć uwagę na działanie operatora dzielenia dla różnych typów argumentów, np:

- int i = 7; float j = i/2; //zmienna j ma wartość 3
- float i = 7; float j = i/2; //zmienna j ma wartość 3.5

Przykład dla n = 5: $\frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \frac{4}{5} + \frac{5}{6} = 3.55$

Zad.8. Napisz program sumujący losowe liczby naturalne z przedziału $[0, 20]$ do momentu, gdy kolejna wylosowana liczba będzie taka sama jak poprzednia. Wykorzystaj pętlę *do...while*.

Przykład: $6, 10, 3, 17, 2, 2$; suma $= 40$.

Zad.9. Napisz program sprawdzający, czy liczba naturalna n jest liczbą doskonałą. Liczba doskonała to taka, której suma dzielników (nie licząc samej liczby n) jest równa tej liczbie.

Przykład 1: 6, bo $1 + 2 + 3 = 6$.
Przykład 2: 28, bo $1 + 2 + 4 + 7 + 14 = 28$.

Zad.10. Napisz program sprawdzający, czy dana liczba naturalna n jest pierwsza.

Zad.11. Napisz program, który tworzy tablicę o rozmiarze 10, wypełnia ją losowymi liczbami całkowitymi z przedziału $[0, 20]$, a następnie podaje liczbę elementów nieparzystych.

Przykład: $12, 19, 3, 0, 16, 9, 12, 10, 1, 8$; liczba elementów nieparzystych: 4.

Zad.12. Napisz program, który tworzy macierz wymiaru $3x4$, wypełnia ją losowymi liczbami całkowitymi z przedziału $[10, 50]$, a następnie znajduje element maksymalny.

Przykład:

23 11 30 36
41 19 28 33
27 48 26 19
Element maksymalny: 48.

Klasy, obiekty, pola i metody

Zad.13. Uruchom Eclipse i utwórz nowy projekt.

1. Utwórz klasę o nazwie **Punkt**, zawierającą dwa publiczne pola typu int (współrzędną x i współrzędną y punktu).
2. W tym samym projekcie utwórz klasę **Test**. W metodzie main tej klasy utwórz dwa obiekty typu Punkt. Ustaw wartości ich współrzędnych (np. (1,2) i (5,3)) i wypisz je w postaci:

   Wspolrzedne punktu to (<wartosc x>, <wartosc y>).
3. Do klasy Punkt dodaj metodę **wyswietl**, która wyświetla wartości współrzędnych x i y punktu. Zmodyfikuj metodę main w klasie Test tak, aby korzystała z nowej metody.
4. Co się stanie, jeżeli oba pola klasy Punkt (współrzędną x i współrzędną y) zmienimy z publicznych na prywatne?
5. Do klasy Punkt dodaj dwie metody: **pobierzX** i **pobierzY**, które będą zwracały odpowiednio wartości współrzędnych x i y. Wykorzystaj te metody w metodzie main.

6. Do klasy Punkt dodaj metodę, która będzie jednocześnie ustawiała pola x i y tej klasy. Nazwij ją **ustawXY**. Metoda przyjmuje dwa parametry. Przetestuj jej działanie w metodzie main.
7. Zmodyfikuj metodę ustawXY tak, aby jako parametr przyjmowała obiekt typu Punkt i przetestuj jej działanie.
8. Do klasy Punkt dodaj metodę **pobierzWsp**, która zwróci w wyniku nowy obiekt klasy Punkt o współrzędnych takich, jakie zostały zapisane w polach obiektu bieżącego. Przetestuj jej działanie w metodzie main.
9. Do klasy Punkt dodaj konstruktor dwuargumentowy inicjalizujący zmienne klasy oraz wypisujący komunikat:

   Utworzono punkt (<wartosc x>,<wartosc y>).

   W metodzie main klasy Test utwórz kilka obiektów typu Punkt. Zaobserwuj wywołania konstruktora.

10. Do klasy Punkt dodaj konstruktor przyjmujący jako argument obiekt klasy Punkt, przepisujący wartości odpowiednich zmiennych. Przetestuj jego działanie.

11. Spróbuj utworzyć obiekt klasy Punkt poprzez wywołanie konstruktora bez żadnego argumentu. Co się stanie?

12. Wykomentuj oba konstruktory klasy Punkt (i miejsca ich wywołania w klasie Test) i ponownie wykonaj poprzednie zadanie. Co się stanie?

Zad.14. Napisz klasę o nazwie **Prostokat** zawierającą dwie składowe typu Punkt - współrzędne lewego górnego oraz prawego dolnego wierzchołka prostokąta (rozpatrujemy prostokąty o bokach równoległych do osi układu współrzędnych).

Zaimplementuj następujące metody dla utworzonej klasy.

1. konstruktor przyjmujący 4 argumenty typu int (kolejne współrzędne punktów), inicjalizujący składowe obiektu (wypisujący odpowiedni komunikat)
2. konstruktor przyjmujący 2 argumenty typu Punkt,
3. metodę wypisującą na ekranie komputera informację o danym prostokącie,
4. metodę obliczającą pole powierzchni prostokąta,
5. metodę, która sprawdza, czy podany punkt leży wewnątrz prostokąta.

Utwórz obiekty zaprojektowanej klasy przy użyciu obu konstruktorów. Sprawdź poprawność działania napisanych metod.

Zad.15. Utwórz klasę **Wymierne** do obsługi liczb wymiernych.

1. Klasa będzie zawierała dwa pola typu int: **licznik** i **mianownik**.
2. Do klasy dodaj dwa konstruktory:
   - dwuargumentowy - ustawiający pola licznik i mianownik,
   - jednoargumentowy - przewidziany do inicjalizacji obiektów o mianowniku 1.
3. Liczba wymierna ma być reprezentowana w postaci nieskracalnej. W tym celu napisz metodę **NWD**, znajdującą największy wspólny dzielnik dwóch liczb całkowitych.
4. Dodaj metodę **Skroc**, która podzieli pola licznik i mianownik przez ich największy wspólny dzielnik. Wykorzystaj metodę NWD.
5. Zmodyfikuj konstruktor dwuargumentowy tak, aby przed nadaniem wartości polom licznik i mianownik skracał je.
6. Napisz metody:
   - **PobierzLicznik** - zwracającą w wyniku wartość pola licznik,
   - **PobierzMianownik** - zwracającą w wyniku wartość pola mianownik,
   - **Wypisz** - wypisującą wartości pól klasy w postaci:
     licznik = <wartość>, mianownik = <wartość>

7. Napisz metodę **Dodaj**, której parametrem będzie obiekt klasy Wymierne. Metoda powinna zwracać referencję do nowo utworzonego obiektu, zawierającego wynik sumowania dwóch liczb wymiernych.

   Np. instrukcja `Wymierne u3 = u1.Dodaj(u2);`
   spowoduje dodanie ułamków $u1$ i $u2$, zapisanie wyniku dodawania w nowym obiekcie, do którego referencja będzie przypisana zmiennej $u3$.

8. W podobny sposób utwórz metody **Odejmij** i **Pomnoz**.

9. Kolejna metoda, **DodajDoSiebie**, nie tworzy nowego obiektu, ale wynik dodawania zapisuje w obiekcie bieżącym (tzn. w obiekcie na rzecz którego została wywołana).

   Np. wywołanie `u1.DodajDoSiebie(u2);`
   spowoduje dodanie ułamków $u1$ i $u2$ i zapisanie wyniku dodawania w obiekcie $u1$ (czyli ułamek $u1$ zostanie zmieniony).

10. Podobnie napisz metody **OdejmijOdSiebie** i **PomnozPrzezSiebie**.

11. Przeciąż metody **Dodaj** i **Pomnoz** tak, aby dodawały do ułamka liczby typu int i mnożyły ułamki przez liczby typu int.

12. Utwórz klasę **WymierneTest** z metodą main, w której przetestujesz wszystkie napisane metody klasy Wymierne.

Zad.16. Utwórz klasę **Student** z następującymi składowymi

- pola przechowujące:
  1. imię,
  2. nazwisko,
  3. numer indeksu,
  4. rok studiów,
  5. listę przedmiotów (np. w tablicy napisów (String); zakładamy, że nie ma więcej niż 50 przedmiotów; zapamiętujemy, do którego miejsca tablica jest wypełniona);
- konstruktory:
  1. 5-argumentowy, przyjmujący imię, nazwisko, numer indeksu, rok, przedmioty;
  2. 4-argumentowy, przyjmujący imię, nazwisko, numer indeksu, rok; natomiast na listę przedmiotów wstawiający jeden przedmiot: "Programowanie obiektowe";
- metody:
  1. wypisująca komplet danych, np. w postaci:
     (Jan Kowalski, 234567, rok 2, Programowanie obiektowe, Analiza matematyczna, Rachunek prawdopodobieństwa),
  2. zwracająca imię i nazwisko,
  3. zwracająca numer indeksu,
  4. zwracająca rok studiów,
  5. zwiększająca rok studiów,
  6. dodająca przedmiot (przyjmująca parametr typu String).

Utwórz klasę **Main** z metodą main i przetestuj wszystkie metody klasy Student.