Nauka programowania i język Java – ćwiczenia do domu

0.

Rozwiązujcie i przysyłajcie na adres **p.czarnik@alx.pl**

1. Zmienne, if-y i pierwsze pętle

Zadanie 1.1 Pole trójkąta

Program, który odczytuje trzy liczby, sprawdza czy liczby te mogą stanowić boki trójkąta (np. z 2, 2 i 5 nie da się ułożyć trójkąta, prawa?), a jeśli mogą, oblicza pole powierzchni trójkąta o takich bokach.

Wzór Herona: $\sqrt{p(p-a)(p-b)(p-c)}$, gdzie p jest połową obwodu: $(a+b+c)/2$.

Tutaj użyj jednego z poznanych sposobów komunikacji z użytkownikiem. Pierwiastek kwadratowy to metoda Math.sqrt( ).

Zadanie 1.2 Szewc

Napisz taki program: użytkownik ma podać, w jaki dzień tygodnia oddał buty do szewca (numer od 1 do 7). Ma też podać, ile dni będzie trwała naprawa. Program ma wypisać, w jaki dzień tygodnia buty będą gotowe do odbioru. W podstawowej wersji możesz wypisywać dzień odbioru też jako numer. Postaraj się obsłużyć także sytuacje, że naprawa trwa dłużej niż 7 dni. Na końcu zrób wersję, w której program wypisuje dzień odbioru słownie.

Zadanie 1.3 Firma remontowa

Firma remontowa posiada taki cennik usług (nie mam pojęcia czy realistyczny ;)):

- gipsowanie ścian: 100 zł za metr kwadratowy ściany
- malowanie ścian i sufitów: 30 zł za metr kwadratowy
- położenie paneli podłogowych: 50 zł za metr kwadratowy podłogi
- położenie listew przypodłogowych: 40 zł za metr bieżący

Napisz program, który pomaga wycenić pracę na podstawie wymiarów pomieszczenia. Zakładając, że pomieszczenie ma kształt prostokąta, program powinien zapytać o dwa wymiary poziome (w metrach) oraz o wysokość i na tej podstawie obliczyć powierzchnię podłogi, sufitu oraz łącznie wszystkich ścian, a także obwód pokoju (listy podłogowe).

Wersja 1: Przyjmij, że zawsze wykonywany jest komplet robót: gipsowanie ścian, malowanie ścian i sufitu, panele podłogowe, listwy przypodłogowe. Program na podstawie danych wejściowych oblicza sumaryczny koszt prac.

Wersja 2: Zapytaj użytkownika jakie elementy prac są wykonywane. Jedną z opcji jest skorzystanie z JOptionPane.showConfirmDialog – popatrz na moje proste przykłady, przeczytaj dokumentację, poszukaj wskazówek w internecie.

Jako domyślną wysokość pomieszczenia można wpisać 2.50, ale tak, aby użytkownik mógł zmienić. Można to zrobić tak: JOptionPane.showInputDialog("Podaj wysokość", "2.50");

Zadanie 1.4 Zgadnij liczbę z zakresu

Random r = new Random(); int x = r.nextInt(1000);

Program losuje liczbę z zakresu od 0 do 999 (jak wyżej). Użytkownik ma zgadnąć tę liczbę nie widząc jej. Kiedy użytkownik poda nieprawidłowy wynik, program podpowiada pisząc czy podana liczba była za duża, czy za mała. Gdy użytkownik poda właściwą liczbę, program wypisuje gratulacje jednocześnie informując, w której próbie udało się zgadnąć liczbę.

Nawiasem mówiąc technika wyszukiwania oparta o „podpowiedzi" *za dużo/za mało* nazywa się **bisekcją** i pełni w informatyce bardzo ważną rolę. Umiejętnie ją stosując powinno się te zagadki rozwiązywać w 9-10 próbach (bo $2^{10} = 1024$).

Zadanie 1.5 Podzielne przez 3 lub przez 5

Użytkownik podaje liczbę całkowitą „limit". Następnie program wypisuje, bez powtórzeń, liczby z zakresu od 1 do podanego limitu włącznie, które są podzielne przez 3 lub przez 5. Wypisz także jak dużo takich liczb wystąpiło w tym przedziale.

Przykładowe działanie programu:

```
Podaj limit: 20
3 5 6 9 10 12 15 18 20
Takich liczb było: 9
```

Zadanie 1.6 Choinka

Napisz program, który wczytuje liczbę całkowitą, a następnie na konsolę wypisuje w tylu liniach „choinkę" ze znaków * . Np. dla parametru 3 powinno się wypisać:

```
    *
   ***
  *****
```

Podpowiedź: System.out.print(...) wypisuje i nie przechodzi do nowej linii,
System.out.println() przechodzi do nowej linii.

Zadanie 1.7 Skarb

Napisz grę tekstową polegającą na poszukiwaniu skarbu na dwuwymiarowej planszy o rozmiarach 10 na 10. Program na początku losuje pozycję skarbu oraz pozycję gracza.

Następnie użytkownik może wprowadzać komendy zmieniające położenie postaci o jedną pozycję w górę/dół/lewo/prawo (np zgodnie z konwencją WSAD) – normalnie za pomocą Scannera.

Gdy gracz wejdzie na pole, na którym kryje się skarb – wygrywa.

Gdy wyjdzie poza planszę – przegrywa.

Po każdym ruchu użytkownik powinien otrzymywać informację o tym, czy zmierza w dobrym kierunku (zbliżasz się / oddalasz się). Po znalezieniu skarbu wypisz liczbę ruchów wykorzystanych przez użytkownika na dojście do celu.

Zadanie 1.8 Range

Jedną z często używanych możliwości języka Python jest łatwe przechodzenie przez zakresy liczbowe za pomocą funkcji `range` oraz wyrażeń listowych. Jeśli masz dostęp do interpretera Pythona, spróbuj np. następujących wyrażeń:

list(range(10)) list(range(5, 10)) list(range(10, 20, 4)) list(range(20, 10, -3))

Zaimplementuj podobną funkcjonalność w Javie jako **program**, który przyjmuje parametry wiersza poleceń (`String[] args` w `main`) i wypisuje ciąg liczb rozdzielonych spacją na standardowe wyjście.

Program przyjmuje od 1 do 3 parametrów wiersza poleceń – liczb całkowitych. Ich ilość jest dostępna jako `args.length`, a kolejne parametry jako `args[0]`, `args[1]` i `args[2]`.

- Pierwszy parametr, o ile podano co najmniej dwa, oznacza początek zakresu. Jeśli podano tylko jeden parametr, to domyślnym początkiem zakresu jest 0.
- Drugi parametr, a jedyny w przypadku gdy podano tylko jeden, oznacza koniec zakresu, czyli liczbę, przed którą program ma zakończyć wypisywanie.
- Trzeci parametr oznacza wartość kroku, o jaki różnią się kolejne liczby ciągu. Domyślną wartością kroku jest 1. Krok może być ujemny.

Ten program możesz spróbować napisać w zwykłym edytorze (typu Notepad++, a nie Eclipse/IntelliJ). Program najłatwiej przetestować w konsoli. Przykładowe wywołania i ich wyniki (zakładają, że klasa nie należy do żadnego pakietu - jak nasze przykłady z pierwszego dnia):

```
javac Range.java
java Range 10
0 1 2 3 4 5 6 7 8 9

java Range 0

java Range 5 10
5 6 7 8 9

java Range 10 20 4
10 14 18

java Range 20 10 -3
20 17 14 11
```