Nauka programowania i język Java – ćwiczenia do domu

1. Konfiguracja środowiska (ok. 2 godz.)

Na własnym komputerze skonfiguruj środowisko potrzebne do tworzenia programów w Javie. Postępując analogicznie do pracy w labie ALX, zainstaluj:

1. Java Development Kit w najnowszej stabilnej wersji
2. Środowisko programistyczne (IDE), w którym pracowaliśmy na zajęciach (NetBeans / Eclipse / IntelliJ w zależności od wyboru prowadzącego).

W swoim IDE załóż nowy projekt, napisz i uruchom prosty program typu „Hello World".

Zaimportuj projekty wykonywane podczas pierwszych zajęć, spróbuj je uruchomić. Popraw bądź uzupełnij przykłady tego wymagające.

2. Interakcja z użytkownikiem

Zadanie 2.1 Buty do szewca :)

Napisz taki program: użytkownik ma podać, w jaki dzień tygodnia oddał buty do szewca (poniedziałek to dzień 1, wtorek to dzień 2 itp.). Ma też podać, ile dni będzie trwała naprawa. Program ma wypisać, w jaki dzień tygodnia buty będą gotowe do odbioru. Jeśli tak będzie ci wygodniej, możesz założyć, że naprawa butów nigdy nie będzie trwała dłużej niż siedem dni.

Zadanie 2.2 Opłata hotelowa

Potem napisz taki program: użytkownik podaje swój wiek i ile nocy spędzi w hotelu, a program wylicza, ile trzeba zapłacić. Zasady są takie:

- nieletni (poniżej 18 roku życia) płacą 100 zł za noc

- dorośli (ci, którzy mają przynajmniej 18 lat ale mniej niż 65 lat) płacą:

  - 200 zł za noc, jeśli nocują jedną noc

  - 180 zł za noc, jeśli nocują przynajmniej dwie ale mniej niż pięć nocy

  - 150 zł za noc, jeśli nocują pięć lub więcej nocy

- emeryci (ci, którzy mają przynajmniej 65 lat), płacą jak dorośli, ale z 10% zniżki

Przykładowo: jeśli użytkownik ma 70 lat i spędzi w hotelu dziesięć nocy, zapłaci 150 zł za noc z 10% zniżki, czyli 150-15 zł za noc, czyli 135 zł za noc, czyli 1350 zł

Program można zrealizować jako pytanie-odpowiedź, albo jako aplikację okienkową w Swingu.

Zadanie 2.3 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: $\sqrt{p(p-a)(p-b)(p-c)}$, gdzie p jest połową obwodu: $(a+b+c)/2$.

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

3. Pisanie „funkcji”

(czyli w Javie „metod statycznych”)

Zadanie 3.1 Funkcje liczbowe

Napisz w formie metod statycznych, które przyjmują parametry typu double i zwracają wynik typu double (można pominąć te, które zrobiliśmy na zajęciach) takie funkcje matematyczne:

1. stopyNaMetry - przelicza odległość wyrażoną w stopach na metry,
2. max - zwraca większą z dwóch liczb,
3. srednia - oblicza średnią z dwóch liczb,
4. poleKola - oblicza pole koła o podanym promieniu (jeden parametr)
   podpowiedź: wartość Π jest dostępna jako Math.PI
5. bmi - oblicza współczynnik BMI dla podanych dwóch parametrów: wzrostu w metrach i wagi w kg
6. poleTrojkata - z trzema parametrami - oblicza pole trójkąta o podanych bokach z wzoru Herona: $\sqrt{p(p-a)(p-b)(p-c)}$, gdzie p jest połową obwodu: $(a+b+c)/2$.
   podpowiedź: pierwiastek jest dostępny jako Math.sqrt
7. przekatnaProstokata – oblicza długość przekątnej prostokąta o podanych bokach

Dla wszystkich funkcji dodaj przykładowe wywołania, żeby sprawdzić czy działają poprawnie. Dla wybranych napisz też interaktywne programy, które pytają użytkownika o dane i wypisują wynik.

4. Interakcja w pętli

Zadanie 4.1 Zagadka matematyczna

(to robiliśmy)

Program losuje dwie liczby z zakresu od 0 do 99 (patrz poniżej). Podaje te dwie liczby i pyta jaka jest ich suma (nie podaje jej). Użytkownik ma odgadnąć (no, policzyć w głowie) wynik. Program pyta o wynik wielokrotnie, tak długo, aż użytkownik poda prawidłowy wynik.

Do interakcji można użyć dowolnej poznanej techniki, np. JOptionPane albo Scanner(System.in). Do losowanie można użyć klasy java.util.Random:

```
Random r = new Random();
// ...
int liczba = r.nextInt(100); // pseudolosowa liczba od 0 do 99
```

Zamiast dodawania może być mnożenie (program sprawdzający znajomość tabliczki mnożenia...)

Zadanie 4.2 Zgadnij liczbę z zakresu

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 4.3 Pytania o język

Program pyta użytkownika jaki jest jego ulubiony język programowania. Jeśli użytkownik wpisze Java, program wyświetla "Dobry wybór!" i kończy się; jeśli użytkownik wpisze coś innego, program pyta dalej.

Uwaga, w zadaniu kryje się pułapka związana z porównywaniem napisów w Javie – poszukajcie na ten temat, jeśli jeszcze tego nie znacie...

5. Pętle i tablice, czyli „pisanie algorytmów”

Każde z tych ćwiczeń należy zrealizować jako metodę statyczną, która otrzymuje w parametrze tablicę (int[] dla liczb całkowitych); jeśli ktoś zna albo pojawiły się już na zajęciach, można także użyć list (List<Integer>). Niektóre z metod będą wymagały podania dodatkowych parametrów.

Dla każdego z tych ćwiczeń minimum to jest implementacja samej metody („logiki”) i uruchomienie dla przykładowych danych w metodzie main. **Dla chętnych** (być może tylko dla kilku wybranych zadań) można napisać program, który pobiera dane od użytkownika interaktywnie, na jeden z wcześniej poznanych sposobów; wówczas bardziej wygodne od tablic będą na pewno listy.

Zadanie 5.1 Choinka

Napisz program, który (np. za pomocą Scanner) wczytuje liczbę całkowitą, a następnie na konsolę wypisuje w tylu liniach „choinkę” ze znaków * . Np. dla parametru 3 powinno się wypisać:

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

Zadanie 5.2 Operacje na tablicach liczb

(te, których nie było na zajęciach, ew. jeszcze raz samodzielnie dla powtórzenia)

- int **suma**(int[] tab) (o ile nie zrobiliśmy na zajęciach)
- double **srednia**(int[] tab) lub double srednia(double[] tab)
- int **max**(int[] tab) – zwraca największą wartość z tablicy
  - Wynikiem może być Integer i wtedy w przypadku pustej tablicy można zwrócić null
- int **roznicaMinMax**(int[] tab) – różnica pomiędzy największą a najmniejszą liczbą w tablicy; 0 jeśli tablica jest pusta.
- void **wypiszWieksze**(int[] tab, int x) – wypisuje na System.out wszystkie te liczby z tablicy tab, które są większe od x
- Integer **pierwszaWieksza**(int[] tab, int x) – zwraca (return) pierwszą znalezioną w tab liczbę większą od x; zwraca null, jeśli takiej liczby tam nie ma.
- int **sumaWiekszych**(int[] tab, int x) – zwraca (return) sumę liczb z tablicy tab, które są większe niż x.
- int **ileWiekszych**(int[] tab, int x) – liczy ile elementów tablicy tab jest większych od liczby x.
- void **wypiszPodzielne**(int[] tab, int x) – wypisuje na System.out wszystkie te liczby z tablicy tab, które są podzielne przez x (warunek do sprawdzenia: element % x == 0)
- Integer **pierwszaPodzielna**(int[] tab, int x) – zwraca (return) pierwszą znalezioną w tab liczbę podzielną przez x; zwraca null, jeśli takiej liczby tam nie ma.
- int **ileWiekszych**(int[] tab, int x) – liczy ile elementów tablicy tab jest większych od liczby x.
- int **ileKwardatow**(int[] t) – ile jest kwadratów liczb naturalnych w tablicy.
- Integer **znajdzWspolny**(int[] t1, int[] t2) – zwraca element (liczbę), który występuje zarówno w tablicy t1, jak i t2; zwraca null, jeśli takiego elementu nie ma.

6. Aplikacje okienkowe (Swing)

Stwórz jedną lub więcej aplikacji okienkowych w technologii Swing. Wygląd interfejsu przygotuj w edytorze wizualnym, np. w Netbeans (New JframeForm) lub w Eclipse z doinstalowaną wtyczką Window Builder (New WindowBuilder > Application Window).

Można stworzyć aplikację według własnego pomysłu (mile widziane), albo wybrać coś poniższych propozycji. Możliwe, że część z nich została już zrobiona jako przykłady podczas zajęć.

Zadanie 6.1 Koszt paliwa

Użytkownik wpisuje: ile średnio pali w trasie jego samochód (l/100 km), ile kosztuje paliwo (zł/l) oraz długość trasy w km. Program wylicza ile kosztuje taka trasa.

Dodatkowo można dodać pole (najlepiej JSpinner), w którym określa się liczbę osób, a program dodatkowo wylicza koszt podróży podzielony na te osoby.

Zadanie 6.2 Konwerter jednostek

Napisz program, który służy do przeliczania wartości między różnymi systemami miar. Minimum jest program, który przelicza jedną konkretną rzecz, np. mile na kilometry. Ale można spróbować zrobić bardziej rozbudowaną aplikację, np. po jednej stronie ekranu umieść pola na dane w jednostkach metrycznych (centymetry, metry, kilometry, kilogramy, Celcjusze), a z drugiej brytyjskich (cale, stopy, mile, funty, Farenheity), a za pomocą przycisków można przeliczać w jedną lub w drugą stronę. Własne pomysły na układ okna i sposób działania mile widziane.

Zadanie 6.3 BMI

Człowiek podaje swój wzrost i wagę, a otrzymuje wyliczony współczynnik BMI i informację tekstową czy jest w normie, czy ma się odchudzać, czy raczej przytyć.

Niektóre źródła na temat BMI rozróżniają normy ze względu na wiek lub płeć. Opcjonalnie możesz w swoim programie uwzględniać także te informacje.

Zadanie 6.4 500+ ;)

Program wyliczający ile się należy z programu 500+ w zależności od liczby dzieci (polecam komponent Spinner) i dochodów w rodzinie (przy dochodach <800 zł / osobę należy się także na pierwsze dziecko; chyba, że coś się zmieniło ;)).

Zadanie 6.5 Przedszkole

W przedszkolu opłata za pierwsze dziecko wynosi 300 zł, ale każde kolejne 200 zł. Za pomocą Spinner podajemy liczbę dzieci, a program wylicza ile wynosi łączna opłata za dzieci.

Niech wspomniane w treści wartości 300 i 200 także będą parametrami, które można zmienić w okienku, ale wartości 300 i 200 będą początkowymi wartościami gdy uruchomi się program.