JAVA – PODSTAWY
Programowanie I - Zadania

Paweł Dudko

WYRAŻENIA REGULARANE
DATA I CZAS
OOP

Zadanie 1

Napisz program, który sprawdzi czy podany adres email ma poprawny format.

Przykład:

If email address **aaa@com.pl** has correct format: true

If email address **aaa** has correct format: false

Autor: Paweł Dudko

Zadanie 2

Napisz program, który sprawdzi czy podany polski numer IBAN ma poprawny format.

Przykład:

If PL IBAN **PL10105000997603123456789123** has correct format: true

If PL IBAN **1324** has correct format: false

Autor: Paweł Dudko

Zadanie 3

Napisz program, który sprawdzi czy liczba wejściowa (typu String) ma poprawny format.

Przykład:

If digit **1234,1234** has correct format: true
If digit **123a123** has correct format: false

Autor: Paweł Dudko

Zadanie 4

Napisz program, który pobierze od użytkownika tekst i sprawdzi, czy użytkownik kichnął, tzn. czy w podanym tekście znajduje się „aaaa psik” z dowolnie wieloma, ale minimum jedną literą 'a' na początku wyrażenia.

Przykład:

Please insert your text: **a psik**
Result: true

Please insert your text: **aaaaa psik**
Result: true

Please insert your text: **psik**
Result = false

Autor: Paweł Dudko

Zadanie 5

Napisz program, działający jak stoper.
Program powinien wyświetlić komunikat, że pomiar czasu rozpoczyna się od wciśnięcia klawisza Enter. Po uruchomieniu pomiaru powinien pojawić się kolejny komunikat informujący o tym, że w celu zatrzymania pomiaru należy ponownie wcisnąć klawisz Enter.
Po zakończeniu pomiaru program powinien wyświetlić informacje o całkowitym czasie pomiaru.

Podpowiedź: Użyj klasy *Duration* do wyliczenia czasu pomiaru.

Przykład:

To start timing press ENTER
To stop timing press ENTER
Total time: 6 sec

Autor: Paweł Dudko

Zadanie 6

Napisz program, który pobierze od użytkownika datę najbliższych Twoich zajęć w SDA i obliczy ile dni do nich pozostało.

Przykład:

Please insert date [in format [yyyy-MM-dd HH:mm:ss]: **1900-01-01 09:00:00**
You have 19 days to the next lesson.

* Spróbuj rozszerzyć program, tak aby wynik był uzupełniony równie o liczbę godzin i minut.

Przykład:

Please insert date [in format [yyyy-MM-dd HH:mm:ss]: **1900-01-01 09:00:00**
You have Days: 19 Hours : 10 Minutes : 38 Secs to the next lesson.

Autor: Paweł Dudko

Zadanie 7

Napisz klasę `Cat`, która będzie posiadała:
a) prywatne pole przechowujące imię kota (typu String);
b) konstruktor, który będzie przyjmować imię kota;
c) metodę `public void makeSound()`, która wypisywać będzie wydawany przez niego dźwięk;
d) metodę `public void eatMice(int mice)`, która będzie zliczała zjedzone przez kota myszy i wypisywała komunikat: „I ate x mice”;
e) metodę `public void print()`, która będzie wypisywała informacje o kocie (imię).

Następnie napisz nową klasę `Main`, w której:
a) utworzysz tablicę kotów;
b) dodasz do niej parę utworzonych obiektów typu `Cat`;
c) dla wszystkich obiektów wywołasz metody `makeSound`, `eatMice` i `print`.

Autor: Paweł Dudko

Zadanie 7

Przykład:

Cat:
name=Mruczek
Miau miau
I ate 6 mice.

Cat:
name=Filemon
Miau miau
I ate 12 mice.

Cat:
name=Bonifacy
Miau miau
I ate 20 mice.

Autor: Paweł Dudko

Zadanie 8.1

Napisz klasę `Author`, reprezentującą autora – pisarza wierszów, która będzie posiadała:
a) prywatne pola (typu String) przechowujące informacje o nazwisku (surname) i narodowości (nationality);
b) konstruktor, który będzie przyjmować oba pola;
c) metodę `public void print()`, która będzie wypisywała informacje o autorze.

Napisz klasę `Poem`, reprezentującą wiersz, która będzie posiadała:
a) prywatne pola przechowujące informacje o autorze (creator, typu `Author`) i ilości zwrotek (stropheNumbers, typu int)
b) konstruktor, który będzie przyjmować oba pola;
c) metodę `public void print()`, która będzie wypisywała informacje o pojedynczym wierszu.

Autor: Paweł Dudko

Zadanie 8.2

Napisz klasę `Main`, w której:
a) utworzysz tablicę wierszy;
b) dodasz do niej 3 obiekty klasy `Poem`;
c) wypiszesz nazwisko tego autora, który napisał wiersz o jak największej liczbie zwrotek;
d) wypiszesz wszystkie informacje nt. autora, który napisał wiersz o jak największej liczbie zwrotek.

Przykład wypisania autora z największa liczbą zwrotek:

```
Author:
  surname=Chotomska
  nationality=PL
```


Autor: Paweł Dudko

Zadanie 9.1

Napisz klasę `OrderItem` (pozycja zamówienia), która będzie posiadała:
a) prywatne pola przechowujące informacje o produkcie (productName, typu String), ilość zamówienia (quantity, typu int) oraz cenie jednostkowej (price, typu double);
b) konstruktor, który będzie przyjmować wszystkie pola;
c) metodę `public double getAmount()`, która będzie obliczała wartość danej pozycji zamówienia;
d) metodę `public boolean isCorrect()`, która będzie sprawdzała czy pozycja jest prawidłowa, tzn. ilość i cena są większe od zera;
e) Metodą `public void print()`, która będzie wyświetlała daną pozycję zamówienia, np.

Cukier | 4,00 zł | 3 pcs | 12,00 zł

Autor: Paweł Dudko

Zadanie 9.2

Napisz klasę `Order` (zamówienie), która będzie posiadała:
a) prywatne pole przechowujące informacje o maksymalnej ilość pozycji w zamówieniu (maxOrderItem, typu int);
b) prywatną tablicę obiektów (items, typu OrderItem) z zamówieniami;
c) konstruktor, który utworzy zamówienie o zadanej liczbie pozycji (maxOrderItem);
d) metodę `public boolean addItem(OrderItem orderItem)`, dodającą pozycję zamówienia, ale tylko wtedy, gdy w zamówieniu jest jeszcze miejsce i jeżeli pozycja zamówienia jest prawidłowa;
e) metodę `public double getTotalAmount()`, obliczającą łączną wartość zamówienia;
f) metodę `public int getItemsCount()`, obliczającą łączną liczbę zamówionych produktów;
g) metodę `public void print()`, wyświetlającą dane zamówienia, np.

Chleb | 3,50 zł | 1 pcs | 3,50 zł
Cukier | 4,00 zł | 3 pcs | 12,00 zł
Total amount: 15,50 zł

Autor: Paweł Dudko

Zadanie 9.2

Przykład wyniku działania programu:

```
Cukier   | 3.5 zł  | 1 pcs  | 3.5 zł
Mąka     | 4.0 zł  | 2 pcs  | 8.0 zł
Chleb    | 4.0 zł  | 3 pcs  | 12.0 zł
Jaja     | 4.0 zł  | 4 pcs  | 16.0 zł
Mleko    | 4.0 zł  | 5 pcs  | 20.0 zł
Total amount: 59.5 zł
Count: 15 pcs
```


Autor: Paweł Dudko

Zadanie 10

Stwórz klasę `Warrior` posiadającą 3 pola prywatne: name, strength oraz hp. Następnie zaimplementuj w niej metodę symulującą walkę dwóch wojowników. Metoda powinna sprawdzać czy wojownik żyje i czy może wykonać swoje uderzenie (dla uproszczenia możesz przyjąć że sprawdzanie odbywa się raz na turę).

Następnie stwórz nową klasę `Main` w której stworzysz dwóch wojowników i zasymulujesz walkę pomiędzy nimi.

Autor: Paweł Dudko

Zadanie 11

Stwórz klasę `Email` posiadającą 4 pola prywatne: receiver, title, bodyMsg oraz attachment. Następnie zaimplementuj w niej zagnieżdżoną klasę `Builder` za pomocą której będzie możliwe tworzenie (budowanie) nowych obiektów typu Email.

Następnie stwórz nową klasę `Main` w której stworzysz nowy obiekt typu Email.

Przykładowy sposób tworzenia nowego obiektu typu Email w klasie `Main`:

```
Email email = new Email.Builder()
                .receiver("sda")
                .title(„New JAVA course")
                .bodyMsg(„Hello!")
                .attachment(„introduction.pdf")
                .build();
```


Autor: Paweł Dudko

Zadanie 12

Napisz klasę `Computer`, zawierającą klasy zagnieżdżone jako kolejne komponenty komputera:
a) `Processor`, posiadający prywatne pola przechowujące informacje o rdzeniach (cores, typu int) i producencie (producer, typu String);
b) `Memory`, posiadający prywatne pola przechowujące informacje o ilości pamięci (memory, typu int), producencie (producer, typu String) oraz typie (type, typu String);
c) `ExternalPort`, jako klasę statyczną, posiadającą prywatne pole przechowujące informacje o typie (type, typu String).

Następnie w klasie `Main` spróbuj stworzyć obiekt nowego komputera, wykorzystując wszystkie komponenty.

Autor: Paweł Dudko

Przydatne źródła

Strony z zadaniami

- https://codesignal.com
- https://www.codewars.com
- https://app.codility.com/programmers/lessons/1-iterations/binary_gap/

Przykłady best practise w Javie

- https://google.github.io/styleguide/javaguide.html
- https://medium.com/@rhamedy/a-short-summary-of-java-coding-best-practices-31283d0167d3

Książki

- Czysty kod (Clean Code) - Robert C. Martin
- Effective Java - Joshua Bloch
- Mistrz czystego kodu (The Clean Coder) - Robert C. Martin
- Pragmatic Programmer - Andrew Hunt, David Thomas

Autor: Paweł Dudko