Warsztaty I Java

v3.1

Jak zacząć?

- Stwórz repozytorium dla projektu o nazwie: **`Warsztaty_numerWarsztatu`**
- Utwórz projekt za pomocą Twojego IDE. Dla każdego zadania utwórz oddzielny pakiet.
- Zainicjuj repozytorium w swoim folderze z projektem (wystarczy folder **src**). Instrukcja krok po kroku zostanie wyświetlona na Github, po utworzeniu pustego repozytorium.
- Rozwiąż zadania i zakomituj zmiany do swojego repozytorium. Użyj do tego komendy:

```
git add nazwa_pliku
```

Jeżeli chcesz dodać wszystkie zmienione pliki użyj:

```
git add .
```

Pamiętaj że kropka na końcu jest ważna!

Jeśli nie chcesz dodawać plików ukrytych, które uległy zmianie, zastosuj komendę: **`git add *`**, lub stwórz plik **`.gitignore`**, w którym wskażesz, jakie pliki mają być pomijane.

Plik **`.gitignore`** możesz łatwo utworzyć klikając w link: https://www.gitignore.io/

Jak zacząć?

- Następnie wykonaj commit zmian komendą:

```
git commit -m "nazwa_commita"
```

- Wypchnij zmiany do swojego repozytorium na GitHubie. Użyj do tego komendy

```
git push origin master
```

Podstawy programowania w Javie

Zadania

1. Gra w zgadywanie liczb
2. Symulator LOTTO
3. Gra w zgadywanie liczb 2
4. Kostka do gry
5. Wyszukiwarka najpopularniejszych słów

Gra w zgadywanie liczb

Napisz prostą grę w zgadywanie liczb. Komputer ma wylosować liczbę w zakresie od 1 do 100.
Następnie:

1. wypisać: "Zgadnij liczbę" i pobrać liczbę z klawiatury;
2. sprawdzić, czy wprowadzony napis, to rzeczywiście liczba i w razie błędu wyświetlić komunikat: "To nie jest liczba", po czym wrócić do pkt. 1;
3. jeśli liczba podana przez użytkownika jest mniejsza niż wylosowana, wyświetlić komunikat: "Za mało!", po czym wrócić do pkt. 1;
4. jeśli liczba podana przez użytkownika jest większa niż wylosowana, wyświetlić komunikat: "Za dużo!", po czym wrócić do pkt. 1;
5. jeśli liczba podana przez użytkownika jest równa wylosowanej, wyświetlić komunikat: "Zgadłeś!", po czym zakończyć działanie programu.

Gra w zgadywanie liczb

Przykład:

```
Zgadnij liczbę: cześć
To nie jest liczba.
Zgadnij liczbę: 50
Za mało!
Zgadnij liczbę: 75
Za dużo!
Zgadnij liczbę: 63
Zgadłeś!
```

Symulator LOTTO

Jak wszystkim wiadomo, LOTTO to gra liczbowa polegająca na losowaniu 6 liczb z zakresu od 1 do 49. Zadaniem gracza jest poprawne wytypowanie losowanych liczb. Nagradzane jest trafienie 3, 4, 5 lub 6 poprawnych liczb.

Napisz program, który:

- zapyta o typowane liczby, przy okazji sprawdzi następujące warunki:
  - czy wprowadzony ciąg znaków jest poprawną liczbą,
  - czy użytkownik nie wpisał tej liczby już poprzednio,
  - czy liczba należy do zakresu 1-49,
- po wprowadzeniu 6 liczb, posortuje je rosnąco i wyświetli na ekranie,
- wylosuje 6 liczb z zakresu i wyświetli je na ekranie,
- poinformuje gracza, czy trafił przynajmniej "trójkę".

Symulator LOTTO

Aby wylosować 6 liczb z zakresu 1-49 bez powtórzeń możemy utworzyć tablicę z wartościami z tego zakresu, wymieszać jej zawartość i pobrać pierwsze 6 elementów.

Poniższy kod powinien Ci pomóc:

```
Integer[] arr = new Integer[49];
for (int i = 0; i < arr.length; i++) {
  arr[i] = i + 1;
}
System.out.println(Arrays.toString(arr));
Collections.shuffle(Arrays.asList(arr));
System.out.println(Arrays.toString(arr));
```

Możesz również losować liczby z określonego zakresu przy użyciu klasy **`Random`** (sprawdź w snippetach jak to wykonać) – jeżeli wybierzesz takie rozwiązanie, pamiętaj o sprawdzaniu czy dana wartość nie została wcześniej wylosowana.

Gra w zgadywanie liczb 2

Odwróćmy teraz sytuację z warsztatu "Gra w zgadywanie liczb": to użytkownik pomyśli sobie liczbę z zakresu 1-1000, a komputer będzie zgadywał i zrobi to maksymalnie w 10 ruchach (pod warunkiem, że gracz nie będzie oszukiwał).

Zadaniem gracza będzie udzielanie odpowiedzi "więcej", "mniej", "trafiłeś".

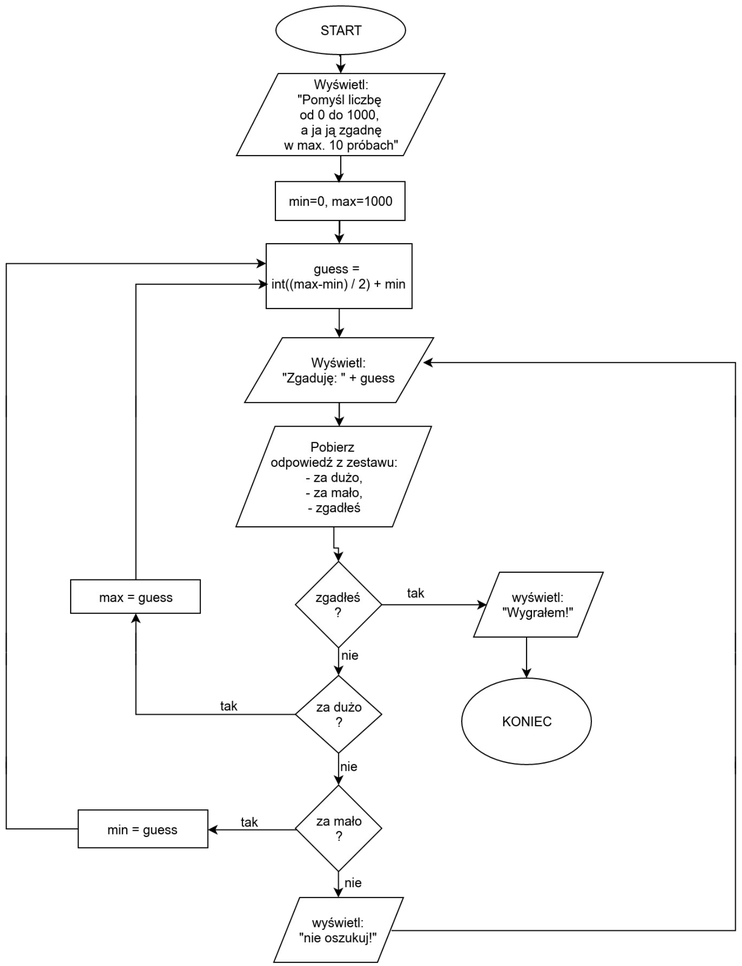
Na następnym slajdzie znajduje się schemat blokowy algorytmu.

Dostępny jest także pod adresem:

https://gist.github.com/arek-jozwiak-coderslab/4783d45e75a71793a123673cc0998ae3

Zaimplementuj go w Javie.

Gra w zgadywanie liczb 2



Kostka do gry

W grach planszowych i papierowych RPG używa się wielu rodzajów kostek do gry, nie tylko tych dobrze znanych, sześciennych. Jedną z popularniejszych kości jest np. kostka dziesięciościenna, a nawet stuścienna!

Ponieważ w grach kośćmi rzuca się często, pisanie za każdym razem np. "rzuć dwiema kostkami dziesięciościennymi, a do wyniku dodaj 20", byłoby żmudne, czasochłonne i marnowałoby ogromne ilości papieru.

W takich sytuacjach używa się kodu skracającego polecenie np.:

"**rzuć 2D10+20**".

Kostka do gry

Kod takiej kostki wygląda następująco:

xDy+z

gdzie:

- **y** – rodzaj kostek, których należy użyć (np. D6, D10),
- **x** – liczba rzutów kośćmi (jeśli rzucamy raz, ten parametr jest pomijalny),
- **z** – (opcjonalnie) liczba, którą należy dodać (lub odjąć) do wyniku rzutów.

Przykłady:

- **2D10+10** – 2 rzuty D10, do wyniku dodaj 10,
- **D6** – zwykły rzut kostką sześcienną,
- **2D3** – rzut dwiema kostkami trójściennymi,
- **D12-1** – rzut kością D12, od wyniku odejmij 1.

Kostka do gry

Napisz funkcję, która:

1. przyjmie w parametrze taki kod w postaci String,
2. rozpozna wszystkie dane wejściowe:
   - rodzaj kostki,
   - liczbę rzutów,
   - modyfikator,
3. wykona symulację rzutów i zwróci wynik.

Typy kostek występujące w grach:

D3, D4, D6, D8, D10, D12, D20, D100.

Wyszukiwarka najpopularniejszych słów

- Zaimportuj do projektu bibliotekę **jsoup**, możesz ją pobrać z adresu: https://jsoup.org/download
- Wyszukaj w popularnych serwisach internetowych nagłówków artykułów, a następnie zapisz pojedyncze słowa w nich występujące do pliku o nazwie **`popular_words.txt`**. Przykład pobrania tytułów z tagu html **`span`** z atrybutem class o wartości **`title`**:

```
Connection connect = Jsoup.connect("http://www.onet.pl/");
try {
    Document document = connect.get();
    Elements links = document.select("span.title");
    for (Element elem : links) {
        System.out.println(elem.text());
    }
} catch (IOException e) {
    e.printStackTrace();
}
```

Wyszukiwarka najpopularniejszych słów

- Wywołaj pobieranie dla wybranych serwisów internetowych.
- Pomiń wszystkie elementy krótsze niż 3-znakowe.
- Utwórz tablicę elementów wykluczonych np. **oraz**, **ponieważ**
- Wczytaj utworzony plik **`popular_words.txt`** i na jego podstawie utwórz plik **`filtered_popular_words.txt`**, który zawierać będzie wszystkie znalezione słowa, pomijając słowa wykluczone.