



Podstawy programowania

mgr. inż Filip Rynkiewicz
f.rynkiewicz@vizja.net

AGENDA

1. Tablice
 - ❖ Co to są tablice?
 - ❖ Jak tworzyć tablice w Javie?
 - ❖ Kontenery
 - ❖ ArrayList
2. Działania na plikach
 - ❖ Wczytywanie z pliku
 - ❖ Zapisywanie do pliku
3. Generator liczb losowych
 - ❖ Jak stworzyć?
4. Zadanie zaliczające

TABLICE

Tablica – array

Ogólna składnia tworzenia tablic

```
typ [] nazwa = new type [rozmiar]
```

Przykład wykorzystania

```
double array[] = new double[20];
```

```
double[] array2 = new double[20];
```

```
int rozmiar = 20;  
double[] array2 = new double[rozmiar];
```

```
double[] array = {1.032, 2.1213, 3.32, 4.54};
```

Metody

Pobieranie element na n miejscu

```
int[] array = {3, 2, 4, 1};  
int wartosc = array[1];
```

Zmiana wartosci elementu na n miejscu

```
int[] array = {3, 2, 4, 1};  
array[1] = 11;
```

Pobieranie rozmiaru tablicy

```
int[] array = {3, 2, 4, 1};  
int dlugosc = array.length;
```

Printowanie elementow tablicy

```
int[] array = {3, 2, 4, 1};  
int dlugosc = array.length;  
for( int i = 0; i < dlugosc ; i++){  
    System.out.println(array[i]);  
}
```

```
int[] array = {3, 2, 4, 1};  
for( int element : array){  
    System.out.println(element);  
}
```

Zadanie

1. Stworzyc program ktory z podanej tablicy obliczy srednia liczb
 1. Tablica = [1,4,2.3,11.34,342,545,0.1,22.3]
2. Stworzyc program ktory pokaze czesc wspolna dwoch tablic
 1. Tablica1 = [1,2,3,4,5,6,7,8,9]
 2. Tablica2 = [3,4,11,35,64,2,8,5]
3. Stworzyc program ktory obliczy ile jest liczb parzystych I nieparzystych w tablicy
 1. Tablica = [1,2,3,4,5,7,5,23,654,123,321]
4. Stworzyc program ktory znajdzie najmniejsza oraz największa liczbe w tablicy
 1. Tablica = [2513,-1234,-923,55421,1]

KONTENERY

ArrayList

Ogólna składnia tworzenia tablic

```
import java.util.ArrayList;  
ArrayList<klasa> nazwa = new ArrayList<klasa>();
```

Wykorzystanie

```
ArrayList<String> listZakupow = new ArrayList<String>();
```

W ArrayList nie można wykorzystywać typów wbudowanych (POD)!

Metody

Dodawanie element do ArrayList

```
ArrayList<String> listZakupow = new ArrayList<String>();  
listZakupow.add("Mleko");
```

Pobieranie elementu z ArrayList na *n* miejscu

```
ArrayList<String> listZakupow = new ArrayList<String>();  
listZakupow.add("Mleko");  
String element = listZakupow.get(0);
```

Zmiana elementu z ArrayList na *n* miejscu

```
ArrayList<String> listZakupow = new ArrayList<String>();  
listZakupow.add("Mleko");  
listZakupow.set(0, "Krowie mleko");
```

Usuniecie elementu z ArrayList na *n* miejscu

```
ArrayList<String> listZakupow = new ArrayList<String>();  
listZakupow.add("Mleko");  
listZakupow.remove(0);
```

Metody cd

Ilość elementów ArrayList

```
ArrayList<String> listZakupow = new ArrayList<String>();  
listZakupow.add("Mleko");  
listZakupow.size();
```

Usunięcie wszystkich elementów z ArrayList

```
ArrayList<String> listZakupow = new ArrayList<String>();  
listZakupow.add("Mleko");  
listZakupow.add("Mleko krowie");  
listZakupow.add("Mleko UHT");  
listZakupow.clear();
```

Iterator

Iteracja z użyciem iteratora

```
ArrayList<Double> tablica = new ArrayList<Double>();  
Iterator<Double> iterator = tablica.iterator();  
while (iterator.hasNext()) {  
    System.out.println(iterator.next());  
}
```

Usuwanie elementu

```
Iterator<Double> iterator = tablica.iterator();  
while (iterator.hasNext()) {  
    Double number = iterator.next();  
    if (number == 1) {  
        iterator.remove();  
    }  
}
```

Zadanie

1. Stworzyc program ktory z podanej tablicy usunie tylko parzyste element
 1. Tablica = [-2,32,11,3,4,1124556,2]
2. Stworzyc program ktory z elementow tablicy stworzy nowa tablice, skopuje te element ktore spełniają następujące warunki
 1. Element jest podzielny przez 5 oraz przez 6
 2. Reszta z dzielenia wartosci element przez 7 jest rowna 1 (element % 7 == 1)
3. Stworzyc program ktory do podanej tablicy doda element rowne kwadratowi kazdego element .Tablica = [1,3,5]
 1. Iteracja 1 = [1,3,5,1,9,25]
 2. Iteracja 2 = [1,3,5,1,9,25,1,9,25,1,81,625]

Obsługa plików

Zapisywanie do pliku

```
public static void main(String[] args) throws IOException {  
    try {  
        FileWriter myWriter = new FileWriter("filename.txt");  
        myWriter.append("Tekst ktory ma byc w pliku");  
        myWriter.close();  
        System.out.println("Zapisanie do pliku bez bledow");  
    } catch (IOException e) {  
        System.out.println("BLAD!");  
    }  
}
```

Wyjątek

Nazwa pliku

Dodaj tekst do pliku

Zamknij plik

Złap wyjątek

Zadanie

Stworzyc program ktory:

1. Stworzy tablice (czy w formie array czy ArrayList) liczb parzystych od 1 do 10.000.000.
2. Zapisze kazda z tych liczb do pliku w osobnej lini.

Wynikiem program powinien byc plik z wszystkimi liczbami parzystymi z przedzialu 1 do 10.000.000

Czytanie linia po lini z pliku

Nazwa pliku

Wyjątek

Wczytana linia z pliku

Zamknięcie pliku

```
public static void main(String[] args) throws IOException {  
    String file = "file.txt";  
    BufferedReader bufferedReader = new BufferedReader(new FileReader(file));  
  
    String curLine;  
    while ((curLine = bufferedReader.readLine()) != null){  
        System.out.println(curLine);  
    }  
    bufferedReader.close();  
}
```


Zadanie

Wczytać plik stworzony przy poprzednim zadaniu. Z wczytanych liczb stworzyć listę i przefiltrować ją aby zawierała tylko liczby podzielne przez:

- ❖ 4
- ❖ 11
- ❖ 23

Zapisać liczby do innego pliku.

Generator liczb losowych (RNG)

Klasy RNG

RANDOM

```
int min = 10;  
int max = 120;  
Random r = new Random();  
int liczbaLosowa = r.nextInt((max-min)+1)+min;  
System.out.println(liczbaLosowa);
```

MATH.RANDOM

```
int min = 10;  
int max = 120;  
int liczbaLosowa = (int)(Math.random()*((max-min)+1))+min;  
System.out.println(liczbaLosowa);
```

Zadanie

Stworzyc program ktory wygeneruje 30.000 losowych liczb z przedzialu 0 do 10.000.000 a nastepnie przefiltruje je tak jak w poprzednim zadaniu.

Następne zajęcia. Zadanie zaliczeniowe

Na zajęcia 4

❖ • Implementacja sortowania :

o) Bąbelkowego

1) Przez wstawianie

2) Przez wybieranie

❖ Musi wykorzystywać tablice/ArrayList

❖ Musi wykorzystywać wczytywanie liczb z pliku

❖ Musi wykorzystywać sortowanie liczb wygenerowanych losowo

Algorytm wyboru wariantów:

Zsumowanie wszystkich cyfr w numerze indeksu modulo 3.

```
int numerIndeksu = 173186;  
int wariantSortowania = (1 + 7 + 3 + 1 + 8 + 6) % 3 ;  
// wariantSortowania = 2
```

Dziękuję za uwagę!

