

Algorytmy i struktury danych

LABORATORIUM

Środowiska wykorzystywane na zajęciach laboratoryjnych

- **Java Development Kit (JDK)**
<https://www.oracle.com/technetwork/java/javase/downloads/index.html>
- **Eclipse**
<https://www.eclipse.org/downloads/>



Literatura

- [CLRS] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein. Wprowadzenie do algorytmów. PWN, Warszawa, 2018 (pierwsze wydanie: 1990).
- [BDR] L. Banachowski, K. Diks, W. Rytter. Algorytmy i struktury danych. WNT, Warszawa, 2018 (pierwsze wydanie: 1996).
- [SW] R. Sedgewick, K. Wayne. Algorytmy. Wyd. IV, Helion, Gliwice, 2018.

Prowadzący laboratorium

- **Dr hab. Dariusz Barbucha, prof. UMG** (Katedra Systemów Informatycznych)
- **Dr hab. inż. Włodzimierz Filipowicz, prof. UMG** (Katedra Systemów Informatycznych)

Sale laboratoryjne

Katedra Systemów Informatycznych, Budynek F

- **Lab 1:** sala F-13 (parter)
- **Lab 2:** sala F-112 (piętro)
- **MS Teams**

Algorytmy i struktury danych

LABORATORIUM

Zajęcia 1.

Środowisko Eclipse oraz struktur programu w języku Javie.

Wybrane problemy algorytmiczne i ich implementacja w języku Java.

Struktura danych tablicy.

Podstawowe operacje na tablicach jedno- i dwuwymiarowych liczb całkowitych/rzeczywistych.

Cel zajęć

1. Zapoznanie studentów ze środowiskiem Eclipse oraz strukturą programu w języku Java, jak również implementacja wybranych problemów algorytmicznych w tym języku.
2. Zapoznanie studentów ze strukturą danych tablicy, implementacja podstawowych operacji na tablicach, jak również implementacja wybranych problemów algorytmicznych wykorzystujących strukturę tablicy.

Struktura programu w języku Java (bardzo ogólnie) na przykładzie BinarySearch [SW]:

Kod programu

```
import java.util.Arrays;

public class BinarySearch {

    public static int rank(int key, int[] a) {
        int lo = 0;
        int hi = a.length - 1;
        while (lo <= hi) {
            int mid = lo + (hi - lo) / 2;
            if (key < a[mid]) hi = mid - 1;
            else if (key > a[mid]) lo = mid + 1;
            else return mid;
        }
        return -1;
    }

    public static void main(String[] args) {

        int[] whitelist = In.readInts(args[0]);

        Arrays.sort(whitelist);

        while (!StdIn.isEmpty()) {
            int key = StdIn.readInt();
            if (rank(key, whitelist) == -1)
                StdOut.println(key);
        }
    }
}
```

Uruchomienie

```
% java BinarySearch largeW.txt < largeT.txt

499569
984875
```

Zadania

1. Wczytaj dwie liczby całkowite z klawiatury. Wydrukuj większą z nich.
2. Wczytaj trzy liczby całkowite z klawiatury. Wydrukuj najmniejszą z nich.
3. Wczytaj z klawiatury liczby całkowite tak długo, aż użytkownik poda liczbę 0. Następnie wydrukuj sumę i średnią arytmetyczną podanych liczb.
4. Wczytaj liczbę naturalną z klawiatury. Sprawdź, czy podana liczba jest liczbą pierwszą, a następnie wydrukuj na ekranie stosowny komunikat.
5. Wczytaj dwie liczby naturalne z klawiatury. Wyznacz:
 - największy wspólny dzielnik tych liczb (NWD),
 - najmniejszą wspólną wielokrotność tych liczb (NWW).
6. Wyznacz i wydrukuj na ekranie sumę pierwszych 100 liczb naturalnych dodatnich.
7. Wyznacz i wydrukuj na ekranie sumę pierwszych 11 liczb – potęg dwójki, tj.: 2^0 , 2^1 , 2^2 , ..., 2^{10} .
8. Jak jest złożoność czasowa zaproponowanych w zad. 6 i 7 algorytmów? Zaproponuj algorytmy o niższej złożoności.
9. Wczytaj z klawiatury 6 liczb całkowitych i zapisz je w strukturze tablicy. Wydrukuj powstałą tablicę.
10. Wygeneruj tablicę 12 liczb całkowitych z przedziału $[0, 100)$, a następnie wydrukuj powstałą tablicę. Zlicz, ile jest liczb parzystych w tej tablicy, wydrukuj stosowny komunikat.
11. Wygeneruj tablicę 30 liczb całkowitych z przedziału $[10, 50)$, a następnie wydrukuj powstałą tablicę. Sprawdź czy liczba 31 występuje w tablicy. Wydrukuj stosowny komunikat. Zastosuj algorytm:
 - wyszukiwania sekwencyjnego,
 - wyszukiwania binarnego.

Jaka jest złożoność obu algorytmów? Co to oznacza dla tablicy 100, 1000, 10000, 100000 elementów?