

## **Podstwy Javy, wejście/wyjście**

sprawozdanie z ćwiczeń laboratoryjnych 5:

**Agnieszka Majkut**

**nr indeksu 286116**

## Wprowadzenie:

Celem laboratoriów było zapoznanie się z podstawami dotyczącymi środowiska programowania jakim jest Java, tj wejście/wyjście.

Do kodu z poprzednich zajęć należało dodać:

- wprowadzenie danych z klawiatury
- zapis danych do pliku tekstowego
- wprowadzenie danych z pliku zawierającego liczby zmiennoprzecinkowe
- zapis i odczyt obiektów z użyciem serializacji

Do wybranego tematu 9, dotyczącego drukarni, dodałam powyższe zapisy i odczyty. Zmodyfikowałam również kod programu na potrzeby dodania zapisu i odczytu obiektów za pomocą serializacji. Program na te zajęcia również napisałam w środowisku IntelliJ IDEA 2017.3.4. Użytkownik podaje jaką czynność chce wykonać (dodać zamówienie, sprawdzić stan zamówienia czy sprawdzić nowości w drukarni), wybiera również ilość kopii, które chce wydrukować oraz format kopii.

Zmienne zadeklarowane w programie:

- **priceBulk** – cena hurtowa druku w drukarni
- **price** – cena detaliczna druku w drukarni
- **order** – zmienna umożliwiająca sprawdzenie statusu zamówienia
- **format** – zmienna umożliwiająca wybór formatu druku
- **count** – zmienna zapoamiętujuca ilość kopii
- **finish** – tablica jednowymiarowa do wypisania na ekranie końca procesu przy zakończeniu działania programu
- **index** – tablica dwuwymiarowa, w której zapisane jest położenie zamówienia, pierwszy indeks oznacza numer drukarki, natomiast drugi położenie zamówienia klienta wśród innych zamówień z danej drukarki

Klasy zaimplementowane w programie:

- **Press** – klasa główna, w której stworzyłam obiekty znajdujące się w pakiecie *press*
- **Book** – klasa książek, implementuje interfejs Print, rozszerza klasę abstrakcyjną Payment znajdującą się w pakiecie *Books*
- **Magazine** – klasa magazynów, implementuje interfejs Print, rozszerza klasę abstrakcyjną Payment znajdującą się w pakiecie *Magazines*
- **Poster** – klasa plakatów zachęcających do zakupu nowo wydanej książki, dziedziczy po klasie Book znajdującą się w pakiecie *Books*
- **Advertisement** – klasa reklam nowego nakładu magazynu znajdującą się w pakiecie *Magazines*
- **Payment** – klasa abstrakcyjna zawierająca metodę calculateOfPayment(), która oblicza wartość do zapłaty znajdującą się w pakiecie *press*
- **Print** – interfejs wyświetlania znajdujący się w pakiecie *press*

W projekcie utworzyłam trzy pakiety:

- **Books** – odpowiada za drukowanie zamówień na książki
- **Magazines** – odpowiada za drukowanie zamówień na magazyny/gazety
- **press** – zawiera interfejs Print, klasę Advertisement oraz główną klasę Press, która symuluje działanie drukarni

- Do programu należało dodać  
➔ wprowadzanie z klawiatury:

```
System.out.println("Podaj ilość kopii:"); // wprowadzenie danych z klawiatury
count = scanner.nextInt();
if (count >= 0) {
    throw new IllegalArgumentException("Podana wartość musi być większa od 0.");
}
else if (count >= 1000) {
    System.out.println("Zniżka 30%. Do zapłaty: " + 0.3 * (count * priceBulk));
    return 0.3 * (count * priceBulk);
} else {
    System.out.println("Cena detaliczna. Do zapłaty:" + count * price);
    return count * price;
}
```

rys.1 Użyłam obiekty klasy scanner i przypiąłam ją do count, jest to ilość kopii jaką klient chce wydrukować

- ➔ zapis danych do pliku tekstowego:

```
/* ZAPIS DO PLIKU */
PrintWriter write = new PrintWriter( fileName: "Book.txt");
Book book = new Book( author: "Stephen King", title: "Smentarz dla zwierzat", publishingHouse: "Prószyński i Sp.", count: 1000);
book.print();
write.println("Autor: Stephen King" + "\t Tytuł: Smentarz dla zwierzat" + "\t Wydawnictwo: Prószyński i Sp." + "\t Ilość: " + 1000);
write.println("Autor: Stephen King");
write.println("Tytuł: Smentarz dla zwierzat");
write.println("Wydawnictwo: Prószyński i Sp.");
write.println("Ilość: " + 1000);
write.close();

/* ZAPIS DO PLIKU */
Magazine magazine = new Magazine( nameOfMagazine: "Świat Wiedzy", publishingHouseMagazine: "Bauer", countMagazine: 100);
magazine.print();
PrintWriter write2 = new PrintWriter( fileName: "Magazine.txt");
write2.println("Nazwa magazynu: Świat Wiedzy" + "\t Wydawnictwo: Bauer" + "\t Ilość: " + 100);
write2.println("Nazwa magazynu: Świat Wiedzy");
write2.println("Wydawnictwo: Bauer");
write2.println("Ilość: " + 100);
write2.close();
```

rys.2 Utworzyłam pliki Book.txt, następnie utworzyłam obiekt book typu Book, wypisuje go na ekranie oraz za pomocą write.println(...) zapisuję do pliku i zamykam strumień(write.close());  
znalognicznie zrobiłam dla obiektu magazine klasy Magazine

- ➔ wprowadzenie danych z pliku zawierającego liczby zmiennoprzecinkowe

```
public static void zmiennoprzecinkowe() {
    // odczyt = pliku, który zawiera liczby zmiennoprzecinkowe
    System.out.print("Odczyt = pliku zawierający liczby przecinkowe: ");
    try {
        FileInputStream fileInputStream = new FileInputStream( name: "D:\\_Bogusława Stachura Inteligencja\\LABORATORIA\\lab_5\\Zmiennoprzecinkowe.txt");
        // Odczytywanie jednego bajtu z pliku
        int b = fileInputStream.read();
        while(b != -1) { // dopóki jest bajt
            // Przekształcanie na typ zmiennoprzecinkowy
            System.out.print((char) b);
            b = fileInputStream.read();
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
    catch (IOException e) {
        e.printStackTrace();
    }
    System.out.println("\n");
}
}
```

rys.3 Wprowadzanie liczb zmiennoprzecinkowych napisałam w metodzie, którą wywołuję na początku funkcji main()

- ➔ zapis i odczyt obiektów z użyciem serializacji

```
/* SERIALIZACJA */
Books.Poster poster = new Books.Poster( author: "Stephen King", title: "Smentarz dla zwierzat", publishingHouse: "Prószyński i Sp.", count: 1000, posterFormat: "A3");
poster.getPosterFormat(); // sprawdzanie rozszerzenia klasy Poster przez metody klasy Book
poster.print();

// ZAPIS DO PLIKU
PrintWriter serialWrite = new PrintWriter( fileName: "Poster(serial).txt");
serialWrite.println(posters.getAuthor() + " " + poster.getTitle() + " " + poster.getPublishingHouse() + " " + poster.getCount() + " " + poster.getPosterFormat());
serialWrite.close();
System.out.println("Sprawdzenie rozszerzenia klas dla klasy 'Poster': \n użycie dziedziczonej metody 'getTitle' = klasy 'Book': \n" + poster.getTitle() + "\n");

// odczyt = pliku zapisany w wyniku serializacji
File file = new File( fileName: "Poster(serial).txt");
Scanner po = new Scanner(file);

String serialRead = po.nextLine();
System.out.println(serialRead + "\n");
```

Jak można zauważyć język Java różni się od znanego już języka programowania, tj C++, ale widać duże podobieństwa. Język Java jest trochę bardziej złożony niż C++. Wszystko zawiera się w klasach – o czym trzeba pamiętać.

Serializacja jest wbudowanym mechanizmem zapisywania obiektów, który pozwala na zapis binarny całego drzewa obiektów (zapisuje obiekty wraz z ich referencjami w strumieniu wyjściowym). Dzięki niej można również przysyłać obiekty przez sieć.

Do wprowadzania danych z klawiatury należy dodać bibliotekę (zaimportować) *java.util.Scanner*. Natomiast, aby móc zapisać dane do pliku zaimportowałam *java.io.PrintWriter*.

Przy wczytywaniu z klawiatury należy pamiętać, by stworzyć obiekt typu Scanner, co umożliwi nam wpisywanie z klawiatury, np.

```
static Scanner scanner = new Scanner(System.in).
```