



# Przetwarzanie danych - I/O

Łukasz Bednarski

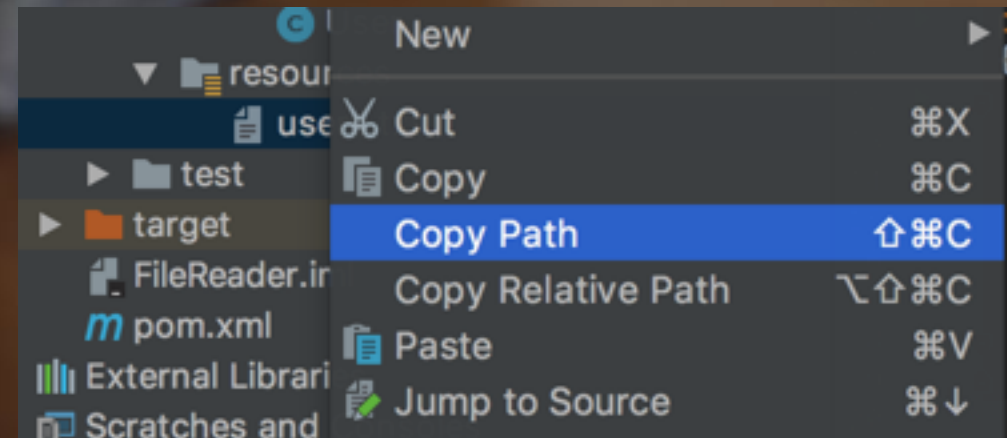


1. Strumienie w Javie pozwalają na odczyt i zapis danych z/do pliku.
2. Służą do tego klasy pozwalające na operowanie na strumieniach danych.
3. Klasa `InputStream` jest podstawową klasą abstrakcyjną, definiującą metody odczytu bajtów ze strumienia.
4. Klasa `OutputStream` jest podstawową klasą abstrakcyjną, definiującą metody zapisu bajtów ze strumienia.
5. Istnieje kilka sposobów dostępu do pliku.

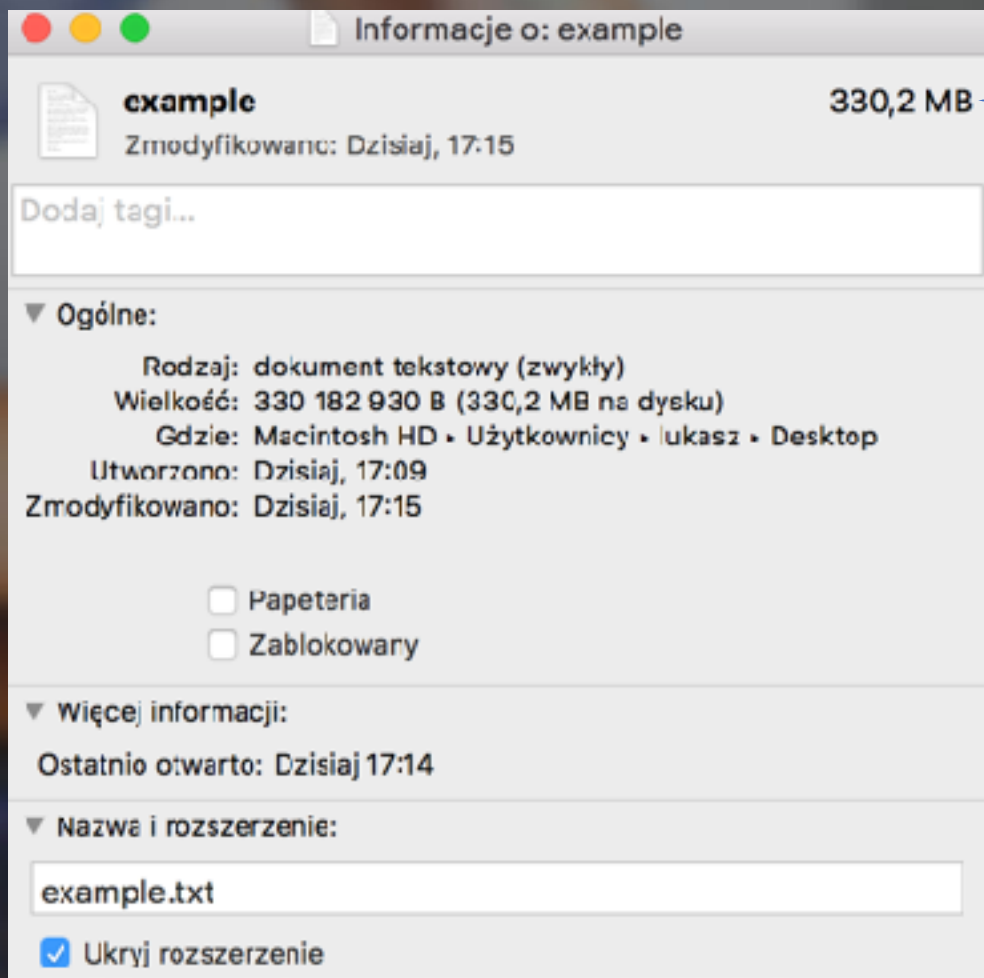


# Wczytywanie pliku do pamięci

```
public static void main( String[] args ) throws IOException {  
    File file = new File("/Users/lukasz/IdeaProjects/FileReader/src/main/  
resources/example.txt");  
    //Otwiera strumień dostępu do pliku  
    FileInputStream fis = new FileInputStream(file);  
    // Tworzy tablicę bajtów aby zapisać zawartość pliku  
    byte[] data = new byte[(int) file.length()];  
    //Wczytuje dane z pliku ze strumienia do tablicy data  
    fis.read(data);  
    //Zamyka strumień  
    fis.close();  
  
    String str = new String(data, "UTF-8");  
}
```



# Wczytywanie pliku do pamięci



```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at java.util.Arrays.copyOf(Arrays.java:3332)
    at java.lang.StringCoding.safeTrim(StringCoding.java:89)
    at java.lang.StringCoding.access$100(StringCoding.java:50)
    at java.lang.StringCoding$StringDecoder.decode(StringCoding.java:154)
    at java.lang.StringCoding.decode(StringCoding.java:193)
    at java.lang.String.<init>(String.java:426)
    at java.lang.String.<init>(String.java:491)
    at pl.lbednarski.App.main(App.java:14)
```





# Wczytywanie pliku po bajcie

```
public static void main( String[] args ) throws IOException {  
    File file = new File("/Users/lukasz/IdeaProjects/FileReader/src/main/resources/  
example.txt");  
    //Otwiera strumień dostępu do pliku  
    FileInputStream fis = new FileInputStream(file);  
    //Wczytanie pierwszego bajtu  
    int bajt = fis.read();  
    //metoda read zwraca wartość -1 po ostatnim wczytanym bajcie z pliku  
    while(bajt != -1){  
        //bajt jest rzutowany na znak i wypisywany w konsoli  
        System.out.print((char)bajt);  
        //Wczytywany jest kolejny bajt  
        bajt = fis.read();  
    }  
}
```

App X

- keeping the register of persons involved in the processing of personal data (Art. 33 paragraph 1 of the GDPR) pursuant to Art. 24 paragraph 4 of the Regulation (this register)
- Appointing a person referred to as "an administrator of information security", who is responsible for personal data security within the computer system. The scope of duties
- Determining individual job specifications and responsibilities for the persons authorised to process personal data. The scope of job and responsibilities should specify the
- Marking out an instruction of conduct in cases when personal data protection has been violated; the instruction is designed for employees working on personal data processing
- the violation of the computer system security has been revealed,
- the state of the appliances, contents of the personal data file, revealed methods of work, procedures of programme functioning or the quality of communication within the
- Determining the buildings, premises or their parts where personal data are processed by means of stationary computer equipment.
- Marking out an instruction that would define the way in which the computer systems used for personal data processing are to be managed. The instruction should, in particular
- definition of how to lay down methods of passwords distribution between users and frequency of their changes and an indication of a person who is responsible for the afore
- a definition of how to lay down methods in which users will be logged in and out (measuring the beginning and the end of work),
- methods and frequency of making emergency copies,
- methods and frequency of detecting and deleting computer viruses,
- methods and period of information media storing including data copies and printouts,
- methods of performing system and personal file service routine,



# Wczytywanie pliku linijka po linijce

```
public static void main( String[] args ) throws IOException {  
    File file = new File("/Users/lukasz/IdeaProjects/FileReader/src/main/resources/  
example.txt");  
    //Otwiera strumień znakowy do pliku  
    FileReader fis = new FileReader(file);  
    String linia = "";  
    //Otwiera bufor do czytania znaków z pliku  
    BufferedReader bfr = new BufferedReader(fis);  
    //dopóki metoda readLine() nie zwraca null (koniec pliku) wczytuj kolejne linie  
    while((linia = bfr.readLine()) != null){  
        System.out.println(linia);  
    }  
}
```

App x

keeping the register of persons involved in the processing of personal data (Art. 32 paragraph 1 of the GDPR); pursuant to Art. 24 paragraph 4 of the Regulation data register

Appointing a person referred to as "an administrator of information security", who is responsible for personal data security within the computer system. The scope of duties

Determining individual job specifications and responsibilities for the persons authorised to process personal data. The scope of job and responsibilities should specify the

Working out an instruction of conduct in cases when personal data protection has been violated; the instruction is designed for employees working on personal data processing

the violation of the computer system security has been revealed,

the state of the appliances, contents of the personal data file, revealed methods of work, procedures of programme functioning or the quality of communication within the

Determining the buildings, premises or their parts where personal data are processed by means of stationary computer equipment.

Working out an instruction that would define the way in which the computer systems used for personal data processing are to be managed. The instruction should, in particular

definition of how to lay down methods of passwords distribution between users and frequency of their changes and an indication of a person who is responsible for the afore

a definition of how to lay down methods in which users will be logged in and out and an indication of a person responsible for the aforesaid activities,

procedures of clocking employees in and out (measuring the beginning and the end of work),

methods and frequency of making emergency copies,

methods and frequency of detecting and deleting computer viruses,

methods and period of information media storing including data copies and printouts,

methods of performing system and personal file service routine,

# Apache Commons IO



```
public static void main( String[] args ) throws IOException {  
    File file = new File("/Users/lukasz/IdeaProjects/FileReader/src/  
main/resources/example.txt");
```

```
    LineIterator fileContents= FileUtils.lineIterator(file, "UTF-8");  
    while(fileContents.hasNext()){  
        System.out.println(fileContents.nextLine());  
    }  
}
```

```
<dependency>  
  <groupId>commons-io</groupId>  
  <artifactId>commons-io</artifactId>  
  <version>2.6</version>  
</dependency>
```

# Zapis do pliku linia po linii



```
public static void main( String[] args ) throws IOException {  
    List<String> textLines = Arrays.asList("Jan Nowak", "Marek Kowalski", "Janina Dąbrowska");  
    File file = new File("/Users/lukasz/IdeaProjects/FileReader/src/main/resources/example.txt");  
    //Otwiera strumień dostępu do pliku  
    FileWriter fileWriter = new FileWriter(file);  
    //Otwiera bufor zapisywania znaków do pliku  
    BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);  
  
    try {  
        for (String line : textLines) {  
            //Zapis linijki do pliku  
            bufferedWriter.write(line);  
            //Dodanie znaku końca linii  
            bufferedWriter.newLine();  
        }  
    } finally {  
        bufferedWriter.close();  
    }  
}
```

App.java	users.txt	example.txt
1		Jan Nowak
2		Marek Kowalski
3		Janina Dąbrowska
4		



# Zapis do pliku Apache IO



```
public static void main( String[] args ) throws IOException {  
    List<String> textLines = Arrays.asList("Jan Nowak", "Marek Kowalski", "Janina  
Dąbrowska");  
    File file = new File("/Users/lukasz/IdeaProjects/FileReader/src/main/resources/  
example.txt");  
  
    //Zapis listy wszystkich linijek do pliku  
    FileUtils.writeLines(file, textLines);  
}
```

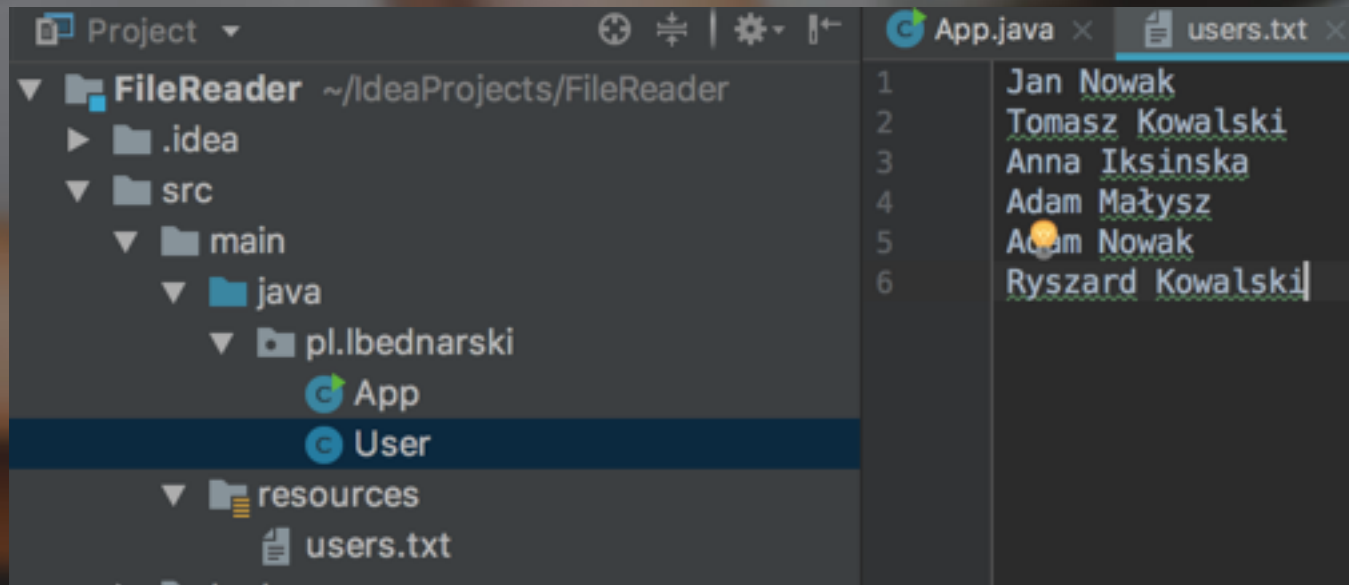
Line	Content
1	Jan Nowak
2	Marek Kowalski
3	Janina Dąbrowska

# Które metody stosować?



1. Wszystko zależy od tego jaki plik przetwarzamy
2. Plik o niedużych rozmiarów można wczytać do pamięci aby łatwiej było go przetwarzać.
3. Pliki bardzo dużych rozmiarów, warto wczytywać bajt po bajcie.
4. Wczytywanie bajt po bajcie utrudnia analizowanie zawartości pliku (np. wyodrębnianie słów do tworzenia obiektów) dlatego jeśli jest to możliwe, warto wczytywać plik linia po linii.
5. Analogicznie sytuacja wygląda z zapisem.

# Tworzenie obiektów



```
public class User {
    private String firstName;
    private String lastName;

    public User(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getLastName() {
        return lastName;
    }

    @Override
    public String toString() {
        return "User{" +
            "firstName='" + firstName + '\'' +
            ", lastName='" + lastName + '\'' +
            '}';
    }
}
```

# Tworzenie obiektów



```
public static void main( String[] args ) throws IOException {  
    File file = new File("/Users/lukasz/IdeaProjects/FileReader/src/main/resources/  
users.txt");  
    List<User> users = new ArrayList<User>();
```

```
    LineIterator fileContents = FileUtils.lineIterator(file, "UTF-8");  
    while(fileContents.hasNext()){  
        //Wczytanie linijki z pliku i utworzenie tablicy  
        //Każdy element tablicy to kolejne słowo z linijki  
        String[] line = fileContents.nextLine().split(" ");  
        //Utworzenie obiektu klasy User i dodanie go do listy  
        users.add(new User(line[0], line[1]));  
    }
```

```
    for (User user : users) {  
        System.out.println(user);  
    }  
}
```

```
Run App x  
User{firstName='Jan', lastName='Nowak'}  
User{firstName='Tomasz', lastName='Kowalski'}  
User{firstName='Anna', lastName='Iksinska'}  
User{firstName='Adam', lastName='Małysz'}  
User{firstName='Adam', lastName='Nowak'}  
User{firstName='Ryszard', lastName='Kowalski'}
```



# Zadanie (przetwarzanie danych)



1. Utwórz projekt `FileReaderExamples` w którym zaimplementujesz obsługę plików. Do utworzonego projektu dodaj katalog `resources` (po utworzeniu kliknij PPM na nazwę projektu, wybierz z menu Maven a następnie Reimport - ikona katalogu powinna się zmienić).
2. Do katalogu `resources` dodaj plik `simpleExample.txt` i wyświetl jego zawartość w konsoli.
3. Do katalogu `resources` dodaj plik `users.txt` Plik zawiera listę imion oraz nazwisk wraz z wiekiem. Utwórz obiekt `User` i utwórz listę obiektów z danymi z pliku.
4. Rozbuduj poprzedni algorytm. Utwórz 2 listy. Na jednej zapisz kobiety, natomiast na drugiej mężczyzn (dla ułatwienia przyjmujemy że każda osoba z nazwiskiem kończącym się na „a” jest kobietą). Na listach zapisuj tylko osoby pełnoletnie.
5. Wczytaj plik `books.csv` w którym znajduje się lista książek. Utwórz listę obiektów, posortuj według dostępności w magazynie i ceny. Na koniec wyświetl listę książek w konsoli.
6. Zmodyfikuj poprzedni algorytm w taki sposób aby posortowana lista książek została zapisana w pliku `sortedBooks.csv`.

# Zadanie (przeszukiwanie)



1. Dodaj do projektu plik weather-data.csv. Zawiera on informacje pogodowe z jednej stacji meteorologicznej na przestrzeni prawie 70 lat (25 tysięcy rekordów).
2. Napisz program który pobierze od użytkownika datę i zwróci minimalną, średnią i maksymalną temperaturę tego dnia. Spróbuj zoptymalizować swój algorytm.
3. Dodaj metodę która przyjmie 2 daty a następnie wyświetli jaka była najwyższa, najniższa i średnia temperatura w tym okresie czasu.
4. Dodaj metodę która przyjmie liczbę całkowitą i zwróci liczbę dni w których średnia temperatura była wyższa lub równa przekazanej.
5. Dodaj metodę która przeanalizuje i zwróci 2 informacje o datach oraz różnice temperatur:
  1. Statystycznie który rok był najcieplejszy a który najchłodniejszy
  2. Statystycznie który miesiąc był najcieplejszy a który najchłodniejszy
  3. Statystycznie który kalendarzowy dzień był najcieplejszy a który najchłodniejszy

# Zadanie (przetwarzanie danych)



1. Otwórz plik `userActions.csv`. Plik zawiera dane które należy wprowadzić do systemu dotyczące użytkowników. Login użytkownika jest zawsze unikatowy. Plik ma strukturę: `[METODA][USER]`, np.
  1. `CREATE jnowak password Jan Nowak user`
  2. `CREATE anowak password2 Adam Nowak user`
  3. `UPDATE jnowak ; ; admin`
  4. `REMOVE anowak`
2. Akcja `CREATE` zawiera zawsze wypełnione wszystkie pola. `UPDATE` zawiera tylko login użytkownika (który jest unikatowy) oraz pole które powinno być aktualizowane. `REMOVE` zawiera tylko login użytkownika do usunięcia.
3. Utwórz klasę `UserService` która zawiera listę użytkowników. Zaimplementuj metody `addUser`, `updateUser` oraz `removeUser`. W zależności od akcji zdefiniowanej w pliku, wykonaj odpowiednią operację.

# Zadanie (analiza danych)



1. Wczytaj plik flights.csv - plik zawiera dane dotyczące pasażerów pewnej firmy lotniczej działającej w latach 1949 - 1960. Przeprowadź analizę danych a wyniki zapisz w nowym pliku.
2. Twoja analiza powinna zawierać:
  1. Ilu pasażerów przetransportowała firma przez cały okres działania?
  2. Zsumuj i zapisz liczbę pasażerów korzystających z usług firmy z podziałem na lata.
  3. Zsumuj i zapisz liczbę pasażerów korzystających z usług firmy z podziałem na każdy miesiąc.
  4. Podaj rok i miesiąc w którym pasażerów było najwięcej oraz najmniej (podaj także różnicę liczby pasażerów).
  5. O ile procent zmieniała się liczba pasażerów które korzystały z usług firmy każdego roku w stosunku do poprzedniego roku?