

# Sprawozdanie z Podstaw Programowania

---

Sprawozdanie z Podstawy Programowania

Grupa Patryk Piorunowicz/ Wiktor Malarczyk

## **Zadanie 3 „Niesforne dane”**

Wykonanie:

`dos2unix dane.txt` | (Konwertuje plik tekstowy na format UNIX).

`paste -d „/t” - - - <dane.txt > danePoukladane.txt` | (sortuje dane w pliku dane.txt w trzy kolumny i wkleja je do pliku danePoukladane.txt).

#### Zadanie 4 „Dodawanie poprawek”

Wykonanie:

Diff -u lista.txt lista-pop.txt > lista.patch | (komenda diff -u sprawdza zawartość podanych plików tekstowych i kopiuje różnicę między nimi do pliku lista.patch).

Patch lista.txt < lista.patch | (komenda patch wprowadza poprzednią wyodrębnioną różnicę z pliku poprawkowego do pliku docelowego).

Md5sum lista-pop.txt lista.txt | (md5sum sprawdza integralność między nowo poprawionym plikiem lista.txt a lista-pop.txt. Ciągi liczb i cyfr zgadzają się w obu plikach co oznacza że zadanie zostało poprawnie wykonane).

```
skd_s2@LAB318-2 UCRT64 ~  
$ diff -u lista.txt lista-pop.txt > lista.patch  
  
skd_s2@LAB318-2 UCRT64 ~  
$ dos2unix lista.txt  
dos2unix: converting file lista.txt to Unix format...  
  
skd_s2@LAB318-2 UCRT64 ~  
$ dos2unix lista-pop.txt  
dos2unix: converting file lista-pop.txt to Unix format...  
  
skd_s2@LAB318-2 UCRT64 ~  
$ patch lista.txt < lista.patch  
  
skd_s2@LAB318-2 UCRT64 ~  
$ md5sum lista-pop.txt lista.txt  
683c1c85343c7337adfb13acb7598237 *lista-pop.txt  
683c1c85343c7337adfb13acb7598237 *lista.txt
```

## Zadanie 5.1 „Z CSV do SQL”

Wykonanie:

Tail steps-2sql.csv | (komenda tail wyświetla ostatnie 10 linijek pliku steps-2sql.csv).

Awk -F „;” ‘{print „INSERT INTO stepsData(time,intensity,steps) Values (“\$1”,“\$2”,“\$3”)”}’ <steps-2sql.csv | (Awk -F oddziela kolumny separatorom ';' w pliku steps-2sql.csv oraz na początku każdej lini dopisuje „INSERT INTO stepsData(time,intensity,steps) Values (“\$1”,“\$2”,“\$3”)” gdzie \$1, \$2, \$3 to dane kolumny z poprzedniego pliku).

```
skd_s2@LAB318-2 UCRT64 ~  
$ tail steps-2sql.csv |awk -F ";" '{print "INSERT INTO stepsData(time,intensity,steps) Values (" $1 "," $2 "," $3 ")"}' <steps-2sql.csv  
INSERT INTO stepsData(time,intensity,steps) Values (time,intensity,steps)  
INSERT INTO stepsData(time,intensity,steps) Values (1562001120,19,0)  
INSERT INTO stepsData(time,intensity,steps) Values (1562001180,23,0)  
INSERT INTO stepsData(time,intensity,steps) Values (1562001240,13,0)  
INSERT INTO stepsData(time,intensity,steps) Values (1562004900,0,0)  
INSERT INTO stepsData(time,intensity,steps) Values (1562004960,53,26)  
INSERT INTO stepsData(time,intensity,steps) Values (1562005020,57,15)  
INSERT INTO stepsData(time,intensity,steps) Values (1562005080,22,0)  
INSERT INTO stepsData(time,intensity,steps) Values (1562005140,44,0)  
INSERT INTO stepsData(time,intensity,steps) Values (1562005200,30,0)  
INSERT INTO stepsData(time,intensity,steps) Values (1562005260,41,0)  
INSERT INTO stepsData(time,intensity,steps) Values (1562005320,9,0)  
INSERT INTO stepsData(time,intensity,steps) Values (1562005380,41,0)  
INSERT INTO stepsData(time,intensity,steps) Values (1562005440,72,24)
```

## Zadanie 5.2”Z SQL do CSV”

Wykonanie:

echo time; intensity; steps | (wypisuje nagłówki).

cat steps-2csv.sql | (przekazuje nagłówki do pliku steps-2csv.sql).

sed -e 's/INSERT INTO stepsData (dateTime, steps, synced) VALUES (//g' | (Usuwa początek [INSERT INTO stepsData (dateTime, steps, synced) VALUES] zostawiając tylko dane z wartości).

sed -e 's/)//g' | (Usuwa nawiasy zamykające ")").

sed -e 's/;/;/g' | (Usuwa wszystkie średniki ";").

sed -e 's/,/;/g' | (zamienia przecinki na średniki ";").

sed -e 's/ //g' | (Usuwa spacje w pliku).

```
1673038800000;7796;0  
1673039400000;8002;0  
1673040000000;8105;0  
1673041200000;8136;0  
1673041800000;8202;0  
1673042400000;8354;0  
1673043000000;8364;0  
1673043600000;8448;0  
Piorun@DESKTOP-RDIO48F UCRT64 ~  
$ echo time; intensity; steps |cat steps-2csv.sql | sed -e 's/INSERT INTO stepsData (dateTime, steps, synced) VALUES (//g' | sed -e 's/)//g' | sed -e 's/;/;/g' | sed -e 's/,/;/g' | sed -e 's/ //g'
```

## Zadanie 6 „Marudny tłumacz”

Wykonanie:

gsub - funkcja AWK do globalnego zmieniania tekstu

gsub(/^[ \t]+/, "", \$0); | (usuwa spacje i tabulatory w każdej linii, usuwa "" puste znaki oraz usuwa, globalnie dopasowuje spacje i tabulatory na początku linii ).

if (\$0 ~ /^[^".\*":.\*",?\$/ ) | (sprawdza czy jest para wartości w pliku Json5 gdzie [ ^" - zaczyna się od cudzysłowu. .\*^ - znajduje się dowolny tekst w cudzysłowie, : " - spacja po dwukropku, .\*" - dowolny tekst, ,?\$ - z opcjonalnym przecinkiem na końcu linii]).

print "/" / " \$0 RS \$0; | (JEŚLI linia spełnia warunek to wstawia komentarz z całą treścią linii, RS tworzy nową linię, i wkleja oryginalną linię)

else print \$0 | (Przekierowuje wynik kodu AWK do nowego pliku pl-7\_2.json5 , jeżeli warunek nie został spełniony).

```
$ cat pl-7_2.json5
{
  // "401.help": "You're not authorized to access this page. You can use the button below to get back to the home page.",
  "401.help": "You're not authorized to access this page. You can use the button below to get back to the home page.",
  // "401.link.home-page": "Take me to the home page",
  "401.link.home-page": "Take me to the home page",
  // "401.unauthorized": "unauthorized",
  "401.unauthorized": "unauthorized",
  // "403.help": "You don't have permission to access this page. You can use the button below to get back to the home page.",
  "403.help": "You don't have permission to access this page. You can use the button below to get back to the home page.",
  // "403.link.home-page": "Take me to the home page",
  "403.link.home-page": "Take me to the home page",
  // "403.forbidden": "Forbidden",
  "403.forbidden": "forbidden",
  // "500.page-internal-server-error": "Service Unavailable",
  "500.page-internal-server-error": "Service Unavailable",
  // "500.help": "The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.",
  "500.help": "The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.",
  // "500.link.home-page": "Take me to the home page",
  "500.link.home-page": "Take me to the home page",
  // "404.help": "We can't find the page you're looking for. The page may have been moved or deleted. You can use the button below to get back to the home page.",
  "404.help": "We can't find the page you're looking for. The page may have been moved or deleted. You can use the button below to get back to the home page."
}
```

cat en-7.2.json5 | (wczytuje plik json5).

awk -F ':' '{print \$1}' | (rozdziela linie na części dzielone dwukropkiem ":" oraz wpisuje zawartość \$1).

sed 's/"//g' | (szuka i kasuje wszystkie cudzysłowy).

sed 's/[{}]/g' | (wyszukiwuje i usuwa nawiasy klamrowe "{}").

awk NF | (filtruje wiersze i usuwa puste).

sed 's/ //g' | (szuka i usuwa wszystkie spacje).

> var.txt | (zapisuje wynik kodu jako plik Txt).

grep -f var.txt | (szuka ścieżki i używa pliku var.txt jako klucza).

-v en-7.4.json5 | (wypisuje tylko linie z pliku json5 które nie posiadają żadnego z kluczy z pliku var.txt).

awk NF | (filtruje wiersze i usuwa puste).

sed 's/"//g' | (szuka i kasuje wszystkie cudzysłowy).

sed 's/[{}]/g' | (wyszukiwuje i usuwa nawiasy klamrowe "{}").

```
skd_s2@LAB318-2 UCRT64 ~  
$ cat en-7.2.json5 | awk -F ':' '{print $1}' | sed 's/"//g' | sed 's/[{}]/g' | awk NF | sed 's/ //g' > var.txt|  
skd_s2@LAB318-2 UCRT64 ~  
$ grep -f var.txt -v en-7.4.json5 | awk NF | sed 's/"//g' | sed 's/[{}]/g'  
researcher.profile.public.visibility : PUBLIC,  
researcher.profile.status: Status:,  
researcherprofile.claim.not-authorized: You are not authorized to claim this item. For more details contact the  
researcherprofile.error.claim.body : An error occurred while claiming the profile, please try again later,  
researcherprofile.success.claim.body : Profile claimed with success,  
person.orcid.sync.setting: ORCID Synchronization settings,  
person.orcid.registry.queue: ORCID Registry Queue,  
person.orcid.registry.auth: ORCID Authorizations,  
home.recent-submissions.head: Recent Submissions,
```

## Zadanie 7 „Fotografik gamoń”

Wykonanie:

unzip \\*.zip | (rozpakuje plik o zip).

rm kopie-2.zip kopie-1.zip | (usuwa oba pliki o nazwach kopie-1 i kopie-2).

unzip \\*.zip | (rozpakuje plik zip).

rm \*.zip | (usuwa wszystkie pliki zip).

for f in \*.png; do magick "\$f" "\${f%.png}.jpg"; done | (wykonuje tą operację na każdym pliku .png i konwertuje go na plik jpg, \${f%.png}.jpg zamienia nazwę na taką z końcówką .jpg).

rm \*.png | (Usuwa wszystkie pliki png).

mkdir -p Przekonwertowane | (tworzy katalog "Przekonwertowane").

for f in \*.jpg; do magick "\$f" -resize x720 -units PixelsPerInch -density 96 "Przekonwertowane/\$f"; done |  
(zaznacza każdy plik jpg, ustawia wysokość każdego zdjęcia do 720 pikseli, zamienia jednostkę na "PixelsPerInch" i rozdzielczość na 96"dpi"

, zapisuje wykonaną pracę na nowym pliku w folderze "katalogu" Przekonwertowane).

## Zadanie 8 „Wszędzie to PDF-y”

Wykonanie:

montage \*.jpg | (tworzy kolaże do wszystkich plików jpg).

-tile 2x4 | (Tworzy i układa obrazy w tablice 2 kolumn na 4 wierszy).

-geometry 400x200+30+40 | (obrazki zostają przeskalowane do wielkości 400x200 pikseli, z odstępami 30 pikseli w poziomie i 40 w pionie między każdym obrazem).

-font C:/Windows/fonts/arial.ttf | (ustawia czcionkę etykiet na windowsowy arial).

-label '%f' | (pod każdym obrazem zostaje dodana etykieta).

-pointsize 14 | (nadaje rozmiar czcionki na 14).

-background white | (Zmienia kolor tła strony na biały).

-page A4 | (USTAWIA rozmiar strony dla ImageMagick na A4).

-density 150 | (nadaje rozdzielczość pdf-a na 150DPI).

output2.pdf | (nazwa pliku gdzie pdf ma zostać zapisany).

```
Piorun@DESKTOP-RDIO48F UCRT64 ~/Zadanie 7/Przekonwertowane
$ montage *.jpg -tile 2x4 -geometry 400x200+30+40 -font C:/Windows/fonts/arial.ttf -label '%f' -pointsize 14 -background white -page A4 -density 150 output2.pdf
```

## Zadanie 9 „Porządki w kopiach zapasowych”

Wykonanie:

- mkdir kopie | (tworzy katalog/folder o nazwie kopie).
- cd kopie | (wchodzi w katalog kopie).
- unzip \*.zip | (wypakowuje zawartość pliku zip w obecnym katalogu kopie).
- cd | (wraca do poprzedniego katalogu).
- bash SortowanieDrzewkowe.bat | (Otwiera i uruchamia skrypt batchowy zawarty w pliku .bat):
- for file in \*.zip; do | (wykonuje pętlę "do" dla każdego pliku .zip).
- y=\$(echo "\$file" | cut -d'-' -f1) | (znajduje i wyciąga rok z nazwy pliku).
- m=\$(echo "\$file" | cut -d'-' -f2) | (powtarzamy to samo tylko z miesiącem).
- mkdir -p "Rok: \$y/ Miesiac: \$m" | (tworzy katalog z rokiem i podfolder zawarty w tym katalogu o numerze miesiąca).
- mv "\$file" "Rok: \$y/ Miesiac: \$m/" | (przenosi plik do odpowiedniego katalogu bazując na roku i miesiącu w nazwie pliku).
- done | (zamyka pętlę "do").

```
Rok: 2023
├── Miesiac: 01
│   ├── 2023-01-02.zip
│   ├── 2023-01-05.zip
│   └── 2023-01-17.zip
├── Miesiac: 03
│   └── 2023-03-13.zip
├── Miesiac: 04
│   ├── 2023-04-05.zip
│   └── 2023-04-08.zip
├── Miesiac: 05
│   ├── 2023-05-03.zip
│   └── 2023-05-31.zip
├── Miesiac: 06
│   └── 2023-06-23.zip
├── Miesiac: 07
│   ├── 2023-07-02.zip
│   ├── 2023-07-03.zip
│   ├── 2023-07-10.zip
│   └── 2023-07-23.zip
├── Miesiac: 10
│   ├── 2023-10-16.zip
│   └── 2023-10-31.zip
├── Miesiac: 11
│   └── 2023-11-10.zip
├── Miesiac: 12
│   ├── 2023-12-07.zip
│   └── 2023-12-21.zip
└── Rok: 2024
    ├── Miesiac: 01
    │   ├── 2024-01-06.zip
    │   ├── 2024-01-17.zip
    │   └── 2024-01-29.zip
    ├── Miesiac: 03
    │   ├── 2024-03-10.zip
    │   ├── 2024-03-26.zip
    │   └── 2024-03-27.zip
    ├── Miesiac: 04
    │   ├── 2024-04-08.zip
    │   └── 2024-04-11.zip
    ├── Miesiac: 05
    │   ├── 2024-05-13.zip
    │   └── 2024-05-27.zip
    ├── Miesiac: 06
    │   └── 2024-06-21.zip
    ├── Miesiac: 07
    │   └── 2024-07-14.zip
    ├── Miesiac: 10
    │   ├── 2024-10-01.zip
    │   ├── 2024-10-03.zip
    │   └── 2024-10-09.zip
```



## Zadanie 10 „Galeria dla grafika”

### Wykonanie:

Wystarczy otworzyć plik .sh komenda

**-bash Zad10.sh**

```
-<!DOCTYPE html> > galeria.html | (Tworzy plik o nazwie galeria i nadaje mu typ html).
-<html lang="pl"> >> galeria.html | (ustawia język domyślny plikowi galeria na polski).
-<head> >> galeria.html... | (Wstawia do pliku zestaw polskich znaków, nadaje tytuł strony "Moja Galeria").
-</head> >> galeria.html | (zamyka sekcje head).
-<body> >> .. | (otwiera sekcje body).
-<h1>Moja Galeria Obrazów</h1> .. | (tworzy nagłówek h1 z tekstem "Moja Galeria obrazów").
-<p>Projekt by Piorun</p> | (tworzy paragraf p z tekstem "Projekt by Piorun").
-<div style="overflow:auto"> | (Tworzy sekcje która umożliwia automatyczne przemieszczanie się obrazków bez
zniekształcania układu strony w zależności od wielkości okna strony).
-for plik in *.png; do | (Tworzy pętlę "do" dla każdego pliku .png)
  -<div class="obrazek-blok">' >> galeria.html | (otwiera sekcje definiowania której nadaję klasę o nazwie "obrazek-
blok").
  -<div class="ramka">' >> galeria.html | (otwiera sekcje definiowania której nadaję klasę o nazwie "ramka").
  -<a target="_blank" href="\$plik\"> >> galeria.html | (tworzy sekcje hiperłącza w html gdzie po kliknięciu
obrazka otwiera go na nowej stronie).
  - >> galeria.html | (Ustawia tekst alternatywny na wypadek gdy obrazek się
nie ładuje).
  -</a> >> galeria.html | (zamyka sekcje hiperłącza).
  -<div class="podpis">$plik</div> >> galeria.html | (tworzy sekcje definiowania i łączy ją z nazwą "podpis").
  -</div>' >> galeria.html | (zamyka sekcje z "podpis").
  -</div>' >> galeria.html | (zamyka sekcje z "ramka").
-done | (zamyka petle).
-</div> | (zamyka sekcje "obrazek-blok").
-</body> | (zamyka sekcje body).
-</html> | (zamyka plik galeria.html)
```

```

echo '<!DOCTYPE html>' > galeria.html
echo '<html lang="pl">' >> galeria.html
echo '<head>' >> galeria.html
echo '  <meta charset="UTF-8">' >> galeria.html
echo '  <title>Moja Galeria</title>' >> galeria.html
echo '</head>' >> galeria.html
echo '<link rel="stylesheet" href="style.css">' >> galeria.html
echo '<body>' >> galeria.html
echo '  <h1>Moja Galeria Obrazów</h1>' >> galeria.html
echo '  <p>Projekt by Piorun</p>' >> galeria.html
echo '<div style="overflow:auto">' >> galeria.html

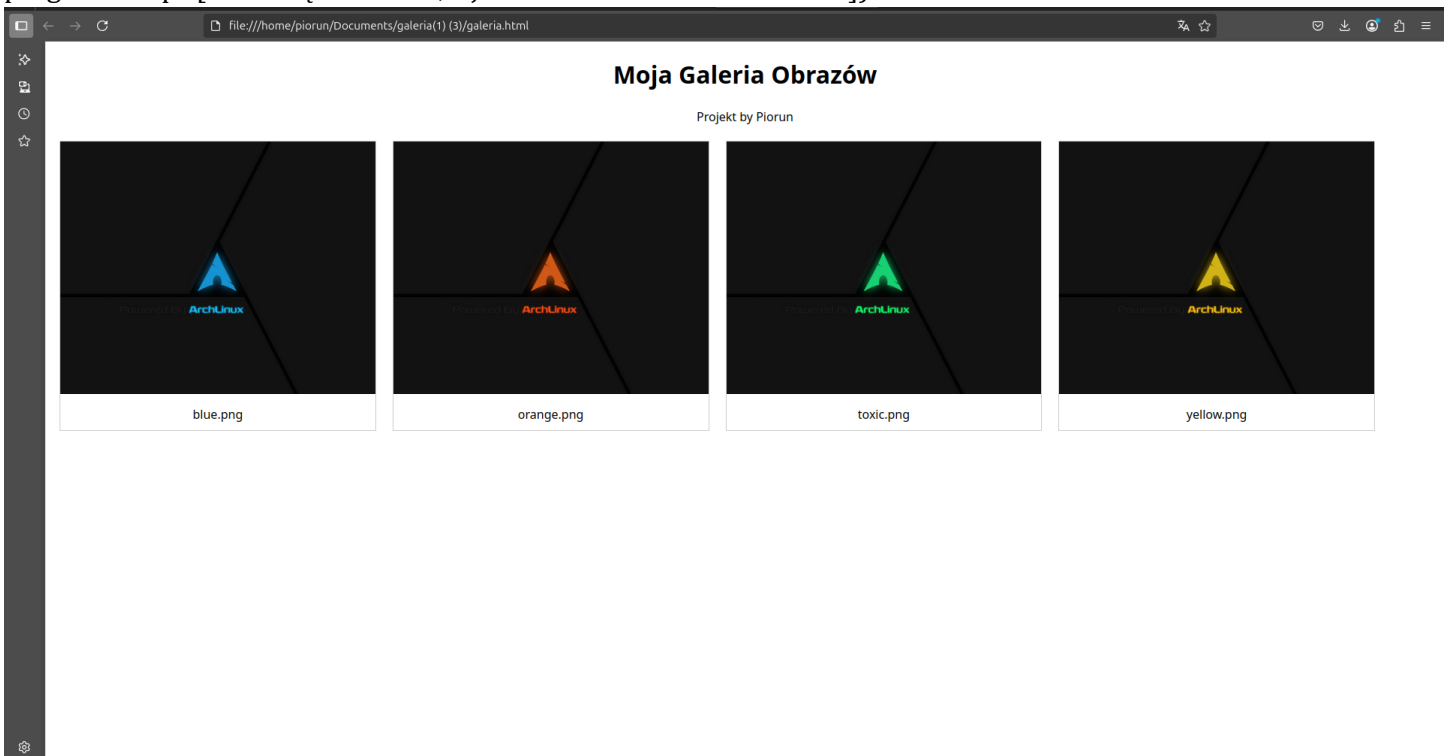
for plik in *.png; do
  echo '    <div class="obrazek-blok">' >> galeria.html
  echo '      <div class="ramka">' >> galeria.html
  echo '        <a target="_blank" href="$plik">' >> galeria.html
  echo '          ' >> galeria.html
  echo '        </a>' >> galeria.html
  echo '        <div class="podpis">$plik</div>' >> galeria.html
  echo '      </div>' >> galeria.html
done

echo '</div>' >> galeria.html
echo '</body>' >> galeria.html
echo '</html>' >> galeria.html

```

Na końcu recznie zostały dodane style, aby odzwierciedlić przykład na stronie

style.css (plik style zawiera kod css używany do formatowania pliku. Przypisuje nazwę .obrazek-blok[odległość od ramki 6px, przyrównywanie do lewej, z szerokością 24%], .ramka[pełną obramówkę o grubości 1px, z marginesem 5px], .ramka img[szerokość 100%, z automatyczną wielkością w zależności od wielkości okna strony], .podpis[odległość od ramki 10px, tekst na środku, z czcionką sans.serif] oraz każdemu nagłówkowi "h1" oraz pragrafowi "p" [czcionkę sans.serif, z justowaniem tekstu na środek]).



```
* { box-sizing: border-box; }
.obrazek-blok {
  padding: 0 6px;
  float: left;
  width: 24%;
}
.ramka {
  border: 1px solid #ccc;
  margin: 5px;
}
.ramka img {
  width: 100%;
  height: auto;
}
.podpis {
  padding: 10px;
  text-align: center;
  font-family: sans-serif;
}
h1, p {
  font-family: sans-serif;
  text-align: center;
}
```