

# Języki Skryptowe

dokumentacja projektu Rozwiązanie funkcji kwadratowej, w ciele liczb zespolonych

Jakub Stachurski, grupa 2 sekcja 3

3 stycznia 2024

# Część I

## Opis programu

Skrypt batch'owy jest podzielony na 3 części:

### 1

Uruchomienie dwóch skryptów napisanych w języku Python. Pierwszy czytuje wartości z pliku "wejsciowe.txt". A,B,C dla funkcji kwadratowej oraz wartości x, dla której ma zostać obliczony wynik. Wyniki zapisuje do pliku "wyjsciowe.txt". Drugi skrypt obsługuje tworzenie raportów, zapisując wyniki z pliku "wyjsciowe.txt" w formie tabelki do pliku "raport-(aktualna data).html"

### 2

Wyświetlenie informacji o programie

### 3

Stworzenie backupu raportu do pliku "backup-raport-(aktualna data).html" przy wykorzystaniu drugiego skryptu Python.

## Instrukcja obsługi

Program należy uruchomić poprzez plik "Projekt - Batch.bat". Dane, dla których obliczane są wartości, znajdują się w pliku "wejsciowe.txt". Pierwsze 3 wartości to kolejne zmienne A,B,C. Każda kolejna jest wartością x, tych zmiennych może być nielimitowana ilość.

### 0.1 Instrukcja wdrożenia

#### Dodatkowe informacje

Skrypt wymaga istnienia pliku "wejsciowe.txt" z minimum czterema wartościami (A,B,C,X), każda zmienna napisana w nowej linii.

Pierwszy skrypt obsługuje czytanie z pliku "wejsciowe.txt", zmianę wartości na zmiennoprzecinkowe, wykonanie obliczenia, a na końcu zapisanie wyniku wraz z punktem X, dla którego został obliczony, do pliku "wyjsciowe.txt".

Drugi skrypt odczytuje wartości z pliku "wyjsciowe.txt" i przepisuje je do pliku raport.html w formie tabeli, wraz z funkcją  $f(X)$ , zapisaną na początku raportu.

Trzeci skrypt służy do obsługi całości, w nim następuje wybór, co chcemy wykonać, dodatkowo do wykonania backupu raportu potrzebny jest raport.

## Część II

### Opis działania

Cały program opiera się o poniżej opisany wzór matematyczny

$$f(X) = AX^2 + BX + C, \quad (1)$$

gdzie  $A$ ,  $B$ ,  $C$  to współczynniki, a  $X$  to punkt, dla którego jest obliczana funkcja.

### Implementacja systemu

Pierwszy plik Python "funkcja.py" pobiera wartości z pliku "wejsciove.txt", zamienia je na liczby zmiennoprzecinkowe, oblicza wartość funkcji w punkcie  $X$  a następnie zapisuje wszystkie wyniki do nowego pliku "wyjsciove.txt".

Drugi plik Python "uRaport.py" pobiera linie z pliku "wyjsciove.txt" i przepisuje je do nowego pliku "raport.html" w formie tabeli. Dodatkowo na początku raportu pokazuje funkcję  $f(X)$ . Trzeci plik "Projekt - Batch.bat" zajmuje się obsługą całości. On odpowiada za uruchomienie powyższych plików, czy utworzenie backupu raportu.

### Plik funkcja.py

Ta część kodu odpowiada za odczytanie poszczególnych linii z pliku wejściowego, czyli wartości  $A, B, C$  oraz  $X$ . Następnie wpisuje je do tabeli

```
1  try:
2      with open(nazwa_pliku, 'r') as plik:
3          linie = plik.readlines()
4          return [linia.strip() for linia in linie]
5  except FileNotFoundError:
6      print(f"Nie mozna odnalezc pliku o nazwie '{nazwa_pliku}'")
7      return None
```

---

Obliczanie wartości funkcji

```
1  def oblicz_funkcje_kwadratowa(a, b, c, x):
2      return a * x**2 + b * x + c
```

---

Kod obsługuje zapis do pliku. Najpierw zapisuje funkcję  $f(x)$ , a następnie punkt  $x$  wraz z odpowiadającym mu wynikiem.

```
1  with open(nazwa_pliku, 'w') as plik_txt:
2      plik_txt.write(f"Funkcja kwadratowa: f(x) = {a}x^2 + {b}x + {c}\n")
3      for argument, wynik in wyniki:
4          plik_txt.write(f"{argument}: {wynik}\n")
```

---

W tej części program sprawdza czy została wprowadzona odpowiednia ilość zmiennych. Następnie konwertuje wartości i pokazuje funkcję  $f(X)$ . Następnie tworzy tablicę wynik oraz argumenty, którą wypełnia od 3 miejsca tablicy dane. W kolejnym kroku wywołuje pętlę która wykona się tyle razy, ile argumentów zostało przekazanych. W niej uruchamia formułę

obliczającą funkcję kwadratową, a wynik wpisuje do tablicy wyniki. Na końcu uruchamia funkcję wpisującą wyniki do pliku "wyjsciowe.txt".

```
1  if dane is not None and len(dane) >= 4:
2      try:
3          # Konwertuj współczynniki na liczby zmiennoprzecinkowe
4          a = float(dane[0])
5          b = float(dane[1])
6          c = float(dane[2])
7
8          print(f"Funkcja kwadratowa:  $f(x) = \{a\}x^2 + \{b\}x + \{c\}$ ")
9
10         # Wczytaj argumenty
11         argumenty = [complex(arg) for arg in dane[3:]]
12
13         wyniki = []
14
15         # Oblicz wartości funkcji dla argumentów
16         for argument in argumenty:
17             wynik = oblicz_funkcje_kwadratowa(a, b, c, argument)
18             print(f"Wartosc funkcji dla {argument}: {wynik}")
19
20             wyniki.append((argument, wynik))
21
22         zapisz_do_txt(wyniki, "wyjsciowe.txt", a, b, c)
23         print(f"Wyniki zostały zapisane do pliku wyjsciowe.txt")
24
25     except ValueError:
26         print("Bład konwersji współczynników na liczby")
```

---

## Plik uRaport.py

Ta część pliku odpowiada za sczytywanie wartości z pliku tekstowego i wpisywania go do tablicy

```
1  with open(nazwa_pliku, 'w') as plik_txt:
2      plik_txt.write(f"Funkcja kwadratowa:  $f(x) = \{a\}x^2 + \{b\}x + \{c\}$ \n")
3
4      for argument, wynik in wyniki:
5          plik_txt.write(f"{argument}: {wynik}\n")
```

---

Ta część pliku odpowiada za sczytywanie wartości z pliku tekstowego i wpisywania go do tablicy

```
1  try:
2      with open(nazwa_pliku, 'r') as plik:
3          linie = plik.readlines()
4          return [linia.strip() for linia in linie]
5  except FileNotFoundError:
6      print(f"Nie można odnaleźć pliku o nazwie '{nazwa_pliku}'")
7      return None
```

---

Kod, którego zadaniem jest zapisanie wyników do pliku o rozszerzeniu html

```

1  with open(nazwa_pliku, 'w') as plik_html:
2      plik_html.write("<html>\n")
3      plik_html.write("<head><title>"+nazwa_pliku+"</title></head>\n")
4      plik_html.write("<body>\n")
5      plik_html.write("<h1>"+nazwa_pliku+"</h1>\n")
6
7      # Dodaj funkcje kwadratowa do raportu
8      plik_html.write(f"<p>{funkcja_kwadratowa}</p>\n")
9
10     plik_html.write("<table border='1'>\n")
11     plik_html.write("<tr><th>Argument</th><th>Wynik</th></tr>\n"
12                     )
13
14     for wynik in wyniki:
15         argument, wartosc = wynik.split(": ")
16         plik_html.write(f"<tr><td>{argument}</td><td>{wartosc}</td></tr>\n")
17
18     plik_html.write("</table>\n")
19     plik_html.write("</body>\n")
20     plik_html.write("</html>\n")

```

Główna część odpowiadająca za obsługę całego skryptu. Na samym początku sprawdza ile wartości znajduje się w pliku wyjściowym. Następnie wywołuję jedną z powyższych funkcji. Jeśli w tablicy dane-obliczen znajduje się tam więcej niż 2 rekordy przystępuje do wywołania funkcji zapisz-do-html.

```

1  REM Uzyskanie aktualnej daty i godziny
2      for /f "delims=" %%a in ('wmic OS Get localdatetime ^| find ".")
3          do set datetime=%%a
4
5      set "year=%datetime:~0,4%"
6      set "month=%datetime:~4,2%"
7      set "day=%datetime:~6,2%"
8
9      for /f "tokens=1-2 delims=: " %%a in ('time /t') do set "godzina=%%a%%b"
10
11     REM Utworzenie nazwy raportu z daty i godziny
12     set "nazwa_backup=backup-raport-%year%-%month%-%day%-%godzina%"
13     set "nazwa_raport=raport-%year%-%month%-%day%-%godzina%"

```

## Plik Projekt - Batch.bat

Kod zczytuje aktualną datę i godzinę a następnie konwertuje w taki sposób by można było go przypisać do pliku raport i backup-raport

```

1  if len(sys.argv) != 2:
2      sys.exit(1)
3
4      nazwa_raportu = sys.argv[1]
5      dane_obliczen = czytaj_dane_z_pliku("wyjsciowe.txt")

```

```

6
7     if dane_obliczen is not None and len(dane_obliczen) >= 2:
8         try:
9             funkcja_kwadratowa = dane_obliczen[0]
10            wyniki_obliczen = dane_obliczen[1:]
11
12            zapisz_do_html(funkcja_kwadratowa, wyniki_obliczen,
13                           nazwa_raportu+".html")
14            print(f"Raport zostal zapisany do pliku {nazwa_raportu}.
15                  html")
16
17        except ValueError:
18            print("Blad konwersji danych na odpowiedni format")

```

---

ta część odpowiada za pełną obsługę pliku, w zależności od wybranej opcji uruchomi odpowiednie skrypty .py, wyświetli informacje o całym skrypcie, zakończy działanie programu

```

1  :menu
2      cls
3      echo MENU:
4      echo 1. Start - Uruchomienie pierwszego i drugiego programu
5      echo 2. Informacje - Wyszwietlenie informacji o skrypcie
6      echo 3. Backup - Stworzenie i wyswietlenie backupu z raportu
7      echo 4. Zakoncz - Zakoncz prace skryptu
8
9      set /p wybor="Wybierz opcje: "
10
11     if "%wybor%"=="1" goto start
12     if "%wybor%"=="2" goto informacje
13     if "%wybor%"=="3" goto backup
14     if "%wybor%"=="4" goto koniec
15
16     :start
17     echo Uruchamianie pierwszego programu...
18     python funkcja.py
19
20     echo.
21     echo Uruchamianie drugiego programu...
22     python uRaport.py %nazwa_raport%
23
24     pause
25     goto menu
26
27     :informacje
28     echo Informacje o skrypcie:
29     echo - Skrypt ma za zadanie obliczyc wartosci funkcji
30           kwadratowej, zadanej w pliku txt,
31     echo utworzenie pliku wyjsciowego z wynikami, a nastepnie
32     echo stowrzenie raportu w pliku html.
33     echo - Mozliwosc stworzenia bakupu wygenerowanego wczesniej
34           raportu.
35     pause
36     goto menu
37
38     :backup

```

```
37         echo Tworzenie i wyswietlanie backupu z raportu w formacie HTML
38         ...
39         python uRaport.py %nazwa_backup%
40         start %nazwa_backup%.html
41         pause
42         goto menu
43     :koniec
44     echo Zakonczono prace skryptu.
45     exit /b
```

---

## Testy

Dla przykładowej funkcji

$$f(x)=2x^2 + 3x + 1$$

$$x=5+0j: f(x)=66+0j$$

$$x=0+4j: f(x)=-31+12j$$

$$x=0+3j: f(x)=-17+9j$$

$$x=5+1j: f(x)=64+23j$$

# Pełen kod aplikacji

Pierwszy skrypt Python: funkcja.py

```
1  -*- coding: utf-8 -*-
2
3  import cmath
4
5  def czytaj_dane_z_pliku(nazwa_pliku):
6      try:
7          with open(nazwa_pliku, 'r') as plik:
8              linie = plik.readlines()
9              return [linia.strip() for linia in linie]
10     except FileNotFoundError:
11         print(f"Nie można odnaleźć pliku o nazwie '{nazwa_pliku}'")
12         return None
13
14 def oblicz_funkcje_kwadratowa(a, b, c, x):
15     return a * x**2 + b * x + c
16
17 def zapisz_do_txt(wyniki, nazwa_pliku, a, b, c):
18     with open(nazwa_pliku, 'w') as plik_txt:
19         plik_txt.write(f"Funkcja kwadratowa: f(x) = {a}x^2 + {b}x + {c}\n")
20         for argument, wynik in wyniki:
21             plik_txt.write(f"{argument}: {wynik}\n")
22
23 def main_obliczenia():
24
25     # Wczytaj współczynniki i argumenty z pliku
26     dane = czytaj_dane_z_pliku("wejscowe.txt")
27
28     if dane is not None and len(dane) >= 4:
29         try:
30             # Konwertuj współczynniki na liczby zmiennoprzecinkowe
31             a = float(dane[0])
32             b = float(dane[1])
33             c = float(dane[2])
34
35             print(f"Funkcja kwadratowa: f(x) = {a}x^2 + {b}x + {c}")
36
37             # Wczytaj argumenty
38             argumenty = [complex(arg) for arg in dane[3:]]
39
40             wyniki = []
41
42             # Oblicz wartości funkcji dla argumentów
43             for argument in argumenty:
44                 wynik = oblicz_funkcje_kwadratowa(a, b, c, argument)
45                 print(f"Wartość funkcji dla {argument}: {wynik}")
46
47                 wyniki.append((argument, wynik))
48
49             zapisz_do_txt(wyniki, "wyjscowe.txt", a, b, c)
50             print(f"Wyniki zostały zapisane do pliku wyjscowe.txt")
```



```

51
52         except ValueError:
53             print("Bład konwersji współczynników na liczby")
54
55 if __name__ == "__main__":
56     main_obliczenia()

```

---

Drugi skrypt Python: uRaport.py

```

1  -*- coding: utf-8 -*-
2  import datetime
3  import sys
4
5  def czytaj_dane_z_pliku(nazwa_pliku):
6      try:
7          with open(nazwa_pliku, 'r') as plik:
8              linie = plik.readlines()
9              return [linia.strip() for linia in linie]
10     except FileNotFoundError:
11         print(f"Nie można odnalezc pliku o nazwie '{nazwa_pliku}'")
12         return None
13
14 def zapisz_do_html(funkcja_kwadratowa, wyniki, nazwa_pliku):
15     with open(nazwa_pliku, 'w') as plik_html:
16         plik_html.write("<html>\n")
17         plik_html.write("<head><title>"+nazwa_pliku+"</title></head>\n")
18         plik_html.write("<body>\n")
19         plik_html.write("<h1>"+nazwa_pliku+"</h1>\n")
20
21         # Dodaj funkcje kwadratowa do raportu
22         plik_html.write(f"<p>{funkcja_kwadratowa}</p>\n")
23
24         plik_html.write("<table border='1'>\n")
25         plik_html.write("<tr><th>Argument</th><th>Wynik</th></tr>\n")
26
27         for wynik in wyniki:
28             argument, wartosc = wynik.split(": ")
29             plik_html.write(f"<tr><td>{argument}</td><td>{wartosc}</td>
30                             ></tr>\n")
31
32         plik_html.write("</table>\n")
33         plik_html.write("</body>\n")
34         plik_html.write("</html>\n")
35
36 if __name__ == "__main__":
37     # Wczytaj funkcje kwadratowa i wartosci z pliku wyjsciowego
38     if len(sys.argv) != 2:
39         sys.exit(1)
40
41     nazwa_raportu = sys.argv[1]
42     dane_obliczen = czytaj_dane_z_pliku("wyjsciowe.txt")
43
44     if dane_obliczen is not None and len(dane_obliczen) >= 2:
45         try:
46             funkcja_kwadratowa = dane_obliczen[0]

```

```

46         wyniki_obliczen = dane_obliczen[1:]
47
48         zapisz_do_html(funkcja_kwadratowa, wyniki_obliczen,
49                       nazwa_raportu+".html")
50         print(f"Raport zostal zapisany do pliku {nazwa_raportu}.html")
51
52     except ValueError:
53         print("Blad konwersji danych na odpowiedni format")

```

---

## Skrypt Projekt - Batch.bat

```

1 @echo off
2 setlocal enabledelayedexpansion
3
4 REM Uzyskanie aktualnej daty i godziny
5 for /f "delims=" %%a in ('wmic OS Get localdatetime ^| find "."') do set
6     datetime=%%a
7
8 set "year=%datetime:~0,4%"
9 set "month=%datetime:~4,2%"
10 set "day=%datetime:~6,2%"
11
12 for /f "tokens=1-2 delims=: " %%a in ('time /t') do set "godzina=%%a%%b"
13
14 REM Utworzenie nazwy raportu z daty i godziny
15 set "nazwa_backup=backup-raport-%year%-%month%-%day%-%godzina%"
16 set "nazwa_raport=raport-%year%-%month%-%day%-%godzina%"
17
18 :menu
19 cls
20 echo MENU:
21 echo 1. Start - Uruchomienie pierwszego i drugiego programu
22 echo 2. Informacje - Wyświetlenie informacji o skrypcie
23 echo 3. Backup - Stworzenie i wyświetlenie backupu z raportu
24 echo 4. Zakoncz - Zakoncz prace skryptu
25
26 set /p wybor="Wybierz opcje: "
27
28 if "%wybor%"=="1" goto start
29 if "%wybor%"=="2" goto informacje
30 if "%wybor%"=="3" goto backup
31 if "%wybor%"=="4" goto koniec
32
33 :start
34 echo Uruchamianie pierwszego programu...
35 python funkcja.py
36
37 echo .
38 echo Uruchamianie drugiego programu...
39 python uRaport.py %nazwa_raport%
40
41 pause
42 goto menu

```

```
43 :informacje
44 echo Informacje o skrypcie:
45 echo - Skrypt ma za zadanie obliczyc wartosci funkcji kwadratowej,
      zadanej w pliku txt,
46 echo utworzenie pliku wyjsciowego z wynikami, a nastepnie
47 echo stowrzenie raportu w pliku html.
48 echo - Mozliwosc stworzenia bakupu wygenerowanego wczesniej raportu.
49 pause
50 goto menu
51
52 :backup
53 echo Tworzenie i wyswietlanie backupu z raportu w formacie HTML...
54 python uRaport.py %nazwa_backup%
55 start %nazwa_backup%.html
56 pause
57 goto menu
58
59 :koniec
60 echo Zakonczono prace skryptu.
61 exit /b
```

---