**AKADEMIA GÓRNICZO-HUTNICZA**

**im. Stanisława Staszica w Krakowie**

**WYDZIAŁ ZARZĄDZANIA**

**STUDIA STACJONARNE II stopnia**

**KIERUNEK: Informatyka i Ekonometria**

**PRZEDMIOT: Przetwarzanie i analiza danych w języku Python**

**PROWADZĄCY: prof. dr hab. inż. Oleksandr Petrov**

# Sprawozdanie 6

Agnieszka Pytel

**Kraków, 2018/2019**

# 1) Zadanie 14.1.

**Napisz program ‘parking2.py’ różniący się od programu ‘parking.py’ tym, że klienci parkingu mogą posiadać miesięczne karty abonentowi (dodać funkcję sprzedaży). Wjazdy i wyjazdy samochodów takich klientów są rejestrowane, jednak przy wyjeździe z parkingu opłata nie jest pobierana, o ile nie abonament nie skończył się (wyświetla się tylko informacja o liczbie pozostałych dni). Jeżeli abonament danego auta skończył się, przy jego najbliższym wjeździe lub wyjeździe użytkownik jest o tym informowany i może wykupić nowy abonament lub z niego zrezygnować (i w konsekwencji wnieść standardową opłatę).**

In [48]:
```
import shelve, sys
from time import *
from datetime import datetime
```

In [49]:
```
#inicjalizacja bazy danych
def init():
    global baza, parking, abonamenty, stawka, okres
    try:
        baza = shelve.open ('baza_parkingowa', writeback = True) # otwarcie
    except:
        print("Błąd krytyczny! Baza danych nie została otwarta!")
        sys.exit(0) # wyjście z programu
    print("Inicjalizacja udana. Baza danych została otwarta.")
    if '_stawka' in baza.keys(): # czy już istnieje?
        (stawka,okres)=baza['_stawka'] # tak - kopiujemy stawki
    else:
        (stawka,okres)=(0.0,1) # nie -
        zmiana_stawki() # wczytujemy od użytkownika
    if 'parking' not in baza.keys():
        baza['parking'] = {}
    parking = baza['parking']
    if 'abonamenty' not in baza.keys():
        baza['abonamenty'] = {}
    abonamenty = baza['abonamenty']
    aktualizuj_abonamenty()
```

In [50]:
```
# menu główne programu
def menu():
    while True:
        print ()
        print ('-'*70)
        print ('PARKING'.center(70))
        print ('-'*70 )
        print ('[W] Wjazd [E] Wyjazd [P] Pojazdy [S] Stawka [A] Abonament [K] Koniec'.center(70))
        print ('-'*70)
        w=input()[0].upper() # pierwszy znak (duza litera)
        if w in 'WEPSAK': # znany?
            return w # tak - zwracamy go
        print ('Nieznane polecenie -')
```

In [51]:
```
# realizacja wyboru użytkownika
def wybor():
    while True:
        w=menu()
        if w=='K':
            break
        elif w=='S':
            zmiana_stawki()
        elif w=='P':
            pojazdy()
        elif w=='E':
            wyjazd()
        elif w=='W':
            wjazd()
        elif w=='A':
            abonament()
```

In [52]:
```
# zmiana opłat
def zmiana_stawki():
    global baza,stawka,okres # zmienne globalne
    print("\nZmiana wysokości opłat\n")
    print("Bieżąca stawka wynosi %.2f zł za %i minut(y)\n" % (stawka,okres))
    try:
        s=float(input("Podaj nową wysokość opłat: "))
        o=int(input("Podaj nowy czas naliczania w minutach: "))
    except:
        print("Błąd wprowadzania danych! Stawka nie została zmieniona!")
        return
    try:
        baza['_stawka']=(s,o) # zapisujemy w bazie
    except:
        print("Błąd zapisu danych! Stawka nie została zmieniona!")
    else:
        stawka=s # kopiujemy do
        okres=o # zmiennych globalnych
```

In [53]:
```
# lista pojazdów na parkingu
def pojazdy():
    global parking
    print ()
    print ('Lista pojazdów na parkingu'.center(33))
    print ('-'*33)
    print ('|'+'Nr rej.'.center(10)+'|'+'Godz. parkowania'.center(20)+'|' )
    print( '-'*33 )
    for rej,godz in parking.items():
        if rej!='_stawka':
            print ("|%9s |" % rej,strftime("%H:%M (%Y-%m-%d)",godz),'|' )
    print( '-'*33)
```

In [54]:
```
# rejestracja wjazdu pojazdu
def wjazd():
    global parking, abonamenty
    godz=localtime()
    print ('Wjazd pojazdu - godzina',strftime("%H:%M (%Y-%m-%d)",godz))
    rej=input('Podaj numer rejestracyjny pojazdu: ')[:9]
    if rej=='_stawka':
        return # zabezpieczenie
    if rej not in parking.keys(): # nie jest zaparkowany?
        parking[rej]=godz
        print ("Wprowadzono." )
    else:
        print ("Błąd! Taki pojazd już jest na parkingu!")
```

In [59]:
```
# rejestracja wyjazdu pojazdu
def wyjazd():
    global parking, abonament, stawka, okres
    print ('Wyjazd pojazdu - godzina',strftime("%H:%M (%Y-%m-%d)"))
    rej=input('Podaj numer rejestracyjny pojazdu: ')
    if rej in parking.keys(): # czy taki był zaparkowany?
        godz=parking[rej]
        print ("Godzina wjazdu:",strftime("%H:%M (%Y-%m-%d)",godz))
        if rej in abonamenty.keys():
            pozostale_dni = 30 - (datetime.now() - abonamenty[rej][0]).days
            if pozostale_dni:
                print("Pozostało",pozostale_dni,"dni abonamentu. Dziękujemy.")
            else:
                nowy_abonament = input('Twój abonament się skończył. Czy chcesz wykupić nowy? (T/N): ')
                del abonamenty[rej]
                if nowy_abonament == 'T': abonament()
                elif nowy_abonament == 'N':
                    print("Zrezygnowano z abonamentu. Naliczona zostanie standardowa opłata.")
                    minuty=int(mktime(localtime())-mktime(godz))/60
                    jednostki=(minuty+okres-1)/okres # naliczamy za rozpoczeta
                    print ("\nDo zapłaty: %.2f zł" % (jednostki*stawka))
        else:
            minuty=int(mktime(localtime())-mktime(godz))/60
            jednostki=(minuty+okres-1)/okres # naliczamy za rozpoczeta
            print ("\nDo zapłaty: %.2f zł" % (jednostki*stawka))
        del parking[rej] # usuwamy wpis
    else:
        print ("Błąd! Takiego pojazdu nie ma na parkingu!" )
```

In [56]:
```
# wykupienie abonamentu miesięcznego
def abonament():
    global abonamenty
    print ('Zakup abonamentu miesięcznego - godzina',strftime("%H:%M (%Y-%m-%d)"))
    rej=input('Podaj numer rejestracyjny pojazdu: ')
    data = datetime.now()
    if rej not in abonamenty.keys():
        abonamenty[rej] = [data, 1] # drugi argument wskazuje na to, czy abonament jest aktywny
        print ("Wykupiono abonament." )
    else:
        print ("Błąd! Ten pojazd już posiada abonament!")
```

In [57]:
```
# aktualizacja abonamentów
def aktualizuj_abonamenty():
    global abonamenty
    now = datetime.now()
    for abonament in abonamenty.items():
        if (now-abonament[0]).days > 30:
            abonament[1] = 0
            print("Abonament się skończył!")
```

In [58]:
```
# program główny
init() # otwarcie bazy
try:
     wybor() # interfejs użytkownika
except:
     print ("Wystąpił poważny błąd." )
baza.close() # zamknięcie bazy
```

```
Inicjalizacja udana. Baza danych została otwarta.

---------------------------------------------------------------------
                              PARKING
---------------------------------------------------------------------
 [W] Wjazd [E] Wyjazd [P] Pojazdy [S] Stawka [A] Abonament [K] Koniec
---------------------------------------------------------------------
A
Zakup abonamentu miesięcznego - godzina 17:47 (2019-01-14)
Podaj numer rejestracyjny pojazdu: KWAG717
Wykupiono abonament.

---------------------------------------------------------------------
                              PARKING
---------------------------------------------------------------------
 [W] Wjazd [E] Wyjazd [P] Pojazdy [S] Stawka [A] Abonament [K] Koniec
---------------------------------------------------------------------
W
Wjazd pojazdu - godzina 17:47 (2019-01-14)
Podaj numer rejestracyjny pojazdu: KWAG717
Wprowadzono.

---------------------------------------------------------------------
                              PARKING
---------------------------------------------------------------------
 [W] Wjazd [E] Wyjazd [P] Pojazdy [S] Stawka [A] Abonament [K] Koniec
---------------------------------------------------------------------
W
Wjazd pojazdu - godzina 17:47 (2019-01-14)
Podaj numer rejestracyjny pojazdu: KRAM345
Wprowadzono.

---------------------------------------------------------------------
                              PARKING
---------------------------------------------------------------------
P

    Lista pojazdów na parkingu
---------------------------------
| Nr rej.  |  Godz. parkowania  |
---------------------------------
|  KWAG717 | 17:47 (2019-01-14) |
|  KRAM345 | 17:47 (2019-01-14) |
---------------------------------

---------------------------------------------------------------------
                              PARKING
---------------------------------------------------------------------
 [W] Wjazd [E] Wyjazd [P] Pojazdy [S] Stawka [A] Abonament [K] Koniec
---------------------------------------------------------------------
W
Wjazd pojazdu - godzina 17:48 (2019-01-14)
Podaj numer rejestracyjny pojazdu: BIAF4567
Wprowadzono.
```

```
E
Wyjazd pojazdu - godzina 17:48 (2019-01-14)
Podaj numer rejestracyjny pojazdu: KWAG717
Godzina wjazdu: 17:47 (2019-01-14)
Pozostało 30 dni abonamentu. Dziękujemy.

----------------------------------------------------------------------
                              PARKING
----------------------------------------------------------------------
 [W] Wjazd [E] Wyjazd [P] Pojazdy [S] Stawka [A] Abonament [K] Koniec
----------------------------------------------------------------------
K
```

## 2) Zadanie 14.2.

**Napisz program 'narzedzia.py' służący do obsługi narzędziowni. Program ma umożliwiać wykonywanie następujących funkcji: dopisanie nowego narzędzia na listę, zmiana liczby sztuk narzędzia na liście (w tym możliwość kompletnego usunięcia narzędzia, gdy liczba sztuk spadnie do zera), wydanie narzędzia (zapamiętanie godziny i nazwiska pracownika biorącego narzędzie), zwrot narzędzia, lista wszystkich narzędzi, lista dostępnych narzędzi, lista wydanych narzędzi.**

(przyjęto założenie, że jeden pracownik może wypożyczyć tylko jedno narzędzie danego typu)

In [1]:
```
import shelve, sys
from time import *
from datetime import datetime
```

In [2]:
```
#inicjalizacja bazy danych
def init():
    global baza, wszystkie, wydane, dostepne
    try:
        baza = shelve.open ('baza_narzedziowa', writeback = True) # otwarcie
    except:
        print("Błąd krytyczny! Baza danych nie została otwarta!")
        sys.exit(0) # wyjście z programu
    print("Inicjalizacja udana. Baza danych została otwarta.")
    if 'wszystkie' not in baza.keys():
        baza['wszystkie'] = {}
    if 'wydane' not in baza.keys():
        baza['wydane'] = {}
    if 'dostepne' not in baza.keys():
        baza['dostepne'] = {}
    wszystkie = baza['wszystkie']
    wydane = baza['wydane']
    dostepne = baza['dostepne']
```

In [3]:
```
# menu główne programu
def menu():
    while True:
        print ('\n')
        print ('-'*100)
        print ('NARZĘDZIOWNIA'.center(100))
        print ('-'*100 )
        print ('[+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec'.center(1
00))
        print ('-'*100)
        w=input()[0].upper() # pierwszy znak (duza litera)
        if w in '+N-ZADWK': # znany?
            return w # tak - zwracamy go
        print ('Nieznane polecenie -')
```

In [4]:
```
# realizacja wyboru użytkownika
def wybor():
    while True:
        w=menu()
        if w=='K':
            break
        elif w=='+':
            dodaj()
        elif w=='N':
            zmien_liczbe()
        elif w=='-':
            wydaj()
        elif w=='Z':
            zwroc()
        elif w=='A':
            pokaz_wszystkie()
        elif w=='D':
            pokaz_dostepne()
        elif w=='W':
            pokaz_wydane()
```

In [5]:
```
# dodanie narzędzia do listy
def dodaj():
    global wszystkie, dostepne
    print("\nDodanie narzędzia do listy\n")
    try:
        nazwa=input("Podaj nazwę nowego narzędzia: ").lower()
        ilosc=int(input("Podaj liczbę sztuk nowego narzędzia: "))
    except:
        print("Błąd wprowadzania danych! Narzędzie nie zostało dodane!")
        return
    try:
        if nazwa not in wszystkie.keys():
            wszystkie[nazwa] = ilosc
            dostepne[nazwa] = ilosc
            print("Dodano narzędzie",nazwa,"w ilości",ilosc,"sztuk.")
        else:
            print("Takie narzędzie już jest w narzędziowni. Możesz zmienić liczbę sztuk w menu głównym.")
    except:
        print("Błąd zapisu danych! Narzędzie nie zostało dodane!")
```

In [6]:
```
# zmiana liczby sztuk wybranego narzędzia
def zmien_liczbe():
    global wszystkie, wydane, dostepne
    print("\nZmiana liczby sztuk narzędzia\n")
    try:
        nazwa=input("Podaj nazwę narzędzia: ").lower()
        ilosc=int(input("Podaj liczbę sztuk narzędzia: "))
    except:
        print("Błąd wprowadzania danych! Liczba sztuk nie została wczytana!")
        return
    try:
        if nazwa in wszystkie.keys():
            ile_wydanych = sum([1 for element in wydane.keys() if element == nazwa])
            if ilosc >= ile_wydanych:
                wszystkie[nazwa] = ilosc
                dostepne[nazwa] = ilosc - ile_wydanych
                if dostepne[nazwa] == 0:
                    usun = input("Aktualna liczba dostępnych sztuk tego narzędzia to 0. Czy chcesz usunąć je całkowicie z
bazy dostępnych narzędzi? [T/N]")
                    if usun == 'T':
                        del dostepne[nazwa]
                    elif usun == 'N':
                        print("Narzędzie nie zostało usunięte z listy dostępnych narzędzi.")
                    else:
                        print("Nieznane polecenie. Narzędzie nie zostało usunięte z listy.")
            else:
                print("Nie wprowadzono zmiany. Narzędzi musi być co najmniej tyle, ile wydanych sztuk!")
        else:
            print("Nie ma takiego narzędzia. Liczba sztuk nie została zmieniona!")
    except:
        print("Błąd zapisu danych! Liczba sztuk nie została zmieniona!")
```

In [7]:
```
# wydanie narzędzia
def wydaj():
    global wszystkie, wydane, dostepne
    print("\nWydanie narzędzia\n")
    try:
        nazwa=input("Podaj nazwę narzędzia: ").lower()
        nazwisko=input("Podaj imię i nazwisko wypożyczającego: ")
    except:
        print("Błąd wprowadzania danych! Dane nie zostały wczytane!")
        return
    try:
        godzina = datetime.now()
        if sum([1 for narz, attr in wydane.items() if narz == nazwa and attr['nazwisko'] == nazwisko]) == 0:
            if nazwa in dostepne.keys():
                wydane[nazwa] = {}
                wydane[nazwa]['nazwisko'] = nazwisko
                wydane[nazwa]['godzina'] = godzina
                dostepne[nazwa] -= 1
                if dostepne[nazwa] == 0:
                    usun = input("Aktualna liczba dostępnych sztuk tego narzędzia to 0. Czy chcesz usunąć je całkowicie z
bazy dostępnych narzędzi? [T/N]")
                    if usun == 'T':
                        del dostepne[nazwa]
                    elif usun == 'N':
                        print("Narzędzie nie zostało usunięte z listy dostępnych narzędzi.")
                    else:
                        print("Nieznane polecenie. Narzędzie nie zostało usunięte z listy.")
                print("Wydano narzędzie.")
            else:
                print("Narzędzie nie jest dostępne. Nie można go wydać!")
        else:
            print("Ten pracownik wypożyczył już",nazwa,'.')
    except:
        print("Błąd zapisu danych! Nie wydano narzędzia!")
```

In [8]:
```
# zwrot narzędzia
def zwroc():
    global wydane, dostepne
    print("\nZwrot narzędzia\n")
    try:
        nazwa=input("Podaj nazwę narzędzia: ").lower()
        nazwisko=input("Podaj imię i nazwisko osoby, która wypożyczyła narzędzie: ")
    except:
        print("Błąd wprowadzania danych! Dane nie zostały wczytane!")
        return
    try:
        if nazwa in wydane.keys():
            for narz, attr in wydane.items():
                if narz == nazwa and attr['nazwisko'] == nazwisko:
                    if nazwa in dostepne.keys():
                        dostepne[nazwa] += 1
                    else:
                        dostepne[nazwa] = 1
                    do_usuniecia = narz
                    print("Narzędzie zostało zwrócone, dziękujemy.")
                else:
                    print("To narzędzie wypożyczyła inna osoba. Zwrot nie jest możliwy.")
            del wydane[do_usuniecia]
        else:
            print("Nie wydano takiego narzędzia. Zwrot nie jest możliwy.")
    except:
        print("Błąd zapisu danych! Narzędzie nie zostało zwrócone!")
```

In [9]:
```
# lista wszystkich narzędzi
def pokaz_wszystkie():
    global wszystkie
    print ()
    print ('Lista wszystkich narzędzi'.center(38))
    print ('-'*38)
    print ('|'+'Nazwa'.center(20)+'|'+'Liczba sztuk'.center(15)+'|' )
    print( '-'*38 )
    if len(wszystkie):
        for narz, ilosc in wszystkie.items():
            print ("|%20s|%15s|" % (narz.center(20), str(ilosc).center(15)) )
            print( '-'*38)
    else:
        print('|'+"Brak narzędzi.".center(36)+'|')
        print( '-'*38)
```

In [11]:
```
# lista wydanych narzędzi
def pokaz_wydane():
    global wydane
    print ()
    print ('Lista wydanych narzędzi'.center(65))
    print ('-'*65)
    print ('|'+'Nazwa'.center(20)+'|'+'Imię i nazwisko'.center(20)+'|'+'Godzina'.center(21)+'|')
    print( '-'*65 )
    if len(wydane.keys()):
        for narz, attr in wydane.items():
            print ("|%20s|%20s|%21s|" % (narz.center(20), attr['nazwisko'].center(20), attr['godzina'].strftime("%H:%M (%
Y-%m-%d)").center(21)) )
            print( '-'*65)
    else:
        print("|"+'Brak narzędzi'.center(63)+"|")
        print( '-'*65)
```

In [12]:
```
# program główny
init() # otwarcie bazy
try:
     wybor() # interfejs użytkownika
except:
     print ("Wystąpił poważny błąd." )
baza.close() # zamknięcie bazy
```

```
Inicjalizacja udana. Baza danych została otwarta.


-----------------------------------------------------------------------------------------------------
                                          NARZĘDZIOWNIA
-----------------------------------------------------------------------------------------------------
  [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
-----------------------------------------------------------------------------------------------------
+

Dodanie narzędzia do listy

Podaj nazwę nowego narzędzia: grabie
Podaj liczbę sztuk nowego narzędzia: 2
Dodano narzędzie grabie w ilości 2 sztuk.


-----------------------------------------------------------------------------------------------------
                                          NARZĘDZIOWNIA
-----------------------------------------------------------------------------------------------------
  [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
-----------------------------------------------------------------------------------------------------
N

Zmiana liczby sztuk narzędzia

Podaj nazwę narzędzia: grabie
Podaj liczbę sztuk narzędzia: 3


-----------------------------------------------------------------------------------------------------
                                          NARZĘDZIOWNIA
-----------------------------------------------------------------------------------------------------
  [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
-----------------------------------------------------------------------------------------------------
-

Wydanie narzędzia

  Podaj nazwę narzędzia: grabie
  Podaj imię i nazwisko wypożyczającego: Agnieszka Pytel
  Wydano narzędzie.


  -----------------------------------------------------------------------------------------------------
                                            NARZĘDZIOWNIA
  -----------------------------------------------------------------------------------------------------
    [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
  -----------------------------------------------------------------------------------------------------
  -

  Wydanie narzędzia

  Podaj nazwę narzędzia: grabie
  Podaj imię i nazwisko wypożyczającego: Agnieszka Pytel
  Ten pracownik wypożyczył już grabie .


  -----------------------------------------------------------------------------------------------------
                                            NARZĘDZIOWNIA
  -----------------------------------------------------------------------------------------------------
    [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
  -----------------------------------------------------------------------------------------------------
  a

        Lista wszystkich narzędzi
  ---------------------------------------
  |       Nazwa        |  Liczba sztuk |
  ---------------------------------------
  |       grabie       |       3       |
  ---------------------------------------
```

```
-------------------------------------------------------------------------------------------------------
                                        NARZĘDZIOWNIA
-------------------------------------------------------------------------------------------------------
  [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
-------------------------------------------------------------------------------------------------------
d
      Lista dostępnych narzędzi
--------------------------------------
|       Nazwa        |  Liczba sztuk |
--------------------------------------
|       grabie       |       2       |
--------------------------------------


-------------------------------------------------------------------------------------------------------
                                        NARZĘDZIOWNIA
-------------------------------------------------------------------------------------------------------
  [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
-------------------------------------------------------------------------------------------------------
w

                     Lista wydanych narzędzi
-------------------------------------------------------------------
|       Nazwa        |  Imię i nazwisko   |       Godzina       |
-------------------------------------------------------------------
|       grabie       |  Agnieszka Pytel   |  16:26 (2019-01-15) |
-------------------------------------------------------------------


-------------------------------------------------------------------------------------------------------
                                        NARZĘDZIOWNIA
-------------------------------------------------------------------------------------------------------
  [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
-------------------------------------------------------------------------------------------------------
z

 Zwrot narzędzia

 Podaj nazwę narzędzia: grabie
 Podaj imię i nazwisko osoby, która wypożyczyła narzędzie: Agnieszka Pytel
Narzędzie zostało zwrócone, dziękujemy.


-------------------------------------------------------------------------------------------------------
                                        NARZĘDZIOWNIA
-------------------------------------------------------------------------------------------------------
  [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
-------------------------------------------------------------------------------------------------------
a

      Lista wszystkich narzędzi
--------------------------------------
|       Nazwa        |  Liczba sztuk |
--------------------------------------
|       grabie       |       3       |
--------------------------------------


-------------------------------------------------------------------------------------------------------
                                        NARZĘDZIOWNIA
-------------------------------------------------------------------------------------------------------
  [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
-------------------------------------------------------------------------------------------------------
d
      Lista dostępnych narzędzi
--------------------------------------
|       Nazwa        |  Liczba sztuk |
--------------------------------------
|       grabie       |       3       |
--------------------------------------


-------------------------------------------------------------------------------------------------------
                                        NARZĘDZIOWNIA
-------------------------------------------------------------------------------------------------------
  [+] Dodaj [N] Zmień liczbę [-] Wydaj [Z] Zwróć [A] Wszystkie [D] Dostępne [W] Wydane [K] Koniec
-------------------------------------------------------------------------------------------------------
K
```