

**Anna Kaleta**  
**Piotr Chojnacki**  
IV rok, informatyka chemiczna  
Liceum Ogólnokształcące nr 10 we Wrocławiu

Wrocław, dn. 19 kwietnia 2006 roku

Czas trwania zajęć: 90 minut, przedmiot: informatyka

**Temat lekcji:** Zastosowanie funkcji w Pascalu

Cele ogólne lekcji:

1. Zapoznanie z budową programów i podprogramów w języku Pascal.
2. Przedstawienie sposobów wywoływania podprogramów w programie głównym.
3. Omówienie korzyści wynikających ze stosowania podprogramów.
4. Omówienie procedur i funkcji.

Cele operacyjne:

**Uczeń wie:**

1. Jak zbudowany jest program w języku Pascal.
2. Co to są parametry procedury/funkcji.
3. Jakie zadania spełniają zmienne lokalne i globalne.

**Uczeń umie:**

1. Prawidłowo wywołać podprogram w programie głównym.
2. Wymienić korzyści płynące z użycia podprogramów.
3. Prawidłowo zapisać procedurę i funkcję.

Metody pracy:

- wykład, pogadanka,
- ćwiczenia praktyczne

Forma pracy:

- grupowa i indywidualna;

Środki dydaktyczne:

- tablica, rzutnik, komputer z zainstalowanym kompilatorem języka Pascal;
- prezentacja Power Point;
- programy: *graj.pas*, *calc.pas* i *silnia.pas*;

## Przebieg lekcji:

### 1. Wstęp:----- (30 minut)

- ❖ Sprawdzenie obecności.
- ❖ Sprawdzenie pracy domowej:  
Pytamy uczniów czy przerobili program „choinka.pas” i procedurę „losowanie” (bez powtórzeń). Jeden chętny uczeń prezentuje swój pomysł. Resztę programów sprawdzają Agnieszki (prowadzące z poprzedniej lekcji).
- ❖ Przypomnienie wiadomości z ostatniej lekcji. **Sprawdzenie wcześniej pisanego programu: „graj.pas”**
  - kompilujemy program na laptopie;
  - analizujemy jego fragmenty: uczniowie odpowiadają na pytania
    - Wskażcie procedury wykorzystywane w programie totolotek;
    - Z czego zbudowany jest nagłówek procedury;
    - Gdzie znajduje się treść procedury;
    - Wskażcie listę parametrów formalnych;
    - W jaki sposób wywołujemy procedurę.

**(wartość lub nazwa zmiennej)**
- ❖ Ogólna postać procedury jest zatem następująca: (prezentacja Power Point)

```
procedure nazwa(lista parametrów formalnych);  
.  
. {deklaracja stałych, zmiennych i typów}  
.  
begin  
.  
. {treść procedury}  
.  
end;
```

Deklaracja procedury zawiera, zatem nagłówek i blok, na który składają się definicje i deklaracje wykorzystywanych obiektów oraz instrukcja złożona opisująca wykonywany algorytm.

Po nazwie procedury może wystąpić, ale nie musi, lista parametrów formalnych. Wywołanie procedury, czyli żądanie wykonania instrukcji zawartych w treści procedury polega na podaniu nazwy procedury wraz z listą parametrów aktualnych, jeżeli istnieją.

❖ Dyskusja z uczniami na temat przydatności procedur podsumowana wyświetleniem slajdu o następującej treści:

1. Dzielenie zadania na mniejsze moduły – każdy może być realizowany w oddzielnej procedurze.
2. Wprowadzenie porządku do programu – zwiększenie jego czytelności i przejrzystości.
3. Unikanie powtórzeń – fragmenty programu, które się powtarzają, są wyodrębnione w postaci procedur. **Np. użycie procedury z\_tablicy**
4. Łatwiejsze wyszukiwanie błędów i dokonywanie poprawek – w procedurze można zlokalizować błędy szybciej niż w długiej liście instrukcji programu głównego. **Np. testowaliśmy osobno procedurę losowanie.**
5. Programowanie zespołowe – po uzgodnieniu postaci procedur każdy z członków zespołu „może” zająć się pracą nad swoimi procedurami. **Sami to robiliście w zespołach.**

❖ Sformułowanie tematu lekcji: „Zastosowanie funkcji w Pascalu”

2. Część właściwa:----- (50 minut)

❖ **Funkcje – prezentacja Power Point** (10 min)

Ogólna postać funkcji w języku Pascal jest następująca:

```
function nazwa(lista parametrów formalnych): typ
wyniku;
.
.{deklaracja stałych, zmiennych i typów}
.
begin
.
.{treść funkcji}
.
end;
```

Deklaracja funkcji zawiera, zatem nagłówek (różniący się o procedury) i blok z definicjami i deklaracjami wykorzystywanych obiektów oraz instrukcję złożoną opisującą wykonywany algorytm.

W treści funkcji musi być umieszczona przypisanie

```
nazwa := wynik;
```

gdzie *wynik* jest zmienną lub wyrażeniem określającym zwracaną wartość. Ponadto w odróżnieniu od procedury nagłówek funkcji musi zawierać typ wyniku. Wywołanie funkcji ma postać:

```
zmienna := nazwa(lista parametrów aktualnych);
```

gdzie *zmienna* jest dowolną zmienną (zgodna z typem wyniku funkcji), a *nazwa* nazwą funkcji.

```
lub   writeln(nazwa(lista parametrów aktualnych));
```

- ❖ Modyfikacja wcześniej pisanego programu „silnia” na program z użyciem funkcji. Prawidłowy program pod nazwą silnia.pas. Kod programu podany na końcu konspektu. **(10 min)**

**Uczniowie otwierają swoje poprzednie programy silnia i przerabiają je na programy z użyciem funkcji.**

**Po 10 minutach sprawdzamy wykonanie zadania i prezentujemy na rzutniku naszą wersję.**

- ❖ Tworzenie „kalkulatora”, który korzystając z już napisanej funkcji silnia oblicza kolejno: silnie, podaje liczbę podzielników i sprawdza czy podana liczba jest kwadratem liczby naturalnej. Prawidłowy program pod nazwą calc.pas. Kod programu podany na końcu konspektu. **(30 min)**

**Uczniowie dzielą się na zespoły dwuosobowe. Co drugie stanowisko wykonuje funkcje kwadrat reszta funkcje podzielnik. Po 20 minutach sprawdzamy ich pracę. Prezentujemy na rzutniku naszą wersję i omawiamy.**

- ❖ **Przydatność użycia funkcji:**

**Takie same, co w przypadku procedur.**

**3. Podsumowanie:------(10 minut)**

- ❖ Najważniejsze korzyści wynikające ze stosowania podprogramów są następujące:
  1. Program napisany z podziałem na podprogramy jest znacznie bardziej czytelny i zrozumiały.
  2. Uzyskujemy bezpośrednią odpowiedniość między algorytmem a tekstem programu.
  3. Pewne powtarzające się fragmenty lub realizujące ściśle określone operacje powinny być wyodrębnione i zapisane w postaci jednego podprogramu.
  4. Podczas testowania i uruchamiania programu można oddzielnie testować poszczególne podprogramy, a następnie uruchomić cały program.

Programy, na których będziemy pracować to:

-----TOTOLOTEK-----GRAJ.pas-----

```
program totolotek;
  uses crt;

type
  lotto = array [1..6] of integer;

var
  i,j,k: integer;
  t,m,s: lotto;

procedure z_tablicy(a:lotto);
var i:integer;
begin
  for i:=1 to 6 do
    write(a[i], ' ');
  writeln;
end;

procedure gra(var x:lotto);
var i:integer;
begin
  writeln('Wybierz 6 liczb z zakresu od 1 do 49');
  for i:=1 to 6 do
    begin
      write('Podaj ', i , ' liczbe:');
      readln(x[i]);
    end;
  writeln;
  writeln('Oto liczby, ktore skresliles:');
  z_tablicy(x);
  writeln;
end;

procedure losowanie(var y:lotto);
var i:integer;
begin
  writeln('Losowanie..... wcisnij ENTER');
  readln;
  for i:=1 to 6 do
    y[i]:=1+random(49);
  writeln;
  writeln('Oto liczby, ktore wylosowano:');
  z_tablicy(y);
  writeln;
end;

procedure sprawdzenie(x:lotto; y:lotto);
var i,j:integer;
var z:lotto; var q:integer;
begin
  q:=0;
  for i:=1 to 6 do
    for j:=1 to 6 do
      begin
        if y[i]=x[j] then
          begin
```

```

        z[q+1]:=y[i];
        q:=q+1;
    end;
end;
writeln('liczba trafien: ', q);
writeln;
for i:=1 to q do write (z[i], '');
readln;
end;

begin
    clrscr;
    Randomize;
    gra(t);
    losowanie(m);
    sprawdzenie(t,m);

end.

```

-----Silnia-----silnia.pas-----

```

program silnia;
uses crt;

var n:integer;

function silnia_for(n:integer):longint;
var i:integer;
    wynik:longint;
begin
    wynik:=1;
    for i:=1 to n do wynik:=wynik*i;
    silnia_for:=wynik;
end;

begin
    clrscr;

    write('Podaj n: ');
    readln(n);

```



```

{---- obliczenie silni wywolujac funkcje -----}

if n>=0 then
    writeln('Silnia obliczona za pomoca funkcji z ',n,' wynosi:
',silnia_for(n))

else
    writeln('Musisz podac liczbe naturalna');

repeat until keypressed;
end.

```

-----CALC-----CALC.pas-----

```

program calc;
uses crt;

var n:byte;

function silnia_for(n:byte):longint;
var i:byte;
    wynik:longint;

begin
    wynik:=1;
    for i:=1 to n do wynik:=wynik*i;
    silnia_for:=wynik;
end;

function podzielnik(n:byte):byte;
var i,wynik:byte;

begin
    wynik:=0;
    for i:=2 to n-1 do
        begin
            if (n mod i)=0 then wynik:=wynik+1;
            end;
        podzielnik:=wynik;
    end;

function kwadrat(n:byte):boolean;

begin
    if (sqrt(n)-round(sqrt(n)))=0 then kwadrat:=true
    else kwadrat:=false;
end;

begin
    clrscr;

    write('Podaj n: ');
    readln(n);

```

```

{---- obliczenie silni wywolujac funkcje -----}

if n>=0 then
  writeln('Silnia obliczona za pomoca funkcji z ',n,' wynosi:
',silnia_for(n))

else
  writeln('Musisz podac liczbe naturalna');

{----- obliczenie liczby dzielnikow -----}

  writeln;
  writeln('Liczba ',n,' ma ',podzielnik(n),' dzielnikow roznych od 1 i
jej samej');

{----- obliczenie czy liczba jest kwadratem-----}

  writeln;
  if kwadrat(n)=true then writeln('Liczba ',n,' jest kwadratem liczby
naturalnej')
  else write('Liczba ',n,' NIE jest kwadratem liczby naturalnej');

  repeat until keypressed;
end.

```