uses crt;

var s,ps,st,pd,dv,de,pa,dk,ko:integer; {s, oznacenie hodnoty platidiel}

procedure ROZKLAD (a:integer; var d,c,l,y,x,v,z,i:integer);

begin

d:= a div 500; {pocet 500} a:= a mod 500;

c:= a div 100; a:= a mod 100;

l:= a div 50; a:= a mod 50;

y:= a div 20; a:= a mod 20;

x:= a div 10; a:= a mod 10;

v:= a div 5; a:= a mod 5;

z:= a div 2; i:= a mod 2

end;

begin

clrscr;

writeln ('zadaj sumu');

readln (s);

ROZKLAD (s,ps,st,pd,dv,de,pa,dk,ko); {hodnota s vstupy,ostatnym prem.}

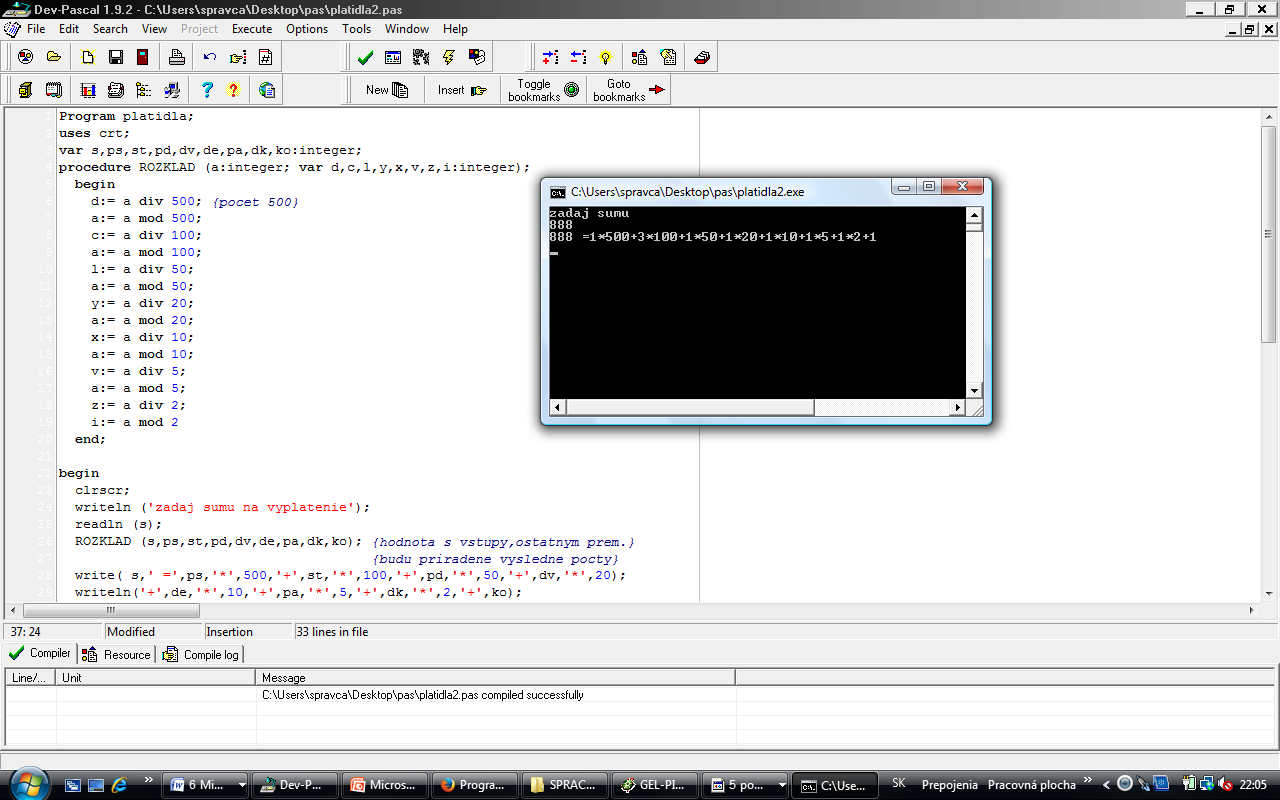
{budu priradene vysledne pocty}

write( s,' =',ps,'\*',500,'+',st,'\*',100,'+',pd,'\*',50,'+',dv,'\*',20);

writeln('+',de,'\*',10,'+',pa,'\*',5,'+',dk,'\*',2,'+',ko);

readln

end.

****

náhľad na hotový program 2. riešenie

1. **Riešenie úlohy (pomocou podmienky )**

**Toto riešenie bude nosným pre uplatnenie neúplného podmieneného príkazu**

**Pre program na riešenie úlohy o výplate peňazí, v ktorom treba určiť, koľko 500, 200, 100, 50, 20, 10, 5, 2, 1 - euroviek potrebujeme na vyplatenie danej sumy tak, aby sme použili čo najmenší možný počet platidiel.**

V prvom kroku si ujasníme čo má program vykonať, aké bude jeho členenie, ktoré

dátové typy, príkazy, funkcie použijeme,

( názov programu , deklarácia premenných, deklarácia poľa, určenie jeho prvkov, členenie v programe, postupnosť krokov – algoritmizáciu )

Formulácia - slovné zadanie úlohy ( definovanie vstupov a výstupov )

Analýza – úlohu zovšeobecňujeme, určujú sa podmienky postupu ( popis ako získať z daných vstupov dané výstupy)

Zostavenie algoritmu- presné vyjadrenie stavby, logiky a postupu riešenia.

**program mincovka;** *pomenovanie programu*

**uses crt;** *knižničné funkcie zabezpečujúce riadenie vstupov, výstupov*

Učiteľ sa pýta, prečo je výhodné použiť pole a čo bude tvoriť prvky poľa,

ako budeme pristupovať k prvkom poľa a ako ich načítame (žiaci už poznajú pojem: pole, jednoduchý cyklus pre určený počet opakovaní

**var b, pocet: array [1..9]of integer ;** *deklarácia premenných, počtu prvkov poľa,*

*pole obsahuje deväť prvkov rovnakého typu*

**suma, i:integer;** *deklarácia premennej pre požadovanú sumu*

*a priradenie premennej do prvku poľa(počet b*

začiatok programu a načítavanie hodnôt pre jednotlivé prvky poľa

**begin**

**b[1]:=500; b[2]:=200; b[3]:=100; b[4]:=50; b[5]:=20;**

**b[6]:=10; b[7]:=5; b[8]:=2; b[9]:=1;** *príkaz pre načítanie prvkov poľ*a

po načítaní jednotlivých prvkov poľa nasleduje požiadavka na zápis sumy,

ktorá je určená pre vyplatenie

**writeln ('Zadajte sumu v eurach ktoru chcete vyplatit');** *výzva**na zadanie sumy*

**-** *výpis na monitore*

program načíta z klávesnice zápis a zobrazí požadovanú sumu na monitore počítača

v tomto príklade som použil sumu 888 Eur (z ďalšieho vyplynie prečo)

**readln (suma);** *načítanie zadanej sumy*

nasleduje krok pre stanovený počet opakovaní ( deväť opakovaní pre načítanie

hodnôt pre jednotlivé bankovky resp. mince – učiteľ sa pýta, prečo práve deväť opakovaní,

žiaci vedia, že **for** súvisí s  poľom a jeho prvkami

**for i:= 1 to 9 do** *prebehne cyklus pre načítanie hodnôt prvkov poľa*

v ďalšom kroku nasleduje celočíselné delenie jednotlivých hodnôt prvkov poľa od

najväčšej hodnoty, a každé ďalšie delenie zbytku po predošlom delení, až bude výsledkom

delenia nedeliteľný zbytok

tu je dôležité pripomenúť, a položiť žiakom otázku ako rozumejú zbytku po delení

máme sumu pre vyplatenie v hodnote 888 eur (suma je zvolená tak, aby bola

zastúpená každá hodnota bankoviek, resp. mincí a aby bol viditeľný aj ich počet

**begin**

**pocet[i]:=suma div b[i];** *celočíselné delenie**i je v sume*

**suma:= suma mod b[i];** *zbytok po delení*

*výsledok delenia je*

888:500 ( 1), zbytok je hodnota 500 Eur bankovky

388:200 (1), zbytok je

188:100 (1), zbytok je

88:50 (1), zbytok je

38:20 (1), zbytok je

18:10 (1), zbytok je

8:5 (1), zbytok je

3:2 (1), zbytok je

1 (1), prebehol určený počet opakovaní

**end;**

**writeln('Pocty bankoviek a minci v eur na vyplatenie sumy:');** *výpis na monitore*

program vypisuje na základe prebehnutia cyklu a stanovenej podmienky neúplného príkazu vetvenia, ak počet hodnôt pre jednotlivé prvky poľa nie je rovný nule, tak vypíš hodnotu priradenú prvku v poli a vypíš počet kusov tejto hodnoty nachádzajúcej sa v poli,

v tomto prípade sú to nominálne hodnoty bankoviek a mincí po jednom kuse

**for**

**i:=1 to 9 do**

*prebehne cyklus pre výpis počtu hodnôt jednotlivých prvkov poľa*

**if pocet[i]<>0 then** *ak počet prvkov nerovná sa nule potom*

**writeln (b[i],' je potrebnych ',pocet[i],' kusov');** *vypíš hodnotu prvku poľa a počet hodnôt*

**readln;** *zápis prázdneho riadku čaká na stlač enter*

**end.** *koniec programu*

Žiaci si robia poznámky súbežne s vysvetľovaním postupu riešenia tejto úlohy, pričom učiteľ do tohto riešenia zapája aj samotných žiakov, nabáda ich, aby vyjadrovali aj svoje vlastné myšlienky, aby sa pýtali a skúmali, prečo beží program práve takto.

Zdrojový kód pre zadanú úlohu vyzerá takto: [(v pracovnom liste je upravený pre prácu žiaka)](file:///C:\Users\EVSINF1\AppData\Local\Temp\praclist_did2.docx)

**program mincovka;**

**uses crt;**

**var b, pocet: array [1..9]of integer ;**

**suma, i:integer;**

**begin**

**b[1]:=500; b[2]:=200; b[3]:=100; b[4]:=50; b[5]:=20;**

**b[6]:=10; b[7]:=5; b[8]:=2; b[9]:=1;**

**writeln ('Zadajte sumu v eurach ktoru chcete vyplatit');**

**readln (suma);**

**for i:= 1 to 9 do**

**begin**

**pocet[i]:=suma div b[i];**

**suma:= suma mod b[i];**

**end;**

**writeln('Pocty bankoviek a minci v eur na vyplatenie sumy:');**

**for**

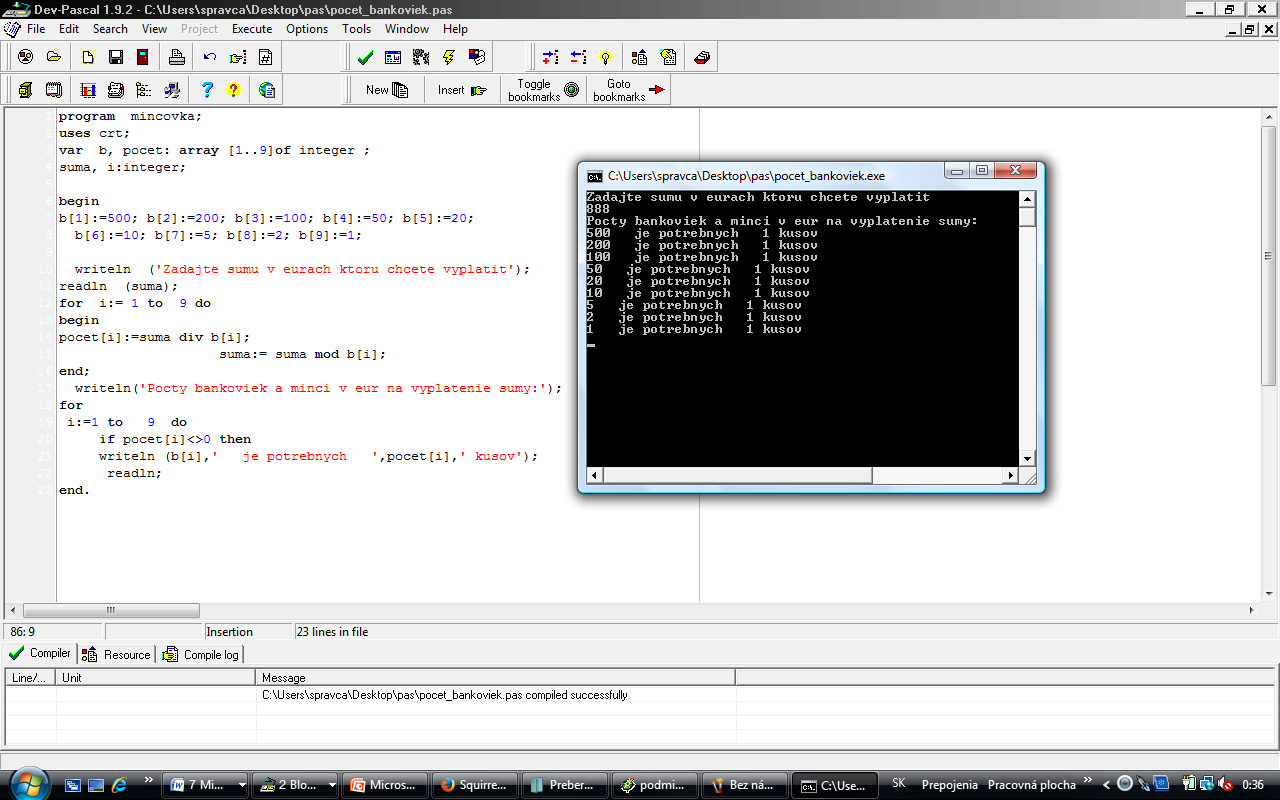
**i:=1 to 9 do**

**if pocet[i]<>0 then**

**writeln (b[i],' je potrebnych ',pocet[i],' kusov');**

**readln;**

**end.**

****

náhľad na hotový program 3. Riešenie

**Užívateľom definované typy.**

V Programovacom jazyku Pascal, môžeme typy rozdeliť do štyroch skupín:

**jednoduché typy**, reťazové typy, štruktúrované typy, typ ukazovateľ

Jednoduché typy definujú usporiadanie množiny hodnôt. Patria sem ordinálne typy a typ *real.*

Ordinálne typy: ***integer****,* ***shorint****,* ***longint****,* ***byte****,* ***word****,* ***boolean***a***char****.*

Okrem toho môžeme použiť ďalšie, užívateľom definované ordinálne typy, a to

***vymenovaný typ***a*typ* ***interval****.*

**Užívateľom definované typy patria medzi jednoduché** **dátové** **typy** - celočíselné, reálne, znakové, logické

#### VYMENOVANÝ DÁTOVÝ TYP

Z typu definovaného vymenovaním hodnôt je to hlavne použitie takých hodnôt typu, ktoré spracovávaný objekt nadobúda v reálnom prostredí. Program sa tak stáva prehľadnejším, čitateľnejším a ľahšie spracovateľným.

Užívatel si ho definuje podľa vlastných požiadaviek, jednotlivé hodnoty majú svoje ordinálne čísla podľa poradia v zápise a ich počítanie začína nulou.

Umiestňuje sa do súhrnu definícií typov, ktorý má tvar:

|  |
| --- |
| **type**  *tyzden* = (*pondelok*, *utorok*, *streda*, s*tvrtok*, *piatok*,*sobota*,*nedela*);  *euro = ( 500, 200, 100, 50, 20, 10, 5, 2, 1 )*  **var**  *den*: *tyzden*;  *doba*: *obdobie*;  *hodnota: euro* |

Potom platia nasledujúce vzťahy:

Succ (pondelok) = utorok

pred (jesen) = leto

ord(piatok) = 4

#### DATOVÝ TYP INTERVAL

Tento datový typ špecifikuje súvislú neprázdnu podmnožinu hodnôt určitého ordinálneho typu. Dolnú a hornú hranicu tejto podmnožiny udávajú dve konštanty daného ordinálneho typu.

Napr.: zápis 1 .. 100 definuje interval, špecifikujúcí množinu celočíselných hodnôt. Typ interval však nemusí být definováný iba nad číselnými typmi, môže být definovaný aj nad typom vymenovaným či char,

Ako ukazuje nasledujúcí príklad:

|  |
| --- |
| **type** |
| *pracovne\_dni* = *pondelok* .. *piatok*; |
| *velke\_pismeno* = 'A' .. 'Z'; |

Potom platia nasledujúce vzťahy:

ord(piatok) = 4

ord('A') = 65

'A' < 'B' < 'C' < ... < 'Z'

**Záver hodiny:**

Učiteľ vyhodnotí priebeh vyučovacej hodiny, v krátkosti zhrnie naplnenosť cieľov hodiny, upozorní

na význam využitia príkazu a funkcie vetvenia v programovaní.

Kontrolné otázky k prebranému učivu:

Kedy použijete príkaz vetvenia v programe?

Akým spôsobom sa zapisuje neúplný podmienený príkaz vetvenia?

V ktorej časti programu umiestňujeme neúplný podmienený príkaz vetvenia ?

Definujte dátový typ interval, uveďte príklad.

Aký typ by zte zvolili pre zadanie mesiacov v roku? Napíšte jeho tvar.

Námet na domácu úlohu:

Skúste doplniť program tak, aby dokázal od už vyplatenej sumy odpočítať ďalšiu sumu a znova prepočítal

počet bankoviek a mincí, ktoré po odpočítaní zostanú.

**ZDROJE:** [**https://www.gymrk.sk/Direr/pascal/obsah.htm#ucivo**](https://www.gymrk.sk/Direr/pascal/obsah.htm#ucivo)

[**http://cec.truni.sk/stoffov/programovanie/ZSProg/Sites/803.html**](http://cec.truni.sk/stoffov/programovanie/ZSProg/Sites/803.html)

[**http://www.sosdca.edu.sk/programovanie/vetvenie.htm**](http://www.sosdca.edu.sk/programovanie/vetvenie.htm)

[**http://referaty.atlas.sk/ostatne/informatika/15729/?print=1**](http://referaty.atlas.sk/ostatne/informatika/15729/?print=1)

[**http://www.isd.cz/pascal/3dattypy.html**](http://www.isd.cz/pascal/3dattypy.html)

**POMOCNÉ ZDROJE:**

[**Algoritmy a programovanie v jazyku Pascal**](file:///C:\Users\EVSINF1\AppData\Local\Temp\prezentacia_pascal.ppt)  **prezentácia stiahnutá zo zdroja: https://www.google.sk/search?q=algoritmy+a+programovanie+v+jazyku+pascal&ie=utf-8&oe=utf-8&client=firefox-b&gfe\_rd=cr&ei=V6gMWdPyJ6\_08AeLsqewAQ**

**Pripomeňme si základné prvky jazyka Pascal**

**Písmená** A..Z, a..z **Číslice** 0..9 **Špeciálne symboly**/ ( ) \* + - = := < > <> . ; [ ] { } ' ' , # $ @ ^ : \_

**Vyhradené slová** – Pozor !, nie je možné používať ich, ako premenné  
  
AND, ARRAY, BEGIN, CASE, CONST, DIV, DOWNTO, DO, **ELSE**, END, FILE, FOR, FUNCTION, GOTO, IN, **IF**, INHERITED, INTERFACE, LABEL, MOD, UNTIL, NOT, OF, OR, PROCEDURE,PROGRAM, RECORD, REPEAT, SET, STRING, **THEN**, TO, TYPE, USES, VAR, WHILE, WITH, XOR (modrou vyznačené pre podmienený príkaz)

**Direktívy** - nie je možné používať ich, ako premenné   
ABSOLUTE, ASSEMBLER, EXTERNAL, FAR, FORWARD, INTERRUPT, NEAR, VIRTUAL

**Identifikátory** - doposiaľ popísané symboly nestačia na vytvorenie programu (je ich málo a majú špeciálny význam). Preto si programátor vytvára vlastné prvky, ktorým dáva vlastný názov a sú to takzvané identifikátory - sú to mená premenných, funkcií, tried, vlastných údajových typov, ktoré si vytvára sám programátor. <p**Poznámky -** vlastný komentár autora slúžiaci na sprehľadnenie a okomentovanie programu. Forma ich zápisu v Pascal-e je (možno ich do seba vnárať):   
  
{ komentár }  
(\* komentár \*)  
{ ...  
(\* ... \*)  
(\* ... \*)  
}  
</p

**Návestia -** slúžia na predznačenie príkazov t.j. odvolanie sa na daný príkaz na inom mieste v programe a vyzvať ho k činnosti tým, že mu odovzdáme riadenie príkazom skoku GOTO

**Znakové reťazce -** špecifikum jazyka Pascal. Sú to postupnosti ľubovoľných znakov z tabuľky ASCII ohraničených znakom apostrof - ( ).

**Oddeľovače -** medzera, poznámka, end of line, end of file.

**Poznámka: Pravidlá pre tvorbu identifikátorov:**

1. identifikátor sa skladá z písmen, číslic a podčiarkovníka (na začiatku môže byť len písmeno),
2. nie je rozdiel medzi malými a VEĽKÝMI písmenami (v Pascale) - insenzitívny jazyk,
3. identifikátor nesmie byť totožný s vyhradeným slovom,
4. identifikátor musí byť jednoznačný v rámci programu alebo jeho časti (procedúra, funkcia, ...) t.j. nemožno označiť rovnakým identifikátorom viacero funkcií, premenných, typov, tried,
5. dĺžka postupnosti znakov identifikátora môže byť neobmedzená, avšak Turbo Pascal rozlišuje iba prvých 64 znakov.