

```
Module Module1
```

```
    Public Structure rozdeleni  
        Public rucne As Boolean  
        Public hodnota As Double  
        Public popis As String
```

```
    End Structure
```

```
    Public Structure rychlost  
        Public rucne As Boolean  
        Public hodnota As Double  
        Public popis As String  
        Public na_useku As Double
```

```
    End Structure
```

```
    Public nazev_prejezdu As String  
    Public kmEvidencni As rozdeleni  
    Public kmSkutecna As rozdeleni  
    Public kmOPL As rozdeleni  
    Public kmOPS As rozdeleni  
    Public zavory As Boolean  
    Public zavory_cele As Boolean
```

```
        'A.1.1 VÝCHOZÍ DÉLKY V METRECH
```

```
    Public d(0 To 20) As rozdeleni  
    Public dk As rozdeleni  
    Public dn As rozdeleni  
    Public ds As rozdeleni  
    Public sj As rozdeleni  
    Public ss As rozdeleni
```

```
        'A.1.2 VÝSLEDNÉ DÉLKY V METRECH
```

```
    Public dp As rozdeleni  
    Public dt As rozdeleni  
    Public dz As rozdeleni  
    Public sp As rozdeleni
```

```
        'A.1.3 ÚHLY VE STUPNÍCH
```

```
    Public alfa As rozdeleni  
    Public beta1 As rozdeleni  
    Public beta2 As rozdeleni
```

```
        'B.1
```

```
    Public a As rozdeleni
```

```
        'B.1.1 VÝCHOZÍ DOBY V SEKUNDÁCH
```

```
    Public tb1 As rozdeleni  
    Public tb2 As rozdeleni  
    Public td As rozdeleni  
    Public te As rozdeleni  
    Public tf As rozdeleni  
    Public tgA As rozdeleni
```

```
Public tg1 As rozdeleni
Public tg2 As rozdeleni
Public tk As rozdeleni
Public t0 As rozdeleni
Public tr As rozdeleni
Public tt As rozdeleni
Public tu As rozdeleni
Public tu1 As rozdeleni
Public tu2 As rozdeleni
Public tx As rozdeleni
    'B.1.2 VÝLEDNÉ DOBY V SEKUNDÁCH
Public tA As rozdeleni
Public tL As rozdeleni
Public tM As rozdeleni
Public tM1 As rozdeleni
Public tM2 As rozdeleni
Public tn As rozdeleni
Public tV As rozdeleni
Public tZ As rozdeleni
Public tZZ As rozdeleni
Public tZV As rozdeleni
    'B.1.3 DÉLKY V METRECH
Public dNN As rozdeleni
Public dv As rozdeleni
Public lD As rozdeleni
Public lp As rozdeleni 'rozlišit lichý a sudý směr
Public lv As rozdeleni ' rozlišit lichý a sudý směr
Public lzv As rozdeleni
Public lz As rozdeleni
    'B.1.4 RYCHLOSTI V KM/H
Public vs As rozdeleni
Public vt1(0 To 5) As rychlost
Public vts(0 To 5) As rychlost
Public vv As rozdeleni
    'C.1 VÝZNAM ZNAČENÍ
Public lpp As rozdeleni
Public lsv As rozdeleni
Public trp As rozdeleni
'moje
Public zav_rovnoB As Boolean
Sub NactiPopis()
    kmEvidencni.popis = "Kilometrická poloha přejezdu evidenční"
    kmSkutecna.popis = "Kilometrická poloha přejezdu skutečná"
    kmOPL.popis = "Km poloha okraje přejezdu ve směru od začátku tratě"
    kmOPS.popis = "Km poloha okraje přejezdu ve směru od konce tratě"
```

'A.1.1 VÝCHOZÍ DÉLKY V METRECH

d(0).popis = "Průmět délky nebezpečného pásma do vnějšího okraje jízdního pruhu pozemní komunikace"

d(1).popis = "Délka d1 je průmět délky nebezpečného pásma do vnějšího okraje jízdního pruhu pozemní komunikace."

d(2).popis = "Délka d2 je vzdálenost kolmého průmětu osy samostatného stožáru výstražníku před přejezdem do vnějšího okraje jízdního pruhu pozemní komunikace a průsečíku tohoto okraje s hranicí nebezpečného pásma." [↗](#)

d(3).popis = "Délka d3 je vzdálenost průsečíku roviny závory za přejezdem s vnějším okrajem jízdního pruhu pozemní komunikace za přejezdem a průsečíku tohoto okraje s hranicí nebezpečného pásma. Pokud není za přejezdem jízdní pruh přehrazován závorou, stanoví se $d3=0$ " [↗](#)

d(4).popis = "Délka d4 je průmět části sklopeného břevna závory přehrazující jízdní pruhy pro jízdu na přejezd do vnějšího okraje jízdního pruhu pozemní komunikace." [↗](#)

d(5).popis = "Délka d5 je průmět části sklopeného břevna závory, přehrazující jízdní pruhy za přejezdem do vnějšího okraje jízdního pruhu pozemní komunikace." [↗](#)

d(6).popis = "Délka d6 je průmět šířky jízdního pruhu (jízdních pruhů) za přejezdem do vnějšího okraje jízdního pruhu pozemní komunikace; přichází v úvahu, nejsou-li jízdní pruhy za přejezdem přehrazeny závorou." [↗](#)

d(7).popis = "Délka d7 je vzdálenost čelních ploch světel výstražníku od osy jeho stožáru měřená ve směru pozemní komunikace; je závislá na konstrukčním provedení výstražníku." [↗](#)

d(8).popis = "Délka d8 je vzdálenost čela silničního vozidla od čelní plochy světel výstražníku do vnějšího okraje jízdního pruhu pozemní komunikace, nebo u přejezdu s tupým úhlem křížení od nejbližší části závory, zasahuje-li před výstražník. Uvažuje se, že silniční vozidlo, případně chodec zastaví jeden metr od čelních ploch světel výstražníku, nebo od nejbližší části břevna závory. Proto se stanoví $d8 = 1m$." [↗](#)

d(9).popis = "Délka d9 je vzdálenost průsečíku roviny závory před přejezdem s vnějším okrajem jízdního pruhu pozemní komunikace ve směru jízdy na přejezd a průsečíku tohoto okraje s hranicí nebezpečného pásma." [↗](#)

d(10).popis = "Délka d10 je průmět části sklopeného břevna závory přehrazující jízdní pruhy pro jízdu na přejezd do vnějšího okraje jízdního pruhu pozemní komunikace." [↗](#)

d(11).popis = "Délka d11 je vzdálenost průsečíku vnějšího okraje jízdního pruhu pozemní komunikace s hranicí nebezpečného pásma a průmětu nejvzdálenější části výstražníku nebo závory do tohoto okraje." [↗](#)

dk.popis = "Vzdálenost os krajních kolejí; u jednokolejového přejezdu $dk = 0$ "

dn.popis = "Vzdálenost hranic nebezpečného pásma; u jednokolejového přejezdu $dn = 5$, u vícekolejového přejezdu $dn = dk + 5$ "

ds.popis = "Délka silničního vozidla (22 m) / chodce (3 m)"

sj.popis = "Celková šířka všech jízdních pruhů pro daný směr jízdy"

ss.popis = "Celková šířka všech jízdních pruhů pozemní komunikace"

'A.1.2 VÝSLEDNÉ DÉLKY V METRECH

dp.popis = "Délka pásma přejezdu dp je délka pozemní komunikace, ve které se nesmí nacházet silniční vozidlo (popř. jiný účastník provozu na pozemní komunikaci) v době jízdy drážního vozidla." [↗](#)

dt.popis = "Do délky směrodatné pro výpočet vyklizovací doby dT je třeba zahrnout kromě délky pásma přejezdu i délku silničního vozidla ds." [↗](#)

dz.popis = "Délka směrodatná pro výpočet předzváněcí doby pro závoru před přejezdem dz je vzdálenost od konce silničního vozidla před přejezdem k nejvzdálenější části břevna závory před přejezdem. Uvažuje se u PZS s polovičními závorami a u PZS s celými závorami, kde závora před přejezdem se sklápí dříve než závora za přejezdem. Pro přejezdové zařízení s celými závorami, kde se závory před přejezdem i za ním sklápějí současně, nebo jen pro závoru za přejezdem, sklápí-li se závora před přejezdem dříve, je $dz=dt$." [↗](#)

sp.popis = "Šířka přejezdu sp je délka koleje, ve které drážní vozidlo znemožňuje provoz po pozemní komunikaci; je to vzdálenost průsečíku osy koleje s vnějším okrajem jízdního pruhu pozemní komunikace na přejezdu." [↗](#)

'A.1.3 ÚHLY VE STUPNÍCH

alfa.popis = "Úhel křížení pozemní komunikace s dráhou; měří se v kladném smyslu (tj. proti směru otáčení hodinových ručiček) od osy pozemní komunikace k ose koleje" [↗](#)

beta1.popis = "Úhel břevna závory s osou pozemní komunikace; pokud jsou tyto úhly před přejezdem a za ním různé, označuje se úhel břevna závory [↗](#)

před přejezdem $\beta 1$ "

beta1.popis = "Úhel břevna závory s osou pozemní komunikace; pokud jsou tyto úhly před přejezdem a za ním různé, označuje se úhel břevna závory za přejezdem $\beta 2$ "

'B.1

a.popis = "Průměrné zrychlení nebo zpomalení v $m.s^{-2}$ "

'B.1.1 VÝCHOZÍ DOBY V SEKUNDÁCH

tb1.popis = "Základní bezpečnostní doba tb1 je doba, která musí uplynout od skončení měření vyklizovací doby do příjezdu čela drážního vozidla na přejezd a do výpočtu se bere $tb1=6s$ "

tb2.popis = "Přídavná bezpečnostní doba tb2 je doba, která vyplývá z tolerancí při měření a zaokrouhlování. U nově projektovaných PZS se do výpočtu bere $tb2=3s$, u stávajících PZS, projektovaných podle norem platných do října 1998, se do výpočtu bere $tb2=0s$."

td.popis = "Doba průjezdu drážního vozidla přejezdem td závisí na šířce přejezdu, délce a rychlosti nejpomalejšího drážního vozidla."

te.popis = "Celková doba plánovaného stání mezi dopravnou a přejezdem"

tf.popis = "Doba přípravy jízdní cesty a výpravy drážního vozidla"

tgA.popis = "Doba pravidelného plánovaného stání drážního vozidla ve vzdalovacím úseku stanovená provozovatelem dráhy"

tg1.popis = "Doba pravidelného plánovaného stání drážního vozidla v části přibližovacího úseku před přejezdníkem stanovená provozovatelem dráhy, tato doba je vždy součástí doby tg2"

tg2.popis = "Doba pravidelného plánovaného stání drážního vozidla v části přibližovacího úseku stanovená provozovatelem dráhy"

tk.popis = "Kritická doba"

t0.popis = "Doba zvedání břevna závory"

tr.popis = "Doba reakce zařízení tr je doba ovlivnění ovládacího prvku závislého na jízdě drážního vozidla do zahájení výstrahy, pokud by se neuplatnilo odložení zahájení výstrahy. Závisí od druhu prvků, použitých na zjištění přítomnosti drážního vozidla v ovládacím úseku, a od druhu prvků použitých ve vlastním přejezdovém zařízení. Pokud není pro jednotlivé typy ovládacích prvků a přejezdových zařízení udáno jinak, bere se do výpočtu jednotná doba $tr=1$."

tt.popis = "Doba průjezdu drážního vozidla vzdalovacím úsekem"

tu.popis = "Doba sklápění jednoho břevna závory"

tu1.popis = "Doba od povelu ke sklápění břevna závory do povelu ke sklápění posledního břevna závory před přejezdem"

tu2.popis = "Doba od povelu ke sklápění břevna závory za přejezdem do povelu ke sklopení posledního břevna závory"

tx.popis = "Přídavná doba na úplné sklopení břevna závory"

'B.1.2 VÝSLEDNÉ DOBY V SEKUNDÁCH

tA.popis = "Mezní doba anulace"

tL.popis = "Přibližovací doba"

tM.popis = "Mezní výstražná doba"

tM1.popis = "Mezní výstražná doba/tM1 - první část"

tM2.popis = "Mezní výstražná doba/tM2 - druhá část"

tn.popis = "Doba, za kterou se po spuštění výstrahy smí rozsvítit povolující návěstní znak"

tV.popis = "Vyklizovací doba tv je nejkratší doba, během níž může ještě bezpečně projet přejezdem nejdelší a nejpomalejší silniční vozidlo, nebo cyklista, nebo projít chodec, kteří jsou při spuštění výstrahy ve vzdálenosti d8 před výstražníkem, nebo břevnem závory, je-li blíže."

tZ.popis = "Předzváněcí doba pro břevno závory před přejezdem"

tZZ.popis = "Předzváněcí doba pro břevno závory za přejezdem"

tZV.popis = "Doba odložení výstrahy"

'B.1.3 DÉLKY V METRECH

dNN.popis = "Vzdálenost mezi přejezdem a návěstidlem, případně přejezdem a místem (před návěstidlem), po které lze zařízením zjistit volnost části přibližovacího úseku"

dv.popis = "Délka nejdelší soupravy drážních vozidel "

lD.popis = "Vzdálenost mezi středem přejezdu a nejvzdálenějším místem v dopravně, kde může stát drážní vozidlo, kterému bude povolen odjezd"

```
(většinou odjezdové, nebo cestové návěstidlo pro opačný směr jízdy)"
lp.popis = "Délka přibližovacího úseku"
lv.popis = "Délka vzdalovacího úseku (obvykle je rovna délce přibližovacího úseku opačného směru)"
lzv.popis = "Délka úseku trati mezi místem vypočteného počátku přibližovacího úseku a skutečným místem ovlivnění přejezdu"
lzv.popis = "Vzdálenost místa na trati od přejezdu, kde již strojvedoucí nemůže přijmout informaci o stavu přejezdu při přenosu informace (např. ↗
vzdálenost přejezdníku od přejezdu)"
'B.1.4 RYCHLOSTI V KM/H
vs.popis = "Rychlost nejpomalejšího silničního vozidla, není-li rychlost nejpomalejšího silničního vozidla určena, použije se pro chodce a ↗
cyklisty Vs = 3 km/h, pro ostatní uživatele vs = 5 km/h."
vtl(0).popis = "Nejvyšší dovolená rychlost v rozhodujícím úseku před přejezdem v lichém směru, tj. traťová rychlost dovolená návěstidlem, ↗
rychlostníkem, popř. i nejvyšší rychlost v obvodu výhybek"
vts(0).popis = "Nejvyšší dovolená rychlost v rozhodujícím úseku před přejezdem v sudém směru, tj. traťová rychlost dovolená návěstidlem, ↗
rychlostníkem, popř. i nejvyšší rychlost v obvodu výhybek"
vv.popis = "Rychlost nejpomalejšího drážního vozidla, není-li rychlost nejpomalejšího drážního vozidla určena, použije se Vv = 20 km/h"
'C.1 VÝZNAM ZNAČENÍ
lpp.popis = "Délka přibližovacího úseku PZS s přejezdníky"
lsv.popis = "Vzdálenost místa na trati od přejezdu, kde lze poprvé spatřit přejezdník v m"
trp.popis = "Doba reakce přejezdníku v s, tj. doba od ovlivnění ovládacího prvku závislého na jízdě drážního vozidla do rozsvícení návěsti ↗
*Uzavřený přejezd*, pokud by se neuplatnilo zpoždění rozsvícení této návěsti. Není-li stanoveno jinak, bere se trp = 3"
End Sub
Sub NactiKonst()
ds.hodnota = 22
dv.hodnota = 400
vs.hodnota = 5
vv.hodnota = 20
tr.hodnota = 1
tu.hodnota = 10
t0.hodnota = 10
tu1.hodnota = 0
tu2.hodnota = 0
tb1.hodnota = 6
tb2.hodnota = 3
End Sub
Sub AktualizujObrazek()
If alfa.hodnota = 90 Then
    Form1.obrPřejezd.Image = Prejezdy.My.Resources.Resources.kolmej500
ElseIf alfa.hodnota > 90 Then
    Form1.obrPřejezd.Image = Prejezdy.My.Resources.Resources.tupej500
ElseIf alfa.hodnota < 90 And alfa.hodnota > 0 Then
    Form1.obrPřejezd.Image = Prejezdy.My.Resources.Resources.ostrej500
End If
If alfa.hodnota = 0 Then Form1.obrPřejezd.Image = Nothing
End Sub
Function FiltrHodnot(hodnota As String)
    Select Case hodnota
```

```
Case " "
    hodnota = "x"
Case ""
    hodnota = "x"
End Select
If IsNumeric(hodnota) = False And hodnota <> "x" Then
    MsgBox("Zadaná hodnota '" & hodnota & "' není platná. Zadejte číslo (s desetinnou čárkou) nebo znak 'x', mezeru nebo prázdný znak.",
        vbCritical, "Chyba")
    hodnota = "x"
End If
Return hodnota
```

End Function

```
Function FiltrHodnotNastaveni(hodnota As String)
```

```
    Select Case hodnota
```

```
        Case " "
```

```
            hodnota = "0"
```

```
        Case "x"
```

```
            hodnota = "0"
```

```
    End Select
```

```
    If IsNumeric(hodnota) = False And hodnota <> "" Then
```

```
        MsgBox("Zadaná hodnota '" & hodnota & "' není platná. Zadejte číslo (s desetinnou čárkou).", vbCritical, "Chyba")
```

```
        hodnota = "0"
```

```
    End If
```

```
    Return hodnota
```

```
End Function
```

```
End Module
```

```
Public Class VstupniData
```

```
Sub UlozData()
```

```
nazev_prejezdu = txtNazevPZS.Text  
kmEvidencni.hodnota = tabulka.Rows.Item(0).Cells(2).Value  
kmSkutecna.hodnota = tabulka.Rows.Item(1).Cells(2).Value  
kmOPL.hodnota = tabulka.Rows.Item(2).Cells(2).Value  
kmOPS.hodnota = tabulka.Rows.Item(3).Cells(2).Value  
dk.hodnota = tabulka.Rows.Item(4).Cells(2).Value  
ss.hodnota = tabulka.Rows.Item(5).Cells(2).Value  
alfa.hodnota = tabulka.Rows.Item(6).Cells(2).Value  
beta1.hodnota = tabulka.Rows.Item(7).Cells(2).Value  
beta2.hodnota = tabulka.Rows.Item(8).Cells(2).Value  
ds.hodnota = tabulka.Rows.Item(9).Cells(2).Value  
dv.hodnota = tabulka.Rows.Item(10).Cells(2).Value  
vs.hodnota = tabulka.Rows.Item(11).Cells(2).Value  
vv.hodnota = tabulka.Rows.Item(12).Cells(2).Value 'rychlost z jedné i druhé strany???  
vtl(0).hodnota = tabulka.Rows.Item(13).Cells(2).Value 'rychlost z jedné i druhé strany???  
vts(0).hodnota = tabulka.Rows.Item(14).Cells(2).Value  
tr.hodnota = tabulka.Rows.Item(15).Cells(2).Value  
tb1.hodnota = tabulka.Rows.Item(16).Cells(2).Value  
tb2.hodnota = tabulka.Rows.Item(17).Cells(2).Value  
tgA.hodnota = tabulka.Rows.Item(18).Cells(2).Value  
AktualizujObrazek()
```

```
End Sub
```

```
Sub Prepocitat_Data()
```

```
VyplnTabulku(tabulka, kmEvidencni.popis, "", kmEvidencni.hodnota, "km")  
VyplnTabulku(tabulka, kmSkutecna.popis, "", kmSkutecna.hodnota, "km")  
VyplnTabulku(tabulka, kmOPL.popis, "", kmOPL.hodnota, "km")  
VyplnTabulku(tabulka, kmOPS.popis, "", kmOPS.hodnota, "km")  
VyplnTabulku(tabulka, dk.popis, "dk", dk.hodnota, "m")  
VyplnTabulku(tabulka, ss.popis, "ss", ss.hodnota, "m")  
VyplnTabulku(tabulka, alfa.popis, "alfa", alfa.hodnota, "o")  
VyplnTabulku(tabulka, beta1.popis, "beta1", beta1.hodnota, "o")  
VyplnTabulku(tabulka, beta2.popis, "beta2", beta2.hodnota, "o")  
VyplnTabulku(tabulka, ds.popis, "ds", ds.hodnota, "m")  
VyplnTabulku(tabulka, dv.popis, "dv", dv.hodnota, "m")  
VyplnTabulku(tabulka, vs.popis, "vs", vs.hodnota, "km/h")  
VyplnTabulku(tabulka, vv.popis, "vv", vv.hodnota, "km/h")  
VyplnTabulku(tabulka, vtl(0).popis, "vtl", vtl(0).hodnota, "km/h")  
VyplnTabulku(tabulka, vts(0).popis, "vts", vts(0).hodnota, "km/h")  
VyplnTabulku(tabulka, tr.popis, "tr", tr.hodnota, "s")  
VyplnTabulku(tabulka, tb1.popis, "tb1", tb1.hodnota, "s")  
VyplnTabulku(tabulka, tb2.popis, "tb2", tb2.hodnota, "s")  
VyplnTabulku(tabulka, tgA.popis, "tgA", tgA.hodnota, "s")  
Form1.Text = "Přejezdy - " & txtNazevPZS.Text
```

```
End Sub
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Dalsi.Click
    Call UlozData()
    tabulka.Rows.Clear()
    Call Prepocitat_Data()
    Form1.butRucne.PerformClick()
    Me.Dispose()
End Sub
Sub VypInTabulku(tab As Object, popis As String, promenna As String, hodnota As Double, jednotka As String)
    Dim radek = tab.rows.count
    tab.rowcount = tab.rowcount + 1
    With tab.rows
        .item(radek).cells(0).value = popis
        .item(radek).cells(1).value = promenna
        .item(radek).cells(2).value = hodnota
        .item(radek).cells(3).value = jednotka
    End With
End Sub
Private Sub VstupniData_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    txtNazevPZS.Text = nazev_prejezdu
    chbZavory.Checked = zavory
    rbCele.Checked = zavory_cele
    rbPolovicni.Checked = Not (zavory_cele)
    chkRovnobezne.Checked = zav_rovnob
    Call Prepocitat_Data()
End Sub
Private Sub rbCele_CheckedChanged(sender As Object, e As EventArgs) Handles rbCele.CheckedChanged
    zavory_cele = True
End Sub
Private Sub rbPolovicni_CheckedChanged(sender As Object, e As EventArgs) Handles rbPolovicni.CheckedChanged
    zavory_cele = False
End Sub
Private Sub chbZavory_CheckedChanged(sender As Object, e As EventArgs) Handles chbZavory.CheckedChanged
    zavory = chbZavory.Checked
    If chbZavory.Checked = True Then
        rbCele.Enabled = True
        rbPolovicni.Enabled = True
        chkRovnobezne.Enabled = True
    If tabulka.Rows.Count > 0 Then
        REM tabulka.Rows.Item(7).Cells(2).Value = alfa.hodnota
        REM tabulka.Rows.Item(8).Cells(2).Value = alfa.hodnota
        Call chkRovnobezne_CheckedChanged(chbZavory, e)
    End If
Else
```

```
    rbCele.Enabled = False
    rbPolovicni.Enabled = False
    chkRovnobezne.Enabled = False
    If tabulka.Rows.Count > 0 Then
        tabulka.Rows.Item(7).Cells(2).Value = 0
        tabulka.Rows.Item(8).Cells(2).Value = 0
    End If
End If
End Sub
Private Sub chkRovnobezne_CheckedChanged(sender As Object, e As EventArgs) Handles chkRovnobezne.CheckedChanged
    If tabulka.Rows.Count > 0 Then
        If chkRovnobezne.Checked = True And chbZavory.Checked = True Then
            beta1.hodnota = alfa.hodnota
            beta2.hodnota = alfa.hodnota
            tabulka.Rows.Item(7).Cells(2).Value = beta1.hodnota
            tabulka.Rows.Item(8).Cells(2).Value = beta2.hodnota
        Else
            beta1.hodnota = 90
            beta2.hodnota = 90
            tabulka.Rows.Item(7).Cells(2).Value = beta1.hodnota
            tabulka.Rows.Item(8).Cells(2).Value = beta2.hodnota
        End If
    End If
End Sub
Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles Button1.Click
    Call UlozData()
    tabulka.Rows.Clear()
    Call Prepocitat_data()
End Sub
Private Sub tabulka_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles tabulka.CellClick
    Dim zprava As String
    If e.ColumnIndex = 0 Then
        zprava = tabulka.Item(e.ColumnIndex, e.RowIndex).Value
        MsgBox(zprava, vbInformation, "Informace o proměnné")
    End If
End Sub
Private Sub Tabulka_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles tabulka.CellContentClick
End Sub
Private Sub tabulka_CellEndEdit(sender As Object, e As DataGridViewCellEventArgs) Handles tabulka.CellEndEdit
    Dim radek = tabulka.CurrentRow.Index
    Dim sloupek = tabulka.CurrentCell.ColumnIndex
```

```
Dim hodnota = tabulka.Rows.Item(radek).Cells(sloupek).Value  
hodnota = FiltrHodnotNastaveni(hodnota)  
tabulka.Rows.Item(radek).Cells(sloupek).Value = hodnota
```

```
End Sub
```

```
End Class
```

```
Imports Microsoft.Office.Interop
Imports System.Runtime.InteropServices
Public Class Form1
    Sub VypInTabulku(tab As Object, norma As String, promenna As String, vzorec As String, dosazeni As String, hodnota As Double, Optional barva As Boolean = False)
        Dim radek = tab.rows.count
        tab.rowcount = tab.rowcount + 1
        If barva = True Then
            dosazeni = "zadáno ručně"
        End If
        With tab.rows
            .item(radek).cells(0).value = norma
            .item(radek).cells(1).value = promenna
            .item(radek).cells(2).value = vzorec
            .item(radek).cells(3).value = dosazeni
            If hodnota = -1 Then
                .item(radek).cells(4).value = ""
            Else
                .item(radek).cells(4).value = hodnota
            End If
        End With
        If barva = True Then tab.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
    End Sub
    Sub VypInRychlosti(tab As Object, rychlost As Int16, delka As Double, zbyva As Double, Optional barva As Boolean = False)
        Dim radek = tab.rows.count
        tab.rowcount = tab.rowcount + 1
        zbyva = 3600 * (delka / 1000) / rychlost
        If barva = True Then
            'barva
        End If
        With tab.rows
            .item(radek).cells(0).value = rychlost
            .item(radek).cells(1).value = delka
            .item(radek).cells(2).value = zbyva
        End With
        If barva = True Then tab.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
    End Sub
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        NactiPopis() 'načte popis proměnných, které jsou definovány v Modulu1
        NactiKonst() 'načte do proměnných hodnoty, které se nemění
        Button9.PerformClick() 'načtení hodnot ze souboru
        If navez_prejezdu <> "" Then Me.Text = "Přejezdy - " & navez_prejezdu
        te.hodnota = 60 'dáno na zastávku
        tf.hodnota = 120 'empiricky dáno
    End Sub
End Class
```

```
'sp.hodnota = 5.7 'šířka přejezdu sp je délka koleje, ve které drážní vozidlo znemožňuje provoz po pozemní komunikaci
'ss.hodnota = 5 'celková šířka všech jízdních pruhů pozemní komunikace
'sj.hodnota = 2.5 'celková šířka všech jízdních pruhů pro daný směr jízdy
'dk.hodnota = 0 'vzdálenost os krajních kolejí; u jednokolejového přejezdu dk = 0
'dv.hodnota = 140 'největší délka kolejového vozidla
'vtl(0).hodnota = 55 'max. rychlost lichá
'vts(0).hodnota = 55 'max. rychlost sudá
```

```
AktualizujObrazek()
buttRucne.PerformClick()
```

```
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles buttNastaveni.Click
    VstupniData.Show()
```

```
End Sub
```

```
Sub PocitejA()
```

```
'Private Sub Button2_Click(sender As Object, e As EventArgs) Handles buttDelky.Click
```

```
Dim dosazeni, vzorec, vstup As String
```

```
gridA.Rows.Clear() 'vymaže data v tabulce před novým přepočítáním
```

```
gridC.Rows.Clear()
```

```
'A.2.1
```

```
dn.hodnota = dk.hodnota + 5
```

```
'A.2.2
```

```
vzorec = "d1=dn/(sin alfa)=(dk+5)/(sin alfa)"
```

```
dosazeni = "d1=(" & dk.hodnota & "+5)/sin(" & alfa.hodnota & ")"
```

```
If d(1).rucne = False Then d(1).hodnota = (dk.hodnota + 5) / (Math.Sin((alfa.hodnota * Math.PI / 180)))
```

```
VyplnTabulku(gridA, "A.2.2) " & d(1).popis, "d1", vzorec, dosazeni, d(1).hodnota, d(1).rucne)
```

```
'A.2.3
```

```
vzorec = "naměřený údaj"
```

```
dosazeni = "nutno zadat"
```

```
If d(2).rucne = False Then
```

```
    If d(2).hodnota = 0 Then
```

```
        vstup = InputBox("Zadej proměnnou d2" & Chr(13) & d(2).popis)
```

```
        If IsNumeric(vstup) = True Then
```

```
            d(2).hodnota = vstup
```

```
            d(2).rucne = True
```

```
        Else
```

```
            MsgBox("Tato hodnota není číslo.", vbCritical, "Chyba vstupu")
```

```
        End If
```

```
    End If
```

```
End If
```

```
VyplnTabulku(gridA, "A.2.3) " & d(2).popis, "d2", vzorec, dosazeni, d(2).hodnota, d(2).rucne)
```

```
'A.2.4
```

```
If zavory_cele = True And zavory = True Then
```

```
    vzorec = "za přejezdem je pruh přehrazen závorou"
```

```
    dosazeni = "nutno zadat"
```

```
    If d(3).hodnota = 0 And d(3).rucne = False Then
```

```
        vstup = InputBox("Zadej proměnnou d3" & Chr(13) & d(3).popis)
```

```
        If IsNumeric(vstup) = True Then
```

```
            d(3).hodnota = vstup
```

```
            d(3).rucne = True
```

```
        Else
```

```
            MsgBox("Tato hodnota není číslo.", vbCritical, "Chyba vstupu")
```

```
        End If
```

```
    End If
```

```
Else
```

```
    vzorec = "za přejezdem není pruh přehrazen závorou"
```

```
    dosazeni = "d3=0"
```

```
    d(3).hodnota = 0
```

```
End If
```

```
VyplnTabulku(gridA, "A.2.4) " & d(3).popis, "d3", vzorec, dosazeni, d(3).hodnota, d(3).rucne)
```

```
'A.2.5 pokud zvora přehrazuje všechny pruhy, počítá se s celou šířkou ss
```

```
If zavory = True Then
```

```
    If beta1.hodnota > 90 Then
```

```
        vzorec = "Pro beta1 > 90°: d4=sj.tg(beta1-90)"
```

```
        dosazeni = "d4=" & sj.hodnota & ".tg(" & beta1.hodnota & "-90)"
```

```
        If d(4).hodnota = 0 And d(4).rucne = False Then d(4).hodnota = sj.hodnota * (Math.Tan((beta1.hodnota - 90) * Math.PI / 180))
```

```
    Else
```

```
        vzorec = "Pro beta1 <= 90°: d4=0"
```

```
        dosazeni = "d4=0"
```

```
        If d(4).rucne = False Then d(4).hodnota = 0
```

```
    End If
```

```
Else
```

```
    vzorec = "nejsou závory"
```

```
    dosazeni = "neuvažuje se"
```

```
End If
```

```
VyplnTabulku(gridA, "A.2.5) " & d(4).popis, "d4", vzorec, dosazeni, d(4).hodnota, d(4).rucne)
```

```
'A.2.6
```

```
If zavory = True Then
```

```
    If beta2.hodnota < 90 Then
```

```
        vzorec = "Pro beta2 < 90°: d5=sj.tg(90-beta2)"
```

```
        dosazeni = "d5=sj.tg(90-" & beta1.hodnota & ")"
```

```
If d(5).rucne = False Then d(5).hodnota = sj.hodnota * Math.Tan((90 - beta2.hodnota) * Math.PI / 180))
Else
    vzorec = "Pro beta2 >= 90°: d5=0"
    dosazeni = "d5=0"
    If d(5).rucne = False Then d(5).hodnota = 0
End If
Else
    vzorec = "nejsou závory"
    dosazeni = "neuvažuje se"
End If
VyplnTabulku(gridA, "A.2.6) " & d(5).popis, "d5", vzorec, dosazeni, d(5).hodnota, d(5).rucne)

'A.2.7
If zavory_cele = True And zavory = True Then
    vzorec = "pruh za přejezdem přehrazen"
    dosazeni = "neuvažuje se"
Else
    If alfa.hodnota < 90 Then
        vzorec = "Pro alfa <90°:d6=sj.tg(90-alfa)"
        dosazeni = "d6=sj.tg(90-" & alfa.hodnota & ")"
        If d(6).rucne = False Then d(6).hodnota = sj.hodnota * Math.Tan((90 - alfa.hodnota) * Math.PI / 180)
    Else
        vzorec = "Pro alfa >= 90°: d6=0"
        dosazeni = "d6=0"
        If d(6).rucne = False Then d(6).hodnota = 0
    End If
End If
VyplnTabulku(gridA, "A.2.7) " & d(6).popis, "d6", vzorec, dosazeni, d(6).hodnota, d(6).rucne)

'A.2.8
vzorec = "určuje se podle konstrukčního provedení výstražníku"
dosazeni = ""
If d(7).rucne = False Then d(7).hodnota = 1
VyplnTabulku(gridA, "A.2.8) " & d(7).popis, "d7", vzorec, dosazeni, d(7).hodnota, d(7).rucne)

'A.2.9
vzorec = "vzdálenost zastavení chodce nebo vozidla od čelní plochy světelných výstražníků"
dosazeni = ""
If d(8).rucne = False Then d(8).hodnota = 1
VyplnTabulku(gridA, "A.2.9) " & d(8).popis, "d8", vzorec, dosazeni, d(8).hodnota, d(8).rucne)

'A.2.10
If zavory = True Then
    vzorec = "přejezd se závory"
    dosazeni = "nutno zadat"
```

```
If d(9).rucne = False And d(9).hodnota = 0 Then
    vstup = InputBox("Zadej proměnnou d9" & Chr(13) & d(9).popis)
    If IsNumeric(vstup) = True Then
        d(9).hodnota = vstup
        d(9).rucne = True
    Else
        MsgBox("Tato hodnota není číslo.", vbCritical, "Chyba vstupu")
    End If
End If
Else
    vzorec = "nejsou závory"
    dosazeni = "neuvažuje se"
    If d(9).rucne = False Then d(9).hodnota = 0
End If
VyplnTabulku(gridA, "A.2.10" & d(9).popis, "d9", vzorec, dosazeni, d(9).hodnota, d(9).rucne)

'A.2.11
If zavory = True Then
    If beta1.hodnota < 90 Then
        vzorec = "Pro beta2 < 90°: d10=sj.tg(90-beta1)"
        dosazeni = "d10=sj.tg(90-" & beta1.hodnota & ")"
        If d(10).rucne = False Then d(10).hodnota = sj.hodnota * Math.Tan(((90 - beta1.hodnota * Math.PI / 180)))
    Else
        vzorec = "Pro beta1 >= 90°: d10=0"
        dosazeni = "d10=0"
        If d(10).rucne = False Then d(10).hodnota = 0
    End If
End If
Else
    vzorec = "nejsou závory"
    dosazeni = "neuvažuje se"
End If
VyplnTabulku(gridA, "A.2.11" & d(10).popis, "d10", vzorec, dosazeni, d(10).hodnota, d(10).rucne)

If d(2).hodnota + d(7).hodnota >= d(4).hodnota + d(9).hodnota Then
    vzorec = "platí, že d2+d7 >= d4+d9"
    dosazeni = "netřeba vyznačit místo zastavení siln. vozidla"
Else
    vzorec = "neplatí, že d2+d7 >= d4+d9"
    dosazeni = "NUTNO vyznačit místo zastavení siln. vozidla"
End If
VyplnTabulku(gridA, "A.2.11 - dodatek", "", vzorec, dosazeni, -1)
gridA.Rows(gridA.Rows.Count - 1).DefaultCellStyle.BackColor = Color.GhostWhite

'A.2.12
If d(2).hodnota + d(7).hodnota >= d(4).hodnota + d(9).hodnota Then
```

```

vzorec = "d11=d2+d7"
dosazeni = "d11=" & d(2).hodnota & "+" & d(7).hodnota
If d(11).rucne = False Then d(11).hodnota = d(2).hodnota + d(7).hodnota
Else
vzorec = "d11=d4+d9"
dosazeni = "d11=" & d(2).hodnota & "+" & d(7).hodnota
If d(11).rucne = False Then d(11).hodnota = d(4).hodnota + d(9).hodnota
End If
VyplnTabulku(gridA, "A.2.12) " & d(11).popis, "d11", vzorec, dosazeni, d(11).hodnota, d(11).rucne)

'A.2.13
vzorec = "délka vozidla/chodec"
dosazeni = "stanoveno v Nastavení"
VyplnTabulku(gridA, "A.2.13) " & ds.popis, "ds", vzorec, dosazeni, ds.hodnota, ds.rucne)
' VyplnTabulku(gridC, "A.2.13", "ds", vzorec, dosazeni, ds.hodnota, ds.rucne)

'A.3
If sp.hodnota = 0 Then 'občas se odměřuje ručně z polohopisu
vzorec = "sp=ss/sin(alfa)"
dosazeni = "sp=" & ss.hodnota & "/sin(" & alfa.hodnota & ")"
sp.hodnota = ss.hodnota / Math.Sin(alfa.hodnota * Math.PI / 180)
Else
vzorec = "zadáno ručně"
dosazeni = "sp=" & sp.hodnota
End If
VyplnTabulku(gridA, "A.3) " & sp.popis, "sp", vzorec, dosazeni, sp.hodnota, sp.rucne)
VyplnTabulku(gridC, "A.3", "sp", vzorec, dosazeni, sp.hodnota, sp.rucne)

'A.4
If zavory = True And zavory_cele = True Then
vzorec = "dp=d1+d3+d5+d8+d11"
dosazeni = "dp=" & Math.Round(d(1).hodnota, 2) & "+" & Math.Round(d(3).hodnota, 2) & "+" & Math.Round(d(5).hodnota, 2) & "+" & Math.Round(d(8).hodnota, 2) & "+" & Math.Round(d(11).hodnota, 2)
If dp.rucne = False Then dp.hodnota = d(1).hodnota + d(3).hodnota + d(5).hodnota + d(8).hodnota + d(11).hodnota
Else
vzorec = "dp=d1+d6+d8+d11"
dosazeni = "dp=" & Math.Round(d(1).hodnota, 2) & "+" & Math.Round(d(6).hodnota, 2) & "+" & Math.Round(d(8).hodnota, 2) & "+" & Math.Round(d(11).hodnota, 2)
If dp.rucne = False Then dp.hodnota = d(1).hodnota + d(6).hodnota + d(8).hodnota + d(11).hodnota
End If
VyplnTabulku(gridA, "A.4) " & dp.popis, "dp", vzorec, dosazeni, dp.hodnota, dp.rucne)
VyplnTabulku(gridC, "A.4", "dp", vzorec, dosazeni, dp.hodnota, dp.rucne)

If dp.hodnota > 25.5 Then 'DOVYMYSLET, JAK TO MĚNIT V PRŮBĚHU VÝPOČTU
'vložit hodnoty cyklisty
vs.hodnota = 3

```

```
ds.hodnota = 3
vzorec = "dp>25.5 (hodnoty cyklisty/chodce)"
dosazeni = "vs=3, ds=3"
Else
    'vložit hodnoty auta
    vs.hodnota = 5
    ds.hodnota = 22
    vzorec = "dp<=25.5 (hodnoty vozidla)"
    dosazeni = "vs=5, ds=22"
End If
VyplnTabulku(gridA, "A.4 - dodatek", "dp", vzorec, dosazeni, -1)
VyplnTabulku(gridC, "A.4", "dp", vzorec, dosazeni, -1)

gridA.Rows(gridA.Rows.Count - 1).DefaultCellStyle.BackColor = Color.GhostWhite
'A.5
vzorec = "dt=dp+ds"
dosazeni = "dt=" & Math.Round(dp.hodnota, 2) & "+" & ds.hodnota
If dt.rucne = False Then dt.hodnota = dp.hodnota + ds.hodnota
VyplnTabulku(gridA, "A.5" & dt.popis, "dt", vzorec, dosazeni, dt.hodnota, dt.rucne)
VyplnTabulku(gridC, "A.5", "dt", vzorec, dosazeni, dt.hodnota, dt.rucne)

'A.6 (nutno ještě dořešit / překontrolovat!!!)
If zavory = True Then
    If zavory_cele = False Then
        vzorec = "dz=ds+d8-d9+d10+d11"
        dosazeni = "dz=" & ds.hodnota & "+" & d(8).hodnota & "-" & d(9).hodnota & "+" & d(10).hodnota & "+" & d(11).hodnota
        If dz.rucne = False Then dz.hodnota = ds.hodnota + d(8).hodnota - d(9).hodnota + d(10).hodnota + d(11).hodnota
    Else
        vzorec = "dz=dt"
        dosazeni = "dz=" & dt.hodnota
        If dz.rucne = False Then dz.hodnota = dt.hodnota
    End If
Else
    vzorec = "nejsou závory"
    dosazeni = "neuvažuje se"
    If dz.rucne = False Then dz.hodnota = 0
End If
VyplnTabulku(gridA, "A.6" & dz.popis, "dz", vzorec, dosazeni, dz.hodnota, dz.rucne)
VyplnTabulku(gridC, "A.6", "dz", vzorec, dosazeni, dz.hodnota, dz.rucne)
End Sub
Private Sub gridA_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles gridA.CellClick
    Dim zprava As String
    If e.ColumnIndex = 0 Then
        zprava = gridA.Item(e.ColumnIndex, e.RowIndex).Value
        MsgBox(zprava, vbInformation, "Informace o proměnné")
    End If
End Sub
```

```
End If
End Sub

Sub PocitejB()
    Dim vzorec, dosazeni As String
    gridB.Rows.Clear()

    'B.2.1
    vzorec = "tr=1"
    dosazeni = "není-li uvedeno jinak"
    tr.hodnota = 1
    VyplnTabulku(gridB, "B.2.1) " & tr.popis, "tr", vzorec, dosazeni, tr.hodnota)

    'B.2.2 ZKONTROLVOAT PODMÍNKU, JESTLI JSOU NEBO NEJSOU ZÁVORY!!!!
    If zavory = False Then
        vzorec = "žádné závory"
        dosazeni = "neuvažuje se"
        tu.hodnota = 0
    Else
        vzorec = "tu=10"
        dosazeni = "není-li uvedeno jinak"
        tu.hodnota = 10
    End If
    VyplnTabulku(gridB, "B.2.2) " & tu.popis, "tu", vzorec, dosazeni, tu.hodnota)

    If zavory = False Then
        vzorec = "žádné závory"
        dosazeni = "neuvažuje se"
        t0.hodnota = 0
    Else
        vzorec = "t0=10"
        dosazeni = "není-li uvedeno jinak"
        t0.hodnota = 10
    End If
    VyplnTabulku(gridB, "B.2.2) " & t0.popis, "t0", vzorec, dosazeni, t0.hodnota)

    If zavory = False Then
        vzorec = "žádné závory"
        dosazeni = "neuvažuje se"
    Else

        vzorec = "tu1=0"
        dosazeni = "není-li uvedeno jinak"
        tu1.hodnota = 0
    End If
End If
```

```
VyplnTabulku(gridB, "B.2.2) " & tu1.popis, "tu1", vzorec, dosazeni, tu1.hodnota)
```

```
If zavory = False Then  
    vzorec = "žádné závory"  
    dosazeni = "neuvažuje se"  
    tu2.hodnota = 0
```

```
Else  
    vzorec = "tu2=0"  
    dosazeni = "není-li uvedeno jinak"
```

```
End If
```

```
tu2.hodnota = 0
```

```
VyplnTabulku(gridB, "B.2.2) " & tu2.popis, "tu2", vzorec, dosazeni, tu2.hodnota)
```

```
'B.2.3
```

```
vzorec = "tb1=6"  
dosazeni = "není-li uvedeno jinak"  
tb1.hodnota = 6
```

```
VyplnTabulku(gridB, "B.2.3) " & tb1.popis, "tb1", vzorec, dosazeni, tb1.hodnota)
```

```
'B.2.4
```

```
vzorec = "tb2=3"  
dosazeni = "není-li uvedeno jinak"  
tb2.hodnota = 3
```

```
VyplnTabulku(gridB, "B.2.4) " & tb2.popis, "tb2", vzorec, dosazeni, tb2.hodnota)
```

```
'B.2.5 U PŘEJEZDŮ S POLOVIČNÍMI ZÁVORAMI, ZKONTROLOVAT PODMÍNKU!
```

```
If zavory = False Then  
    vzorec = "žádné závory"  
    dosazeni = "neuvažuje se"  
    tx.hodnota = 0
```

```
Else
```

```
    If alfa.hodnota < 90 Then
```

```
        vzorec = "tx=tu+tu1-3,6*(d6+d1+d9-d10)/Vs"
```

```
        dosazeni = "tx=" & tu.hodnota & "+" & tu1.hodnota & "- 3.6 * (" & d(6).hodnota & "+" & d(1).hodnota & "+" & d(9).hodnota & "-" & d  
            (10).hodnota & ") /" & vs.hodnota
```

```
        tx.hodnota = (tu.hodnota + tu1.hodnota) - 3.6 * (d(6).hodnota + d(1).hodnota + d(9).hodnota - d(10).hodnota) / vs.hodnota
```

```
    Else
```

```
        vzorec = "tx=tu+tu1-3,6*(d1+d9)/Vs"
```

```
        dosazeni = "tx=" & tu.hodnota & "+" & tu1.hodnota & "- 3.6 * (" & d(1).hodnota & "+" & d(9).hodnota & ") /" & vs.hodnota
```

```
        tx.hodnota = (tu.hodnota + tu1.hodnota) - 3.6 * (d(1).hodnota + d(9).hodnota) / vs.hodnota
```

```
    End If
```

```
End If
```

```
VyplnTabulku(gridB, "B.2.5) " & tx.popis, "tx", vzorec, dosazeni, tx.hodnota)
```

```
'B.2.6
```

```

vzorec = "td=3,6 * (dv + sp) / Vv"
dosazeni = "td=3,6 * (" & dv.hodnota & " + " & sp.hodnota & ")/" & vv.hodnota
td.hodnota = Math.Round(3.6 * (dv.hodnota + sp.hodnota) / vv.hodnota, 1)
VyplnTabulku(gridB, "B.2.6) " & td.popis, "td", vzorec, dosazeni, td.hodnota)
VyplnTabulku(gridC, "B.2.6", "td", vzorec, dosazeni, td.hodnota)

```

```
'B.2.8
```

```

vzorec = "stanoveno"
dosazeni = "Lz = 0"
lz.hodnota = 0
VyplnTabulku(gridB, "B.2.8) " & lz.popis, "Lz", vzorec, dosazeni, lz.hodnota)

```

```
'B.2.9
```

```

vzorec = "stanoveno"
dosazeni = "Vs=" & vs.hodnota
VyplnTabulku(gridB, "B.2.9) " & vs.popis, "Vs", vzorec, dosazeni, vs.hodnota)

```

```
'B.2.10
```

```

vzorec = "stanoveno"
dosazeni = "Vv=" & vv.hodnota
VyplnTabulku(gridB, "B.2.10) " & vv.popis, "Vv", vzorec, dosazeni, vv.hodnota)

```

```
'B.3
```

```

vzorec = "tv=3,6*dt/Vs"
dosazeni = "tv=3,6*" & dt.hodnota & "/" & vs.hodnota
tv.hodnota = Math.Round(3.6 * dt.hodnota / vs.hodnota, 2)
VyplnTabulku(gridB, "B.3) " & tv.popis, "tv", vzorec, dosazeni, tv.hodnota)
VyplnTabulku(gridC, "B.3", "tv", vzorec, dosazeni, tv.hodnota)

```

```
'B.4 PŘIBLIŽOVACÍ DOBA (zaokr. na 2 des. místa)
```

```
If zavory = False Then
```

```

    vzorec = "tL=tr+tv+tb1+tb2 (žádné závory)"
    dosazeni = "tL=" & tr.hodnota & "+" & tv.hodnota & "+" & tb1.hodnota & "+" & tb2.hodnota
    tL.hodnota = Math.Round(tr.hodnota + tv.hodnota + tb1.hodnota + tb2.hodnota, 2)

```

```
End If
```

```
If zavory = True And zavory_cele = False Then
```

```

    vzorec = "tL=tr+tv+tb1+tb2+tx (poloviční závory)"
    dosazeni = "tL=" & tr.hodnota & "+" & tv.hodnota & "+" & tb1.hodnota & "+" & tb2.hodnota & "+" & tx.hodnota
    tL.hodnota = Math.Round(tr.hodnota + tv.hodnota + tb1.hodnota + tb2.hodnota + tx.hodnota, 2)

```

```
End If
```

```
If zavory = True And zavory_cele = True Then
```

```

    vzorec = "tL=tr+tv+tb1+tb2+tu+tu2 (celé závory)"
    dosazeni = "tL=" & tr.hodnota & "+" & tv.hodnota & "+" & tb1.hodnota & "+" & tb2.hodnota & "+" & tu.hodnota & "+" & tu2.hodnota
    tL.hodnota = Math.Round(tr.hodnota + tv.hodnota + tb1.hodnota + tb2.hodnota + tu.hodnota + tu2.hodnota, 2)

```

```
End If
```

```
VyplnTabulku(gridB, "B.4) " & tL.popis, "tL", vzorec, dosazeni, tL.hodnota)
VyplnTabulku(gridC, "B.4", "tL", vzorec, dosazeni, tL.hodnota)
```

```
'C.2 (délka přibližovacího úseku Lp, zaokrouhlní na celá čísla nahoru)
```

```
vzorec = "Lp=Vt*tL/3,6 a zároveň Lv=Lp!"
```

```
If lp.rucne = False Then
```

```
    dosazeni = "Lp=" & vtl(0).hodnota & "*" & tL.hodnota & "/3,6"
```

```
    lp.hodnota = Math.Ceiling(vtl(0).hodnota * tL.hodnota / 3.6)
```

```
Else
```

```
    dosazeni = ""
```

```
End If
```

```
VyplnTabulku(gridB, "C.2) " & lp.popis, "Lp", vzorec, dosazeni, lp.hodnota, lp.rucne)
```

```
VyplnTabulku(gridC, "C.2", "Lp", vzorec, dosazeni, lp.hodnota, lp.rucne)
```

```
lv.hodnota = lp.hodnota 'rozlišit lvP = lpS
```

```
'B.2.7 Lv=Lp --> ZÁVISÍ NA ODSTAVCI C.2
```

```
vzorec = "tt=3,6 * Lv / Vv"
```

```
dosazeni = "tt=3,6 * " & lv.hodnota & "/" & vv.hodnota
```

```
tt.hodnota = Math.Round(3.6 * lv.hodnota / vv.hodnota, 1)
```

```
VyplnTabulku(gridB, "B.2.7) " & tt.popis, "tt", vzorec, dosazeni, tt.hodnota)
```

```
'B.5.1 / B.5.2 / B.5.3
```

```
If zavory = False Then
```

```
    vzorec = "žádné závory"
```

```
    dosazeni = "neuvažuje se"
```

```
    tZ.hodnota = 0
```

```
    VyplnTabulku(gridB, "B.5.1) " & tZ.popis, "tZ", vzorec, dosazeni, tZ.hodnota)
```

```
End If
```

```
If zavory = True And zavory_cele = False Then
```

```
    vzorec = "tz=3,6*dz/Vs (poloviční závory)"
```

```
    dosazeni = "tz=3,6*" & dz.hodnota & "/" & vs.hodnota
```

```
    tZ.hodnota = 3.6 * dz.hodnota / vs.hodnota
```

```
    VyplnTabulku(gridB, "B.5.2) " & tZ.popis, "tZ", vzorec, dosazeni, tZ.hodnota)
```

```
End If
```

```
If zavory = True And zavory_cele = True Then
```

```
    vzorec = "tzz=tv (celé závory)"
```

```
    dosazeni = "tzz=" & tv.hodnota
```

```
    tZZ.hodnota = tv.hodnota
```

```
    VyplnTabulku(gridB, "B.5.3) " & tZZ.popis, "tzz", vzorec, dosazeni, tZZ.hodnota)
```

```
End If
```

```
'B.6.1 tgA určeno v Nastavení
```

```
vzorec = "tA=tt+td+tgA"
```

```
dosazeni = "tA=" & tt.hodnota & "+" & td.hodnota & "+" & tgA.hodnota
```

```
tA.hodnota = tt.hodnota + td.hodnota + tgA.hodnota
VyplnTabulku(gridB, "B.6.1) " & tA.popis, "tA", vzorec, dosazeni, tA.hodnota)
VyplnTabulku(gridC, "B.6.1", "tA", vzorec, dosazeni, tA.hodnota)
```

```
'B.6.2 tgA určeno v Nastavení, pro druhou lichou a sudou stranu trati se provede zvlášť výpočet
vzorec = "tgA určeno v Nastavení"
dosazeni = "tgA=" & tgA.hodnota
VyplnTabulku(gridB, "B.6.2) " & tgA.popis, "tgA", vzorec, dosazeni, tgA.hodnota, tgA.rucne)
```

```
'B.7 MEZNÍ VÝSTRAŽNÁ DOBA tM
vzorec = "tM1=tg1+3,6*(Lp-Lz)/Vv"
dosazeni = "tM1=" & tg1.hodnota & " + 3,6*( " & lp.hodnota & "-" & lz.hodnota & " )/" & vv.hodnota
tM1.hodnota = tg1.hodnota + 3.6 * (lp.hodnota - lz.hodnota) / vv.hodnota
VyplnTabulku(gridB, "B.7) " & tM1.popis, "tM1", vzorec, dosazeni, tM1.hodnota)
```

```
vzorec = "tM2=tg2+3,6*(Lp-dv)/Vv"
dosazeni = "tM2=" & tg2.hodnota & " + 3,6*( " & lp.hodnota & "-" & dv.hodnota & " )/" & vv.hodnota
tM2.hodnota = tg2.hodnota + 3.6 * (lp.hodnota - dv.hodnota) / vv.hodnota
VyplnTabulku(gridB, "B.7) " & tM2.popis, "tM2", vzorec, dosazeni, tM2.hodnota)
```

```
' Lzv (délka úseku trati mezi vyp. místem pú a počítačím bodem
vzorec = "ručně zadaná hodnota"
dosazeni = ""
VyplnTabulku(gridB, "Lzv) " & lzv.popis, "Lzv", vzorec, dosazeni, lzv.hodnota, lzv.rucne)
```

```
'B.8 DOBA ODLOŽENÍ VÝSTRAHY v lichém směru!!!, dořešení rovnoměrně zrychleného pohybu
vzorec = "tzv=3,6xLzv/Vt"
dosazeni = "tzv=3,6*" & lzv.hodnota & "/" & vtl(0).hodnota
tZV.hodnota = Math.Round(3.6 * lzv.hodnota / vtl(0).hodnota, 2) 'v lichém směru!!!
VyplnTabulku(gridB, "B.8) " & tZV.popis, "tzv", vzorec, dosazeni, tZV.hodnota)
```

```
'B.9 DOBA ZPOŽDĚNÍ ROZSVÍCENÍ NÁVĚSTIDLA
vzorec = "tn=tL-3,6*dn/Vt"
dosazeni = "tn=" & tL.hodnota & "- 3.6*" & dn.hodnota & "/" & vtl(0).hodnota
tn.hodnota = Math.Round(tL.hodnota - (3.6 * dn.hodnota / vtl(0).hodnota), 2) 'v lichém směru!!!
VyplnTabulku(gridB, "B.9) " & tn.popis, "tn", vzorec, dosazeni, tn.hodnota)
```

```
'LD vzdálenost mezi středem přejezdu a nejvzdálenějším místem od začátku trati:
vzorec = "Odměřeno ze situačního schématu"
dosazeni = "zadáno ručně"
VyplnTabulku(gridB, "LD) " & lD.popis, "LD", vzorec, dosazeni, lD.hodnota, lD.rucne)
```

```
'B.10 KRITICKÁ DOBA (pro všechny koleje zvlášť, nejdelší z nich - možno zaokrouhlit na minuty nahoru)
vzorec = "tk=tf+1,5*te+3,6*Ld+dv/Vv"
dosazeni = "tk=" & tf.hodnota & "+1,5*" & te.hodnota & "+3,6*( " & lD.hodnota & "+" & dv.hodnota & " )/" & vv.hodnota
```

```
tk.hodnota = Math.Round(tf.hodnota + 1.5 * te.hodnota + 3.6 * (lD.hodnota + dv.hodnota) / vv.hodnota, 1)
VyplnTabulku(gridB, "B.10) " & tk.popis, "tk", vzorec, dosazeni, tk.hodnota)
```

```
vzorec = "Kritická doba (s)"
dosazeni = "zaokrouhl. na minuty nahoru"
Dim tks = Math.Ceiling(tk.hodnota / 60) * 60
VyplnTabulku(gridB, "" & tk.popis, "tks", vzorec, dosazeni, tks)
```

```
vzorec = "Kritická doba (m)"
dosazeni = "v minutách"
tks = Math.Ceiling(tk.hodnota / 60)
VyplnTabulku(gridB, "" & tk.popis, "tks", vzorec, dosazeni, tks)
```

End Sub

Sub PocitejC()

```
'Private Sub ButtPribl_Click(sender As Object, e As EventArgs) Handles buttPribl.Click
'Dim vzorec, dosazeni As String
'gridC.Rows.Clear()
'C.2 VÝPOČET DÉLKY PŘIBL. ÚSEKU      /// dodělat pro rovnoměrně zrychleny/zpomalený pohyb
'vzorec = "Lp=Vt*tL/3,6"
'dosazeni = "Lp=" & vtl(0).hodnota & "*" & tL.hodnota & "/3,6"
' lp.hodnota = vtl(0).hodnota * tL.hodnota / 3.6 (počítá se v Běčku)
'VyplnTabulku(gridC, "C.2) " & lp.popis, "Lp", vzorec, dosazeni, lp.hodnota)

'vzorec = "Km = KmOPL - Lp"
'dosazeni = "Km=" & kmOPL.hodnota & "-" & Math.Round(lp.hodnota / 1000, 3)
'VyplnTabulku(gridC, "C.2 začátek přibl. úseku od začátku tr.", "Km", vzorec, dosazeni, (kmOPL.hodnota - Math.Round(lp.hodnota / 1000, 3)))

'vzorec = "Km = KmOPS + Lp"
'dosazeni = "Km=" & kmOPS.hodnota & "+" & Math.Round(lp.hodnota / 1000, 3)
'VyplnTabulku(gridC, "C.2 začátek přibl. úseku od konce tr.", "Km", vzorec, dosazeni, (kmOPS.hodnota + Math.Round(lp.hodnota / 1000, 3)))

'vzorec = "Km = KmOPS - Lp"
'dosazeni = "Km=" & kmOPS.hodnota & "-" & Math.Round(lp.hodnota / 1000, 3)
'VyplnTabulku(gridC, "C.2 začátek přibl. úseku od začátku tr.", "Km", vzorec, dosazeni, (kmOPS.hodnota - Math.Round(lp.hodnota / 1000, 3)))

'vzorec = "Km = KmOPL + Lp"
'dosazeni = "Km=" & kmOPL.hodnota & "+" & Math.Round(lp.hodnota / 1000, 3)
'VyplnTabulku(gridC, "C.2 začátek přibl. úseku od konce tr.", "Km", vzorec, dosazeni, (kmOPL.hodnota + Math.Round(lp.hodnota / 1000, 3)))
```

End Sub

Private Sub ButtRucne_Click(sender As Object, e As EventArgs) Handles buttRucne.Click

```
PocitejA()
PocitejB()
PocitejC()
End Sub
Private Sub gridB_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles gridB.CellClick
    Dim zprava As String
    If e.ColumnIndex = 0 Then
        zprava = gridB.Item(e.ColumnIndex, e.RowIndex).Value
        MsgBox(zprava, vbInformation, "Informace o proměnné")
    End If
End Sub

Private Sub gridC_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles gridC.CellClick
    'Dim zprava As String
    'If e.ColumnIndex = 0 Then
    '    zprava = gridC.Item(e.ColumnIndex, e.RowIndex).Value
    '    MsgBox(zprava, vbInformation, "Informace o proměnné")
    'End If

End Sub

Private Sub gridA_CellEndEdit(sender As Object, e As DataGridViewCellEventArgs) Handles gridA.CellEndEdit
    Dim radek = gridA.CurrentRow.Index
    Dim sloupek = gridA.CurrentRow.ColumnIndex
    Dim hodnota = gridA.Rows.Item(radek).Cells(sloupek).Value
    hodnota = FiltrHodnot(hodnota)
    For i = 0 To 9
        If radek = i Then
            With d(i + 1)
                If hodnota <> "x" Then
                    gridA.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
                    .hodnota = hodnota
                    .rucne = True
                Else
                    gridA.Rows(radek).DefaultCellStyle.BackColor = Color.White
                    .hodnota = 0
                    .rucne = False
                    gridA.Rows.Item(radek).Cells(sloupek).Value = "přepočítat"
                End If
            End With
        End If
    Next i
    If radek = 11 Then
        With d(11)
```

```
    If hodnota <> "x" Then
        gridA.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
        .hodnota = hodnota
        .rucne = True
    Else
        gridA.Rows(radek).DefaultCellStyle.BackColor = Color.White
        .hodnota = 0
        .rucne = False
        gridA.Rows.Item(radek).Cells(sloupek).Value = "přepočítat"
    End If
End With
End If

If radek = 13 Then 'sp
    With sp
        If hodnota <> "x" Then
            gridA.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
            .hodnota = hodnota
            .rucne = True
        Else
            gridA.Rows(radek).DefaultCellStyle.BackColor = Color.White
            .hodnota = 0
            .rucne = False
            gridA.Rows.Item(radek).Cells(sloupek).Value = "přepočítat"
        End If
    End With
End If

If radek = 14 Then 'dp
    With dp
        If hodnota <> "x" Then
            gridA.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
            .hodnota = hodnota
            .rucne = True
        Else
            gridA.Rows(radek).DefaultCellStyle.BackColor = Color.White
            .hodnota = 0
            .rucne = False
            gridA.Rows.Item(radek).Cells(sloupek).Value = "přepočítat"
        End If
    End With
End If

If radek = 16 Then 'dt
    With dt
```

```
    If hodnota <> "x" Then
        gridA.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
        .hodnota = hodnota
        .rucne = True
    Else
        gridA.Rows(radek).DefaultCellStyle.BackColor = Color.White
        .hodnota = 0
        .rucne = False
        gridA.Rows.Item(radek).Cells(sloupek).Value = "přepočítat"
    End If
End With
End If
```

```
If radek = 17 Then 'dz
    With dz
        If hodnota <> "x" Then
            gridA.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
            .hodnota = hodnota
            .rucne = True
        Else
            gridA.Rows(radek).DefaultCellStyle.BackColor = Color.White
            .hodnota = 0
            .rucne = False
            gridA.Rows.Item(radek).Cells(sloupek).Value = "přepočítat"
        End If
    End With
End If
```

```
buttRucne.PerformClick()
gridA.CurrentCell = gridA(4, 0)
```

End Sub

```
Private Sub NaplnA_Click(sender As Object, e As EventArgs) Handles naplnA.Click
    gridOdA.Rows.Clear()
    For x = 0 To gridA.Rows.Count - 1
        If x = 10 Then x = 11
        If x = 12 Then x = 13
        If x = 15 Then x = 16
        gridOdA.RowCount = gridOdA.RowCount + 1
        With gridOdA.Rows
            .Item(gridOdA.RowCount - 1).Cells(0).Value = gridA.Rows.Item(x).Cells(1).Value
            .Item(gridOdA.RowCount - 1).Cells(1).Value = gridA.Rows.Item(x).Cells(4).Value
        End With
    Next x
```

```
If gridOdB.RowCount <> 0 Then 'zvýraznění větší hodnoty
    If gridOdA.Rows.Item(12).Cells(1).Value > gridOdB.Rows.Item(12).Cells(1).Value Then
        gridOdA.Rows(12).DefaultCellStyle.BackColor = Color.Red
        gridOdB.Rows(12).DefaultCellStyle.BackColor = Color.White
    Else
        gridOdB.Rows(12).DefaultCellStyle.BackColor = Color.Red
        gridOdA.Rows(12).DefaultCellStyle.BackColor = Color.White
    End If
    If gridOdA.Rows.Item(12).Cells(1).Value = gridOdB.Rows.Item(12).Cells(1).Value Then
        gridOdA.Rows(12).DefaultCellStyle.BackColor = Color.LightBlue
        gridOdB.Rows(12).DefaultCellStyle.BackColor = Color.LightBlue
    End If
End If
End Sub
```

```
Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles Button1.Click
    gridOdB.Rows.Clear()
    For x = 0 To gridA.Rows.Count - 1
        If x = 10 Then x = 11
        If x = 12 Then x = 13
        If x = 15 Then x = 16
        gridOdB.RowCount = gridOdB.RowCount + 1
        With gridOdB.Rows
            .Item(gridOdB.RowCount - 1).Cells(0).Value = gridA.Rows.Item(x).Cells(1).Value
            .Item(gridOdB.RowCount - 1).Cells(1).Value = gridA.Rows.Item(x).Cells(4).Value
        End With
    Next x
```

```
If gridOdA.RowCount <> 0 Then 'zvýraznění větší hodnoty
    If gridOdB.Rows.Item(12).Cells(1).Value > gridOdA.Rows.Item(12).Cells(1).Value Then
        gridOdB.Rows(12).DefaultCellStyle.BackColor = Color.Red
        gridOdA.Rows(12).DefaultCellStyle.BackColor = Color.White
    Else
        gridOdA.Rows(12).DefaultCellStyle.BackColor = Color.Red
        gridOdB.Rows(12).DefaultCellStyle.BackColor = Color.White
    End If
    If gridOdA.Rows.Item(12).Cells(1).Value = gridOdB.Rows.Item(12).Cells(1).Value Then
        gridOdA.Rows(12).DefaultCellStyle.BackColor = Color.LightBlue
        gridOdB.Rows(12).DefaultCellStyle.BackColor = Color.LightBlue
    End If
End If
End Sub
```

```
Private Sub Button2_Click_1(sender As Object, e As EventArgs) Handles Button2.Click
```

```
        gridOdA.Rows.Clear()
        gridOdB.Rows.Clear()
    End Sub

    Private Sub TabPage1_Click(sender As Object, e As EventArgs) Handles TabPage1.Click

    End Sub

    Private Sub GridC_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles gridC.CellContentClick

    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        gridB_A.Rows.Clear()
        For x = 0 To gridB.Rows.Count - 1
            If x = 0 Then x = 7
            If x = 9 Then x = 12
            If x = 18 Then x = 19
            gridB_A.RowCount = gridB_A.RowCount + 1
            With gridB_A.Rows
                .Item(gridB_A.RowCount - 1).Cells(0).Value = gridB.Rows.Item(x).Cells(1).Value
                .Item(gridB_A.RowCount - 1).Cells(1).Value = gridB.Rows.Item(x).Cells(4).Value
            End With
        Next x
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        gridB_A.Rows.Clear()
        gridB_B.Rows.Clear()
    End Sub

    Private Sub GridB_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles gridB.CellContentClick

    End Sub

    Private Sub gridB_CellEndEdit(sender As Object, e As DataGridViewCellEventArgs) Handles gridB.CellEndEdit
        Dim radek = gridB.CurrentCell().RowIndex
        Dim sloupek = gridB.CurrentCell.ColumnIndex
        Dim hodnota = gridB.Rows.Item(radek).Cells(sloupek).Value
        hodnota = FiltrHodnot(hodnota)
        If radek = 14 Then 'Lp
            With lp
                If hodnota <> "x" Then
                    gridB.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
                    .hodnota = hodnota
                End If
            End With
        End If
    End Sub
```

```
.rucne = True
Else
    gridB.Rows(radek).DefaultCellStyle.BackColor = Color.White
    .hodnota = 0
    .rucne = False
    gridB.Rows.Item(radek).Cells(sloupek).Value = "přepočítat"
End If
End With
End If
If radek = 18 Then 'tgA
    With tgA
        If hodnota <> "x" Then
            gridB.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
            .hodnota = hodnota
            .rucne = True
        Else
            gridB.Rows(radek).DefaultCellStyle.BackColor = Color.White
            .hodnota = 0
            .rucne = False
            gridB.Rows.Item(radek).Cells(sloupek).Value = "přepočítat"
        End If
    End With
End If
If radek = 21 Then 'Lzv
    With lzv
        If hodnota <> "x" Then
            gridB.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
            .hodnota = hodnota
            .rucne = True
        Else
            gridB.Rows(radek).DefaultCellStyle.BackColor = Color.White
            .hodnota = 0
            .rucne = False
            gridB.Rows.Item(radek).Cells(sloupek).Value = "přepočítat"
        End If
    End With
End If
If radek = 24 Then 'LD
    With lD
        If hodnota <> "x" Then
            gridB.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
            .hodnota = hodnota
            .rucne = True
```

```
Else
    gridB.Rows(radek).DefaultCellStyle.BackColor = Color.White
    .hodnota = 0
    .rucne = False
    gridB.Rows.Item(radek).Cells(sloupek).Value = "přepočítat"
End If
End With
End If

buttRucne.PerformClick()
gridB.CurrentCell = gridB(4, 0)
End Sub

Private Sub GridB_B_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles gridB_B.CellContentClick

End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    gridB_B.RowCount = gridB_B.RowCount + 1
    gridB_B.CurrentCell = gridB_B(0, gridB_B.RowCount - 1)
    gridB_B.BeginEdit(0)
End Sub

Private Sub gridB_B_CellEndEdit(sender As Object, e As DataGridViewCellEventArgs) Handles gridB_B.CellEndEdit
    Dim pocet_radek, radek As Int16
    Dim rychlost, vzdalenost, cas, suma_cas As Double
    radek = gridB_B.CurrentCell.RowIndex
    pocet_radek = gridB_B.Rows.Count() - 1
    rychlost = gridB_B.Rows.Item(radek).Cells(0).Value
    vzdalenost = gridB_B.Rows.Item(radek).Cells(1).Value
    cas = Math.Round(3.6 * vzdalenost / rychlost, 2)
    gridB_B.Rows.Item(radek).Cells(2).Value = cas
    For x = 0 To pocet_radek
        gridB_B.Rows(radek).DefaultCellStyle.BackColor = Color.White
    Next
    For x = 0 To pocet_radek
        If suma_cas + gridB_B.Rows.Item(x).Cells(2).Value >= tL.hodnota Then
            gridB_B.Rows(radek).DefaultCellStyle.BackColor = Color.LightBlue
            gridB_B.Rows.Item(radek).Cells(2).Value = tL.hodnota - suma_cas
            gridB_B.Rows.Item(radek).Cells(1).Value = Math.Round(gridB_B.Rows.Item(radek).Cells(0).Value * (tL.hodnota - suma_cas) / 3.6, 2)
        End If
        suma_cas = gridB_B.Rows.Item(x).Cells(2).Value + suma_cas
    Next
End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
```

```
If gridB_B.RowCount > 0 Then gridB_B.RowCount = gridB_B.RowCount - 1
End Sub

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
    Dim suma As Double
    For x = 0 To gridB_B.Rows.Count() - 1
        suma = gridB_B.Rows.Item(x).Cells(1).Value + suma
    Next x
    lp.hodnota = Math.Ceiling(suma)
    gridB.Rows.Item(14).Cells(4).Value = lp.hodnota
    lp.rucne = True
    gridB.Rows(14).DefaultCellStyle.BackColor = Color.LightBlue
    buttRucne.PerformClick()
End Sub

Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
    FileOpen(1, "prejezdy.txt", OpenMode.Output, OpenAccess.Write, OpenShare.Default)
    WriteLine(1, nazev_prejezdu)
    WriteLine(1, kmEvidencni.hodnota)
    WriteLine(1, kmEvidencni.rucne)
    WriteLine(1, kmSkutecna.hodnota)
    WriteLine(1, kmSkutecna.rucne)
    WriteLine(1, kmOPL.hodnota)
    WriteLine(1, kmOPL.rucne)
    WriteLine(1, kmOPS.hodnota)
    WriteLine(1, kmOPS.rucne)
    WriteLine(1, zavory)
    WriteLine(1, zavory_cele)
    'A.1.1 VÝCHOZÍ DÉLKY V METRECH
    For x = 1 To 12
        WriteLine(1, d(x).hodnota)
        WriteLine(1, d(x).rucne)
    Next x
    WriteLine(1, dk.hodnota)
    WriteLine(1, dk.rucne)
    WriteLine(1, dn.hodnota)
    WriteLine(1, dn.rucne)
    WriteLine(1, ds.hodnota)
    WriteLine(1, ds.rucne)
    WriteLine(1, sj.hodnota)
    WriteLine(1, sj.rucne)
    WriteLine(1, ss.hodnota)
    WriteLine(1, ss.rucne)
    'A.1.2 VÝSLEDNÉ DÉLKY V METRECH
    WriteLine(1, dp.hodnota)
```

```
WriteLine(1, dp.rucne)
WriteLine(1, dt.hodnota)
WriteLine(1, dt.rucne)
WriteLine(1, dz.hodnota)
WriteLine(1, dz.rucne)
WriteLine(1, sp.hodnota)
WriteLine(1, sp.rucne)
'A.1.3 ÚHLY VE STUPNÍCH
WriteLine(1, alfa.hodnota)
WriteLine(1, alfa.rucne)
WriteLine(1, beta1.hodnota)
WriteLine(1, beta1.rucne)
WriteLine(1, beta2.hodnota)
WriteLine(1, beta2.rucne)
'B.1
WriteLine(1, a.hodnota)
WriteLine(1, a.rucne)
'B.1.1 VÝCHOZÍ DOBY V SEKUNDÁCH
WriteLine(1, tb1.hodnota)
WriteLine(1, tb1.rucne)
WriteLine(1, tb2.hodnota)
WriteLine(1, tb2.rucne)
WriteLine(1, td.hodnota)
WriteLine(1, td.rucne)
WriteLine(1, te.hodnota)
WriteLine(1, te.rucne)
WriteLine(1, tf.hodnota)
WriteLine(1, tf.rucne)
WriteLine(1, tgA.hodnota)
WriteLine(1, tgA.rucne)
WriteLine(1, tg1.hodnota)
WriteLine(1, tg1.rucne)
WriteLine(1, tg2.hodnota)
WriteLine(1, tg2.rucne)
WriteLine(1, tk.hodnota)
WriteLine(1, tk.rucne)
WriteLine(1, t0.hodnota)
WriteLine(1, t0.rucne)
WriteLine(1, tr.hodnota)
WriteLine(1, tr.rucne)
WriteLine(1, tt.hodnota)
WriteLine(1, tt.rucne)
WriteLine(1, tu.hodnota)
WriteLine(1, tu.rucne)
WriteLine(1, tu1.hodnota)
```

```
WriteLine(1, tu1.rucne)
WriteLine(1, tu2.hodnota)
WriteLine(1, tu2.rucne)
WriteLine(1, tx.hodnota)
WriteLine(1, tx.rucne)
'B.1.2 VÝSLEDNÉ DOBY V SEKUNDÁCH
WriteLine(1, tA.hodnota)
WriteLine(1, tA.rucne)
WriteLine(1, tL.hodnota)
WriteLine(1, tL.rucne)
WriteLine(1, tM.hodnota)
WriteLine(1, tM.rucne)
WriteLine(1, tM1.hodnota)
WriteLine(1, tM1.rucne)
WriteLine(1, tM2.hodnota)
WriteLine(1, tM2.rucne)
WriteLine(1, tn.hodnota)
WriteLine(1, tn.rucne)
WriteLine(1, tV.hodnota)
WriteLine(1, tV.rucne)
WriteLine(1, tZ.hodnota)
WriteLine(1, tZ.rucne)
WriteLine(1, tZZ.hodnota)
WriteLine(1, tZZ.rucne)
WriteLine(1, tZV.hodnota)
WriteLine(1, tZV.rucne)
'B.1.3 DÉLKY V METRECH
WriteLine(1, dNN.hodnota)
WriteLine(1, dNN.rucne)
WriteLine(1, dv.hodnota)
WriteLine(1, dv.rucne)
WriteLine(1, lD.hodnota)
WriteLine(1, lD.rucne)
WriteLine(1, lp.hodnota)
WriteLine(1, lp.rucne)
WriteLine(1, lv.hodnota)
WriteLine(1, lv.rucne)
WriteLine(1, lzv.hodnota)
WriteLine(1, lzv.rucne)
WriteLine(1, lz.hodnota)
WriteLine(1, lz.rucne)
'B.1.4 RYCHLOSTI V KM/H
WriteLine(1, vs.hodnota)
WriteLine(1, vs.rucne)
WriteLine(1, vt1(0).hodnota)
```

```
WriteLine(1, vtl(0).rucne)
WriteLine(1, vts(0).hodnota)
WriteLine(1, vts(0).rucne)
WriteLine(1, vv.hodnota)
WriteLine(1, vv.rucne)
'C.1 VÝZNAM ZNAČENÍ
WriteLine(1, lpp.hodnota)
WriteLine(1, lpp.rucne)
WriteLine(1, lsv.hodnota)
WriteLine(1, lsv.rucne)
WriteLine(1, trp.hodnota)
WriteLine(1, trp.rucne)
'moje
WriteLine(1, zav_rovnob)
FileClose()
MsgBox("Data uložena.", vbInformation, "Info")
```

```
End Sub
```

```
Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
    If My.Computer.FileSystem.FileExists("prejezdy.txt") = False Then
        MsgBox("Soubor prejezdy.txt nenalezen.", vbInformation, "Chyba")
        Exit Sub
    End If
    FileOpen(1, "prejezdy.txt", OpenMode.Input, OpenAccess.Read, OpenShare.Default)
    Input(1, nazev_prejezdu)
    Input(1, kmEvidencni.hodnota)
    Input(1, kmEvidencni.rucne)
    Input(1, kmSkutecna.hodnota)
    Input(1, kmSkutecna.rucne)
    Input(1, kmOPL.hodnota)
    Input(1, kmOPL.rucne)
    Input(1, kmOPS.hodnota)
    Input(1, kmOPS.rucne)
    Input(1, zavory)
    Input(1, zavory_cele)
    'A.1.1 VÝCHOZÍ DÉLKY V METRECH
    For x = 1 To 12
        Input(1, d(x).hodnota)
        Input(1, d(x).rucne)
    Next x
    Input(1, dk.hodnota)
    Input(1, dk.rucne)
    Input(1, dn.hodnota)
    Input(1, dn.rucne)
    Input(1, ds.hodnota)
```

```
Input(1, ds.rucne)
Input(1, sj.hodnota)
Input(1, sj.rucne)
Input(1, ss.hodnota)
Input(1, ss.rucne)
'A.1.2 VÝSLEDNÉ DÉLKY V METRECH
Input(1, dp.hodnota)
Input(1, dp.rucne)
Input(1, dt.hodnota)
Input(1, dt.rucne)
Input(1, dz.hodnota)
Input(1, dz.rucne)
Input(1, sp.hodnota)
Input(1, sp.rucne)
'A.1.3 ÚHLÝ VE STUPNÍCH
Input(1, alfa.hodnota)
Input(1, alfa.rucne)
Input(1, beta1.hodnota)
Input(1, beta1.rucne)
Input(1, beta2.hodnota)
Input(1, beta2.rucne)
'B.1
Input(1, a.hodnota)
Input(1, a.rucne)
'B.1.1 VÝCHOZÍ DOBY V SEKUNDÁCH
Input(1, tb1.hodnota)
Input(1, tb1.rucne)
Input(1, tb2.hodnota)
Input(1, tb2.rucne)
Input(1, td.hodnota)
Input(1, td.rucne)
Input(1, te.hodnota)
Input(1, te.rucne)
Input(1, tf.hodnota)
Input(1, tf.rucne)
Input(1, tgA.hodnota)
Input(1, tgA.rucne)
Input(1, tg1.hodnota)
Input(1, tg1.rucne)
Input(1, tg2.hodnota)
Input(1, tg2.rucne)
Input(1, tk.hodnota)
Input(1, tk.rucne)
Input(1, t0.hodnota)
Input(1, t0.rucne)
```

```
Input(1, tr.hodnota)
Input(1, tr.rucne)
Input(1, tt.hodnota)
Input(1, tt.rucne)
Input(1, tu.hodnota)
Input(1, tu.rucne)
Input(1, tu1.hodnota)
Input(1, tu1.rucne)
Input(1, tu2.hodnota)
Input(1, tu2.rucne)
Input(1, tx.hodnota)
Input(1, tx.rucne)
'B.1.2 VÝSLEDNÉ DOBY V SEKUNDÁCH
Input(1, tA.hodnota)
Input(1, tA.rucne)
Input(1, tL.hodnota)
Input(1, tL.rucne)
Input(1, tM.hodnota)
Input(1, tM.rucne)
Input(1, tM1.hodnota)
Input(1, tM1.rucne)
Input(1, tM2.hodnota)
Input(1, tM2.rucne)
Input(1, tn.hodnota)
Input(1, tn.rucne)
Input(1, tV.hodnota)
Input(1, tV.rucne)
Input(1, tZ.hodnota)
Input(1, tZ.rucne)
Input(1, tZZ.hodnota)
Input(1, tZZ.rucne)
Input(1, tZV.hodnota)
Input(1, tZV.rucne)
'B.1.3 DÉLKY V METRECH
Input(1, dNN.hodnota)
Input(1, dNN.rucne)
Input(1, dv.hodnota)
Input(1, dv.rucne)
Input(1, lD.hodnota)
Input(1, lD.rucne)
Input(1, lp.hodnota)
Input(1, lp.rucne)
Input(1, lv.hodnota)
Input(1, lv.rucne)
Input(1, lzv.hodnota)
```

```
Input(1, lzv.rucne)
Input(1, lz.hodnota)
Input(1, lz.rucne)
'B.1.4 RYCHLOSTI V KM/H
Input(1, vs.hodnota)
Input(1, vs.rucne)
Input(1, vt1(0).hodnota)
Input(1, vt1(0).rucne)
Input(1, vts(0).hodnota)
Input(1, vts(0).rucne)
Input(1, vv.hodnota)
Input(1, vv.rucne)
'C.1 VÝZNAM ZNAČENÍ
Input(1, lpp.hodnota)
Input(1, lpp.rucne)
Input(1, lsv.hodnota)
Input(1, lsv.rucne)
Input(1, trp.hodnota)
Input(1, trp.rucne)
'moje
Input(1, zav_rovnob)
FileClose()
buttRucne.PerformClick()
```

```
End Sub
```

```
Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click
```

```
gridB_A.RowCount = 5
gridB_A.Rows.Item(0).Cells(1).Value = "Otevírám Excel..."
```

```
Dim MyExcel = CreateObject("Excel.Application")
```

```
MyExcel.Workbooks.Add()
MyExcel.Range("A1").Activate()
MyExcel.ActiveCell.Columns.ColumnWidth = 50
gridB_A.Rows.Item(1).Cells(1).Value = "Výpočet dat..."
```

```
' ===== GENEROVÁNÍ TABULKY A VÝPOČTU V EXCELU =====
```

```
MyExcel.ActiveSheet.Range("A1").Value = "Délka přibližovacího úseku"
MyExcel.ActiveSheet.Range("B1").Value = "Lp"
MyExcel.ActiveSheet.Range("C1").Value = lp.hodnota
MyExcel.ActiveSheet.Range("D1").Value = "m"
```

```
If lp.rucne = True Then
```

```
MyExcel.ActiveSheet.Range("E1").Value = "Lp zadána ručně!!!"
```

```
Else
```

```
MyExcel.ActiveSheet.Range("E1").Value = ""
```

```
End If
```

```
MyExcel.ActiveSheet.Range("A2").Value = "Přibližovací doba"
MyExcel.ActiveSheet.Range("B2").Value = "tL"
MyExcel.ActiveSheet.Range("C2").Value = tL.hodnota
MyExcel.ActiveSheet.Range("D2").Value = "s"
MyExcel.ActiveSheet.Range("E2").Value = "=" & lp.hodnota.ToString.Replace(",", ".") & "*3.6/" & vt1(0).hodnota.ToString.Replace(",", ".")
```

```
MyExcel.ActiveSheet.Range("A3").Value = "Vyklizovací doba"
MyExcel.ActiveSheet.Range("B3").Value = "tv"
MyExcel.ActiveSheet.Range("C3").Value = tv.hodnota
MyExcel.ActiveSheet.Range("D3").Value = "s"
MyExcel.ActiveSheet.Range("E3").Value = "=E2-" & tr.hodnota & "-" & tb1.hodnota & "-" & tb2.hodnota
```

```
MyExcel.ActiveSheet.Range("A4").Value = "Délka směrodatná pro výpočet vykl. doby"
MyExcel.ActiveSheet.Range("B4").Value = "dt"
MyExcel.ActiveSheet.Range("C4").Value = dt.hodnota
MyExcel.ActiveSheet.Range("D4").Value = "m"
MyExcel.ActiveSheet.Range("E4").Value = "=E3*" & vs.hodnota.ToString.Replace(",", ".") & "/3.6"
```

```
MyExcel.ActiveSheet.Range("A5").Value = "Délka pásma přejezdu"
MyExcel.ActiveSheet.Range("B5").Value = "dp"
MyExcel.ActiveSheet.Range("C5").Value = dp.hodnota
MyExcel.ActiveSheet.Range("D5").Value = "m"
MyExcel.ActiveSheet.Range("E5").Value = "=E4-" & ds.hodnota.ToString.Replace(",", ".")
```

```
' ===== GENEROVÁNÍ TABULKY V PROGRAMU =====
```

```
' hodnota Lp
```

```
gridB_A.Rows.Item(0).Cells(0).Value = MyExcel.ActiveSheet.Range("B1").Value
gridB_A.Rows.Item(0).Cells(1).Value = MyExcel.ActiveSheet.Range("C1").Value
gridB_A.Rows.Item(0).Cells(2).Value = MyExcel.ActiveSheet.Range("E1").Value
If lp.rucne = True Then gridB_A.Rows.Item(0).Cells(3).Value = "neplatné výsledky"
```

```
' hodnota tL
```

```
gridB_A.Rows.Item(1).Cells(0).Value = MyExcel.ActiveSheet.Range("B2").Value
gridB_A.Rows.Item(1).Cells(1).Value = MyExcel.ActiveSheet.Range("C2").Value
gridB_A.Rows.Item(1).Cells(2).Value = MyExcel.ActiveSheet.Range("E2").Value
gridB_A.Rows.Item(1).Cells(3).Value = Math.Round(MyExcel.ActiveSheet.Range("E2").Value - tL.hodnota, 3)
If Math.Abs(gridB_A.Rows.Item(1).Cells(3).Value) < 1 Then 'stanovení povolené odchylky a výsledek komparace
    gridB_A.Rows.Item(1).Cells(3).Style.BackColor = Color.GreenYellow
```

```
Else
```

```
    gridB_A.Rows.Item(1).Cells(3).Style.BackColor = Color.Red
```

```
End If
```

```
' hodnota tv
```

```
gridB_A.Rows.Item(2).Cells(0).Value = MyExcel.ActiveSheet.Range("B3").Value
```

```
gridB_A.Rows.Item(2).Cells(1).Value = MyExcel.ActiveSheet.Range("C3").Value
gridB_A.Rows.Item(2).Cells(2).Value = MyExcel.ActiveSheet.Range("E3").Value
gridB_A.Rows.Item(2).Cells(3).Value = Math.Round(MyExcel.ActiveSheet.Range("E3").Value - tv.hodnota, 3)
If Math.Abs(gridB_A.Rows.Item(2).Cells(3).Value) < 3 Then 'stanovení povolené odchylky a výsledek komparace
    gridB_A.Rows.Item(2).Cells(3).Style.BackColor = Color.GreenYellow
Else
    gridB_A.Rows.Item(2).Cells(3).Style.BackColor = Color.Red
End If

' hodnota dt
gridB_A.Rows.Item(3).Cells(0).Value = MyExcel.ActiveSheet.Range("B4").Value
gridB_A.Rows.Item(3).Cells(1).Value = MyExcel.ActiveSheet.Range("C4").Value
gridB_A.Rows.Item(3).Cells(2).Value = MyExcel.ActiveSheet.Range("E4").Value
gridB_A.Rows.Item(3).Cells(3).Value = Math.Round(MyExcel.ActiveSheet.Range("E4").Value - dt.hodnota, 3)
If Math.Abs(gridB_A.Rows.Item(3).Cells(3).Value) < 5 * vs.hodnota / 3.6 Then 'stanovení povolené odchylky a výsledek komparace
    gridB_A.Rows.Item(3).Cells(3).Style.BackColor = Color.GreenYellow
Else
    gridB_A.Rows.Item(3).Cells(3).Style.BackColor = Color.Red
End If

' hodnota dp
gridB_A.Rows.Item(4).Cells(0).Value = MyExcel.ActiveSheet.Range("B5").Value
gridB_A.Rows.Item(4).Cells(1).Value = MyExcel.ActiveSheet.Range("C5").Value
gridB_A.Rows.Item(4).Cells(2).Value = MyExcel.ActiveSheet.Range("E5").Value
gridB_A.Rows.Item(4).Cells(3).Value = Math.Round(MyExcel.ActiveSheet.Range("E5").Value - dp.hodnota, 3)
If Math.Abs(gridB_A.Rows.Item(4).Cells(3).Value) < 1 Then 'stanovení povolené odchylky a výsledek komparace
    gridB_A.Rows.Item(4).Cells(3).Style.BackColor = Color.GreenYellow
Else
    gridB_A.Rows.Item(4).Cells(3).Style.BackColor = Color.Red
End If

' ===== UKONČENÍ EXCELU =====
If CheckBox1.Checked = True Then
    MyExcel.WindowState = Excel.XlWindowState.xlMinimized
    MyExcel.Visible = True
Else
    MyExcel.ActiveWorkbook.Close(SaveChanges:=False) 'zavře sešit bez dotazu na uložení

    GC.Collect()
    GC.WaitForPendingFinalizers()
    ' Marshal.FinalReleaseComObject(MyExcel.Worksheets)
    Marshal.FinalReleaseComObject(MyExcel.Workbooks)
    MyExcel.Quit()
    Marshal.FinalReleaseComObject(MyExcel)
End If
```

```
MyExcel = Nothing
End Sub

Private Sub GridA_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles gridA.CellContentClick

End Sub

Private Sub ButtPribl_Click(sender As Object, e As EventArgs)

End Sub

Private Sub GridB_A_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles gridB_A.CellContentClick

End Sub

Private Sub gridB_A_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles gridB_A.CellClick
    Dim zprava, promenna As String
    zprava = ""
    promenna = gridB_A.Item(e.ColumnIndex, e.RowIndex).Value
    promenna = LCase(promenna)
    Select Case promenna
        Case "lp"
            zprava = "Lp - " & lp.popis
        Case "tl"
            zprava = "tL - " & tL.popis
        Case "tv"
            zprava = "tV - " & tV.popis
        Case "dt"
            zprava = "dt - " & dt.popis
        Case "dp"
            zprava = "dp - " & dp.popis
    End Select
    If e.ColumnIndex = 0 Then
        MsgBox(zprava, vbInformation, "Informace o proměnné")
    End If
End Sub

Private Sub gridB_MouseClick(sender As Object, e As MouseEventArgs) Handles gridB.MouseClick
    If e.Button = MouseButton.Right Then
        'MsgBox(gridB.Rows.Item(3).Cells(3).Value)
        'MsgBox(gridB.CurrentRow())
    End If
End Sub

Private Sub Button11_Click(sender As Object, e As EventArgs) Handles Button11.Click
```

```
Dim SaveDLG As New SaveFileDialog
SaveDLG.Filter = "Textový soubor (*.txt)|*.txt"
SaveDLG.Title = "Exportovat výstup"
SaveDLG.InitialDirectory = "."
DialogResult = SaveDLG.ShowDialog
If DialogResult = Windows.Forms.DialogResult.OK Then
    FileOpen(1, SaveDLG.FileName, OpenMode.Output, OpenAccess.Write, OpenShare.Default)
    PrintLine(1, nazev_prejezdu)
    For x = 0 To gridC.Rows.Count - 1
        PrintLine(1, gridC.Item(0, x).Value & " " & gridC.Item(1, x).Value & " " & gridC.Item(2, x).Value & " " & gridC.Item(3, x).Value & " " & gridC.Item(4, x).Value)
    Next
    FileClose()
    MsgBox("Export dat uložený do " & SaveDLG.FileName & ".", vbInformation, "Export")
ElseIf DialogResult = Windows.Forms.DialogResult.Cancel Then
End If

End Sub
End Class
```