**Министерство образования и науки Украины**

**Национальный горный университет**

Институт электроэнергетики

Факультет информационных технологий

**Кафедра ПЗКС**

Лабораторная работа № 3

**по дисциплине**

“Професійна практика програмної інженерії”

**Название** **работы**: Управляющие операторы

**Тема**: Использование управляющих операторов в решении логических и математических задач.

**Цель**: Использование интерпретаторов Ruby для написания и отладки программ.

**Задание**: : Задание для программирования берется из файла с заданиями к Лабораторной работе № 3.

**ВЫПОЛНИЛ:**

студент группы Пиит-15-1

Сенькин Д. Р.

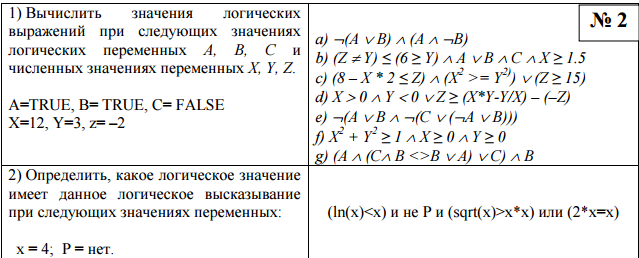
**ПРОВЕРИЛ:**

Коротенко Л.М.

**Днепр**

**2017**

**Условие задачи:**

**1.**

*#****V2  
BoolA***, ***BoolB***, ***BoolC*** = **true**, **true**, **false  
*RealX***, ***RealY***, ***RealZ*** = 12.0, 3.0, -2.0  
  
puts **"A =** #{***BoolA***}**, B =** #{***BoolB***}**, C =** #{***BoolC***}**"**puts **"X =** #{***RealX***}**, Y =** #{***RealY***}**, Z =** #{***RealZ***}**\n\n"**puts **"Task 1"***#****a****)  
res* = !(***BoolA*** || ***BoolB***) && (***BoolA*** || !***BoolB***)  
puts **"a) ¬(A ∨ B) ∧ (A ∧ ¬B) =** #{*res*}**"***#****b****)  
res* = ((***RealZ*** != ***RealY***) ? 1 : 0) <= ((6 >= ***RealY***) ? 1 : 0) && ***BoolA*** || ***BoolB*** && ***BoolC*** && ***RealX*** >= 1.5  
puts **"b) (Z ≠ Y) ⩽ (6 ⩾ Y) ∧ A ∨ B ∧ C ∧ X ⩾ 1.5 =** #{*res*}**"***#****c****)  
res* = (8 - ***RealX*** \* 2 <= ***RealZ***) && (***RealX*** \*\* 2 <= ***RealY*** \*\* 2) || (***RealZ*** >= 15)  
puts **"c) (8 - X \* 2 ⩽ Z) ∧ (X^2 ⩾ Y^2) ∨ (Z ⩾ 15) =** #{*res*}**"***#****d****)  
res* = ***RealX*** > 0 && ***RealY*** < 0 || ***RealZ*** >= (***RealX***\****RealY*** - ***RealY*** / ***RealX***) - (-***RealZ***)  
puts **"d) X > 0 ∧ Y < 0 ∨ Z ⩾ (X\*Y - Y/X) - (-Z) =** #{*res*}**"***#****e****)  
res* = !(***BoolA*** || ***BoolB*** && !(***BoolC*** || (!***BoolA*** || ***BoolB***)))  
puts **"e) ¬(A ∨ B ∧ ¬(C ∨ (¬A ∨ B)) =** #{*res*}**"***#****f****)  
res* = ***RealX*** \*\* 2 + ***RealY*** \*\* 2 >= 1 && ***RealX*** >= 0 && ***RealY*** >= 0  
puts **"f) X^2 + Y^2 ⩾ 1 ∧ X ⩾ 0 ∧ Y ⩾ 0 =** #{*res*}**"***#****g****)  
res* = (***BoolA*** && (***BoolC*** && ***BoolB*** != ***BoolB*** || ***BoolA***) || ***BoolC***) && ***BoolB***puts **"g) A ∧ (C ∧ B <> B ∨ A) ∨ C) ∧ B =** #{*res*}**"**puts **"\nTask 2"  
*IntX*** = 4  
***BoolP*** = **false**puts **"x =** #{***IntX***}**; P =** #{***BoolP***}**"***res* = ***Math***.log(***IntX***) < ***IntX*** && !***BoolP*** && (***Math***.sqrt(***IntX***) > ***IntX***\****IntX***) || (2\****IntX*** == ***IntX***)  
puts **"ln(x) < x и не P и (sqrt(x) > x\*x) или (2\*x=x) =** #{*res*}**"**

**2.**

print **"Enter x = "  
*RealX*** = gets.to\_r  
  
**if** (-4 < ***RealX*** && ***RealX*** <= 0)  
 *realY* = ((***RealX*** - 2).abs / (***RealX*** \*\* 2 \* ***Math***.cos(***RealX***))) \*\* (1.0 / 7)  
**elsif** (0 < ***RealX*** && ***RealX*** <= 12)  
 *realY* = 1.0 / (***Math***.tan(***RealX*** + 1.0 / ***Math***.exp(***RealX***)) / ***Math***.sin(***RealX***) \*\* 2) \*\* (7.0 / 2)  
**else** *realY* = 1.0 / (1 + ***RealX*** / (1 + ***RealX*** / (1 + ***RealX***)))  
**end**puts **"y =** #{*realY*}**"***realY* = **case  
 when** -4 < ***RealX*** && ***RealX*** <= 0  
 ((***RealX*** - 2).abs / (***RealX*** \*\* 2 \* ***Math***.cos(***RealX***))) \*\* (1.0 / 7)  
 **when** 0 < ***RealX*** && ***RealX*** <= 12  
 1.0 / (***Math***.tan(***RealX*** + 1.0 / ***Math***.exp(***RealX***)) / ***Math***.sin(***RealX***) \*\* 2) \*\* (7.0 / 2)  
 **else** 1.0 / (1 + ***RealX*** / (1 + ***RealX*** / (1 + ***RealX***)))  
 **end**puts **"y =** #{*realY*}**"**

**3.**

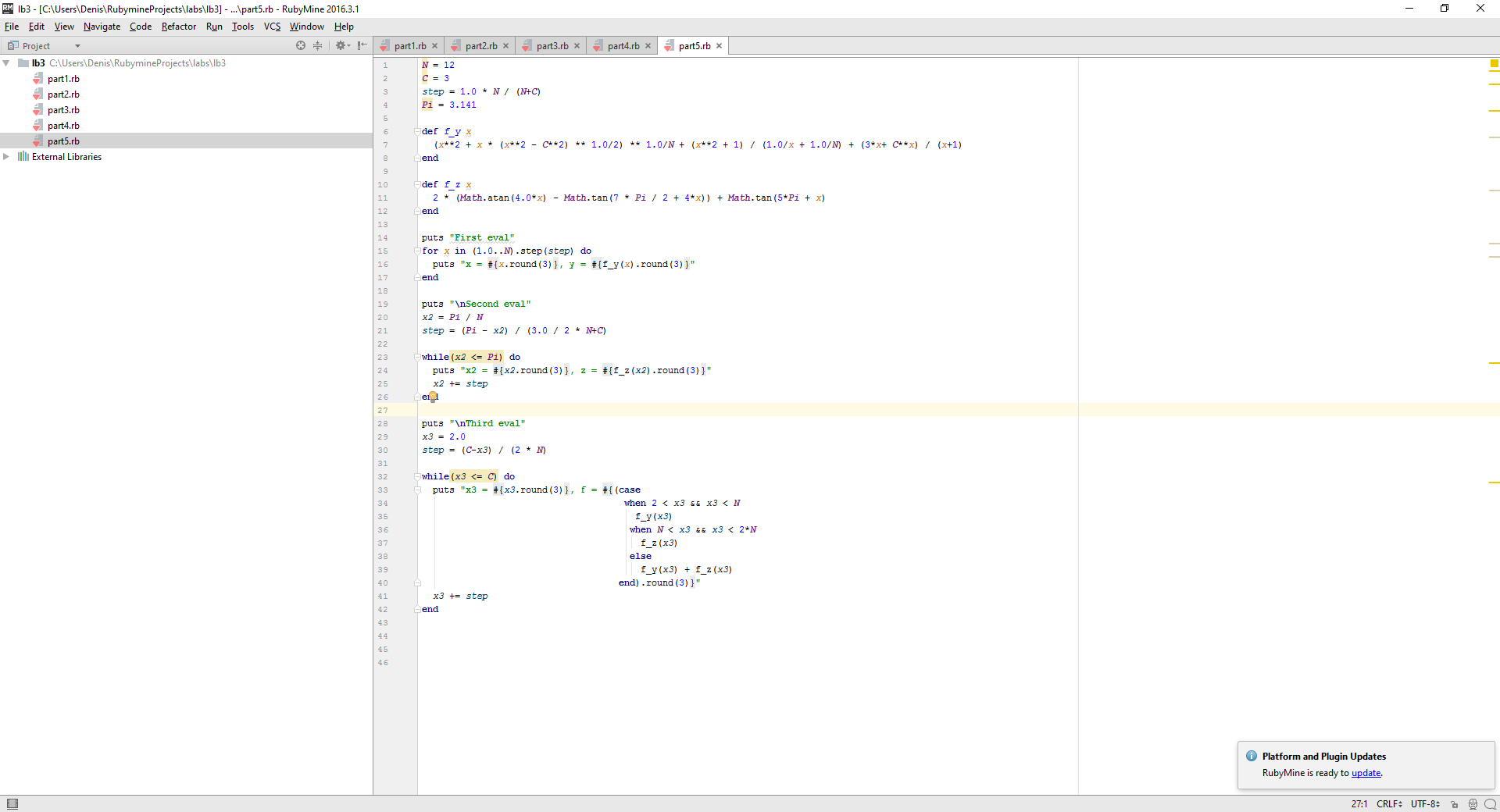
print **"Enter n = "  
*IntN*** = gets.to\_i  
  
abort **"n must be non negative value" if *IntN*** < 0  
*#4)  
res* = 0  
*sum* = 0  
(1..***IntN***).each **do** |*i*|  
 *sum* += ***Math***.sin(*i*)  
 *res* += 1.0 / *sum***end**puts **"result =** #{*res*}**"***#5)  
res* = 0  
(1..***IntN***).each **do** *res* = ***Math***.sqrt(*res* + 2)  
**end**puts **"result =** #{*res*}**"**

**4.**

***Eps*** = 0.00001  
*res* = 0  
*n* = 2  
  
**def** *factorial n  
 n* > 1.0 ? *n* \* factorial(*n* - 1) : 1.0  
**end  
  
def** *firstEval n* (factorial(2 \* *n* - 1) / (2.0\* factorial(*n* + 1)) \*\* (*n* \* (*n* + 1)))  
**end***tmp* = firstEval(*n*)  
**while**(*tmp*.abs >= ***Eps***) **do** *res* += *tmp  
 n* += 1  
 *tmp* = firstEval(*n*)  
**end**puts **"result 1 =** #{*res*.round(5)}**"  
  
def** *pidiv4  
 res* = 0.0  
 *n* = 1.0  
 *tmp* = 1.0 / *n  
 neg* = 1  
 **while**(*tmp*.abs >= ***Eps***) **do** *res* += *tmp  
 n* += 2  
 *neg* = -*neg  
 tmp* = *neg* \* 1.0 / *n* **end** *res***end**puts **"reslut 2:\n\tlang:** #{(***Math***::***PI*** / 4).round(5)}**, my:** #{pidiv4.round(5)}**"  
  
def** *thirdEval n* factorial(3\**n* - 1) \* factorial(2\**n*-1) / (factorial(4\**n*) \* 3 \*\* (2\**n*) \* factorial(2\**n*))  
**end***res* = 0  
*n* = 1  
*tmp* = thirdEval(*n*)  
**while**(*tmp*.abs >= ***Eps***) **do** *res* += *tmp  
 n* += 1  
 *tmp* = thirdEval(*n*)  
**end**puts **"result 3 =** #{*res*.round(5)}**"**

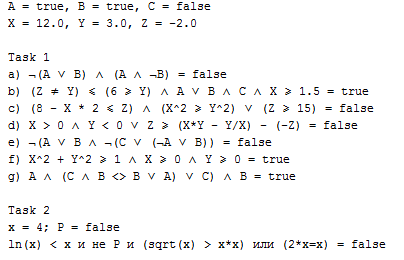
**5.**

***N*** = 12  
***C*** = 3  
*step* = 1.0 \* ***N*** / (***N***+***C***)  
***Pi*** = 3.141  
  
**def** *f\_y x* (*x*\*\*2 + *x* \* (*x*\*\*2 - ***C***\*\*2) \*\* 1.0/2) \*\* 1.0/***N*** + (*x*\*\*2 + 1) / (1.0/*x* + 1.0/***N***) + (3\**x*+ ***C***\*\**x*) / (*x*+1)  
**end  
  
def** *f\_z x* 2 \* (***Math***.atan(4.0\**x*) - ***Math***.tan(7 \* ***Pi*** / 2 + 4\**x*)) + ***Math***.tan(5\****Pi*** + *x*)  
**end**puts **"First eval"  
for** *x* **in** (1.0..***N***).step(*step*) **do** puts **"x =** #{*x*.round(3)}**, y =** #{f\_y(*x*).round(3)}**"  
end**puts **"\nSecond eval"***x2* = ***Pi*** / ***N****step* = (***Pi*** - *x2*) / (3.0 / 2 \* ***N***+***C***)  
  
**while**(*x2* <= ***Pi***) **do** puts **"x2 =** #{*x2*.round(3)}**, z =** #{f\_z(*x2*).round(3)}**"** *x2* += *step***end**puts **"\nThird eval"***x3* = 2.0  
*step* = (***C***-*x3*) / (2 \* ***N***)  
  
**while**(*x3* <= ***C***) **do** puts **"x3 =** #{*x3*.round(3)}**, f =** #{(**case  
 when** 2 < *x3* && *x3* < ***N*** f\_y(*x3*)  
 **when *N*** < *x3* && *x3* < 2\****N*** f\_z(*x3*)  
 **else** f\_y(*x3*) + f\_z(*x3*)  
 **end**).round(3)}**"** *x3* += *step***end**



**5. Результат работы программы**

**1.**



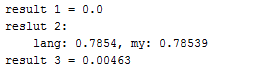
**2.**

****

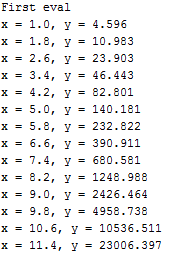
**3.**

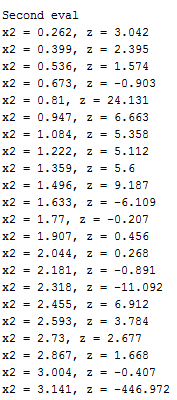


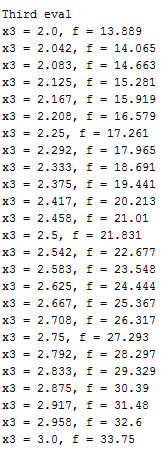
**4.**

****

**5.**

****

****

****