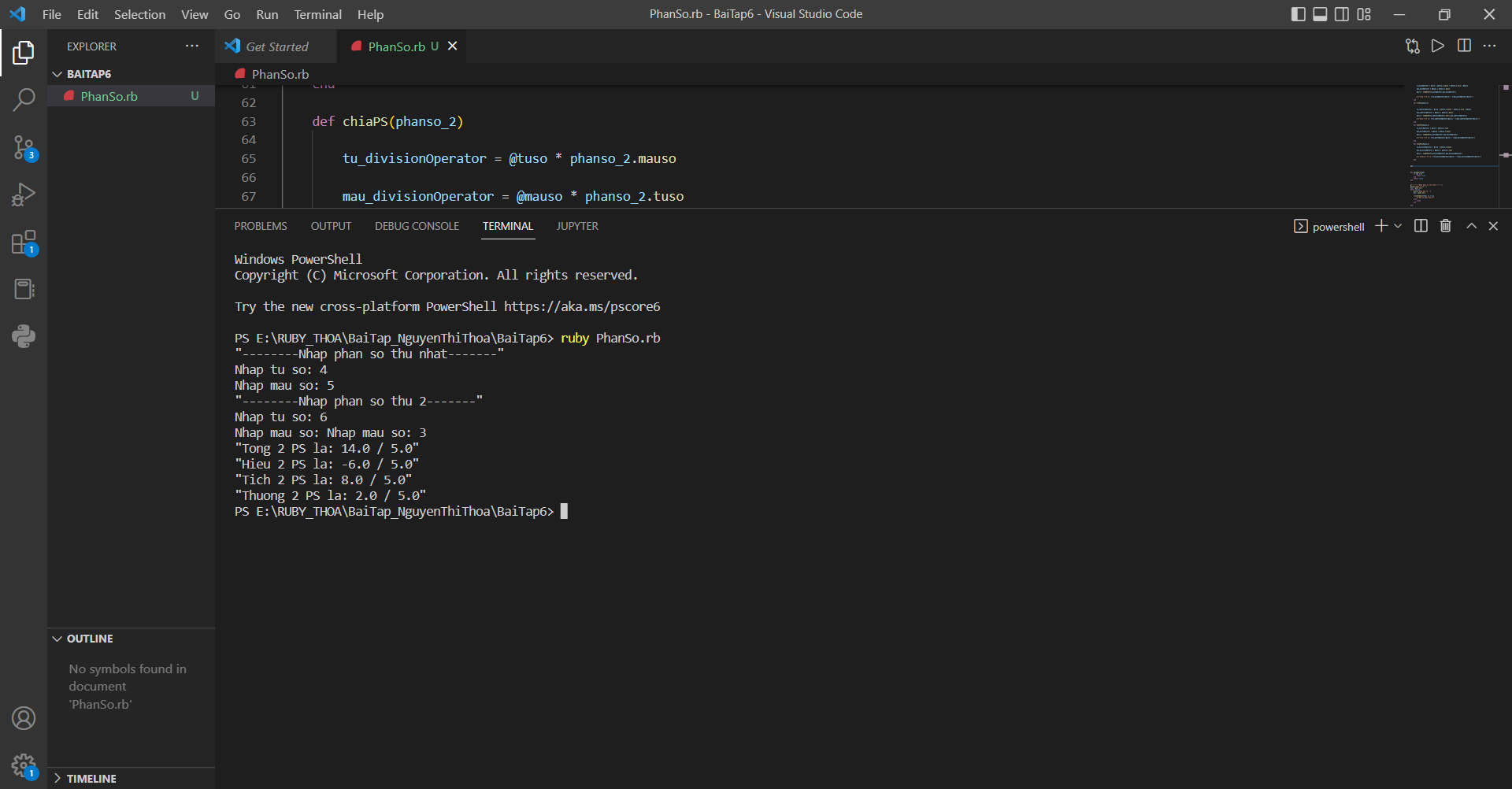
Bài 1: Công trừ nhân chia 2 phân số

* Code:
* class PhanSo
* @tuso
* @mauso
* attr\_accessor :tuso,:mauso
* def initialize(tuso,mauso)
* @tuso = tuso
* @mauso = mauso
* end
* def timUSCLN(a, b)
* while (a != b)
* if (a > b)
* a -= b
* else
* b -= a
* end
* end
* return a
* end
* def congPS(phanso\_2)
* tu\_plusOperator = @tuso \* phanso\_2.mauso + phanso\_2.tuso \* @mauso
* mau\_plusOperator = @mauso \* phanso\_2.mauso
* @ucln = timUSCLN(tu\_plusOperator,mau\_plusOperator)
* p ("Tong 2 PS la: #{tu\_plusOperator/@ucln} / #{mau\_plusOperator/@ucln}")
* end
* def truPS(phanso\_2)
* tu\_subtractOperator = @tuso \* phanso\_2.mauso - phanso\_2.tuso \* @mauso
* mau\_subtractOperator = @mauso \* phanso\_2.mauso
* @ucln = timUSCLN(tu\_subtractOperator.abs(),mau\_subtractOperator)
* p ("Hieu 2 PS la: #{tu\_subtractOperator/@ucln} / #{mau\_subtractOperator/@ucln}")
* end
* def nhanPS(phanso\_2)
* tu\_multiOperator = @tuso \* phanso\_2.tuso
* mau\_multiOperator = @mauso \* phanso\_2.mauso
* @ucln = timUSCLN(tu\_multiOperator,mau\_multiOperator)
* p ("Tich 2 PS la: #{tu\_multiOperator/@ucln} / #{mau\_multiOperator/@ucln}")
* end
* def chiaPS(phanso\_2)
* tu\_divisionOperator = @tuso \* phanso\_2.mauso
* mau\_divisionOperator = @mauso \* phanso\_2.tuso
* @ucln = timUSCLN(tu\_divisionOperator,mau\_divisionOperator)
* p ("Thuong 2 PS la: #{tu\_divisionOperator/@ucln} / #{mau\_divisionOperator/@ucln}")
* end
* end
* def checkZero(num)
* if num == 0
* return true
* end
* return false
* end
* p('--------Nhap phan so thu nhat-------')
* print('Nhap tu so: ')
* ts1 = gets.to\_f
* while (true)
* print('Nhap mau so: ')
* ms1 = gets.to\_f
* if(checkZero(ms1) == true)
* p "Mau so phai khac 0"
* else
* break
* end
* end
* p('--------Nhap phan so thu 2-------')
* print('Nhap tu so: ')
* ts2 = gets.to\_f
* print('Nhap mau so: ')
* while (true)
* print('Nhap mau so: ')
* ms2 = gets.to\_f
* if(checkZero(ms2) == true)
* p "Mau so phai khac 0"
* else
* break
* end
* end
* phanso\_1 = PhanSo.new(ts1,ms1)
* phanso\_2 = PhanSo.new(ts2,ms2)
* phanso\_1.congPS(phanso\_2)
* phanso\_1.truPS(phanso\_2)
* phanso\_1.nhanPS(phanso\_2)
* phanso\_1.chiaPS(phanso\_2)
* Kết quả:



Bài 2 : Overloading:

* Code:
* class PhanSo
* attr\_accessor :tuso,:mauso
* def initialize(tuso,mauso)
* @tuso = tuso
* @mauso = mauso
* end
* def timUSCLN(a, b)
* while (a != b)
* if (a > b)
* a -= b
* else
* b -= a
* end
* end
* return a
* end
* def +(phanso\_2)
* tu\_plusOperator = @tuso \* phanso\_2.mauso + phanso\_2.tuso \* @mauso
* mau\_plusOperator = @mauso \* phanso\_2.mauso
* @ucln = timUSCLN(tu\_plusOperator,mau\_plusOperator)
* return ("Tong 2 PS la: #{tu\_plusOperator/@ucln} / #{mau\_plusOperator/@ucln}")
* end
* def -(phanso\_2)
* tu\_subtractOperator = @tuso \* phanso\_2.mauso - phanso\_2.tuso \* @mauso
* mau\_subtractOperator = @mauso \* phanso\_2.mauso
* @ucln = timUSCLN(tu\_subtractOperator.abs(),mau\_subtractOperator)
* return ("Hieu 2 PS la: #{tu\_subtractOperator/@ucln} / #{mau\_subtractOperator/@ucln}")
* end
* def \*(phanso\_2)
* tu\_multiOperator = @tuso \* phanso\_2.tuso
* mau\_multiOperator = @mauso \* phanso\_2.mauso
* @ucln = timUSCLN(tu\_multiOperator,mau\_multiOperator)
* return ("Tich 2 PS la: #{tu\_multiOperator/@ucln} / #{mau\_multiOperator/@ucln}")
* end
* def /(phanso\_2)
* tu\_divisionOperator = @tuso \* phanso\_2.mauso
* mau\_divisionOperator = @mauso \* phanso\_2.tuso
* @ucln = timUSCLN(tu\_divisionOperator,mau\_divisionOperator)
* return ("Thuong 2 PS la: #{tu\_divisionOperator/@ucln} / #{mau\_divisionOperator/@ucln}")
* end
* end
* def checkZero(num)
* if num == 0
* return true
* end
* return false
* end
* p('--------PHAN SO I--------')
* print('Nhap tu so: ')
* ts1 = gets.to\_f
* while (true)
* print('Nhap mau so: ')
* ms1 = gets.to\_f
* if(checkZero(ms1) == true)
* p "Mau so phai khac 0"
* else
* break
* end
* end
* p('--------PHAN SO II--------')
* print('Nhap tu so: ')
* ts2 = gets.to\_f
* print('Nhap mau so: ')
* while (true)
* print('Nhap mau so: ')
* ms2 = gets.to\_f
* if(checkZero(ms2) == true)
* p "Mau so phai khac 0"
* else
* break
* end
* end
* phanso1 = PhanSo.new(ts1,ms1)
* phanso2 = PhanSo.new(ts2,ms2)
* p phanso1+phanso2
* p phanso1-phanso2
* p phanso1\*phanso2
* p phanso1/phanso2
* Kết quả:

