Задание:



Входные и выходные данные:

Входные:

al poezd = Poezd

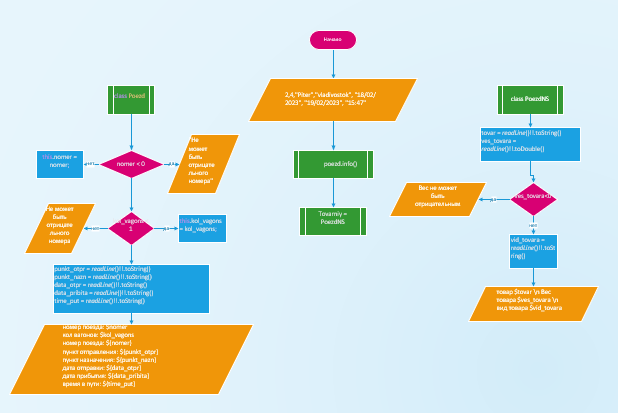
val Tovarniy = PoezdNS

Выходные:

poezd.proverk(nomer, kol\_vagons)  
poezd.info()

Tovarniy.proverk(nomer, kol\_vagons)  
Tovarniy.info()

Блок схема:



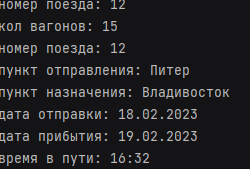
Листинг Программы (если есть)

import Poezd.\*  
import PoezdNS.\*  
  
  
fun main()  
{  
 try  
 {  
 val poezd = Poezd(  
 2,4,"Piter","vladivostok", "18/02/2023", "19/02/2023", "15:47"  
 )  
 *println*("ведите номер поезда")  
 var nomer = *readLine*()!!.toInt()  
  
 *println*("колличество вагонов")  
 var kol\_vagons = *readLine*()!!.toInt()  
 poezd.proverk(nomer, kol\_vagons)  
 poezd.info()  
  
  
 val Tovarniy = PoezdNS(  
 "Апельсины", 1003.34, "Фрукты", 2, 4,"Gelingik","Moskow", "18/02/2023", "19/02/2023", "15:47"  
 )  
  
 Tovarniy.proverk(nomer, kol\_vagons)  
 Tovarniy.info()  
  
  
  
  
 }  
  
 catch (e:Exception)  
 {  
 *println*("Некоректный ввод")  
 }  
  
}

package Poezd  
  
  
open class Poezd (var nomer: Int,var kol\_vagons: Int, var punkt\_otpr:String, var punkt\_nazn: String, var data\_otpr:String, var data\_pribita:String, var time\_put:String)  
{  
  
 open fun proverk (nomer : Int, kol\_vagons:Int ){  
 if (nomer < 0)  
 {  
 *println*("Не может быть отрицательного номера")  
 }  
 else  
 {  
 this.nomer = nomer;  
 }  
  
 if (kol\_vagons < 1)  
 {  
 *println*("Не может быть отрицательного номера")  
 }  
 else  
 {  
 this.kol\_vagons = kol\_vagons;  
 }  
  
 }  
  
  
  
 open fun info ()  
 {  
 *println*("ведите пункт отправления")  
 punkt\_otpr = *readLine*()!!.toString()  
 *println*("ведите пункт Назначения")  
 punkt\_nazn = *readLine*()!!.toString()  
 *println*("ведите дату отправления")  
 data\_otpr = *readLine*()!!.toString()  
 *println*("ведите дату прибытия")  
 data\_pribita = *readLine*()!!.toString()  
 *println*("ведите время в пути")  
 time\_put = *readLine*()!!.toString()  
  
 *println*("номер поезда: $nomer")  
 *println*("кол вагонов: $kol\_vagons")  
 *println*("номер поезда: ${nomer}")  
 *println*("пункт отправления: ${punkt\_otpr}")  
 *println*("пункт назначения: ${punkt\_nazn}")  
 *println*("дата отправки: ${data\_otpr}")  
 *println*("дата прибытия: ${data\_pribita}")  
 *println*("время в пути: ${time\_put}")  
  
 }

import Poezd.\*  
  
class PoezdNS(var tovar: String, var ves\_tovara: Double, var vid\_tovara:String, nsnomer: Int, nskol\_vagons: Int, nspunkt\_otpr:String, nspunkt\_nazn: String, nsdata\_otpr:String, nsdata\_pribita:String, nstime\_put:String) : Poezd(nsnomer, nskol\_vagons, nspunkt\_otpr, nspunkt\_nazn, nsdata\_otpr, nsdata\_pribita, nstime\_put)  
{  
  
  
 override fun info() {  
 super.info()  
 *println*("ведите товар")  
 tovar = *readLine*()!!.toString()  
 *println*("ведите вес товара")  
 ves\_tovara = *readLine*()!!.toDouble()  
 if (ves\_tovara<0)  
 {  
 *println*("Вес не может быть отрицательным")  
 }  
  
 *println*("ведите вид товара")  
 vid\_tovara = *readLine*()!!.toString()  
  
 *println*("товар $tovar \n Вес товара $ves\_tovara \n вид товара $vid\_tovara ")  
  
 }  
}

Тестовые ситуации



Вывод

Мы создали программу используя классы и классы наследники с конструкторами

Программа работает успешно.