Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практика 10**»

Выполнил: Деспижек Булат

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Вариант 1:**

**Входные данные:**

**Имя пк, имя процессора, класс, фирма-строкового типа**

**Тактовая частота процессора, Объём оперативной памяти, объём Винчестера, вес, -Вещественного типа**

**Выходные данные:**

**Информация о ПК-Строкового типа**

**Листинг-Кода:**

import Computers.\*  
fun main() {  
 try {  
 *println*("Введите Класс компьютера (Ноутбук,ПК)")  
 var nameClass: String = *readLine*()!!.toString().*lowercase*()  
 when{  
 (nameClass=="ноутбук")->nameClass="Ноутбук"  
 (nameClass=="пк" || nameClass =="персональный компьютер")->nameClass="Персональный компьютер"  
 else->throw Exception("Неправильный ввод")  
 }  
 *println*("Введите имя процессора ")  
 var nameProc: String = *readLine*()!!.toString().*lowercase*()  
 *println*("Введите Тактовую частоту процессора (МГц)")  
 val tact: Double = *readLine*()!!.*toDouble*()  
 if(tact<2500.0 || tact > 8723.0) throw Exception("Неправильный ввод")  
 *println*("Введите объём оперативной памяти (Mb)")  
 val volumeRam:Double = *readLine*()!!.*toDouble*()  
 if(volumeRam<4.0 || volumeRam > 131072.0) throw Exception("Неправильный ввод")  
  
  
 when(nameClass){  
 "Ноутбук"-> {  
 *println*("Введите имя ноутбука")  
 val name:String= *readLine*().*toString*()  
 *println*("Введите фирму ноутбука")  
 val firma:String= *readLine*().*toString*()  
 *println*("Введите вес ноутбука (kg)")  
 val ves:Double= *readLine*()!!.*toDouble*()  
 if (ves<1.0 || ves > 10) throw Exception("Неправильный ввод")  
 *println*("Введите объём Винчестера (Гб)")  
 val volumeHdd:Double= *readln*().*toDouble*()  
 if (volumeHdd<40.0 || volumeHdd > 1024) throw Exception("Неправильный ввод")  
 val Comp=Laptop(nameClass,nameProc,tact,volumeRam,name,volumeHdd,firma,ves)  
 Comp.Info()  
 }  
 "Персональный компьютер"->{  
 *println*("Введите имя ПК")  
 val name:String= *readLine*().*toString*()  
 *println*("Введите фирму ПК")  
 val firma:String= *readLine*().*toString*()  
 *println*("Введите объём Винчестера (Гб)")  
 val volumeHdd:Double= *readln*().*toDouble*()  
 if (volumeHdd<40.0 || volumeHdd > 1024) throw Exception("Неправильный ввод")  
 val Comp=PC(nameClass,nameProc,tact,volumeRam,name,volumeHdd,firma)  
 Comp.Info()  
 }  
 }  
  
  
 }catch (ex:Exception){  
 *println*(ex.message)  
 }  
  
}

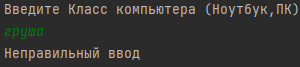
package Computers  
abstract class Computer(val classcomp:String, val nameProc:String, val tact:Double,val volumeRam:Double):Computing\_machine {  
  
 override fun Quality():Double{  
 var quality:Double  
 quality=(0.1\*tact)+volumeRam  
 return quality  
 }  
 override fun Info() {  
 *println*("Класс компьютера: $classcomp " +  
 "\nИмя процессора: $nameProc \nТактовая частота процессора: $tact \nОбъём оперативной памяти: $volumeRam" +  
 "\nКачество: ${Quality()}")  
 }

package Computers  
interface Computing\_machine {  
 fun Info()  
 fun Quality():Double

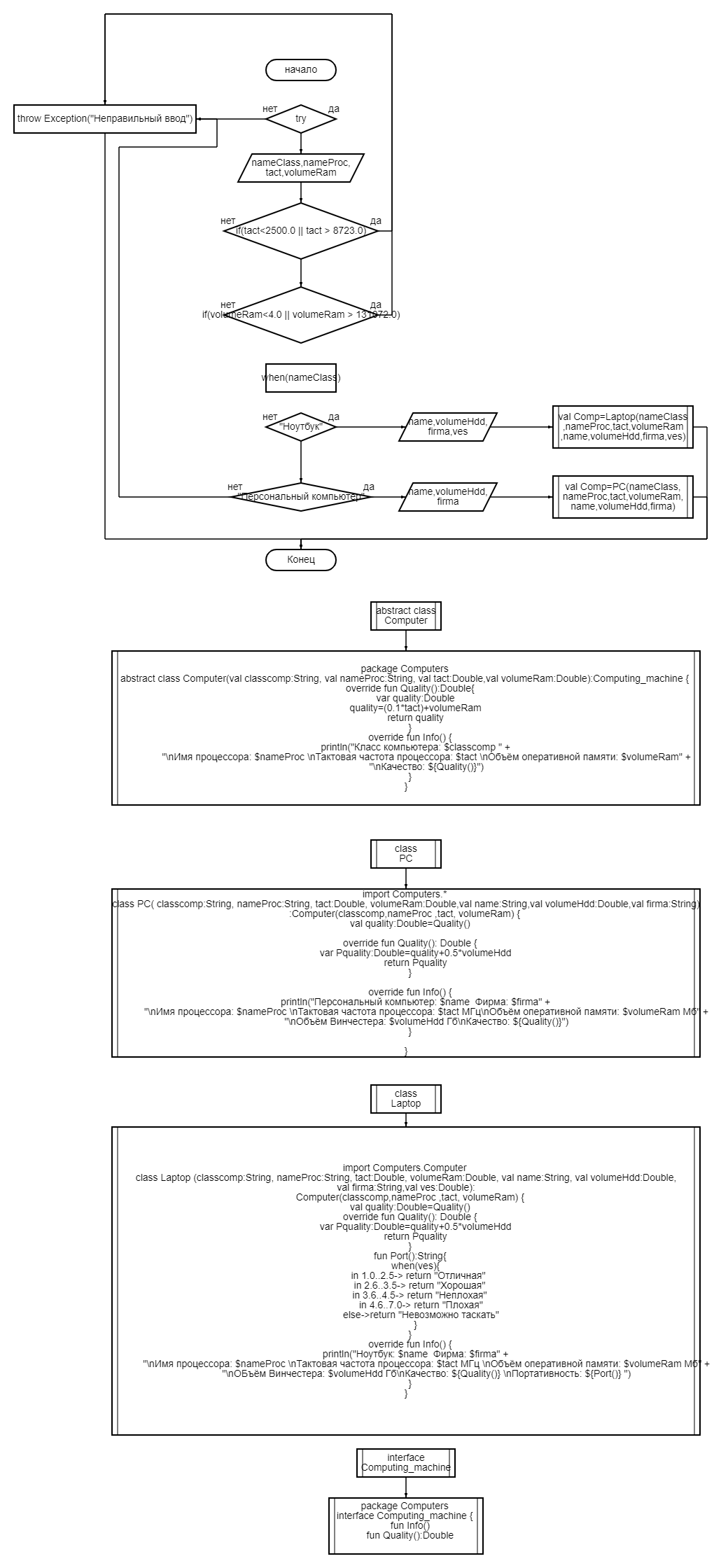
import Computers.\*  
class PC( classcomp:String, nameProc:String, tact:Double, volumeRam:Double,val name:String,val volumeHdd:Double,val firma:String):Computer(classcomp,nameProc ,tact, volumeRam) {  
  
 val quality:Double=Quality()  
  
 override fun Quality(): Double {  
 var Pquality:Double=quality+0.5\*volumeHdd  
 return Pquality  
 }  
  
 override fun Info() {  
 *println*("Персональный компьютер: $name Фирма: $firma" +  
 "\nИмя процессора: $nameProc \nТактовая частота процессора: $tact МГц\nОбъём оперативной памяти: $volumeRam Мб" +  
 "\nОбъём Винчестера: $volumeHdd Гб\nКачество: ${Quality()}")  
 }

import Computers.Computer  
  
class Laptop (classcomp:String, nameProc:String, tact:Double, volumeRam:Double, val name:String, val volumeHdd:Double, val firma:String,val ves:Double):  
 Computer(classcomp,nameProc ,tact, volumeRam) {  
  
 val quality:Double=Quality()  
  
 override fun Quality(): Double {  
 var Pquality:Double=quality+0.5\*volumeHdd  
 return Pquality  
 }  
  
 fun Port():String{  
 when(ves){  
 in 1.0..2.5-> return "Отличная"  
 in 2.6..3.5-> return "Хорошая"  
 in 3.6..4.5-> return "Неплохая"  
 in 4.6..7.0-> return "Плохая"  
 else->return "Невозможно таскать"  
 }  
  
  
 }  
  
 override fun Info() {  
 *println*("Ноутбук: $name Фирма: $firma" +  
 "\nИмя процессора: $nameProc \nТактовая частота процессора: $tact МГц \nОбъём оперативной памяти: $volumeRam Мб" +  
 "\nОБъём Винчестера: $volumeHdd Гб\nКачество: ${Quality()} \nПортативность: ${Port()} ")  
 }  
  
}

**Тестовые ситуации:**

****

**Блок-Схема:**

****