Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра инженерной психологии и эргономики

Современные языки программирования

Отчет по лабораторной работе №1

«Введение в программирование на Kotlin»

Выполнил: Сташук Б.С.

Студент группы 310901

Преподаватель: Усенко Ф.В.

Минск 2024

Цель: Выполнить разработку приложения с использованием языка программирования Kotlin.

Задание 25. Добавить класс Пакет и организовать возможность помещать туда товар. Добавить классу Тележка метод Выдать товар(). Добавить класс Молоко.

fun main() {

val list = mutableListOf<Product>()

val shop = Shop("Evroopt")

val cart = Cart(list)

val packet = Packet()

val buyer = Buyer(0, 0, shop, packet, cart)

println("Вы зашли в магазин" + shop.name)

while(true) {

println("Выберите действие: \n1.Взять хлеб\n2.Взять молоко\n3.Взять спички\n4.Перейти к оплате\n5.Просмотр тележки\n6.Выйти")

val choice: String = readLine().toString()

when(choice){

"1" ->{

println("Напишите количество товара")

val input = readLine()

println("Выберите хлеб\n1.Ржаной\n2.Пшеничный")

val taste = readLine()

when(taste) {

"1" -> {

val product = Bread(input!!.toInt(), "Ржаной")

buyer.Take(product)

buyer . PutInCart (product)

buyer . cart . Accept (product)

}

"2" -> {

val product = Bread(input!!.toInt(), "Пшеничный")

buyer.Take(product)

buyer . PutInCart (product)

buyer . cart . Accept (product)

}

else -> println("Некорректный ввод")

}

}

"2" ->{

println("Напишите количество товара")

val input = readLine()

val product = Milk(input!!.toInt())

buyer.Take(product)

buyer.PutInCart(product)

buyer.cart.Accept(product)

}

"3" ->{

println("Напишите количество товара")

val input = readLine()

println("Выберите длину спичек\n1. 10см\n2. 4см")

val length = readLine()

when(length) {

"1" -> {

val product = Matches(input!!.toInt(), 10)

buyer.Take(product)

buyer . PutInCart (product)

buyer . cart . Accept (product)

}

"2" -> {

val product = Matches(input!!.toInt(), 4)

buyer.Take(product)

buyer . PutInCart (product)

buyer . cart . Accept (product)

}

else -> println("Некорректный ввод")

}

}

"4" ->{

if(buyer.cart.Give(buyer.cart.capacity)) {

val inputMoney = buyer.shop.Sell(buyer.cart.capacity)

val outputMoney = buyer.Buy()

if (shop.TakeMoney(inputMoney, outputMoney)) {

buyer.PutInPacket()

println("Товары положены в пакет")

buyer.money = inputMoney

buyer.amount = inputMoney/10

break

}

}

}

"5" ->{

buyer.cart.Give(buyer.cart.capacity)

}

"6" ->{

return

}

else -> println("Некорректный ввод")

}

}

buyer.packet.Put(cart.capacity)

while(true){

println("Выберите действие: \n1.Зажечь спичку\n2.Попробовать хлеб\n3.Выйти")

val choice: String = readLine().toString()

when(choice) {

"1" -> {

var done: Boolean = false

for(item in buyer.packet.capacity){

if(item is Matches){

if(item.amount > 0) {

item.Burn()

done = true

break

}

}

}

if(!done){

println("Спичек нет")

}

}

"2" -> {

var done: Boolean = false

for(item in buyer.packet.capacity){

if(item is Bread){

if(item.amount > 0) {

item.TastePrint()

done = true

break

}

}

}

if(!done){

println("Хлеба нет")

}

}

"3" -> return

}

}

}

class Buyer(var money: Int, var amount: Int, val shop: Shop, val packet: Packet, val cart: Cart) {

fun Take(product: Product){

println("Вы взяли " + (product.name) + " " + (product.amount))

}

fun Buy() : Int{

//сколько денег даёт покупатель

println("Сколько денег вы даёте")

val output: Int = readLine()!!.toInt()

return output

}

fun PutInPacket(list: MutableList<Product> = mutableListOf<Product>()){

packet.capacity = list

}

fun PutInCart(product: Product){

cart.capacity.add(product)

}

}

class Shop(val name: String) {

fun Sell(list: MutableList<Product> = mutableListOf<Product>()) : Int {

//подсчёт суммы за товары

var sum = 0

for(item in list){

sum += item.amount

}

sum \*= 10

println("Стоимость товаров " + sum)

return sum

}

fun TakeMoney(input: Int, output: Int) : Boolean{

//денежные операции

if(output >= input){

println("Покупка одобрена")

if(output - input > 0)

println("Dам выдана сдача " + (output - input))

return true

} else

println("Недостаточно денег")

return false

}

}

class Packet(var capacity: MutableList<Product> = mutableListOf<Product>()){

fun Put(list: MutableList<Product> = mutableListOf<Product>()){

capacity = list

}

}

class Cart( val capacity: MutableList<Product> = mutableListOf<Product>()){

fun Accept(prod: Product){

println("В корзину поместили " + prod.name + " " + prod.amount + " штук")

}

fun Give(list: MutableList<Product> = mutableListOf<Product>()) : Boolean{

if(list.isEmpty()) {

println("Корзина пуста")

return false

}

println("В корзине лежит:")

for(item in list){

when(item){

is Matches -> println(item.name + " " + item.length + " см " + item.amount)

is Bread -> println(item.taste + " " + item.name + " " + item.amount)

is Milk -> println(item.name + " " + item.amount)

}

}

return true

}

}

abstract class Product(open var amount: Int, open val name: String = "Товар") {

}

class Milk(override var amount: Int, override val name: String = "Молоко") : Product(amount){

}

class Bread(override var amount: Int, val taste: String, override val name: String = "Хлеб") : Product(amount){

fun TastePrint(){

if (amount > 0) {

val range: IntRange = 1..6

val num: Int = range.random()

when(num) {

1 -> println("Хлеб оказался заплесневелым")

2 -> println("Хлеб оказался чёрствым")

else -> println("Хлеб оказался вкусным")

}

amount -= 1

}

}

}

class Matches(override var amount: Int, val length: Int, override val name: String = "Спички") : Product(amount){

fun Burn() {

if (amount > 0) {

println("Спичка зажглась")

amount -= 1

}

}

Результат работы программы представлен на рисунке 1

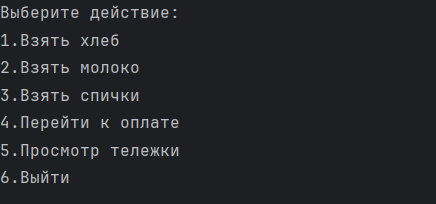


Рисунок 1 –Результат работы программы

Вывод: В ходе выполнения работы разработал программу на Kotlin.