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

«Использование языка программирования Kotlin»

Выполнил: Былёв Н. С.

Студент группы 310901

Преподаватель: Усенко Ф. В.

Минск 2024

Цель: Выполнить разработку приложения с использованием языка программирования Kotlin.

Задание.Вариант 3.

Добавить класс Директор, управляющий классом Завод. Добавить класс Магазин, содержащий ссылку на класс товар, который производит класс Завод. Добавить новые классы, наследующие от класса Продукт.

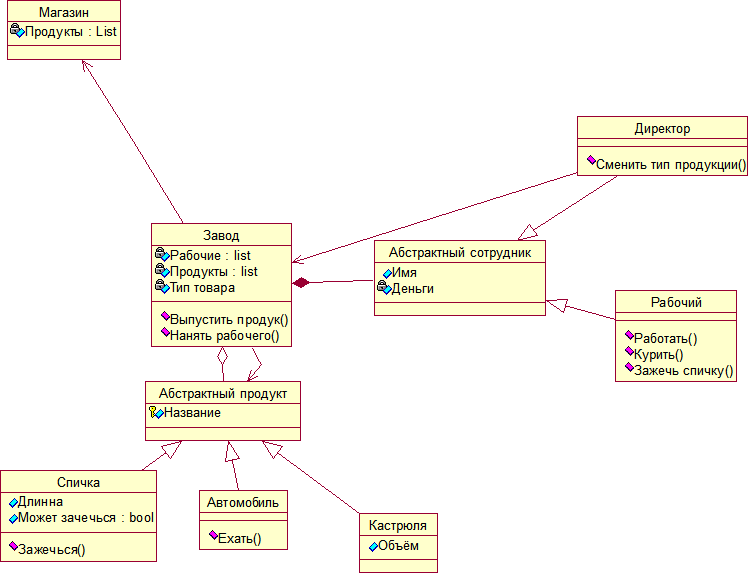


Рисунок 1 – Диаграмма классов

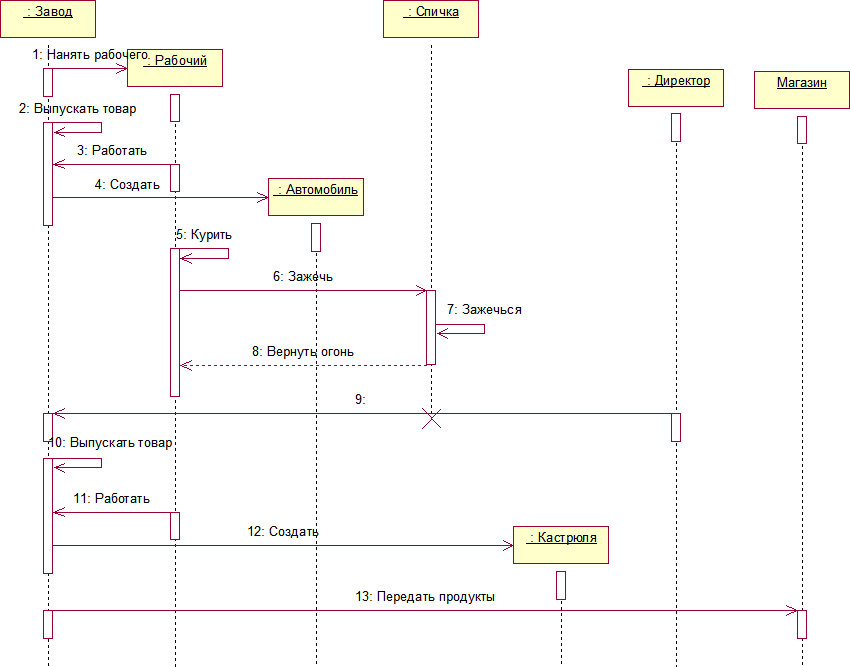


Рисунок 2 – Диаграмма последовательности

Листинг кода:

Файл Main.kt:

abstract class Product(protected val name: String)  
  
  
class Car(name: String) : Product(name) {  
 fun drive() {  
 *println*("$name едет")  
 }  
}  
  
  
class Pot(name: String, val volume: Double) : Product(name)  
  
  
class Match(name: String, val length: Double, val canBeLighted: Boolean) : Product(name) {  
 fun light() {  
 if (canBeLighted) {  
 *println*("$name зажжена")  
 }  
 else {  
 *println*("$name не загорелась")  
 }  
 }  
}  
  
  
abstract class Employee(val name: String, private var money: Double)  
  
  
class Worker(name: String, money: Double) : Employee(name, money) {  
 fun work() {  
 *println*("$name работает")  
 }  
  
 fun smoke() {  
 *println*("$name курит")  
 }  
  
 fun lightAMatch(match: Match) {  
 if (match.canBeLighted) {  
 *println*("$name зажигает спичку")  
 }  
 else{  
 *println*("спичка не горит")  
 }  
 }  
}  
  
  
class Factory {  
 private val workers = *mutableListOf*<Worker>()  
 private val products = *mutableListOf*<Product>()  
 private var productType: String = ""  
  
 fun hireWorker(worker: Worker) {  
 workers.add(worker)  
 }  
  
 fun releaseProduct(product: Product) {  
 products.add(product)  
 }  
  
 fun changeProductType(type: String) {  
 productType = type  
 *println*("тип продукции фабрики сменён на $productType")  
 }  
  
 fun sendToStore(products: List<Product>, store: Store)  
 {  
 store.addProducts(products)  
 }  
}  
  
  
class Director(val factory: Factory) {  
 fun changeProductionType(type: String) {  
 factory.changeProductType(type)  
 }  
}  
  
  
class Store(val products: List<Product>)  
{  
 val inventory = *mutableListOf*<Product>()  
 fun addProducts(products: List<Product>) {  
 inventory.addAll(products)  
 }  
}  
  
fun main() {  
 val factory = Factory()  
 val director = Director(factory)  
 val worker = Worker("Кирюха", 100.0)  
 val match = Match("первая спичка", 12.0, false)  
  
 factory.hireWorker(worker)  
  
 worker.work()  
 worker.smoke()  
 worker.lightAMatch(match)  
 val car = Car("Porsche")  
 factory.releaseProduct(car)  
 match.light()  
  
 director.changeProductionType("электронику")  
}

Результат работы программы представлен на рисунке 3.

# 

Рисунок 3 – Результат выполнения программы

Вывод: в ходе выполнения лабораторной работы номер 1 по языку программирования kotlin мы познакомились с основами языка и научились на практике применять основные его конструкции. Было потрачено время на изучение UML диаграмм для корректного понимая задания лабораторной рабыты.