**O‘zbekiston Respublikasi Raqamli texnologiyalar vazirligi**



**Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti**

**Dasturiy injiniring fakulteti**

1-topshiriq

**Fan: Mobil ilovalarni ishlab chiqish (DI)**

**Bajardi:Oybek Eshonqulov Tekshirdi: Qurbonov B.N. Guruh : MAD001-2**

**Toshkent 2025**

1. Topshiriq
2. Bankomat Simulyatori dasturi. Foydalanuvchidan PIN-kod kiritishni so‘rab,agar to‘g‘ri kiritilsa, bankomat menyusini chiqaradigan dastur yozing; menyuda balansni ko‘rish, pul yechish, pul kiritish va chiqish imkoniyatlari bo‘lsin; noto‘g‘ri PIN 3 marta kiritilsa, kartani bloklasin. Dasturda listlar, classlardan foydalanilmasin. Boshlang’ich o’zgaruvchilar quyidagidan iborat: balance=200000, correct\_pin=”1234” attemps=3.

Dasturga quyiladigan talablar.

1. Dastur faqat bitta main funksiyadan iborat bo’lishi
2. List, Class, Obyekt kabi ma’lumot tuzilmalaridan foydalanilmasin
3. Dars davomida o’tilgan while, do while, if, break, continue kabi operatorlardan foydalaning.

Dastur qanday ishlaydi? Foydalanuvchi PIN-kod kiritadi.

Agar 3 marta noto‘g‘ri kiritsa, dastur tugaydi. To‘g‘ri kiritsa, asosiy menyu ochiladi.

Asosiy menyuda foydalanuvchi 4 ta amaldan birini tanlashi mumkin: 1: Balansni tekshirish

2: Pul yechish (balansdan ortiqcha pul yechish mumkin emas) 3: Pul kiritish

4: Chiqish

While tsikli yordamida foydalanuvchi 4 ni tanlamaguncha menyu qayta Chiqaveradi

fun main() {

val correctPin = "1234" var balance = 200000 var attempts = 3

while (attempts > 0) {

*print*("PIN-kodingizni kiriting: ") val inputPin = *readLine*()

if (inputPin == correctPin) { while (true) {

*println*("\n1: Balansni tekshirish") *println*("2: Pul yechish") *println*("3: Pul kiritish") *println*("4: Chiqish") *print*("Tanlang: ")

when (*readLine*()) {

"1" -> *println*("Sizning balansingiz: $balance so'm") "2" -> {

*print*("Qancha pul yechmoqchisiz? ")

val withdraw = *readLine*()?.*toIntOrNull*()

if (withdraw != null && withdraw > 0 && withdraw <= balance) {

balance -= withdraw

*println*("$withdraw so'm yechildi. Yangi balans: $balance so'm")

} else {

*println*("Xato: Noto'g'ri summa yoki yetarli mablag' yo'q!")

}

}

"3" -> {

*print*("Qancha pul kiritmoqchisiz? ") val deposit = *readLine*()?.*toIntOrNull*() if (deposit != null && deposit > 0) {

balance += deposit

*println*("$deposit so'm qo'shildi. Yangi balans: $balance so'm")

} else {

*println*("Xato: Noto'g'ri summa!")

}

}

"4" -> {

*println*("Dasturdan chiqildi.") return

}

else -> *println*("Noto'g'ri tanlov, qaytadan kiriting.")

}

}

} else {

attempts--

if (attempts == 0) { *println*("Kartangiz bloklandi!") return

} else {

*println*("Noto'g'ri PIN! Qolgan urinishlar: $attempts")

}

}

}

}

iborat

1. Topshiriq
2. Todo List (Topshiriqlarni boshqarish) dasturi

Bu dastur foydalanuvchiga topshiriqlarni qo‘shish, o‘chirish, yangilash va ko‘rish imkoniyatini beradi. Har bir topshiriq nom, sana, soat va holatdan

bo‘ladi.

Dasturga quyiladigan talablar.

1. Dastur funksiyalardan foydalanilsin
2. Class, Obyekt kabi ma’lumot tuzilmalaridan foydalanilmasin
3. Dars davomida o’tilgan list, fun, while, do while, if, break, continue kabi operatorlardan foydalaning.

Dastur qanday ishlaydi?

1. Yangi topshiriq qo‘shish (add\_task)

Foydalanuvchi topshiriq nomini, sanasini, soatini va holatini kiritadi.

1. Topshiriqni o‘chirish (delete\_task)

Mavjud topshiriqni indeks yoki nomi bo‘yicha o‘chirish mumkin.

1. Topshiriqlar ro‘yxatini ko‘rish (list\_of\_task) Barcha saqlangan topshiriqlar ekranga chiqariladi.
2. Topshiriqni yangilash (update\_task)

Mavjud topshiriqning nomi, sanasi, soati yoki holatini o‘zgartirish imkoniyati mavjud.

1. Dasturdan chiqish (exit)

Dasturdan chiqish uchun foydalaniladi. Dastur exit chiqish tugmasini bosmagunicha davom ettirilsin

import java.util.\*

val *tasks* = *mutableListOf*<MutableList<String>>() fun addTask() {

*print*("Task nomi: ")

val task\_Name = *readLine*()!! *print*("Date (year-month-day): ") val date = *readLine*()!! *print*("Soat (soat:minut): ")

val hour = *readLine*()!!

*print*("Status (begin, running, waiting, completed): ") val status = *readLine*()!! *tasks*.add(*mutableListOf*(task\_Name, date, hour, status)) *println*("Task tugatildi!\n")

}

fun updateTask() {

*listOfTasks*()

*print*("Enter task index to update: ")

val index = *readLine*()!!.*toIntOrNull*()?.minus(1) if (index != null && index in *tasks*.*indices*) {

*print*("New Task Name (leave empty to keep current): ")

val taskName = *readLine*()!!.*trim*()

if (taskName.*isNotEmpty*()) *tasks*[index][0] = taskName

*print*("New Date (YYYY-MM-DD, leave empty to keep current): ") val date = *readLine*()!!.*trim*()

if (date.*isNotEmpty*()) *tasks*[index][1] = date

*print*("New Hour (HH:MM, leave empty to keep current): ") val hour = *readLine*()!!.*trim*()

if (hour.*isNotEmpty*()) *tasks*[index][2] = hour

*print*("New Status (begin, running, waiting, completed, leave empty to keep current): ")

val status = *readLine*()!!.*trim*()

if (status.*isNotEmpty*()) *tasks*[index][3] = status

*println*("Task updated successfully!\n")

} else {

*println*("Invalid index!\n")

}

}

fun deleteTask() {

*listOfTasks*()

*print*("Enter task index to delete: ")

val index = *readLine*()!!.*toIntOrNull*()?.minus(1) if (index != null && index in *tasks*.*indices*) {

*tasks*.removeAt(index)

*println*("Task deleted successfully!\n")

} else {

*println*("Invalid index!\n")

}

}

fun listOfTasks() {

*println*("\nTask Menu")

*println*("Index Task Name Date Hour Status")

for (i in 1..50) {

*print*("-")

}

*println*()

for (i in *tasks*.*indices*) { val task = *tasks*[i]

*println*("${i + 1} | ${task[0].*padEnd*(12)} | ${task[1]} | ${task[2]} |

${task[3]}")

}

*println*()

}

fun main() { while (true) {

*println*("Todo List\n") *println*("1. Add Task") *println*("2. Delete Task") *println*("3. List of Tasks") *println*("4. Exit") *println*("5. Update Task") *print*("Enter your choice: ") when (*readLine*()!!) {

"1" -> *addTask*()

"2" -> *deleteTask*()

"3" -> *listOfTasks*()

"4" -> {

*println*("Chiqish...\n") return

}

"5" -> *updateTask*()

else -> *println*("xatolik tugri tanlang 1-5.\n")

}

}

}