|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/05 Современные интеллектуальные**

**программно-аппаратные комплексы**

**ОТЧЕТ**

**по лабораторной работе №8**

|  |  |
| --- | --- |
| Группа | ИУ6-23М |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  |  | И.И. Иванов |
|  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |
| Преподаватель |  |  | П.В. Степанов |
|  |  | (Подпись, дата) | (И.О. Фамилия) |

*Москва, 2023 г.*

**Условие**

1. Реализовать многопоточное приложение “Банк”. Имеется банковский счет. Сделать синхронным пополнение и снятие денежных средств на счет/со счет случайной суммой. При каждой операции (пополнения или снятие) вывести текущий баланс счета. В том случае, если денежных средств недостаточно – вывести сообщение.

**Програмная реализация**

1

|  |
| --- |
| public class Main {  public static void main(String[] args) {  Bank bank = new Bank(10000);  Thread depositThread = new Thread(new BankThread(bank, true));  Thread withdrawalThread = new Thread(new BankThread(bank, false));  depositThread.start();  withdrawalThread.start();  }  } |

2

|  |
| --- |
| import java.util.Random;  public class BankThread implements Runnable {  private Bank bank;  private boolean isDeposit;  private Random random = new Random();  public BankThread(Bank bank, boolean isDeposit) {  this.bank = bank;  this.isDeposit = isDeposit;  }  public void run() {  for (int i = 0; i < 10; i++) {  int amount = random.nextInt(1000);  if (isDeposit) {  bank.deposit(amount);  } else {  bank.withdraw(amount);  }  }  }  } |

3

|  |
| --- |
| public class Bank {  private int balance;  public Bank(int initialBalance) {  this.balance = initialBalance;  }  public synchronized void deposit(int amount) {  balance += amount;  System.out.println("Deposit: " + amount + ", balance: " + balance);  }  public synchronized void withdraw(int amount) {  if (balance >= amount) {  balance -= amount;  System.out.println("Withdrawal: " + amount + ", balance: " + balance);  } else {  System.out.println("Insufficient funds, balance: " + balance);  }  }  } |