|  |  |
| --- | --- |
| **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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа,**

**обработки и интерпретации больших данных.**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 8 |

**Название:**

Потоки

**Дисциплина:** Языки программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | Е.Е. Шморгун |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Вариант 1 Задача 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.

Main.java

import java.util.Random;

public class Main {

public static void main(String[] args) {

Bank bank = new Bank();

int commands\_number = 10;

for (int i = 0; i < commands\_number; i++) {

int command = getRandomInt(1, 3);

int amount = getRandomInt(100, 10000);

BankCommand bankCommand = new BankCommand(bank, command, amount);

bankCommand.run();

}

}

public static int getRandomInt(int min, int max) {

Random random = new Random();

return random.ints(min, max)

.findFirst()

.getAsInt();

}

}

Bank.java

import java.util.regex.Pattern;

public class Bank {

int balance;

public Bank() {

this.balance = 0;

}

public synchronized void increase(int amount) {

this.balance += amount;

print\_balance();

}

public synchronized void decrease(int amount) {

if (amount > this.balance) {

System.out.println("Insufficient funds");

} else {

this.balance -= amount;

}

print\_balance();

}

private void print\_balance() {

System.out.printf("Current balance: %s\n", this.balance);

}

}

BankCommand.java

public class BankCommand implements Runnable {

Bank bank;

int command;

int amount;

BankCommand(Bank bank, int command, int amount) {

this.bank = bank;

this.command = command;

this.amount = amount;

}

public void run() {

if (this.command == 1) {

System.out.printf("Running command: increase. Amount: %d\n", this.amount);

bank.increase(amount);

} else if (this.command == 2){

System.out.printf("Running command: decrease. Amount: %d\n", this.amount);

bank.decrease(amount);

} else {

System.out.printf("Unknown command: %s", this.command);

}

}

}

**Вывод:** В данной работе я познакомился с многопоточностью в Java. Узнал об основных классах, интерфейсах и методах для работы с ними. А также создал приложение, которое работает с несколькими потоками.