|  |  |
| --- | --- |
| 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 (ИУ-6)

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**

# Дисциплина: Языки программирования для работы с большими данными

**Название:** Потоки

Студент гр. **ИУ6-23М \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.П.Завьялова**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ П.В.Степанов**

(Подпись, дата) (И.О. Фамилия)

Москва, 2022

**Условие:**

Вариант 1

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.
2. Реализовать многопоточное приложение “Робот”. Надо написать робота, который умеет ходить. За движение каждой его ноги отвечает отдельный поток. Шаг выражается в выводе в консоль LEFT или RIGHT.

**Фрагмент решения:**

package lab8;  
  
import java.util.Random;  
  
public class Bank {  
 static int *account* = 100;  
  
 public static void main(String[] args) throws InterruptedException {  
 System.*out*.println("Начальная сумма: " + *account*);  
 Random random = new Random();  
 int max = 100;  
 int min = 1;  
  
 BankThread replenishment = new BankThread("+");  
 BankThread debit = new BankThread("-");  
 replenishment.start();  
 debit.start();  
  
 replenishment.changeAccount(random.nextInt(max - min) + min);  
 debit.changeAccount(random.nextInt(max - min) + min);  
 replenishment.changeAccount(random.nextInt(max - min) + min);  
 debit.changeAccount(random.nextInt(max - min) + min);  
 debit.changeAccount(random.nextInt(max - min) + min);  
 debit.changeAccount(random.nextInt(max - min) + min);  
 debit.changeAccount(random.nextInt(max - min) + min);  
  
 replenishment.setFlag(false);  
 System.*out*.println("Replenishment stopped!");  
 debit.setFlag(false);  
 System.*out*.println("Debit stopped!");  
  
 System.*out*.println("Итоговая сумма: " + *account*);  
 }  
}  
  
class BankThread extends Thread {  
  
 private String sign;  
 private boolean flag = true;  
  
 public BankThread(String sign) {  
 this.sign = sign;  
 }  
  
 public void changeAccount(int amount){  
 System.*out*.println(Bank.*account* + " " + this.sign + " " + amount);  
 if (this.sign == "+") Bank.*account* = Bank.*account* + amount;  
 if (this.sign == "-") {  
 if (Bank.*account* < amount){  
 System.*out*.println("Недостаточно средств");  
 } else Bank.*account* = Bank.*account* - amount;  
 }  
 System.*out*.println(" = " + Bank.*account*);  
 }  
  
 public void setFlag(boolean flag) {  
 this.flag = flag;  
 }  
}

**Вывод:**

В ходе выполнения лабораторной работы, получены навыки работы с потоками, а именно: созданы многопоточные приложения в соответствии с вариантом на языке Java.

Репозиторий с кодом: <https://github.com/AnnaZav/Java_labs_bigdata.git>