|  |  |
| --- | --- |
| 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, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ и системы  
 управления**

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

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 |  |  | А.А. Морозова |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

**Цель работы** — ознакомление с механизмами работы потоков в языке Java.

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.

Часть кода задания приведена в листинге 1, результат выполнения – на рисунке 1.

Листинг 1 – Переопределенный метод run()

|  |
| --- |
| @Override  public void run() {  for (int i = 0; i < 10; i++) {  try {  TimeUnit.MILLISECONDS.sleep(random.nextInt() % 100);  } catch (Exception e) {  continue;  }  if ((random.nextInt() % 2) == 0) {  bankAccount.addMoney(Math.abs(random.nextInt() % 100), taskName);  } else {  bankAccount.withdrawMoney(Math.abs(random.nextInt() % 100), taskName);  }  }  } |

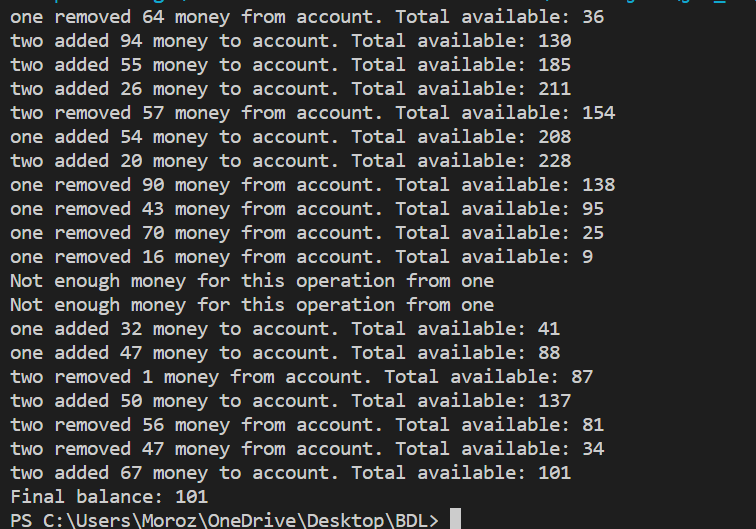


Рисунок 1 – Пример работы программы «Банк»

1. 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.

Часть кода задания приведена в листинге 2, результат выполнения – на рисунке 2.

Листинг 2 – Хождение робота

|  |
| --- |
| public static void main(String[] args) {  Robot robot = new Robot();  Thread leftLegThread = new Thread(() -> {  while (true) {  robot.moveLeftLeg();  }  });  Thread rightLegThread = new Thread(() -> {  while (true) {  robot.moveRightLeg();  }  });  leftLegThread.start();  rightLegThread.start();  robot.startWalking();  } |

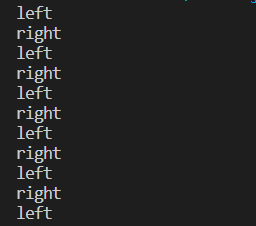


Рисунок 2 – Вывод приложения «Робот»

Полный код заданий размещен в репозитории по ссылке – https://github.com/moroz-matros/BDL.

**Вывод** — в ходе работы были получены навыки создания многопоточных приложений в языке Java.