МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

По лабораторной работе № \_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:

Студент группы ИИ-22

Копанчук Евгений Романович

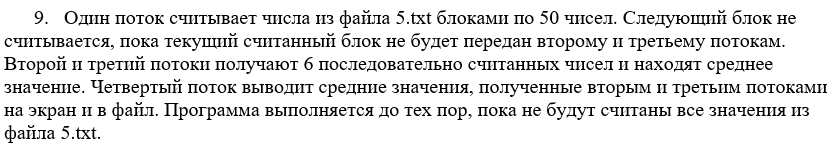
Проверил\_\_:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Брест – 2023

**Ход работы**

TASK



PROGRAM

|  |
| --- |
| void FILL\_FILE\_WITH\_RANDINT\_VALUES(std::string file, int left, int right, int amount) {  std::ofstream in(file);  for (int i = 0; i < amount; i++)  in << (int)(left + rand() % right + 1) << ' ';  in.close();  }  std::mutex mtx;  class ThreadMean {  private:  std::string input\_file\_;  std::string output\_file\_;  int batch\_size\_;  int part\_size\_;  std::vector<int> batch\_;  std::queue<int> buf\_;  std::queue<double> mean\_;  std::queue<int> th\_num\_;  int batch\_number\_;  int part\_number\_;  int last\_part\_;  bool end\_;  public:  ThreadMean(std::string inputfile, std::string outputfile, int batch\_size, int part\_size) : input\_file\_(inputfile), output\_file\_(outputfile), batch\_size\_(batch\_size), part\_size\_(part\_size) {  batch\_number\_ = 0;  part\_number\_ = 0;  }  void th\_LOADNUMBERS() {  std::ifstream in(input\_file\_);  if (!in.is\_open())  std::cout << "INPUT FILE WASN'T OPEN!" << std::endl;  int num;  while (true) {  if (batch\_.size() < batch\_size\_) {  if (in >> num) batch\_.push\_back(num);  else break;  } else if (buf\_.empty()) {  std::cout << "THREAD WITH #0 SEND BATCH TO THREAD 1 AND 2" << std::endl;  for (int i = 0; i < batch\_size\_; i++)  buf\_.push(batch\_[i]);  batch\_.clear();  }  }  end\_ = true;  in.close();  }  void th\_MEANPART(int th) {  while (!end\_ || !buf\_.empty()) {  std::vector<int> part;  mtx.lock();  while (part.size() < part\_size\_ && !buf\_.empty()) {  part.push\_back(buf\_.front()); buf\_.pop();  }  mtx.unlock();  double mean = 0;  for (int i = 0; i < part.size(); i++) {  mean += part[i];  }  if (mean != 0) {  mtx.lock();  mean\_.push(mean / part.size());  th\_num\_.push(th);  mtx.unlock();  }  }  }  void th\_OUTPUT() {  while (!end\_ || mean\_.size() != 0) {  std::ofstream out(output\_file\_, std::ios::app);  if (!out.is\_open())  std::cout << "OUTPUT FILE WASN'T OPEN!" << std::endl;  for (int i = 0; i < mean\_.size(); i++) {  std::cout << "THREAD WITH #" << th\_num\_.front() << " CALC MEAN: " << mean\_.front() << std::endl;  out << mean\_.front() << ' '; mean\_.pop(); th\_num\_.pop();  }  }  }  };  int main() {  srand(time(NULL));  std::string input = "input.txt";  std::string output = "output.txt";  int batch\_size = 50;  int part\_size = 6;  FILL\_FILE\_WITH\_RANDINT\_VALUES(input, 0, 100, 10009);  ThreadMean th\_mean(input, output, batch\_size, part\_size);  std::thread th1(&ThreadMean::th\_LOADNUMBERS, &th\_mean);  std::thread th2(&ThreadMean::th\_MEANPART, &th\_mean, 1);  std::thread th3(&ThreadMean::th\_MEANPART, &th\_mean, 2);  std::thread th4(&ThreadMean::th\_OUTPUT, &th\_mean);  th1.join(); th2.join(); th3.join(); th4.join();  } |

OUTPUT

|  |
| --- |
|  |