

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет «ИТМО» (НИУ ИТМО)

**Лабораторная работа №5 «C++ OOP/Parallel»**  
**по курсу «C++ и UNIX системы»**

**Выполнил студент 3 курса группы К3333:**

Буданцев А.А.

**Проверил:**

Маслов И.Д.

## Цель работы

Познакомить студента с принципами объектно-ориентированного программирования на примере создания сложной синтаксической структуры. Придумать синтаксис своего персонального мини-языка параллельного программирования, а также реализовать его разбор и вычисление.

## Задача

### 1. [C++ PARALLEL LANG] Создать параллельный язык программирования

Требуется создать язык программирования, в котором будет доступна установка следующих команд:

- Установка счетного цикла
- Вывод в консоль
- Вывод в файл в режиме добавления
- Арифметические операции +, -, \*, /

Счетный цикл должен поддерживать дальнейшую установку всех остальных поддерживаемых команд.

Для реализации задачи использовать технологию объектно-ориентированного программирования в части реализации поддерживаемых команд языка.

**В программе должны быть отражены следующие шаги:**

1.1. Текстовый ввод команд. Каждая новая строка – это новый набор команд.

1.2. Ожидание команды на окончание ввода

1.3. Параллельное исполнение введенных строк (наборов команд). Наборы команд должны исполняться параллельно. В консоли фиксировать время запуска / завершения каждого потока. При выводе информации о времени указывать принадлежность потока к строке (набору команд)

### 2. [SAVE] Результат всех вышеперечисленных шагов сохранить в репозиторий (+ отчет по данной ЛР в папку doc)

Фиксацию ревизий производить строго через ветку dev. С помощью скриптов накатить ревизии на stg и на prd. Скрипты разместить в корне репозитория. Также создать скрипты по возврату к виду текущей ревизии (даже если в папке имеются несохраненные изменения + новые файлы).

# Решение

## 1. [C++ PARALLEL LANG] Создать параллельный язык программирования

При написании использовались различные файлы

FileHandler.h

```
#pragma once
#include <iostream>
#include <fstream>
#include <vector>
#include <string>

class FileHandler {
public:
    FileHandler(char *name);
    void handle();
    std::vector<std::string> getCommands();

private:
    std::ifstream file;
    char *file_name;
    std::vector<std::string> commands;

    void open();
    void close();
    bool isOpen();
    void read();
};
```

Threader.h

```
#pragma once
#include "../Parser/Parser.h"

class Threader {
private:
    const int N = 5;
    std::vector<std::string> thread_commands;
    Parser parser;
public:
    Threader(std::vector<std::string> commands);
    void executeThread();
};
```

Object.h

```
#pragma once
#include <string>
#include <map>
#include <iostream>

class Object{
public:
    void insert(std::string key, std::string value);
    std::string getValue(std::string name);
    bool findKey(std::string key);
protected:
    bool isValidKey(std::string key);
    bool isEmpty(std::string value);
    bool isString(std::string value);
private:
    std::map<std::string, std::string> data;
};
```

Integer.h

```
#pragma once
#include "../Object/Object.h"

class Integer: public Object {
private:
    std::map<std::string, int> data;
public:
    void insert(std::string key, std::string value);
    int getValue(std::string name);
    void setValue(std::string name, int value);
    bool findKey(std::string key);
};
```

## Parser.h

```
#pragma once
#include <fstream>
#include <thread>
#include <chrono>
#include <iomanip>
#include <vector>
#include "../Integer/Integer.h"

class Parser {
public:
    static void parse(std::string command, int id);
private:
    static void executeCommands(std::string line);
    static void distributeCommand(std::string command, Object &obj, Integer &int_obj);
    static void getLoopBody(std::string &loop);
    static bool findFirstSubstring(std::string str, std::string substr);
    static void caseCommands(std::string command, Object &obj, Integer &int_obj);
    static std::vector<std::string> split(std::string line, char separator);
    static bool isNumeric(std::string value);
    static void dropSpaces(std::string &line);
    template<typename Type>
    static void initType(std::string command, Type &obj);
    static int getNumber(std::string str, Integer obj);
    static void calculations(std::string command, Integer &int_obj);
    static std::string findBetweenSymbols(std::string line, std::string first_symbol, std::string last_symbol);
    static std::string getLineInBound(std::string value, std::string first, std::string last);
    static void print(std::string command, Object obj, Integer int_obj);
    static void dropBounds(std::string &str);
    static void writeFile(std::string command);
    static std::vector<int> getLoopParams(std::string command);
};
```

Главный файл, который запускает весь код:

```
#include "FileHandler/FileHandler.h"
#include "Threader/Threader.h"

using namespace std;

int main(int argc, char *argv[]) {
    if (argc == 2) {
        FileHandler file_handler(argv[1]);
        file_handler.handle();
        Threader threader(file_handler.getCommands());
        threader.executeThread();
    }
    else {
        cout << "Error! Wrong number of arguments" << "\n";
    }

    return 0;
}
```

Результат выполнения команды

```
1
Thread 0 execution time: 0.090
110
11
Thread 3 execution time: 0.506
11
Thread 2 execution time: 0.618
cnhjrf
Thread 1 execution time: 0.952
Data is written successful.
Thread 4 execution time: 1.295
-130
Thread 6 execution time: 0.044
1
2
-2
Thread 5 execution time: 0.035
4
Thread 7 execution time: 0.094
Thread 8 execution time: 0.002
```

## Вывод

В лабораторной работе создан параллельный язык программирования с использованием объектно-ориентированного программирования.