

Санкт-Петербургский национальный  
исследовательский институт информационных  
технологий, механики и оптики

## Физический факультет



---

### Рабочий протокол и отчет по лабораторной работе №1

#### [C++ & UNIX]: ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

---

Группа: Z33431

Студент: Пуговкин Владимир

Преподаватель: Маслов И.Д.

## 1 Цель работы

Познакомить студента с основными алгоритмическими конструкциями, которые будут использоваться для создания CLI программы. Далее продемонстрировать эффективность использования механизма рекурсии. C++ алгоритмы: CLI Калькулятор вещественных чисел  $+$ ,  $-$ ,  $*$ ,  $/$ . Реализация с использованием только функций, условий, циклов,  $+$  и  $-$ . Вид команд в консоли: `calc plus / minus / power`; Ханойская башня, результат корректной последовательности

- Содержание файла `calc.cpp`

```

1  #include "calc.hpp"
2  #include <vector>
3  #include <iostream>
4  using namespace std;
5
6  int main(int argc, char* argv[])
7  {
8      if(argc < 4) cerr<<"Input error"<<endl;
9      char op = *argv[argc-1];
10     vector<double> vecnew;
11     try
12     {
13         for (int i = 1; i < argc-1; ++i)
14         {
15             vecnew.push_back(convertToDouble(argv[i]));
16         }
17
18         if(op == '^')
19         {
20             for (int i = 1; i < argc-1; ++i)
21             {
22                 if(static_cast<int>(vecnew[i]) != vecnew[i])
23                     throw invalid_argument("Славянский зажим яйцами");
24             }
25             cout << powf(vecnew[0], static_cast<int>(vecnew[1])) << endl;
26
27             return 0;
28         }
29         if(op == '+') cout << sum(vecnew[0], vecnew[1]) << endl;
30         if(op == '*') cout << multiply(vecnew[0], vecnew[1]) << endl;
31         if(op == '-') cout << dev(vecnew[0], vecnew[1]) << endl;
32     }
33     catch (const invalid_argument& ex)
34     {
35         cerr << " Error: " << ex.what() << '\n';
36         return 1;
37     }
38     catch (const exception& ex2) {
39         cerr << "Error: " << ex2.what() << '\n';
40         return 1;
41     }
42     return 0;
43 }

```

- Содержание заголовочного файла calc.hpp, использовался .hpp вместо .h, так как это более современная версия .h, обычно используют такое разрешение заголовка, если в проекте нет языка C, в целом файл .hpp может взаимодействовать с C, но нужно тогда сделать обертку с помощью extern "C"
- Содержимое hanoi.cpp:

```

1  #include "hanoi.hpp"
2  int main() {
3      LinkedList list;
4      int n = 3; // Количество дисков
5      hanoi(n, 1, 3, list);
6      list.printList();
7      return 0;
8  }

```

## 2 Рекурсия как стек

Стек ограничен только внешней памятью нашего компьютера.

## 3 Вывод

Рекурсия удобный способ реализации программ, так как это занимает маленькое количество места и увеличивает читаемость кода (зачастую). Кроме того, в лабораторной работе была произведена работа с указателями и классами. Также ознакомился с структурой двусвязного списка.