

**Правительство Российской Федерации  
Национальный исследовательский университет  
«Высшая школа экономики»**

Факультет компьютерных наук

Департамент программной инженерии

Минипроект “Task 02”

Вариант 2

Выполнил

Студент группы БПИ193

Цыдыпов Агван

[avtsydypov@edu.hse.ru](mailto:avtsydypov@edu.hse.ru)

Москва 2020

## Оглавление

1. Задание .....	3
2. Документация.....	4
2.1 Пример работы.....	4
2.2 Методы.....	4
2.3 Формат ввода-вывода.....	4
3 Текст программы.....	5
4 источники.....	7

## 1. Задание

2. Задача о Винни-Пухе или правильные пчелы. В одном лесу живут  $n$  пчел и один медведь, которые используют один горшок меда, вместимостью  $H$  глотков. Сначала горшок пустой. Пока горшок не наполнится, медведь спит. Как только горшок заполняется, медведь просыпается и съедает весь мед, после чего снова засыпает. Каждая пчела многократно собирает по одному глотку меда и кладет его в горшок. Пчела, которая приносит последнюю порцию меда, будит медведя. Создать многопоточное приложение, моделирующее поведение пчел и медведя.

## 2. Документация

По условию: Требуется создать многопоточное приложение.

О программе : Программа симулирует совместную работу пчел (каждая пчела отдельно работающий поток), когда отдельная пчела находит мёд, то к общему количеству глотков в чаше добавляется один глоток. Для симуляции времени поиска меда пчелами была использована встроенная библиотека <chrono>, для пчел задержка в промежутке от [0,10] секунд, для медведя задержка ровно 1 секунда (симуляция глотка). Когда пчелы заполняют чашу, пчелы прячутся, просыпается медведь и начинает глотать мед. Вся информация о происходящем выводится в консоль.

### 2.1 Пример работы

Результат работы программы:

Программа выводит в консоль информацию о том, что происходит с пчелами и медведем.

### 2.2 Методы

Методы используемые в программе:

1. GetNumber: метод принимающий 3 аргумента, правую границу генерации, левую границу генерации, строку с названием переменной для удобства.
2. Bear: симулирует медведя.

### 2.3 Формат ввода-вывода

Формат ввода:

Вводятся 2 числа:

- 1) Количество пчел
- 2) Максимальное количество глотков в чаше

Формат вывода:

Постоянно выводит текущее состояние всех существ

### 3 Текст программы

```
#include <iostream>
#include <mutex>
#include <thread>
#include <chrono>
#include <string>
#include <omp.h>
#include <time.h>
size_t threads;

std::mutex mtx;

int GetNumber(int left, int right, std::string str)
{
    int temp;
    while (true)
    {
        std::cout << "введите число " << str << " [" << left << ";" << right << "]"
<< std::endl;
        std::cin >> temp;
        if (temp < left || temp > right)
            continue;
        else
            break;
    }
    return temp;
}

void Bear(int& a)
{
    do {
        std::this_thread::sleep_for(std::chrono::milliseconds(1000));
        std::cout << "Медведь делает глоток " << "\n";
        std::lock_guard<std::mutex> guard(mtx);
        a -= 1;
    } while (true);
}

int main()
{
    srand(time(NULL));
    setlocale(LC_ALL, "ru");
    int h = 0;
    std::cout << "Количество пчел = n\n";
    threads = GetNumber(1,10, "n =");
    std::cout << "n = " << threads << std::endl;
    int H;

    std::cout << "Количество глотков в чаше = H\n";
    H = GetNumber(1,100, "H =");
    std::cout << "H = " << H << std::endl;
    int r = rand();
    omp_set_num_threads(threads);
#pragma omp parallel for num_threads(threads)
    for (int i = 0; i < H; i++)
    {
        int time = rand() * r % 1000;
        std::this_thread::sleep_for(std::chrono::milliseconds(time));
        double t = time;
        std::cout << "Пчела_" << std::to_string(omp_get_thread_num()) << " нашла
глоток меда за " << t / 1000 << " секунд\n";
        std::lock_guard<std::mutex> guard(mtx);
        h++;
        std::cout << "глотков в чаше = " << h << std::endl;;
    }
}
```

```
}

    std::cout << "Медведь проснулся\n";
    std::cout << "Пчелы спрятались\n";
    std::cout << "глотков в чаше = " << h << std::endl;
std::thread bear(Bear, std::ref(h));
while (true)
{
    if (h == 0)
        break;
    std::this_thread::sleep_for(std::chrono::milliseconds(1000));
    std::cout << "глотков в чаше = " << h << std::endl;
}
bear.detach();
std::cout << "медведь засыпает";

return 0;
}
```

#### 4 ИСТОЧНИКИ

- 1) <https://docs.microsoft.com/ru-ru/cpp/parallel/openmp/reference/openmp-directives?view=msvc-160> – смотрел OpenMP