

Семинар #4: Шаблоны. Класс `std::vector`. Домашнее задание.

Задача 1. Сумма чётных

Напишите функцию `int sumEven(const std::vector<int>& v)`, которая будет принимать вектор по константной ссылке и возвращать сумму всех чётных чисел этого вектора.

аргумент	возвращаемое значение
{4, 8, 15, 16, 23, 42}	70

Задача 2. Последние цифры

Напишите функции которые должны будут принимать на вход вектор и заменять все его элементы на последние цифры этих элементов. Все функции должны делать одно и то же, но должны будут возвращать результат разными способами.

- Функция `std::vector<int> lastDigits1(const std::vector<int>& v)` должна возвращать вектор.
- Функция `void lastDigits2(std::vector<int>& v)` должна принимать вектор по ссылке и изменять его.
- Функция `void lastDigits3(std::vector<int>* pv)` должна принимать указатель, хранящий адрес некоторого вектора. Функция должна изменять вектор, на который указывает указатель.
- Функция `void lastDigits4(std::span<int> sp)` должна принимать объект типа `std::span` и менять вектор, на который указывает этот `span`.

Задача 3. Простые делители

Напишите функцию `std::vector<std::pair<int, int>> factorization(int n)`, которая будет находить все простые делители числа `n` и их количества.

аргумент	возвращаемое значение
60	{{2, 2}, {3, 1}, {5, 1}}
626215995	{{3, 3}, {5, 1}, {17, 1}, {29, 1}, {97, 2}}
107	{{107, 1}}
1	{{1, 1}}

Задача 4. Времена из строки

- Напишите простой класс для работы со временем:

```
class Time {
private:
    int mHours, mMinutes, mSeconds;
public:
    Time(int hours, int minutes, int seconds);
    Time(std::string_view s); // строка в формате "hh:mm:ss"
    Time operator+(Time b) const;
    int hours() const; int minutes() const; int seconds() const;
    friend std::operator<<(std::ostream& out, Time t);
};
```

- Напишите функцию `std::vector<Time> getTimesFromString(std::string_view s)`, которая будет принимать строку в формате `"hh:mm:ss ... hh:mm:ss"`, где за место букв должны стоять некоторые числа. Например, строка может иметь вид `"12:20:05 05:45:30 22:10:45"`. Функция должна возвращать вектор времен, соответствующий временам в строке.
- Напишите функцию `Time sumTimes(const std::vector<Time>& v)`, которая будет суммировать все времена и возвращать эту сумму.

Задача 5. Максимум

Напишите шаблонную функцию, которая будет принимать на вход вектор и возвращать максимальный элемент вектора.

Задача 6. Разбиение на пары

Напишите шаблонную функцию, которая будет принимать на вход контейнер (`vector`, `array`, `string` или `string_view`) и возвращать вектор пар нечётных и чётных элементов. Протестировать функцию можно в файле `code/test_pairing.cpp`.