МИНИСТЕРСТВО НАУКИ и высшего образования

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**Отчёт по лабораторной работе № 8**

**по учебной дисциплине «Основы алгоритмизации программирования»**

**Тема: «Работа со строками в C++. Потоки ввода-вывода. Файловые операции»**

Выполнил студент

специальности 09.02.07

Информационные системы и

программирование

II курса группы 22919/22

Долгополов Владислав

Валерьевич

Преподаватель

Молькова Лолита Юрьевна

Санкт-Петербург,

2024

Цель работы: Цель работы – ознакомиться с возможностями ввода-вывода языка C++, освоить основные операции работы со строками и файлами.

Задание:

Написать программу, которая считывает текст из файла и выводит на экран только предложения, не содержащие запятых.

**Ход работы**

**Алгоритм**

**Код программы**

#include <cctype>

#include <fstream>

#include <iostream>

#include <string>

#include <vector>

using namespace std;

*vector*<*string*> *split*(const *string* &*text*) {

  vector<string> sentences;

  string sent1;

  for (char ch : text) {

    sent1 += ch;

    if (ch == '.' || ch == '!' || ch == '?') {

      sentences.*push\_back*(sent1);

      sent1.*clear*();

    }

  }

  return sentences;

}

bool *is\_comma*(const *string* &*sentence*) {

  for (char ch : sentence) {

    if (ch == ',') {

      return true;

    }

  }

  return false;

}

int *main*() {

*setlocale*(LC\_ALL, "Russian");

  string filename = "file.txt";

  ifstream *inputFile*(filename);

  string text;

  string line;

  while (*getline*(inputFile, line)) {

    text += line + "\n";

  }

  inputFile.*close*();

  vector<string> sentences = *split*(text);

  for (const string &sentence : sentences) {

    if (!*is\_comma*(sentence)) {

      cout << sentence << endl;

    }

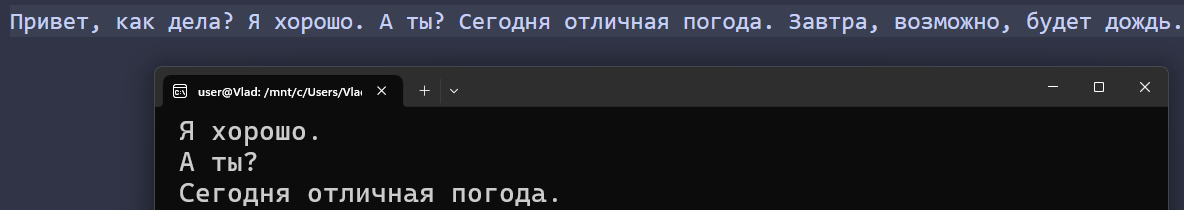
  }

  return 0;

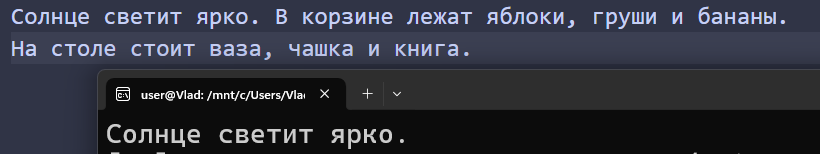
}

**Результат**

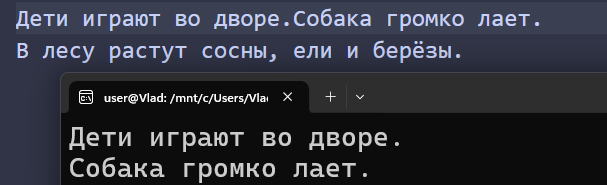
**Рис1**



**Рис2**



**Рис3**



**Контрольные вопросы**

1) Перечислите стандартные потоки ввода-вывода.

cin (istream): стандартный ввод (клавиатура).

cout (ostream): стандартный вывод (экран).

cerr (ostream): стандартный вывод ошибок (без буферизации).

clog (ostream): стандартный вывод ошибок (с буферизацией).

2) Как создать файловый поток?

Для ввода: ifstream inputFile("filename.txt");

Для вывода: ofstream outputFile("filename.txt");

Для ввода и вывода: fstream file("filename.txt", ios::in | ios::out);

3) Какие режимы открытия файлов существуют?

ios::in: открытие для чтения.

ios::out: открытие для записи.

ios::binary: двоичный режим.

ios::ate: установка указателя на конец файла при открытии.

ios::app: запись в конец файла.

ios::trunc: удаление содержимого файла, если он существует.

4) Что такое двоичный режим ввода-вывода?

означает, что данные читаются и записываются без преобразований (например, без замены символов CR/LF на LF). Используется для работы с сырыми данными.

5) Что такое текстовый режим ввода-вывода?

выполняет преобразования символов (например, замену CR/LF на LF). Подходит для работы с текстовыми фалами.

6) Как осуществляется чтение символов и строк?

Для символов:

int get();

istream& get(char& c);

Для строк:

istream& get(char\* buf, long len, char delim = '\n');

istream& getline(char\* buf, long len, char delim = '\n');

7) Как изменить текущую позицию в файле?

Для чтения: seekg(offset, ios::beg/ios::cur/ios::end);

Для записи: seekp(offset, ios::beg/ios::cur/ios::end);

8) Как узнать текущую позицию в файле?

Для чтения: tellg();

Для записи: tellp();

9) Для чего используется метод clear()?

сбрасывает флаги состояния потока (например, eofbit, failbit, badbit), позволяя продолжить работу с потоком после ошибки.