Учебный курс: Технологии программирования

Кафедра: ИС Факультет: ИТиП

Лабораторная работа №4. Классы

Выполнил: Трофимов В.А. Группа: 1511

Преподаватель: Повышев В.В.

# Задание

Согласно варианту описать указанные классы (Текстовый файловый «поток» для записи. Открытие файла. Закрытие файла. Получение имени файла. Проверка, открыт ли файл. Запись в файл строки. Запись в файл целого числа. Запись в файл вещественного числа). Написать программу, использующую описанные классы: инициализация переменных (ввод пользователя), выполнение действий с экземплярами класса (в зависимости от дальнейшего ввода пользователя).

Описания и реализация должны находиться в разных файлах. Доступ к полям класса – только через методы. Внешние функции для работы с данными класса не допускаются. Перегрузка стандартных арифметических операций для класса (только в виде методов класса) – в зависимости от задания и здравого смысла (уместна консультация с преподавателем практики).

# Входные/Выходные данные

## Консоль

CLI for file Printwriter program. Author V.A. Trofimov.

1. Assign and open a file.

2. Show the file name and path.

3. Show the file status.

4. Close a file.

5. Write something to file.

0. Exit.

>>1

Please enter the full path to the target file or just a filename. Notice: less than 256 characters.

c:/temp.txt

File succesfully opened.

1. Assign and open a file.

2. Show the file name and path.

3. Show the file status.

4. Close a file.

5. Write something to file.

0. Exit.

>>2

c:/temp.txt

1. Assign and open a file.

2. Show the file name and path.

3. Show the file status.

4. Close a file.

5. Write something to file.

0. Exit.

>>3

File is open.

1. Assign and open a file.

2. Show the file name and path.

3. Show the file status.

4. Close a file.

5. Write something to file.

0. Exit.

>>5

1. Write an integer.

2. Write the float-point number.

3. Write a string.

0. Back to menu.

>>1

Enter a value.

>>123

Succesfully wrote.

1. Assign and open a file.

2. Show the file name and path.

3. Show the file status.

4. Close a file.

5. Write something to file.

0. Exit.

>>5

1. Write an integer.

2. Write the float-point number.

3. Write a string.

0. Back to menu.

>>2

Enter a value.

>>1.01

Succesfully wrote.

1. Assign and open a file.

2. Show the file name and path.

3. Show the file status.

4. Close a file.

5. Write something to file.

0. Exit.

>>5

1. Write an integer.

2. Write the float-point number.

3. Write a string.

0. Back to menu.

>>3

Please enter a string. Notice: less than 256 characters.

>>test

Succesfully wrote.

1. Assign and open a file.

2. Show the file name and path.

3. Show the file status.

4. Close a file.

5. Write something to file.

0. Exit.

>>4

File succsesfully closed.

1. Assign and open a file.

2. Show the file name and path.

3. Show the file status.

4. Close a file.

5. Write something to file.

0. Exit.

>>0

# Исходный текст

## Файл main.cpp

// Course: Programming technologies (C Plus Plus)

// Lab 4. Classes.

// Student: Trofimov V.A. Group: 1511

// Teacher: Povishev V.V.

// Created 21.02.2013 Modified: 22.02.2013

// Description: Program entry point.

#include "PrintWriter.h"

#include "Menu.h"

using namespace std;

int main() {

Menu();

return 0;

}

## Файл PrintWriter.h

// Course: Programming technologies (C Plus Plus)

// Lab 4. Classes.

// Student: Trofimov V.A. Group: 1511

// Teacher: Povishev V.V.

// Created 21.02.2013 Modified: 22.02.2013

// Description: PrintWriter header file.

#pragma once

#include <string>

using namespace std;

class PrintWriter {

private:

string filename;

bool opened;

FILE \* file;

public:

PrintWriter();

PrintWriter(char \* filename);

int open(char \* filename);

int close();

string getName();

bool isOpen();

void Flush();

int print(char \* input);

int println(char \* input);

int print(char input);

int println(char input);

int print(unsigned char input);

int println(unsigned char input);

int print(int input);

int println(int input);

int print(unsigned int input);

int println(unsigned int input);

int print(short input);

int println(short input);

int print(unsigned short input);

int println(unsigned short input);

int print(long input);

int println(long input);

int print(unsigned long input);

int println(unsigned long input);

int print(long long input);

int println(long long input);

int print(unsigned long long input);

int println(unsigned long long input);

int print(float input);

int println(float input);

int print(double input);

int println(double input);

void ShowError(int ErrorCode);

};

## Файл PrintWriter.cpp

// Course: Programming technologies (C Plus Plus)

// Lab 4. Classes.

// Student: Trofimov V.A. Group: 1511

// Teacher: Povishev V.V.

// Created 21.02.2013 Modified: 22.02.2013

// Description: PrintWriter source code file.

#include "PrintWriter.h"

#include <cstring>

PrintWriter::PrintWriter() {

filename = "";

opened = false;

file = NULL;

}

PrintWriter::PrintWriter(char \* filename) {

this -> filename = filename;

file = fopen(filename, "w");

opened = file;

}

int PrintWriter::open(char \* filename) {

if (opened) {

fflush(file);

fclose(file);

}

this -> filename = filename;

file = fopen(filename, "w");

opened = file;

if (opened) return 0;

else return -1;

}

int PrintWriter::close() {

filename = "";

file = NULL;

opened = false;

if (opened) return fclose(file);

else return 0;

}

string PrintWriter::getName() {

return filename;

}

bool PrintWriter::isOpen() {

return opened;

}

int PrintWriter::print(char \* input) {

if (this -> opened) return fprintf(file, "%s", input);

else return -2;

}

int PrintWriter::println(char \* input) {

if (this -> opened) return fprintf(file, "%s\n", input);

else return -2;

}

int PrintWriter::print(char input) {

if (this -> opened) return fprintf(file, "%d", input);

else return -2;

}

int PrintWriter::println(char input) {

if (this -> opened) return fprintf(file, "%d\n", input);

else return -2;

}

int PrintWriter::print(unsigned char input) {

if (this -> opened) return fprintf(file, "%u", input);

else return -2;

}

int PrintWriter::println(unsigned char input) {

if (this -> opened) return fprintf(file, "%u\n", input);

else return -2;

}

int PrintWriter::print(int input) {

if (this -> opened) return fprintf(file, "%d", input);

else return -2;

}

int PrintWriter::println(int input) {

if (this -> opened) return fprintf(file, "%d\n", input);

else return -2;

}

int PrintWriter::print(unsigned int input) {

if (this -> opened) return fprintf(file, "%u", input);

else return -2;

}

int PrintWriter::println(unsigned int input) {

if (this -> opened) return fprintf(file, "%u\n", input);

else return -2;

}

int PrintWriter::print(short input) {

if (this -> opened) return fprintf(file, "%d", input);

else return -2;

}

int PrintWriter::println(short input) {

if (this -> opened) return fprintf(file, "%d\n", input);

else return -2;

}

int PrintWriter::print(unsigned short input) {

if (this -> opened) return fprintf(file, "%u", input);

else return -2;

}

int PrintWriter::println(unsigned short input) {

if (this -> opened) return fprintf(file, "%u\n", input);

else return -2;

}

int PrintWriter::print(long input) {

if (this -> opened) return fprintf(file, "%ld", input);

else return -2;

}

int PrintWriter::println(long input) {

if (this -> opened) return fprintf(file, "%ld\n", input);

else return -2;

}

int PrintWriter::print(unsigned long input) {

if (this -> opened) return fprintf(file, "%lu", input);

else return -2;

}

int PrintWriter::println(unsigned long input) {

if (this -> opened) return fprintf(file, "%lu\n", input);

else return -2;

}

int PrintWriter::print(long long input) {

if (this -> opened) return fprintf(file, "%lld", input);

else return -2;

}

int PrintWriter::println(long long input) {

if (this -> opened) return fprintf(file, "%lld\n", input);

else return -2;

}

int PrintWriter::print(unsigned long long input) {

if (this -> opened) return fprintf(file, "%llu", input);

else return -2;

}

int PrintWriter::println(unsigned long long input) {

if (this -> opened) return fprintf(file, "%llu\n", input);

else return -2;

}

int PrintWriter::print(float input) {

if (this -> opened) return fprintf(file, "%f", input);

else return -2;

}

int PrintWriter::println(float input) {

if (this -> opened) return fprintf(file, "%f\n", input);

else return -2;

}

int PrintWriter::print(double input) {

if (this -> opened) return fprintf(file, "%lf", input);

else return -2;

}

int PrintWriter::println(double input) {

if (this -> opened) return fprintf(file, "%lf\n", input);

else return -2;

}

void PrintWriter::ShowError(int ErrorCode) {

if (ErrorCode == -1) printf("%s\n", "File could not be created.");

else if (ErrorCode == -2) printf("%s\n", "There is no file to write in.");

else if (ErrorCode == 0) printf("%s\n", "This is not an error.");

else if (ErrorCode < 0) printf("%s\n", "Error while writing.");

}

void PrintWriter::Flush() {

fflush(file);

}

## Файл Menu.h

// Course: Programming technologies (C Plus Plus)

// Lab 4. Classes.

// Student: Trofimov V.A. Group: 1511

// Teacher: Povishev V.V.

// Created 21.02.2013 Modified: 22.02.2013

// Description: Menu header file.

#pragma once

#include "PrintWriter.h"

class Menu {

private:

PrintWriter pw;

void decimal();

void floatPoint();

bool opened();

public:

Menu();

void openFile();

void closeFile();

void getName();

void isOpen();

void write();

};

## Файл Menu.cpp

// Course: Programming technologies (C Plus Plus)

// Lab 4. Classes.

// Student: Trofimov V.A. Group: 1511

// Teacher: Povishev V.V.

// Created 21.02.2013 Modified: 22.02.2013

// Description: Menu source code file.

#include "Menu.h"

#include <stdlib.h>

#include <conio.h>

#include <stdio.h>

#include <iostream>

#include "PrintWriter.h"

void Menu::decimal() {

printf("\n");

printf("Enter a value.\n");

char \* value = new char[256];

int err; char c = 0;

do {

scanf("%s", value);

try {

throw atoi(value);

}

catch (int res) {

if (res != 0) {

err = pw.print(res);

if (err > 0) cout << "Succesfully wrote." << endl;

else pw.ShowError(err);

delete(value); return;

} else {

printf("%s\n", "You entered an incorrect value or value contains zero. Is value a zero?(Y/n)");

c = getch();

if (c == 13 || c == 'y' || c == 'Y' || c == 'н' || c == 'Н') {

err = pw.print(res);

if (err > 0) cout << "Succesfully wrote." << endl;

else pw.ShowError(err);

delete(value); return;

} else printf("Please try again.\n");

}

}

} while (c != 13 && c != 'y' && c != 'Y');

delete(value);

}

void Menu::floatPoint() {

printf("\n");

printf("Enter a value.\n");

char \* value = new char[256];

int err; char c;

do {

scanf("%s", value);

try {

throw atof(value);

}

catch (double res) {

if (res != 0L) {

err = pw.print(res); pw.Flush();

if (err > 0) cout << "Succesfully wrote." << endl;

else pw.ShowError(err);

delete(value); return;

}

else {

printf("%s\n", "You entered an incorrect value or value contains zero. Is value a zero?(Y/n)");

c = getch();

if (c == 13 || c == 'y' || c == 'Y') {

err = pw.print(res); pw.Flush();

if (err > 0) cout << "Succesfully wrote." << endl;

else pw.ShowError(err);

delete(value); return;

}

else printf("Please try again.\n");

}

}

} while (c != 13 && c != 'y' && c != 'Y' && c != 'н' && c != 'Н');

delete(value);

}

void Menu::write() {

char c;

printf("\n");

printf("1. Write an integer.\n");

printf("2. Write the float-point number.\n");

printf("3. Write a string.\n");

printf("0. Back to menu.\n");

do {

c = getch();

switch(c) {

case 49 : {decimal(); break;};

case 50 : {floatPoint(); break;};

case 51 : {

printf("\n");

printf("Please enter a string. Notice: less than 256 characters.\n");

char \* temp = new char[256];

scanf("%255s", temp);

int err = pw.print(temp); pw.Flush();

delete(temp);

if (err > 0) cout << "Succesfully wrote." << endl;

else pw.ShowError(err);

break;

};

case 97 : {decimal(); break;};

case 98 : {floatPoint(); break;};

case 99 : {

printf("\n");

printf("Please enter a string. Notice: less than 256 characters.\n");

char \* temp = new char[256];

scanf("%255s", temp);

int err = pw.print(temp); pw.Flush();

delete(temp);

if (err > 0) cout << "Succesfully wrote." << endl;

else pw.ShowError(err);

break;

};

};

} while (c != 48 && c != 49 && c != 50 && c != 51 && c != 96 && c != 97 && c != 98 && c != 99);

}

Menu::Menu() {

char c;

printf("%s\n", "CLI for file Printwriter program. Author V.A. Trofimov.");

do {

printf("\n");

printf("1. Assign and open a file.\n");

printf("2. Show the file name and path.\n");

printf("3. Show the file status.\n");

printf("4. Close a file.\n");

printf("5. Write something to file.\n");

printf("0. Exit.\n");

do {

c = getch();

switch(c) {

case 49 : {openFile(); break;}

case 50 : {getName(); break;}

case 51 : {isOpen(); break;}

case 52 : {closeFile(); break;}

case 53 : {write(); break;}

case 97 : {openFile(); break;}

case 98 : {getName(); break;}

case 99 : {isOpen(); break;}

case 100 : {closeFile(); break;}

case 101 : {write(); break;}

};

} while (c != 49 && c != 50 && c != 51 && c != 52 && c != 53 && c != 96 &&

c != 97 && c != 98 && c != 99 && c != 100 && c != 101 && c != 48 );

} while (c != 48 && c != 96);

}

void Menu::openFile() {

printf("\n");

printf("Please enter the full path to the target file or just a filename. Notice: less than 256 characters.\n");

char \* temp = new char[256];

scanf("%255s", temp);

int err = pw.open(temp);

delete(temp);

if (err == 0) cout << endl << "File succesfully opened." << endl;

else pw.ShowError(err);

}

void Menu::closeFile() {

printf("\n");

if (opened()) {

int err = pw.close();

if (err == 0) cout << "File succsesfully closed." << endl;

else pw.ShowError(err);

} else cout << "No File - no closing." << endl;

}

void Menu::getName() {

printf("\n");

cout << ((pw.getName().compare("") == 0) ? "File have not name or path." : pw.getName()) << endl;

}

void Menu::isOpen() {

printf("\n");

cout << ((pw.isOpen()) ? "File is open." : "File is not open.") << endl;

}

bool Menu::opened() {

return pw.isOpen();

}