**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Программирование на основе классов и шаблонов»

Отчет по лабораторной работе №8

«Стандартная библиотека шаблонов»

| Выполнил: |  | Проверил: |
| --- | --- | --- |
| студент группы ИУ5-24б |  | преподаватель каф. ИУ5 |
| Чернев Н.А. |  |  |
| Подпись и дата: 29.03.2024 |  | Подпись и дата: |
|  |  |  |

Москва, 2024 г.

## Задание

1. Пользователь вводит текст. Переставить в нем первую букву первого предложения и первую букву второго предложения (сначала найти номер первой точки).
2. Пользователь вводит некоторый текст. Создать новый текст, включив в него только латинские буквы и цифры (не использовать find).
3. Дан текстовый файл. Запишите в другой файл только такие строки исходного файла, если они содержат кавычки.
4. Написать программе для ввода с клавиатуры массива строк (окончание ввода строк – пустая строка), которые записать в вектор. Распечатать введенный массив строк в столбик с указанием номера каждой строки. Удалить из вектора элемент с индексом 3 и снова распечатать вектор. Переделать программу: печатать вектор с помощью функции.
5. Создайте класс с именем Date, содержащий три закрытых поля типа int, предназначенные для хранения дня, месяца и года. (Подсказка: не забудьте для класса Date определить операторы < и ==). Один из конструкторов класса должен инициализировать поля датой 01.01.1970, а другой конструктор — заданным набором значений. Создайте метод класса, который будет выводить значения полей на экран в формате 11.05.2019. Создать массив объектов типа Date с помощью контейнера vector в main() и сразу занести в него информацию о шести датах: (1, 2, 1963), (14, 7, 1995), (7, 12, 2088), (5, 3, 2030), (24, 9, 2013), (19, 9, 2020). Вывести массив на экран таким образом, чтобы каждая дата выводились в отдельной строке. Создайте новый вектор, в который запишите даты первого вектора, которые относятся к будущему. Выведите на экран второй вектор. Напишите функцию вывода массива на экран.
6. Написать программе для ввода с клавиатуры массива строк (окончание ввода строк – пустая строка), которые записать в вектор. Распечатать введенный массив строк в столбик с указанием номера каждой строки. Удалить из вектора элемент с индексом 3 и снова распечатать вектор. Переделать программу: печатать вектор с помощью функции.
7. Создайте класс с именем Time, содержащий три закрытых поля типа int, предназначенные для хранения часов, минут и секунд. (Подсказка: не забудьте для класса Time определить операторы < и ==). Один из конструкторов класса должен инициализировать поля нулевыми значениями, а другой конструктор — заданным набором значений. Создайте метод класса, который будет выводить значения полей на экран в формате 11:59:59.  
   Создать список объектов типа Time с помощью контейнера list в main() и сразу занести в него информацию о шести датах: (1, 2, 63), (14, 57, 19), (7, 32, 20), (5, 13, 23), (10, 19, 45), (19, 9, 59). Вывести список на экран таким образом, чтобы каждая дата выводились в отдельной строке. Создайте новый список, в который запишите даты первого списка, которые относятся к ночи (с 0 до 5 часов). Выведите на экран второй список. Напишите функцию вывода списка на экран.
8. Создать отображение Inventory, которое содержит данные о названии товара и количестве его на складе. Занести в него информацию: ("Отверка", 0), ("Молоток", 10), ("Гайки", 100), ("Профиль", 0), ("Уголок", 9), ("Доска", 17). Вывести отображение на экран таким образом, чтобы каждый товар выводился в отдельной строке, и только такой товар, у которого количество не равно нулю:

Гайки - 100

Доска – 17  
 Молоток - 10  
 Уголок - 9

1. Дано отображение с данными о количестве учащихся в разных классах школы:

{'1а': 21, '1б':20, '2':25, '3а': 20, '3б': 23, '4а':25, '4б':25, '5а': 26, '5б':25, '6a':19, '6б': 18, '7а':26, '7б':25, '8':25, '9':17, '10': 15, '11’: 14}

В программе определить количества учащихся в школе. Распечатать содержимое отображения в консоли, используя функцию.

**Текст программы**

n1.cpp

#include <iostream>

int main() {

std::string str;

getline(std::cin, str);

int ind\_point = str.find('.');

int ind1 = 0, ind2 = ind\_point + 1;

while (str[ind1] == ' ') {

ind1++;

}

while (str[ind2] == ' ') {

ind2++;

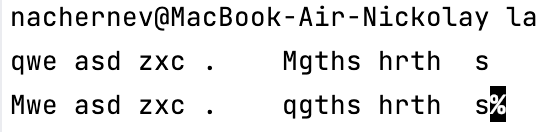
}

std::swap(str[ind1], str[ind2]);

std::cout << str;

return 0;

}



n2.cpp

#include <iostream>

int main() {

std::string str, ans;

getline(std::cin, str);

for (int i = 0; i < str.size(); i++) {

if (isalnum(str[i])) {

ans += str[i];

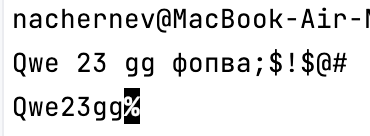
}

}

std::cout << ans;

return 0;

}



n3.cpp

#include <fstream>

int main() {

std::ifstream fin("n3.txt");

std::ofstream fout("n3\_ans.txt");

std::string str;

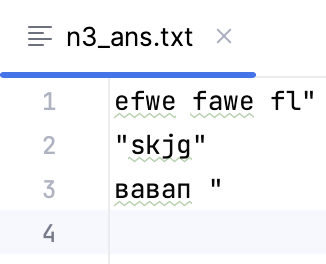
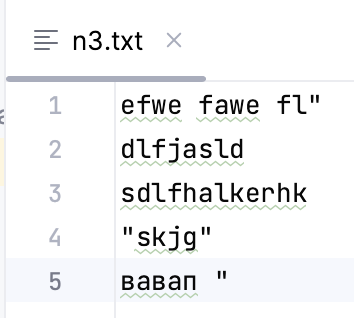
while (getline(fin, str)) {

if (str.find('"') != -1) fout << str << '\n';

}

return 0;

}



n4.cpp

#include <iostream>

void print\_vector(std::vector<std::string> v) {

for (int i = 0; i < v.size(); i++) std::cout << i << ": " << v[i] << '\n';

}

int main() {

std::vector<std::string> v;

std::string str;

getline(std::cin, str);

while (!str.empty()) {

v.push\_back(str);

getline(std::cin, str);

}

print\_vector(v);

std::cout << '\n';

if (v.size() <= 3) {

std::cout << "Too small vector to delete 4th element\n";

return 0;

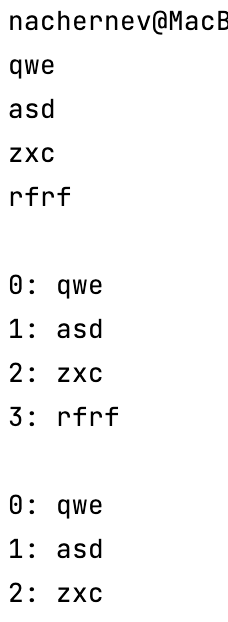
}

v.erase(v.begin() + 3);

print\_vector(v);

return 0;

}



n5.cpp

#include <iostream>

#include <iomanip>

class Date{

int day;

int month;

int year;

public:

Date() : day(1), month(1), year(1970) {}

Date(int d, int m, int y) : day(d), month(m), year(y) {}

void print() {

std::cout << std::setw(2) << std::setfill('0') << day << '.' << std::setw(2) << month << '.' << std::setw(4) << year << std::endl;

}

bool operator == (const Date &date) {

return (day == date.day && month == date.month && year == date.year);

}

bool operator < (const Date &date) {

if (year == date.year) {

if (month == date.month) {

return day < date.day;

}

return month < date.month;

}

return year < date.year;

}

};

void print\_vector(std::vector<Date> v) {

for (Date i : v) i.print();

}

int main() {

Date cur\_date(21, 3, 2024);

std::vector<Date> dates(6), new\_dates;

dates[0] = Date(1, 2, 1963);

dates[1] = Date(14, 7, 1995);

dates[2] = Date(7, 12, 2088);

dates[3] = Date(5, 3, 2030);

dates[4] = Date(24, 9, 2013);

dates[5] = Date(19, 9, 2020);

print\_vector(dates);

for (Date i : dates) {

if (cur\_date < i || cur\_date == i) new\_dates.push\_back(i);

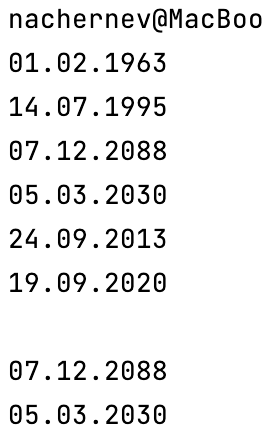
}

std::cout << std::endl;

print\_vector(new\_dates);

return 0;

}



n6.cpp

#include <iostream>

#include <list>

void print\_list(std::list<std::string> list) {

int i = 0;

for (auto val = list.begin(); val != list.end(); val++, i++) {

std::cout << i << ": " << \*val << '\n';

}

}

int main() {

std::list<std::string> list;

std::string str;

getline(std::cin, str);

while (!str.empty()) {

list.push\_back(str);

getline(std::cin, str);

}

print\_list(list);

std::cout << '\n';

if (list.size() <= 3) {

std::cout << "Too small list to delete 4th element\n";

return 0;

}

auto iter = list.begin();

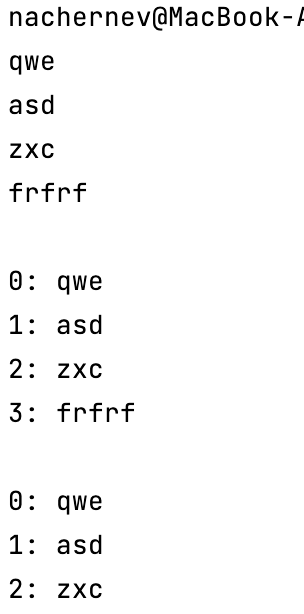
++iter;++iter;++iter;

list.erase(iter);

print\_list(list);

return 0;

}



n7.cpp

#include <iostream>

#include <iomanip>

#include <list>

class Time{

int second;

int minute;

int hour;

public:

Time() : second(0), minute(0), hour(0) {}

Time(int hr, int min, int sec) {

min += sec / 60;

sec %= 60;

hr += min / 60;

min %= 60;

hr %= 24;

second = sec;

minute = min;

hour = hr;

}

void print() {

std::cout << std::setw(2) << std::setfill('0') << hour << ':' << std::setw(2) << minute << ':' << std::setw(2) << second << std::endl;

}

bool operator < (const Time &time) {

if (hour == time.hour) {

if (minute == time.minute) {

return second < time.second;

}

return minute < time.minute;

}

return hour < time.hour;

}

bool operator == (const Time &time) {

return (hour == time.hour && minute == time.minute && second == time.second);

}

};

void print\_list(std::list<Time> list) {

for (auto i : list) {

i.print();

}

}

int main() {

std::list<Time> time, night\_time;

time.push\_back(Time(1, 2, 63));

time.push\_back(Time(14, 57, 19));

time.push\_back(Time(7, 32, 20));

time.push\_back(Time(5, 13, 23));

time.push\_back(Time(10, 19, 45));

time.push\_back(Time(19, 9, 59));

print\_list(time);

Time night\_st(0, 0, 0), night\_end(5, 0, 0);

for (auto i : time) {

if (night\_st < i && i < night\_end) night\_time.push\_back(i);

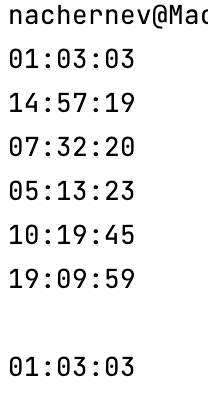
}

std::cout << '\n';

print\_list(night\_time);

return 0;

}



n8.cpp

#include <iostream>

#include <map>

void print\_map(std::map<std::string, int> map) {

for (auto i = map.begin(); i != map.end(); i++) {

if (i->second != 0) {

std::cout << i->first << " - " << i->second << '\n';

}

}

}

int main() {

std::map<std::string, int> inventory;

inventory["Отверка"] = 0;

inventory["Молоток"] = 10;

inventory["Гайки"] = 100;

inventory["Профиль"] = 0;

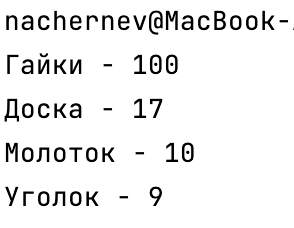
inventory["Уголок"] = 9;

inventory["Доска"] = 17;

print\_map(inventory);

return 0;

}



n9.cpp

#include <iostream>

#include <string>

#include <map>

void print\_map(std::map<std::string, int> map) {

for (auto i = map.begin(); i != map.end(); i++) {

if (i->second != 0) {

std::cout << i->first << " -> " << i->second << '\n';

}

}

}

int main() {

std::map<std::string, int> pupils\_cnt = **{**{"1а", 21}, {"1б", 20}, {"2", 25}, {"3а", 20}, {"3б", 23}, {"4а", 25}, {"4б", 25}, {"5а", 26}, {"5б", 25}, {"6a", 19}, {"6б", 18}, {"7а", 26},{"7б", 25}, {"8", 25}, {"9", 17}, {"10", 15}, {"11", 14}**}**;

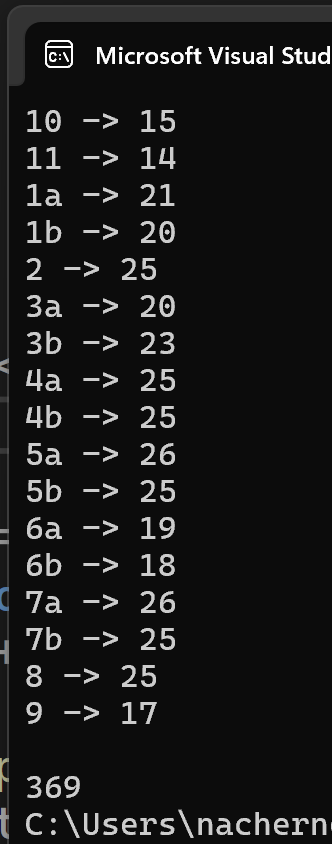
int ans = 0;

for (auto i = pupils\_cnt.begin(); i != pupils\_cnt.end(); i++) {

ans += i->second;

}

print\_map(pupils\_cnt);

std::cout << '\n' << ans;

}