Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа**

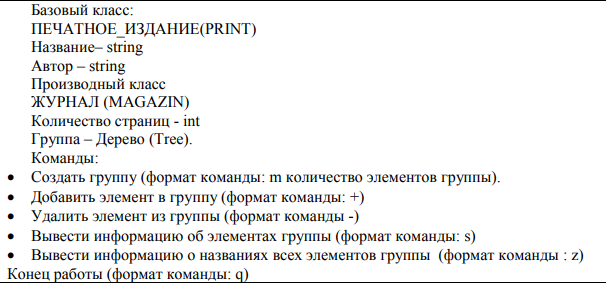
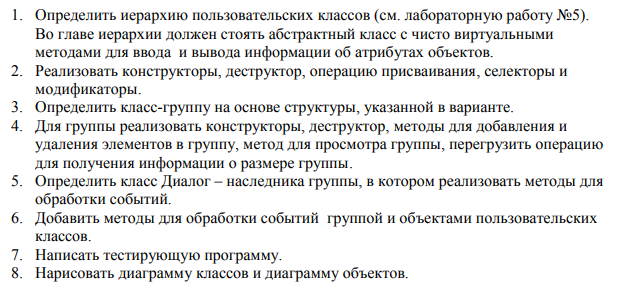
**"Классы №8”**

Выполнил:   
студент группы РИС-23-1б   
Агзамов Артур Альферович

Проверила:   
доцент кафедры ИТАС   
О.А. Полякова

2024 г.

**Задача:**



**Анализ задачи:**

**Код:**

#include <iostream>

#include <string>

#include <vector>

using namespace std;

class Print {

protected:

string title;

string author;

public:

Print() : title(""), author("") {}

Print(string t, string a) : title(t), author(a) {}

virtual ~Print() {}

virtual void input() = 0;

virtual void output() = 0;

void setTitle(string t) {

title = t;

}

string getTitle() const {

return title;

}

void setAuthor(string a) {

author = a;

}

string getAuthor() const {

return author;

}

};

class Magazine : public Print {

private:

int numPages;

public:

Magazine() : Print(), numPages(0) {}

Magazine(string t, string a, int np) : Print(t, a), numPages(np) {}

void input() override {

cout << "Enter title: ";

cin >> title;

cout << "Enter author: ";

cin >> author;

cout << "Enter number of pages: ";

cin >> numPages;

}

void output() override {

cout << "Title: " << title << endl;

cout << "Author: " << author << endl;

cout << "Number of pages: " << numPages << endl;

}

};

class Tree {

private:

vector<Print\*> elements;

public:

Tree() {}

void addElement(Print\* p) {

elements.push\_back(p);

}

void removeElement(int index) {

if (index >= 0 && index < elements.size()) {

delete elements[index];

elements.erase(elements.begin() + index);

}

}

void showInfo() {

for (Print\* p : elements) {

p->output();

}

}

int getSize() const {

return elements.size();

}

friend int main();

};

int main() {

Tree group;

char choice;

int numElements;

cout << "Enter number of elements for the group: ";

cin >> numElements;

for (int i = 0; i < numElements; i++) {

Magazine\* mag = new Magazine();

mag->input();

group.addElement(mag);

}

do {

cout << "Menu:" << endl;

cout << "1. Add element to the group (+)" << endl;

cout << "2. Remove element from the group (-)" << endl;

cout << "3. Show information about elements in the group" << endl;

cout << "4. Show names of all elements in the group" << endl;

cout << "5. Quit (q)" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case '1': {

Magazine\* mag = new Magazine();

mag->input();

group.addElement(mag);

break;

}

case '2': {

int index;

cout << "Enter index of element to remove: ";

cin >> index;

group.removeElement(index);

break;

}

case '3':

group.showInfo();

break;

case '4':

for (Print\* p : group.elements) {

cout << p->getTitle() << endl;

}

break;

case '5':

break;

default:

cout << "Invalid choice. Try again." << endl;

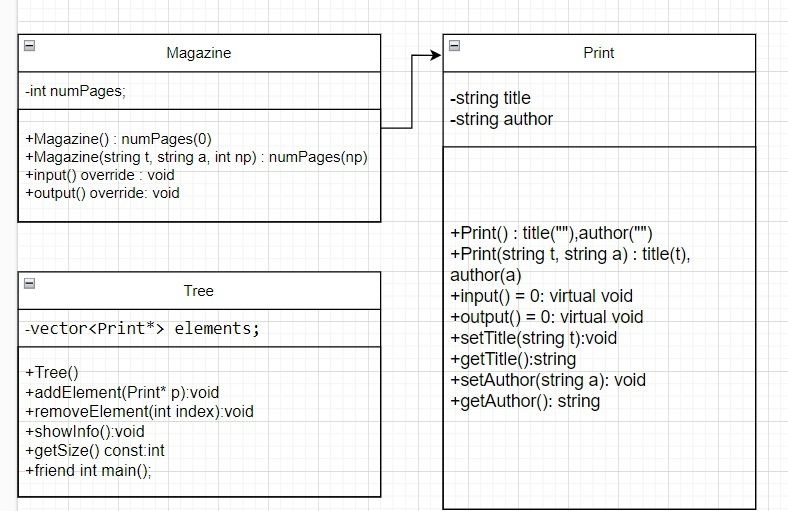
}

} while (choice != '5');

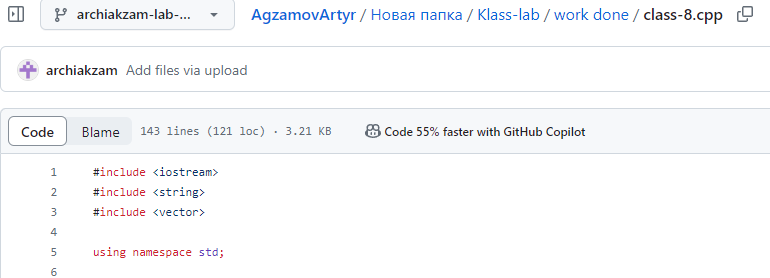
return 0;

}

**UML диаграмма:**



Скрины из git:



**Выводы:** программа сработала корректно и вывела все возможные решения.