

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Пермский национальный исследовательский  
политехнический университет»**

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»  
направление подготовки: 09.03.01 – «Информатика и вычислительная  
техника»

**Отчет по  
лабораторной работе 8 (классы)**

Выполнил студент гр. ИВТ-23-16  
Давыдов Андрей Юрьевич

Проверил:

ст. преп. каф. ИТАС

Яруллин Денис Владимирович

\_\_\_\_\_  
(оценка)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

г. Пермь, 2023

## Постановка задачи

Базовый класс:

ПЕЧАТНОЕ\_ИЗДАНИЕ(PRINT)

Название– string

Автор – string

Производный класс

ЖУРНАЛ (MAGAZIN)

Количество страниц - int

Группа – Дерево (Tree).

Команды:

- Создать группу (формат команды: m количество элементов группы).
  - Добавить элемент в группу (формат команды: +)
  - Удалить элемент из группы (формат команды -)
  - Вывести информацию об элементах группы (формат команды: s)
  - Вывести информацию о названиях всех элементов группы (формат команды : z)
- Конец работы (формат команды: q)

## Код программы:

```
Object.h Print.h Tree.h Cl_8.cpp Dialog.cpp Magazin.h Event.h Dialog.h
Cl_8 TEvent
1 #pragma once
2 using namespace std;
3
4 const int evNothing = 0;
5 const int evMessage = 100;
6 const int cmAdd = 1;
7 const int cmDel = 2;
8 const int cmGet = 3;
9 const int cmShow = 4;
10 const int cmMake = 6;
11 const int cmQuit = 101;
12
13 struct TEvent {
14     int what;
15     int command;
16 };
Object.h Print.h Tree.h Cl_8.cpp Dialog.cpp Magazin.h Event.h Dialog.h
Cl_8 Object
1 #pragma once
2 #include "Event.h"
3 class Object {
4 public:
5     Object() {}
6     virtual void Show() = 0;
7     virtual void Input() = 0;
8     virtual void HandleEvent(const TEvent&) = 0;
9 }
```

Object.hPrint.hTree.hCI\_8.cppDialog.cppMagazin.hEvent.hDialog.h

CL\_8Print

```
1  #pragma once
2  #include<iostream>
3  #include "Object.h"
4
5
6  using namespace std;
7
8  class Print :
9  {
10 public:
11     string name;
12     string author;
13 public:
14     Print();
15     Print(string, string);
16     Print(const Print&);
17     ~Print();
18     void Set_name(string);
19     void Set_author(string);
20     string Get_name();
21     string Get_author();
22     Print& operator=(const Print&);
23     void Show();
24     void Input();
25     void HandleEvent(const TEvent& e);
26 }
```

Object.hPrint.hTree.hCI\_8.cppDialog.cppMagazin.hEvent.hDialog.h

CL\_8Magazin

```
1  #pragma once
2  #include "Print.h"
3
4  class Magazin :
5  {
6 public:
7     int page_count;
8 public:
9     Magazin();
10    Magazin(int);
11    Magazin(const Magazin&);
12    ~Magazin();
13    void Set_page_count(int);
14    int Get_page_count();
15    Magazin& operator=(const Magazin&);
16    void Show();
17    void Input();
18 }
```

Object.hPrint.hTree.hCI\_8.cppDialog.cppMagazin.hEvent.hDialog.h

CL\_8TreeDel()

```
1
2  #pragma once
3  #include<iostream>
4  #include "Object.h"
5  #include "Print.h"
6  #include "Magazin.h"
7
8  class Tree {
9  {
10 protected:
11     Object** beg;
12     int size;
13     int cur;
14 public:
15     Tree();
16     Tree(int);
17     ~Tree();
18     void Add();
19     void Del();
20     void Show();
21     void Find(int);
22     int operator()();
23     void HandleEvent(const TEvent& event);
24 }
```

```
Object.h  Print.h  Tree.h  Cl_8.cpp  Dialog.cpp  Magazin.h  Event.h  Dialog.h  ↗ ✕
CL_8
1  #pragma once
2  #include "Tree.h"
3  #include "Event.h"
4
5  class Dialog : public Tree {
6  public:
7      Dialog();
8      virtual ~Dialog(void);
9      virtual void GetEvent(TEvent& Event);
10     virtual void Execute();
11     virtual void HandleEvent(TEvent& Event);
12     virtual void ClearEvent(TEvent& Event);
13     bool Valid();
14     void EndExec();
15 protected:
16     int EndState;
17 };

Object.h  Print.h  Tree.h  Cl_8.cpp  Dialog.cpp  ↗ ✕  Magazin.h  Event.h  Dialog.h
CL_8
1  #include "Dialog.h"
2  #include "Event.h"
3  #include "Tree.h"
4  #include <iostream>
5  using namespace std;
6
7  Dialog::Dialog() : Tree() {
8      EndState = 0;
9  }
10
11  Dialog::~Dialog(void) {}
12
13  void Dialog::GetEvent(TEvent& Event) {
14      string OpInt = "+m-szq";
15      string s;
16      string param;
17      char code;
18      cout << '>';
19      cin >> s; code = s[0];
20      if (OpInt.find(code) >= 0) {
21          Event.what = evMessage;
22          switch (code) {
23              case 'm': Event.command = cmMake; break;
24              case '+': Event.command = cmAdd; break;
25              case '-': Event.command = cmDel; break;
26              case 's': Event.command = cmShow; break;
27              case 'q': Event.command = cmQuit; break;
28              case 'z': Event.command = cmGet; break;
29          }
30      }
31      else Event.what = evNothing;
32  }
33
34  void Dialog::Execute() {
35      TEvent Event;
36      do {
37          EndState = 0;
38          GetEvent(Event);
39          HandleEvent(Event);
40      } while (Valid());
41  }
42
43  bool Dialog::Valid() { return (EndState == 0); }
44
45  void Dialog::ClearEvent(TEvent& Event) {
46      Event.what = evNothing;
47  }
48  void Dialog::EndExec() {
49      EndState = 1;
50  }
```

```
Object.h  Print.h  Tree.h  Cl_8.cpp  Dialog.cpp  Magazin.h  Event.h  Dialog.h
+ CI_8  Dialog
HandleEvent(TEvent & Event)
52 void Dialog::HandleEvent(TEvent& Event) {
53     if (Event.what == evMessage) {
54         switch (Event.command) {
55             case cmMake:
56                 cout << "размер: ";
57                 cin >> size;
58                 beg = new Object * [size];
59                 cur = 0;
60                 ClearEvent(Event);
61                 break;
62             case cmAdd:
63                 Add();
64                 ClearEvent(Event);
65                 break;
66             case cmDel: Del();
67                 ClearEvent(Event);
68                 break;
69             case cmShow: Show();
70                 ClearEvent(Event);
71                 break;
72             case cmQuit: EndExec();
73                 ClearEvent(Event);
74                 break;
75             default:
76                 Tree::HandleEvent(Event);
77                 break;
78         };
79     };
80 }

Object.h  Print.h  Tree.h  Cl_8.cpp  Dialog.cpp  Magazin.h  Event.h  Dialog.h  Print.cpp  Tree.cpp  Object.cpp  Event.cpp  Magazin.cpp
+ CI_8  (Глобальная область)  main()
1 #include <iostream>
2 #include <string>
3 #include "Print.h"
4 #include "Object.h"
5 #include "Magazin.h"
6 #include "Dialog.h"
7
8 using namespace std;
9
10 int main() {
11     system("chcp 1251 > NULL*");
12
13     cout << "m: Создать группу\n+; Добавить элемент\n-; Удалить элемент\nns: Информация о элементах\nlz: Информация (Названия)\nq: Выход\n";
14     Dialog D;
15     D.Execute();
16
17     return 0;
18 }

Object.h  Print.h  Tree.h  Cl_8.cpp  Dialog.cpp  Magazin.h  Print.cpp  Event.h  Dialog.h  Tree.cpp  Object.cpp  Event.cpp  Magazin.cpp
+ CI_8  Tree  HandleEvent(const TEvent & event)
1 #include "Tree.h"
2 #include <iostream>
3 #include "Dialog.h"
4
5 Tree::Tree() {
6     beg = nullptr;
7     size = 0;
8     cur = 0;
9 }
10 Tree::Tree(int n) {
11     beg = new Object * [n];
12     cur = 0;
13     size = n;
14 }
15 Tree::~Tree(void) {
16     if (beg != 0) { delete[] beg; }
17     beg = 0;
18 }
19 void Tree::Add() {
20     Object* p;
21
22     cout << "\n1.Print\n";
23     cout << "2.Magazin\n";
24     int choise;
25     cin >> choise; cout << "\n";
26
27     if (choise == 1) {
28         Print* a = new (Print);
29         a->Input();
30         p = a;
31         if (cur < size) {
32             beg[cur] = p;
33             cur++;
34         }
35     }
36     else if (choise == 2) {
37         Magazin* b = new (Magazin);
38         b->Input();
39         p = b;
40         if (cur < size) {
41             beg[cur] = p;
42             cur++;
43         }
44     }
45     else return;
46 }
```

```
Object.h  Print.h  Tree.h  Cl_8.cpp  Dialog.cpp  Magazin.h  Print.cpp  Event.h  Dialog.h  Tree.cpp  Object.cpp  Event.cpp  Magazin.cpp  ✕
Cl_8      ↓ Magazin      Input()
1  #include "Magazin.h"
2  Magazin::Magazin() {
3      page_count = 0;
4  };
5  Magazin::Magazin(int p) {
6      page_count = p;
7  }
8  Magazin::Magazin(const Magazin& m) {
9      page_count = m.page_count;
10 }
11 Magazin::~Magazin() {}
12 void Magazin::Set_page_count(int p) {
13     page_count = p;
14 }
15 int Magazin::Get_page_count() {
16     return page_count;
17 }
18 Magazin& Magazin::operator=(const Magazin& m) {
19     if (&m == this) {
20         return *this;
21     }
22     page_count = m.page_count;
23     return *this;
24 }
25 void Magazin::Show() {
26     cout << "Name: " << name;
27     cout << "Author: " << author;
28     cout << "Page_count: " << page_count << endl;
29 }
30 void Magazin::Input() {
31     cout << "Name: "; cin >> name;
32     cout << "Author: "; cin >> author;
33     cout << "Page_count: "; cin >> page_count;
34 }
```