

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №6
по ООПиП

Выполнили:
Студент 2 курса
Группы АС-53
Брашевец Д. Г.
Проверил:
Давидюк Ю.И..

Брест 2020

Вариант 4

4) деталь, механизм, изделие, узел;

Файл main.cpp

```
// Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.  
//
```

```
#include "detail.h"  
#include "knot.h"  
#include "machanism.h"  
#include "prduct.h"  
  
int main() {  
    knot a(23, 45);  
    machanism b(23, 0);  
    prduct c(23, 1, "GOOD");  
    a.add();  
    b.add();  
    c.add();  
    detail::show();  
}
```

Файл prduct.h

```
#pragma once  
#include "machanism.h"  
class prduct :  
    public machanism {  
protected:  
    string aQuality;  
public:  
    void showEl() override;  
    prduct();  
    prduct(int, bool, string);  
    prduct(prduct&);  
    ~prduct();  
};  
prduct::prduct(int size, bool performance, string quality) : aQuality(quality),  
    machanism(size, performance) {}  
prduct::prduct() : prduct(0, 0, "-") {}  
prduct::prduct(prduct &obj) : prduct(obj.aSize, obj.aPerformance, obj.aQuality) {}  
prduct::~~prduct() {}  
void prduct::showEl() {  
    string perf = aPerformance ? "+" : "-";  
    cout << "MECHANISM. SIZE: " << aSize << " PERFORMANCE: " << perf << " QUALITY: "  
    << aQuality;  
}
```

Файл mechanism.h

```
#pragma once  
#include "detail.h"  
  
class machanism :  
    public detail {  
protected:  
    bool aPerformance;
```

```

public:
    void showEl() override;
    machanism();
    machanism(int, bool);
    machanism(machanism&);
    ~machanism();
};
machanism::machanism(int size, bool performance) : aPerformance(performance),
detail(size) {}
machanism::machanism() : machanism(0, 0) {}
machanism::machanism(machanism &obj) : machanism(obj.aSize, obj.aPerformance) {}
machanism::~~machanism() {}
void machanism::showEl() {
    string perf = aPerformance ? "+" : "-";
    cout << "MECHANISM. SIZE: " << aSize << " PERFORMANCE: " << perf;
}

```

Файл knot.h

```

#pragma once
#include "detail.h"
class knot :
    public detail {
protected:
    int aPotential;
public:
    void showEl() override;
    knot();
    knot(int, int);
    knot(knot&);
    ~knot();
};
knot::knot(int size, int potential) : aPotential(potential), detail(size) {}
knot::knot() : knot(0, 0) {}
knot::knot(knot &obj) : knot(obj.aSize, obj.aPotential) {}
knot::~~knot() {}
void knot::showEl() {
    cout << "KNOT. SIZE: " << aSize << " POTENTIAL: " << aPotential;
}

```

Файл detail.h

```

#pragma once
#include <iostream>
#include <string>

using std::string;
using std::cout;
using std::endl;

class detail {
private:
    static detail *apHead, *apTail;
    detail *apNext = nullptr;
protected:
    int aSize;
public:
    void add();
    virtual void showEl() = 0;
    static void show();
    detail();
    detail(int);
    detail(detail&);
    ~detail();
};
detail* detail::apHead = nullptr;
detail* detail::apTail = nullptr;

```

```

detail::detail(int size) : aSize(size) {}
detail::detail() : detail(0) {}
detail::detail(detail &obj) : detail(obj.aSize) {}
detail::~~detail() {}
void detail::add() {
    if (!apTail) apHead = this;
    else apTail->apNext = this;
    apTail = this;
}
void detail::show() {
    detail *pCur = apHead;
    while (pCur) {
        pCur->showEl();
        cout << "\n-----\n";
        pCur = pCur->apNext;
    }
}

```

Вывод

```

KNOT. SIZE: 23 POTENTIAL: 45
-----
MECHANISM. SIZE: 23 PERFORMANCE: -
-----
MECHANISM. SIZE: 23 PERFORMANCE: + QUALITY: GOOD
-----

```