Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №3**

«Паттерны проектирования»

Выполнили:

студенты 3 курса

группы АС-53

Данилюк В. А.

Зайчук Д. Р.

Проверил:

Михняев А.Л.

Брест, 2020

**Вариант 4**

**Цель:** отработка умений и навыков описания паттернов.



**Ход работы:**

**Factory.cpp:**

#include <iostream>

#include <vector>

using namespace std;

class Armchair {

public:

virtual void info() = 0;

virtual ~Armchair() {}

};

class Sofa {

public:

virtual void info() = 0;

virtual ~Sofa() {}

};

class Cupboard {

public:

virtual void info() = 0;

virtual ~Cupboard() {}

};

class Table {

public:

virtual void info() = 0;

virtual ~Table() {}

};

class ArtDecoArmchair : public Armchair {

public:

void info() {

cout << "ArtDecoArmchair" << endl;

}

};

class ArtDecoSofa : public Sofa {

public:

void info() {

cout << "ArtDecoSofa" << endl;

}

};

class ArtDecoCupboard : public Cupboard {

public:

void info() {

cout << "ArtDecoCupboard" << endl;

}

};

class ArtDecoTable : public Table {

public:

void info() {

cout << "ArtDecoTable" << endl;

}

};

class VictorianArmchair : public Armchair {

public:

void info() {

cout << "VictorianArmchair" << endl;

}

};

class VictorianSofa : public Sofa {

public:

void info() {

cout << "VictorianSofa" << endl;

}

};

class VictorianCupboard : public Cupboard {

public:

void info() {

cout << "VictorianCupboard" << endl;

}

};

class VictorianTable : public Table {

public:

void info() {

cout << "VictorianTable" << endl;

}

};

class ModernArmchair : public Armchair {

public:

void info() {

cout << "ModernArmchair" << endl;

}

};

class ModernSofa : public Sofa {

public:

void info() {

cout << "ModernSofa" << endl;

}

};

class ModernCupboard : public Cupboard {

public:

void info() {

cout << "ModernCupboard" << endl;

}

};

class ModernTable : public Table {

public:

void info() {

cout << "ModernTable" << endl;

}

};

class PopArtArmchair : public Armchair {

public:

void info() {

cout << "PopArtArmchair" << endl;

}

};

class PopArtSofa : public Sofa {

public:

void info() {

cout << "PopArtSofa" << endl;

}

};

class PopArtCupboard : public Cupboard {

public:

void info() {

cout << "PopArtCupboard" << endl;

}

};

class PopArtTable : public Table {

public:

void info() {

cout << "PopArtTable" << endl;

}

};

class FurnitureFactory {

public:

virtual Armchair\* createArmchair() = 0;

virtual Sofa\* createSofa() = 0;

virtual Cupboard\* createCupboard() = 0;

virtual Table\* createTable() = 0;

virtual ~FurnitureFactory() {}

};

class ArtDecoFurnitureFactory : public FurnitureFactory {

public:

Armchair\* createArmchair() {

return new ArtDecoArmchair;

}

Sofa\* createSofa() {

return new ArtDecoSofa;

}

Cupboard\* createCupboard() {

return new ArtDecoCupboard;

}

Table\* createTable() {

return new ArtDecoTable;

}

};

class VictorianFurnitureFactory : public FurnitureFactory {

public:

Armchair\* createArmchair() {

return new VictorianArmchair;

}

Sofa\* createSofa() {

return new VictorianSofa;

}

Cupboard\* createCupboard() {

return new VictorianCupboard;

}

Table\* createTable() {

return new VictorianTable;

}

};

class ModernFurnitureFactory : public FurnitureFactory {

public:

Armchair\* createArmchair() {

return new ModernArmchair;

}

Sofa\* createSofa() {

return new ModernSofa;

}

Cupboard\* createCupboard() {

return new ModernCupboard;

}

Table\* createTable() {

return new ModernTable;

}

};

class PopArtFurnitureFactory : public FurnitureFactory {

public:

Armchair\* createArmchair() {

return new PopArtArmchair;

}

Sofa\* createSofa() {

return new PopArtSofa;

}

Cupboard\* createCupboard() {

return new PopArtCupboard;

}

Table\* createTable() {

return new PopArtTable;

}

};

class Furniture {

public:

~Furniture() {

int i;

for (i = 0; i < vecArmchair.size(); ++i) delete vecArmchair[i];

for (i = 0; i < vecSofa.size(); ++i) delete vecSofa[i];

for (i = 0; i < vecCupboard.size(); ++i) delete vecCupboard[i];

for (i = 0; i < vecTable.size(); ++i) delete vecTable[i];

}

void info() {

int i;

for (i = 0; i < vecArmchair.size(); ++i) vecArmchair[i]->info();

for (i = 0; i < vecSofa.size(); ++i) vecSofa[i]->info();

for (i = 0; i < vecCupboard.size(); ++i) vecCupboard[i]->info();

for (i = 0; i < vecTable.size(); ++i) vecTable[i]->info();

}

vector<Armchair\*> vecArmchair;

vector<Sofa\*> vecSofa;

vector<Cupboard\*> vecCupboard;

vector<Table\*> vecTable;

};

class Shop {

public:

Furniture\* createFurniture(FurnitureFactory& factory) {

Furniture\* f = new Furniture;

f->vecArmchair.push\_back(factory.createArmchair());

f->vecSofa.push\_back(factory.createSofa());

f->vecCupboard.push\_back(factory.createCupboard());

f->vecTable.push\_back(factory.createTable());

return f;

}

};

int main() {

int menuNum;

FurnitureFactory\* ff;

bool skip = false;

Shop shop;

PopArtFurnitureFactory pa\_factory;

VictorianFurnitureFactory v\_factory;

ModernFurnitureFactory m\_factory;

ArtDecoFurnitureFactory ad\_factory;

while (!skip) {

cout << "Furniture configurator 1.0.0." << endl << "1. PopArt" << endl << "2. Victorian" << endl << "3. Modern" << endl << "4. ArtDeco" << endl << "0. Exit" << endl;

cin >> menuNum;

switch (menuNum) {

case 1:

ff = new PopArtFurnitureFactory;

break;

case 2:

ff = new VictorianFurnitureFactory;

break;

case 3:

ff = new ModernFurnitureFactory;

break;

case 4:

ff = new ArtDecoFurnitureFactory;

break;

default:

ff = new PopArtFurnitureFactory;

skip = true;

std::cout << "Exited";

break;

}

if (!skip) {

cout << "Furniture configurator 1.0.0." << endl << "1. Armchair" << endl << "2. Sofa" << endl << "3. Cupboard" << endl << "4. Table" << endl;

cin >> menuNum;

switch (menuNum) {

case 1:

ff->createArmchair()->info();

break;

case 2:

ff->createSofa()->info();

break;

case 3:

ff->createCupboard()->info();

break;

case 4:

ff->createTable()->info();

break;

default:

break;

}

}

}

}

Изображение выглядит как текст

Автоматически созданное описание

Вывод: в ходе лабораторной работы мы отработали умение и навык паттернов.