

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій

Звіт
про виконання лабораторної роботи №7
«Шаблони проєктування GoF»

Виконав:
студент групи ФЕП-11с
Качмар Д. Б

Викладач:
Кушнір Олексій
Олександрович

Мета роботи: Засвоїти принципи використання шаблонів проєктування та особливості їх реалізації мовою C++.

Виконання роботи

Посилання на GitHub: <https://github.com/D-Kachm/OOP/tree/main/Lab7>

```
GameFactory.cpp > Elf
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  // Базовий клас персонажа
6  class Character {
7  protected:
8      string name;
9  public:
10     Character(string n) : name(n) {}
11     virtual void display() = 0;
12 };
13
14 // Конкретні персонажі
15 class Elf : public Character {
16 public:
17     Elf() : Character("Elf") {}
18     void display() override {
19         cout << "Created character: " << name << " " << endl;
20     }
21 };
22
23 class Orc : public Character {
24 public:
25     Orc() : Character("Orc") {}
26     void display() override {
27         cout << "Created character: " << name << " " << endl;
28     }
29 };
30
31 // Базовий клас зброї
32 class Weapon {
33 protected:
34     string type;
35 public:
36     Weapon(string t) : type(t) {}
37     virtual void show() = 0;
38 };
39
40 class Bow : public Weapon {
41 public:
42     Bow() : Weapon("Bow") {}
43     void show() override {
44         cout << "Equipped weapon: " << type << " " << endl;
45     }
46 };
47
48 class Axe : public Weapon {
49 public:
50     Axe() : Weapon("Axe") {}
51     void show() override {
52         cout << "Equipped weapon: " << type << " " << endl;
53     }
54 };
55
```

```

57  class GameFactory {
58  public:
59      virtual Character* createCharacter() = 0;
60      virtual Weapon* createWeapon() = 0;
61  };
62
63  // Конкретні фабрики
64  class ElfFactory : public GameFactory {
65  public:
66      Character* createCharacter() override {
67          return new Elf();
68      }
69      Weapon* createWeapon() override {
70          return new Bow();
71      }
72  };
73
74  class OrcFactory : public GameFactory {
75  public:
76      Character* createCharacter() override {
77          return new Orc();
78      }
79      Weapon* createWeapon() override {
80          return new Axe();
81      }
82  };
83
84  int main() {
85      GameFactory* factory;
86
87      // Вибір раси
88      cout << "Choose race (1 - Elf, 2 - Orc): ";
89      int choice;
90      cin >> choice;
91
92      if (choice == 1)
93          factory = new ElfFactory();
94      else
95          factory = new OrcFactory();
96
97      Character* player = factory->createCharacter();
98      Weapon* weapon = factory->createWeapon();
99
100     player->display();
101     weapon->show();
102
103     delete player;
104     delete weapon;
105     delete factory;
106     return 0;
107 }

```

```

GameFacade.cpp > ...
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  // Підсистема 1 – Графіка
7  class GraphicsSystem {
8  public:
9      void loadTextures() {
10         cout << "[Graphics] Textures loaded.\n";
11     }
12     void initializeRenderer() {
13         cout << "[Graphics] Renderer initialized.\n";
14     }
15 };
16
17 // Підсистема 2 – Звук
18 class SoundSystem {
19 public:
20     void loadSounds() {
21         cout << "[Sound] Sound assets loaded.\n";
22     }
23     void initAudio() {
24         cout << "[Sound] Audio system started.\n";
25     }
26 };
27
28 // Підсистема 3 – Мережа
29 class NetworkSystem {
30 public:
31     void connectToServer() {
32         cout << "[Network] Connected to game server.\n";
33     }
34 };
35
36 // Підсистема 4 – Авторизація
37 class LoginSystem {
38 public:
39     void loginPlayer(string username) {
40         cout << "[Login] Player " << username << " logged in.\n";
41     }
42 };
43
44 // Підсистема 5 – Геймплей
45 class GameEngine {
46 public:
47     void startGame() {
48         cout << "[Game] Game has started!\n";
49     }
50 };
51
52 // === ФАСАД ===
53 class GameFacade {
54 private:
55     GraphicsSystem* graphics;
56     SoundSystem* sound;
57     NetworkSystem* network;

```

```

53 class GameFacade {
54     LoginSystem* login;
55     GameEngine* engine;
56
57 public:
58     GameFacade() {
59         graphics = new GraphicsSystem();
60         sound = new SoundSystem();
61         network = new NetworkSystem();
62         login = new LoginSystem();
63         engine = new GameEngine();
64     }
65
66     ~GameFacade() {
67         delete graphics;
68         delete sound;
69         delete network;
70         delete login;
71         delete engine;
72     }
73
74     void launchGame(string username) {
75         graphics->loadTextures();
76         graphics->initializeRenderer();
77         sound->loadSounds();
78         sound->initAudio();
79         network->connectToServer();
80         login->loginPlayer(username);
81         engine->startGame();
82     }
83
84     void playOffline(string username) {
85         graphics->loadTextures();
86         graphics->initializeRenderer();
87         sound->loadSounds();
88         sound->initAudio();
89         login->loginPlayer(username);
90         cout << "[Game] Offline mode.\n";
91         engine->startGame();
92     }
93 };
94
95 // === CLIENT CODE ===
96
97 int main() {
98     GameFacade* game = new GameFacade();
99
100     cout << "=== Starting Online Game ===\n";
101     game->launchGame("Gamer007");
102
103     cout << "\n=== Starting Offline Game ===\n";
104     game->playOffline("SoloPlayer");
105
106     delete game;
107     return 0;
108 }
109
110
111
112

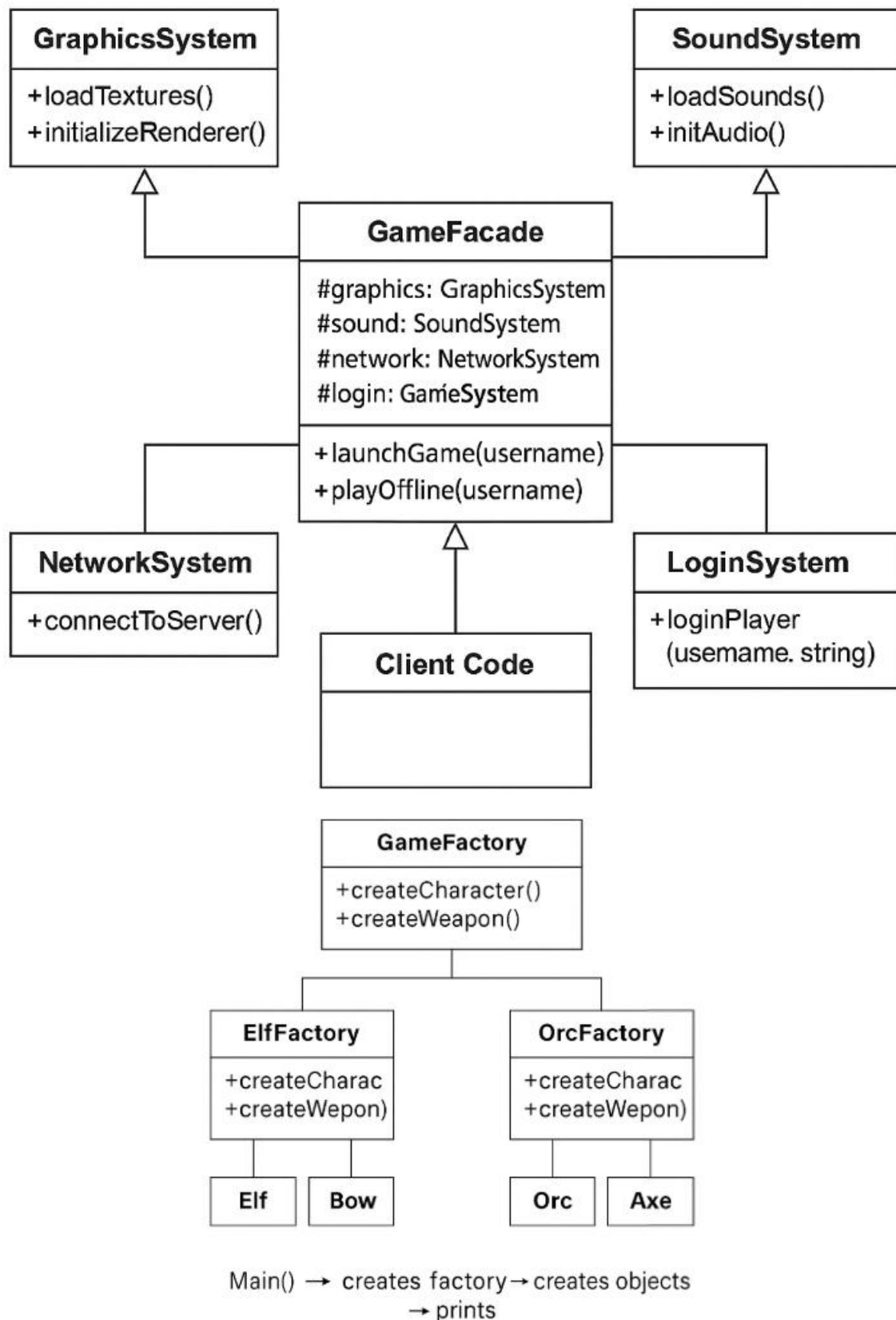
```

Результати

```
PS D:\Github\OOP\Lab7> g++ GameFactory.cpp -o game_factory
PS D:\Github\OOP\Lab7> .\game_factory.exe
Choose race (1 - Elf, 2 - Orc): 1
Created character: Elf
Equipped weapon: Bow
PS D:\Github\OOP\Lab7> 
```

```
PS D:\Github\OOP\Lab7> .\game_facade.exe
=== Starting Online Game ===
[Graphics] Textures loaded.
[Graphics] Renderer initialized.
[Sound] Sound assets loaded.
[Sound] Audio system started.
[Network] Connected to game server.
[Login] Player Gamer007 logged in.
[Game] Game has started!

=== Starting Offline Game ===
[Graphics] Textures loaded.
[Graphics] Renderer initialized.
[Sound] Sound assets loaded.
[Sound] Audio system started.
[Login] Player SoloPlayer logged in.
[Game] Offline mode.
[Game] Game has started!
PS D:\Github\OOP\Lab7> 
```



Висновок: У цій лабораторній роботі я реалізував два шаблони проєктування — Facade та Abstract Factory — на тему комп'ютерної гри.

За допомогою шаблону Facade я створив простий інтерфейс для запуску гри. Він приховує складну логіку таких підсистем, як графіка, звук, мережа, ігровий

рушій і авторизація. Це дозволяє запускати гру одним викликом, не думаючи про внутрішні механізми.

Шаблон Abstract Factory я використав для створення різних типів ігрових персонажів. Я реалізував дві фабрики: фабрику ельфів та фабрику огрів, кожна з яких створює власних героїв і зброю. Завдяки цьому гру можна легко змінювати під різні раси або жанри без переписування основної логіки.

Ця робота допомогла мені краще зрозуміти практичне застосування шаблонів проєктування. Я побачив, як вони допомагають зробити код більш гнучким, зручним для масштабування та підтримки.