

Лабораторна робота №27

Тема: ООП. Поліморфізм

Мета: Навчитися використовувати ООП. Поліморфізм.

Індивідуальне завдання

Загальне завдання

Модернізувати попередню лабораторну роботу шляхом:

- базовий клас зробити абстрактним. Додати абстрактні методи;
- розроблені класи-списки поєднуються до одного класу таким чином, щоб він міг працювати як з базовим класом, так і з його спадкоємцями. При цьому серед полів класу-списку повинен бути лише один масив, що містить усі типи класів ієрархії. Оновити методи, що працюють з цим масивом.
- у функціях базового класу та класів-спадкоємців обов'язкове використання ключових слів `final` та `override`.

Хід роботи

```
class Pupils
{
public:
    int numberSchool;
    string fullname;
    int age;
    string characteristic;
    int grade;
    bool insurance;

    Pupils();
    Pupils(const int numberSchool, const string fullname, const int age, const string characteristic, const int grade, const bool insurance);
    Pupils(const Pupils& obj);
    virtual ~Pupils() = 0 {};

    virtual string print() = 0;
    virtual int whoIAM() = 0;
```

Рисунок 1.1 - Створення абстрактного класу Учень
з абстрактними методами.

```

class JuniorStudent : public Pupils {
private:
    int markExam;
    bool changeSchool;
public:
    JuniorStudent() : markExam(0), changeSchool(false) {};
    JuniorStudent(int numberSchool, string fullname, int age, string characteristic, int grade, bool insurance, int markExam, bool changeSchool);
    ~JuniorStudent() {};

    virtual string print() override final;
    virtual int whoIAM() override final;

    int getMarkExam() { return markExam; }
    int getChangeSchool() { return changeSchool; }

    void setMarkExam(int l) { markExam = l; }
    void setChangeSchool(int s) { changeSchool = s; }
};

class HighSchoolStudent : public Pupils {
private:
    int sportsActivities;
    bool additionalPoints;
public:
    HighSchoolStudent() : sportsActivities(0), additionalPoints(false) {};
    HighSchoolStudent(int numberSchool, string fullname, int age, string characteristic, int grade, bool insurance, int sportsActivities, bool additionalPoints);
    ~HighSchoolStudent() {};

    virtual string print() override final;
    virtual int whoIAM() override final;

    int getSportsActivities() { return sportsActivities; }
    bool getAdditionalPoints() { return additionalPoints; }

    void setSportsActivities(int l) { sportsActivities = l; }
    void setAdditionalPoints(bool s) { additionalPoints = s; }
};

```

Рисунок 1.2 - Приклад використання override та final для методів спадкоємця

```

string HighSchoolStudent::print() {
    std::stringstream ss;
    ss << "\nNumberSchool:" << numberSchool;
    ss << "\nFullname:" << fullname;
    ss << "\nAge:" << age;
    ss << "\nCharacteristic:" << characteristic;
    ss << "\nGrade:" << grade;
    ss << "\nInsurance:" << insurance ? "Yes" : "No";
    ss << "\nSports activities : ";
    switch (sportsActivities)
    {
        case 1:
            ss << "Boxing";
            break;
        case 2:
            ss << "Karate";
            break;
        case 3:
            ss << "Swimming";
            break;
    }
    ss << "\nAdditional points:" << additionalPoints ? "Yes" : "No";

    return ss.str();
}

```

Рисунок 1.3 - - Приклад для реалізації методів виводу

(об'єкт Учень старших класів).

```
public:
    void addPupils();
    void removePupils(const int index);
    void showAll();

    int getSize();
    Pupils& getPupils(const int index);
    Pupils& getPupilsGrade(const int grade);

    string check(regex reg);
    void get6thGradeStudent();
    int chooseInt(int start, int end);

    Pupils& operator[](const int index);
    friend std::ostream& operator << (std::ostream& out, const List& obj) {
        for (int i = 0; i < obj.list.size(); i++)
            out << obj.list[i];

        return out;
    }

    void readFromFile();
    void writeToFile();
};
```

Рисунок 1.4 - Список, який складається з покажчиків на елемент базового класу, який визначає яким спадкоємцем буде елемент. Сам список складається лише з елементів спадкоємців, бо базовий клас є абстрактним. Для його обробки були змінені методи.

```

void List::addPupils() {
    cout << "Do you want to create a Junior Student or High School Student? \n1.Junior Student\n2.High School Student\nYour choose:";
    int choose = chooseInt(1, 2);
    Pupils* p = NULL;
    int digit;
    getchar;

    if (choose == 1) {
        JuniorStudent tmp;
        cout << "\n\nCreate a new pupils now.\n Enter full name : ";

        tmp.setFullname(check(regName));
        cout << "   Enter age : ";
        tmp.setAge(chooseInt(5, 20));
        cout << "   Enter number school : ";
        tmp.setNumberSchool(chooseInt(1, 999));
        cout << "   Describe the positive qualities in 3 words : ";
        tmp.setCharac(check(regName));
        cout << "   Enter grade : ";
        tmp.setGrade(chooseInt(1, 12));
        cout << "   Availability of insurance \n1.Yes\n0.No\n\nYour choose : ";
        tmp.setInsurance(chooseInt(0, 1));
        cout << "   Enter exam score : ";
        tmp.setMarkExam(chooseInt(1, 200));
        cout << "   Have you changed school? \n1.Yes\n2.No\n\nYour choose : ";
        tmp.setChangeSchool(chooseInt(0, 1));
        p = &tmp;
    }
    else if (choose == 2) { ... }
    list.push_back (p);
}

```

Рисунок 1.5 - Приклад методу створення об'єкта класу списку з визначенням яким спадкоємцем буде об'єкт.

```
Your choice: 2

Do you want to create a Junior Student or High School Student?
1.Junior Student
2.High School Student
Your choose:1

Create a new pupils now.
    Enter full name : ALexa
Try again :Alexa
    Enter age : 15
    Enter number school : 485
    Describe the positive qualities in 3 words : energy
Try again :Energy
    Enter grade : 11

Try again : 9
    Availability of insurance
1.Yes
2.No

Your choose : 1
    Enter exam score : 189
    Have you changed school?
1.Yes
2.No

Your choose :0

Try again : 2

Successful.
```

Рисунок 1.6 - Приклад створення елементу

(при створенні використовуються регулярні вирази, для перевірки полей, та вибір за допомогою споміжних методів whoIAm & chooseInt).

```

Your choice: 1

NumberSchool:458
Fullname:Alexa
Age:15
Characteristic:Energy
Grade:9
MarkExam:189
Changed school:0

NumberSchool:142
Fullname:Max
Age:13
Characteristic:Grate
Grade:10
Insurance:1
Sports activities : Boxing
Additional points:1

Successful.

```

Рисунок 1.7 - Приклад виводу на екран списку з різними об'єктами спадкоємців.

```

A list of pupils.Select an action from
1.Write list to screen.
2.Generate and add pupils.
3.Delete pupils by index.
4.Search for pupils with insurance.
5.Reading pupils from a file.
6.Writing pupils in file.
0.Exit.

Your choice: 6

Successful.

```

*text.txt – Блокнот

Файл Правка Формат Вид Справка

```

2
1 458 Alexa 15 Energy 9 189 0
2 142 Max 13 Grate 10 1 1 1

```

Рисунок 1.8 - Приклад запису у файл списку з різними об'єктами спадкоємців.

Your choice: 5

Successful.

A list of pupils.Select an action from the following.

- 1.Write list to screen.
- 2.Generate and add pupils.
- 3.Delete pupils by index.
- 4.Search for pupils with insurance.
- 5.Reading pupils from a file.
- 6.Writing pupils in file.
- 0.Exit.

Your choice: 1

NumberSchool:458
Fullname:Alexa
Age:15
Characteristic:Energy
Grade:9
MarkExam:189
Changed school:0

NumberSchool:142
Fullname:Max
Age:13
Characteristic:Grate
Grade:10
Insurance:1
Sports activities : Boxing
Additional points:1

Successful.

Рисунок 1.9 - Приклад зчитування з файлу.

```
Your choice: 3

Choose index(1-2):2

Successful.

A list of pupils.Select an action from the following.

1.Write list to screen.
2.Generate and add pupils.
3.Delete pupils by index.
4.Search for pupils with insurance.
5.Reading pupils from a file.
6.Writing pupils in file.
0.Exit.

Your choice: 1

NumberSchool:458
Fullname:Alexa
Age:15
Characteristic:Energy
Grade:9
MarkExam:189
Changed school:0

Successful.
```

Рисунок 1.10 - Приклад видалення з файлу за індексом(при цьому видаляється як вміст об'єкту, так і сам об'єкт зі списку)

Висновок: Навчився використовувати ООП. Поліморфізм.