

## **Лабораторна робота №26**

**Тема: ООП. Спадкування**

**Мета: Навчитися використовувати ООП. Спадкування.**

### **Індивідуальне завдання**

#### **Загальне завдання**

Модернізувати попередню лабораторну роботу шляхом:

- додавання класів-спадкоємців з розділу “Розрахункове завдання / Індивідуальні завдання”, котрі будуть поширювати функціонал “базового класу” відповідно до індивідуального завдання;
- додавання ще класу-списку для кожного класу-спадкоємцю, що буде керувати лише елементами стосовного класу-спадкоємця;

**Спадкоємець 1 - Учень початкових-середніх класів. Додаткові умови.**

- Чи змінював ЗОЗ (наприклад: Так, ні)
- Бали ДПА (наприклад: 200, 180, 156)

**Спадкоємець 2 - Учень старших класів. Додаткові умови.**

- Наявність додаткових балів за міжнародні конкурси (наприклад: Так, ні)
- Учні, які займалися видами спорту (один з переліку: бокс, карате, плавання)

**Методи для роботи з колекцією:**

1. Знайти учня, який навчається у 6 класі.
2. Знайти учня, який змінював ЗОЗ.
3. Знайти учня, який займався боксом.

**Хід роботи**

```

class JuniorStudent : public Pupils {
private:
    int markExam;
    bool changeSchool;
public:
    JuniorStudent() :markExam(0), changeSchool(false) {};
    ~JuniorStudent() {};

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

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

```

Рисунок 1.1 - Клас спадкоємець Учень початкових-середніх класів.

```

class HighSchoolStudent : public Pupils {
private:
    int sportsActivities;
    bool additionalPoints;
public:
    HighSchoolStudent() :sportsActivities(0), additionalPoints(false) {};
    ~HighSchoolStudent() {};

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

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

```

Рисунок 1.2 - Клас спадкоємець Учень старших класів.

```

class ListOfJuniorStudent {
private:
    std::vector<JuniorStudent> list;

public:
    void addPupils(JuniorStudent& pupils);
    void addObject();
    void removePupils(const int index);
    void showAll();

    int getSize();
    JuniorStudent& getPupils(const int index);
    JuniorStudent& getPupilsGrade(const int grade);
    string check(regex reg);
    int chooseInt(int start, int end);
    void getWasChangeSchool();

};

```

Рисунок 1.3 - Приклад списку для об'єктів класу Учень початкових-середніх класів.

```

class ListOfHighSchoolStudent {
private:
    std::vector<HighSchoolStudent> list;

public:
    void addPupils(HighSchoolStudent& pupils);
    void addObject();
    void removePupils(const int index);
    void showAll();

    int getSize();
    HighSchoolStudent& getPupils(const int index);
    HighSchoolStudent& getPupilsGrade(const int Grade);
    string check(regex reg);
    int chooseInt(int start, int end);
    void getWasBoxing();

};

```

Рисунок 1.4 - Приклад списку для об'єктів класу Учень старших класів.

```

void List::get6thGradeStudent() {
    for (Pupils& emp : list) {
        if (emp.getGrade() == 6)
            cout << emp.printPupils() << endl;
    }
}

```

Рисунок 1.5 - Приклад пошуку учнів, які навчаються у 6 класі.

```
void ListOfJuniorStudent::getWasChangeSchool() {  
    for (JuniorStudent& emp : list) {  
        if (emp.getChangeSchool() == 0)  
            cout << emp.printPupils() << endl;  
    }  
}
```

Рисунок 1.6 - Приклад пошуку учнів, які змінювали ЗОЗ.

```
void ListOfHighSchoolStudent::getWasBoxing() {  
    for (HighSchoolStudent& emp : list) {  
        if (emp.getSportsActivities() == 1)  
            cout << emp.printPupils() << endl;  
    }  
}
```

Рисунок 1.7 - Приклад пошуку учнів, які займалися боксом.

**Висновок:** Навчився використовувати ООП. Спадкування.