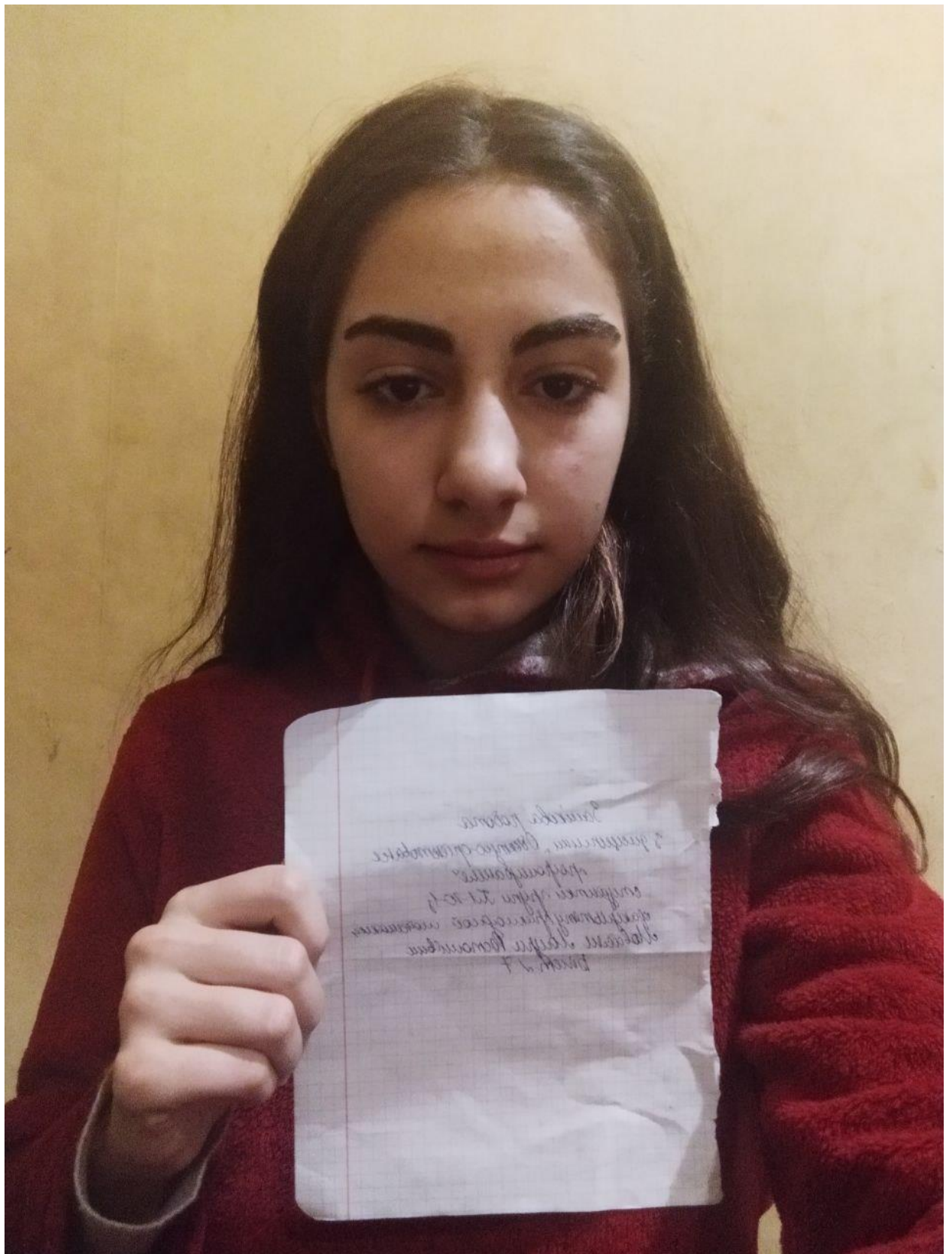


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
УКРАИНЫ
ДНЕПРОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
ИМЕНИ ОЛЕСЯ ГОНЧАРА

Факультет прикладной математики
Зачетная работа
ПО ДИСЦИПЛИНЕ
«Объектно-ориентированное программирование»
студентки группы ПА-20-1з
факультета прикладной математики
Мовсисян Лауры Ростомовны

г. Днепр – 2021 год



Handwritten text on the paper (read from bottom to top):
F. L. M. 1917
The following is a list of the
names of the persons who
were present at the
meeting of the
Committee on
the 1st of
January, 1917.

Залікове завдання № 1

1. Поняття про класи в C++. Класи та принцип ООП. Обмеження доступу до членів класу.
2. Перевантажити операцію «--».
3. Написати програму, яка б містила клас з полями: Номер рейса, прізвище і ініціали пасажера, дата вильоту. Дозволяла виконати наступні дії: Додавання запису, Видалення запису, Редагування запису, Пошук запису по заданому критерію, Виведення списку на екран.

Затверджено на засіданні кафедри комп'ютерних технологій, протокол № 5 від 12.05.2021.

① Класс - это множество объектов, обладающих
одной структурой, поведением и семантикой.
Одним объектом - это экземпляр класса.
Класс представляет собой абстрацию, выража-
ющую свойства объекта.

ООП строится на 3 основных принципах
1) Инкапсуляция - процесс отвлечения друг
от друга элементов объекта, обеспечиваю-
щий безопасность и поведение; инкапсуляция состоит
в том, чтобы скрыть конкретные детали
реализации абстракции и реализовать
ограничение множества функций, которыми
или классом не будет доступен.

Доктор имеет все операции, которые
определены для объекта.

```
class Window {, class dumb-terminal; public Window  
protected: Rectangle inside; public: void prompt;  
get & b
```

Класс Window не наследует от Rectangle
он не является его (protected), но
функции - члены производных классов,
dumb-terminal::prompt(), могут обратиться
к нему и узнать (какого типа объект он
работает). Для других функций или недоступно

Классе составлен на основе списка всех функций,
с группировкой по типу. Для метода обработки
данных метода порождения данных мало
применимы. Решение - удерживать в классе
список списка всех функций, которые
нужны для к. г. В C++ для этого
берут списки методов (private) методов.
Суть языка C++, т.е. расширение методов в
защиту одомов, резко возрастает с ростом
размеров программы.

2- принцип ООП, наследование
- это способность производить новые классы из
старого класса. Производные классы имеют
свойства - наследуются.

3- принцип ООП - полиморфизм
полиморфизм означает представление одних
классов одними, которые имеют форму
во время выполнения программы. Различия
в полиморфизме происходят из-за
наследования классов и виртуальных
функций. Классы имеют атрибуты методов
массовых, а реализация, эти методы
другие, соответствующие специфике класса.
потомки

```

② { class Integer
    { private; int number;
    public: Integer() { number = 0; }
        Integer(int number) { number = number; }
    void Set(int number) { number = number; }
    int Get(void) { return number; }
    Integer operator ++(void)
        { number++;
          return this; }
    Integer operator ++(int d) { number++; return this; }
    Integer operator --(void) { number--; return this; }
    Integer operator --(int d) { number--; return
        this; }
    }

```



```

3) #include <string.h>
#include <stdio.h>
#include <locale.h>
#include <conio.h>
#include <ostream>
using namespace std;
struct AEROTLOT
{
    char* name;
    int n;
    char* tipsomoleto;
};
ostream& operator<< (ostream& out, AEROTLOT& x)
{
    out<<"\n nyukm jukmoleto: " << x.name;
    out<<"\n komep peica: " << x.n;
    return out;
}
const int Amax=255;
int main()
{
    setlocale(LC_ALL, "Russian");
    AEROTLOT k[7], n, t;
    char buf[Amax], buf1[Amax];
    int i;
    for (i=0; i<7; i++)
    {
        printf("blegite nyukm jukmoleto\n");
        k[i].name = new char[Amax];
        scanf("%s", k[i].name, Amax);
        printf("blegite mur camoleto\n");
        k[i].tipsomoleto = new char[Amax];
        scanf("%s", k[i].tipsomoleto, Amax);
        fflush();
        printf("blegite komep camoleto\n");
        scanf("%i", &k[i].n);
        printf("\n");
    }
    bool changes = true;
    while (changes)
    {
        changes = false;
        for (int i=0; i<6; i++)
        {
            if (strcmp(k[i].name, k[i+1].name) > 0)
            {
                n = k[i];
                k[i] = k[i+1];
                k[i+1] = n;
            }
            printf("%s", "cruce nyukm jukmoleto, peica noapobny");
            printf("\n");
            for (i=0; i<7; i++)
                puts(k[i].name);
            t.tipsomoleto = new char[Amax];
            scanf("%s", t.tipsomoleto, Amax);
        }
    }
}

```