Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**Отчет по лабораторной работе “Классы” №1**

**по дисциплине**

**«Теория алгоритмов и структуры данных»**

Выполнил студент гр. ИВТ-21-1б

Ахунов Руслан Булатович

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Денис Владимирович

(оценка) (подпись)

г. Пермь - 2022

**Постановка задачи:**

1.  Реализовать определение нового класса. Для демонстрации работы с объектами написать главную функцию. Продемонстрировать разные способы создания объектов и массивов объектов.

2.  Структура-пара – структура с двумя полями, которые обычно имеют имена first и second. Требуется реализовать тип данных с помощью такой структуры. Во всех заданиях должны присутствовать :

a.  метод инициализации Init (метод должен контролировать значения аргументов на корректность);

b.  ввод с клавиатуры Read;

c.  вывод на экран Show.

3.  Реализовать внешнюю функцию make\_тип(), где тип – тип реализуемой структуры. Функция должна получать значения для полей структуры как параметры функции и возвращать структуру как результат. При передаче ошибочных параметров следует выводить сообщение и заканчивать работу.

Вариант 2:

Поле first – положительное целое число, номинал купюры; номинал может принимать значения 1, 2, 5. 10, 50, 100, 500, 1000, 5000, поле second – положительное целое число, количество купюр данного достоинства. Реализовать метод summa() – вычисление денежной суммы.

**Анализ задачи:**

class ruble

{

private:

    int nominal;

    int count;

public:

    void set(int nominal, int count)

    {

        this->nominal = nominal;

        this->count = count;

    }

    int summa()

    {

        int sum = 0;

            sum = nominal \* count;

            return sum;

    }

};

Класс ruble имеем 2 int поля, а именно nominal – отвечает за номинал купюры, count – отвечает за количество этих купюр. nominal и count находятся private.

Перейдем к модификатору доступа public, в нем имеются 2 метода: set – записывает значения в соответствующие поля, так как переменные имеют одинаковые называния используется this. Метод summa – Умножает номинал купюр на их количество и возвращает результат как переменная типа int.

int main()

{

    setlocale(LC\_ALL, "Russian");

    ruble a;

    int nom, cou;

    cout << "Введите номинал и кол-во купюр этого достоинства:" << endl;

    cin >> nom;

    while (nom != 1 && nom != 2 && nom != 5 && nom != 10 && nom != 50 && nom != 100 && nom != 500 && nom != 1000 && nom != 2000 && nom != 5000)

    {

        cout << "Вы ввели несуществующий номинал купюры." << endl;

        cin >> nom;

    }

    cin >> cou;

    a.set(nom, cou);

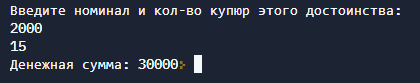
    cout << "Денежная сумма: " << a.summa();

    return 0;

}

В функции main, мы объявляем переменную a, тип данных ruble. Затем объявляем 2 вспомогательные переменные, которые потом перейдут в метод set. Есть проверка на корректность номинала купюр, если он не подходит, то выходит соответствующие ошибка. Если данные верны, то затем проситься указать количество купюр, потом все эти данные уходят в метод summa.

**Результат работы программы**

****

Сперва требуется ввести номинал купюра, а уже потом её количество. Если все правильно, выводится денежная сумма.

**Код программы**

#include <iostream>

using namespace std;

class ruble

{

private:

    int nominal;

    int count;

public:

    void set(int nominal, int count)

    {

        this->nominal = nominal;

        this->count = count;

    }

    int summa()

    {

        int sum = 0;

            sum = nominal \* count;

            return sum;

    }

};

int main()

{

    setlocale(LC\_ALL, "Russian");

    ruble a;

    int nom, cou;

    cout << "Введите номинал и кол-во купюр этого достоинства:" << endl;

    cin >> nom;

    while (nom != 1 && nom != 2 && nom != 5 && nom != 10 && nom != 50 && nom != 100 && nom != 500 && nom != 1000 && nom != 2000 && nom != 5000)

    {

        cout << "Вы ввели несуществующий номинал купюры." << endl;

        cin >> nom;

    }

    cin >> cou;

    a.set(nom, cou);

    cout << "Денежная сумма: " << a.summa();

    return 0;

}

**Вопросы**

1. Что такое класс?  
Классявляется абстрактным типом данных, определяемым пользователем, и представляет собой модель реального объекта в виде данных и функций для работы с ними

2. Что такое объект (экземпляр) класса?  
Объект – переменная класса.

3. Как называются поля класса?  
Поля класса - Данные класса

4. Как называются функции класса?  
Функциями класса - методы.

5. Для чего используются спецификаторы доступа?  
Они управляют видимостью элементов класса.

6. Для чего используется спецификатор public?  
Все поля и методы будут общедоступными.

7. Для чего используется спецификатор private?  
Все методы и поля класса будут скрытыми.

8. Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использоваться по умолчанию?  
Private.

9. Если описание класса начинается со спецификатора struct, то какой спецификатор доступа будет использоваться по умолчанию?  
Public.

10. Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?  
Public, так как без него по умолчанию будут использоваться спецификатор Private, Public нужен для того, чтобы получать доступ к этому элементу из вне.

11. Каким образом можно изменить значения атрибутов экземпляра класса?  
Через методы, которые описаны в Public. Или если поля тоже описаны в Public, к ним напрямую.

12. Каким образом можно получить значения атрибутов экземпляра класса?  
Через методы, которые описаны в Public. Или если поля тоже описаны в Public, к ним напрямую.

13. Класс описан следующим образом:  
Struct Student  
{  
String name;  
Int group;  
…  
};  
Объект класса определен следующим образом  
Student \*s = new Student;  
Как можно обратиться к полю name объекта s?  
  
s[…].name  
  
14.  
Struct Student  
{  
String name;  
Int group;  
…  
};  
Student s;  
Как можно обратиться к полю name объекта s?

s.name  
  
15. class Student  
{  
String name;  
Int group;  
…  
};  
Student \*s = new Student;

Можно обратиться только через методы класса.  
  
16.  
class Student  
{  
String name;  
Int group;  
…  
};  
Объект класса определен следующим образом  
Student s;  
Как можно обратиться к полю name объекта s?  
  
Можно обратиться только через методы класса.

17. Класс описан следующим образом:  
class Student  
{  
Public:  
String name;  
Int group;  
…  
};  
Объект класса определен следующим образом:  
Student\* s = new Student;

Как можно обратиться к полю name объекта s?  
  
s[…].name