Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (ПНИПУ)

Электротехнический факультет

Дисциплина: основы алгоритмизации и программирования, 2 семестр

ОТЧЁТ

"Лабораторная работа ООП № 1. Классы и объекты.

Инкапсуляция"

Выполнил

Студент группы РИС-22-2Б

Хохряков А.С

Проверил

Доцент кафедры ИТАС

Полякова О.А

Пермь 2023

# Содержание отчета

1. Постановка задачи.
2. Анализ задачи.
3. UML-Диаграмма
4. Ответы на вопросы
5. Текст программы

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 3:

Поле first - положительное целое число, цена товара, поле second - положительное целое число, количество единиц данного товара. Реализовать метод cost() -вычисление стоимости данного товара.

Анализ задачи

1. Объявление структуры "Product" с двумя полями типа int - "first" и "second".

2. Метод "Init", который инициализирует поля структуры значениями, переданными в качестве аргументов. Если переданные значения отрицательны, метод выводит сообщение об ошибке и завершает программу.

3. Метод "Read", который запрашивает у пользователя значения цены и количества товара, затем вызывает метод "Init" для инициализации полей структуры.

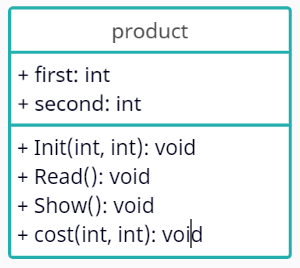
4. Метод "Show", который выводит на экран значения полей структуры.

5. Метод "cost", который вычисляет стоимость товара как произведение цены и количества.

6. Функция "make\_Product", которая создает новый объект структуры "Product" и инициализирует его значениями, переданными в качестве аргументов.

7. В функции "main" создаются два объекта структуры "Product" - "p1" и "p2". Первый объект инициализируется с помощью функции "make\_Product", второй объект заполняется пользователем с помощью метода "Read". Затем для обоих объектов выводятся значения полей и стоимость товара.

UML – Диаграмма



Ответы на вопросы

1. Класс – это шаблон для создания объектов, определяющий состояние (поля) и поведение (методы) объектов.
2. Объект (экземпляр) класса – это конкретный экземпляр класса, созданный на основе его шаблона, имеющий свое состояние и поведение.
3. Поля класса – это переменные, определяющие состояние объектов класса.
4. Функции класса – это методы, определяющие поведение объектов класса.
5. Спецификаторы досту па определяют уровень доступности для полей и методов класса извне.
6. Спецификатор public используется для определения открытых полей и методов класса, которые могут быть доступны извне класса.
7. Спецификатор private используется для определения закрытых полей и методов класса, которые доступны только внутри класса.
8. Если описание класса начинается со спецификатора class, то спецификатор доступа по умолчанию будет private.
9. Если описание класса начинается со спецификатора struct, то спецификатор доступа по умолчанию будет public.
10. При описании интерфейса класса должен использоваться спецификатор доступа public, чтобы методы интерфейса были доступны извне класса.
11. Значения атрибутов экземпляра класса можно изменить, вызвав соответствующий метод – модификатор (сеттер) или обратившись к атрибуту напрямую, если он открыт для доступа извне класса.
12. Значения атрибутов экземпляра класса можно получить, вызвав соответствующий метод – селектор (геттер) или обратившись к атрибуту напрямую, если он открыт для доступа извне класса.
13. Чтобы обратиться к полю name объекта s, нужно использовать оператор "->"
14. Чтобы обратиться к полю name объекта s, нужно использовать оператор "."
15. Аналогично случаю с структурой, чтобы обратиться к полю name объекта s, нужно использовать оператор "->":
16. Чтобы обратиться к полю name объекта s, нужно использовать оператор ".":
17. Чтобы обратиться к полю name объекта s, нужно использовать оператор "->":

Текст программы

#include <iostream>

using namespace std;

struct Product {

int first;

int second;

void Init(int f, int s) {

if (f < 0 || s < 0) {

cout << "Error. Values ​​are incorrect" << endl;

exit(0);

}

first = f;

second = s;

}

void Read() {

cout << "Enter price: ";

cin >> first;

cout << "Enter Quantity: ";

cin >> second;

Init(first, second);

}

void Show() {

cout << "Price - " << first << ", Quantity - " << second << endl;

}

int cost() {

return first \* second;

}

};

Product make\_Product(int f, int s) {

Product pr;

pr.Init(f, s);

return pr;

}

int main() {

Product p1, p2;

p1 = make\_Product(35, 50);

cout << "Product 1: ";

p1.Show();

cout << "Product price 1 - " << p1.cost() << endl;

p2.Read();

cout << "Product 2: ";

p2.Show();

cout << "Product price 2 - " << p2.cost() << endl;

return 0;

}