Рудкевич Д.А.

Объектно-ориентированное программирование на языке c++

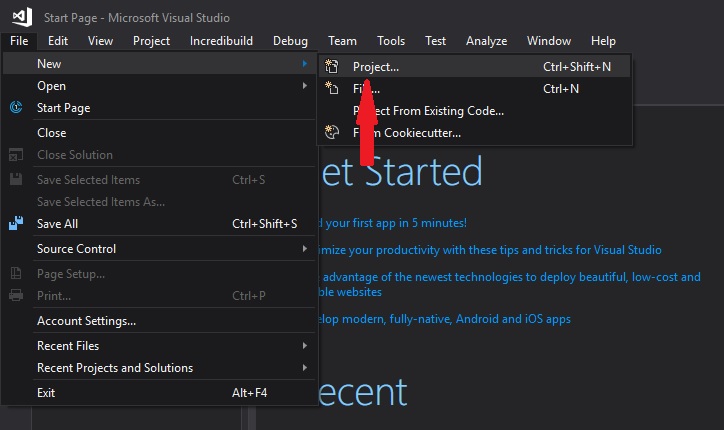
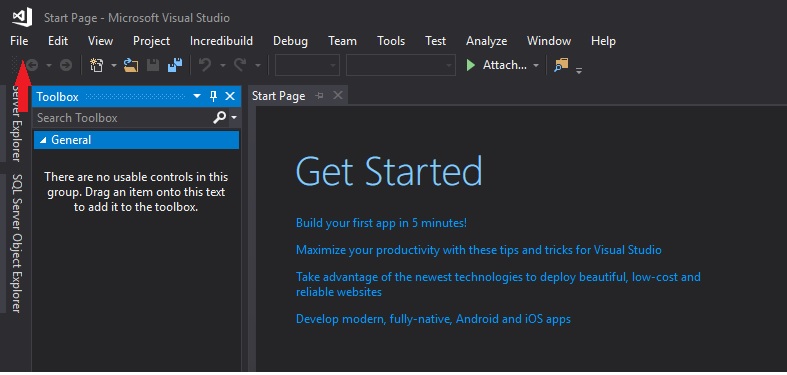
Лабораторная работа №1

Часть 1.Среда разработки Visual Studio 2017.

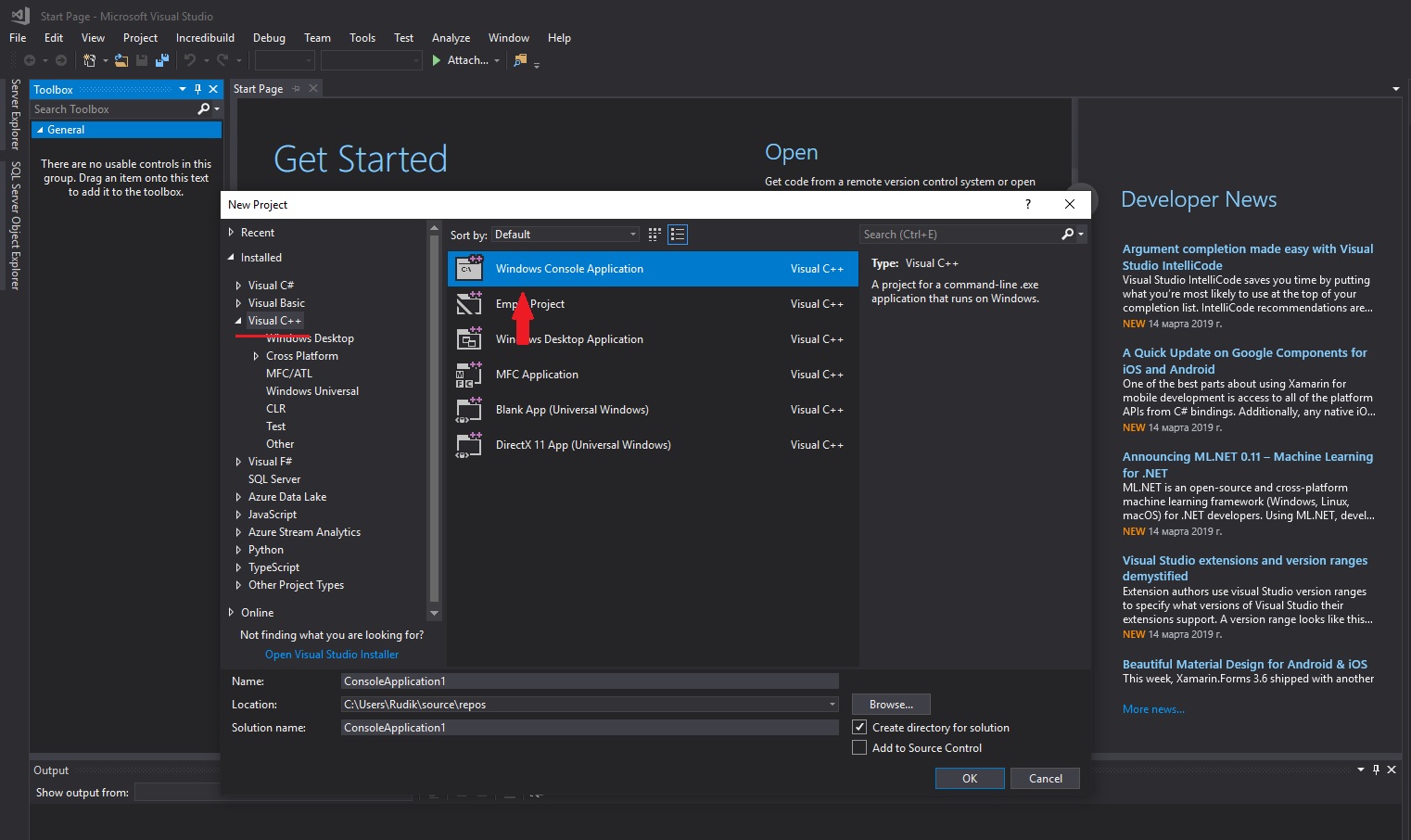
# Создание простейшей консольной программы.

**1. Создание нового проекта. Сначала запустите Visual Studio. Затем проделайте перечисленные ниже действия.**

1) Выберите команду верхнего меню File→New→Project или нажмите сочетание клавиш Ctrl+Shift+N.



2)В открывшемся окне выберите проект типа Windows Console application.

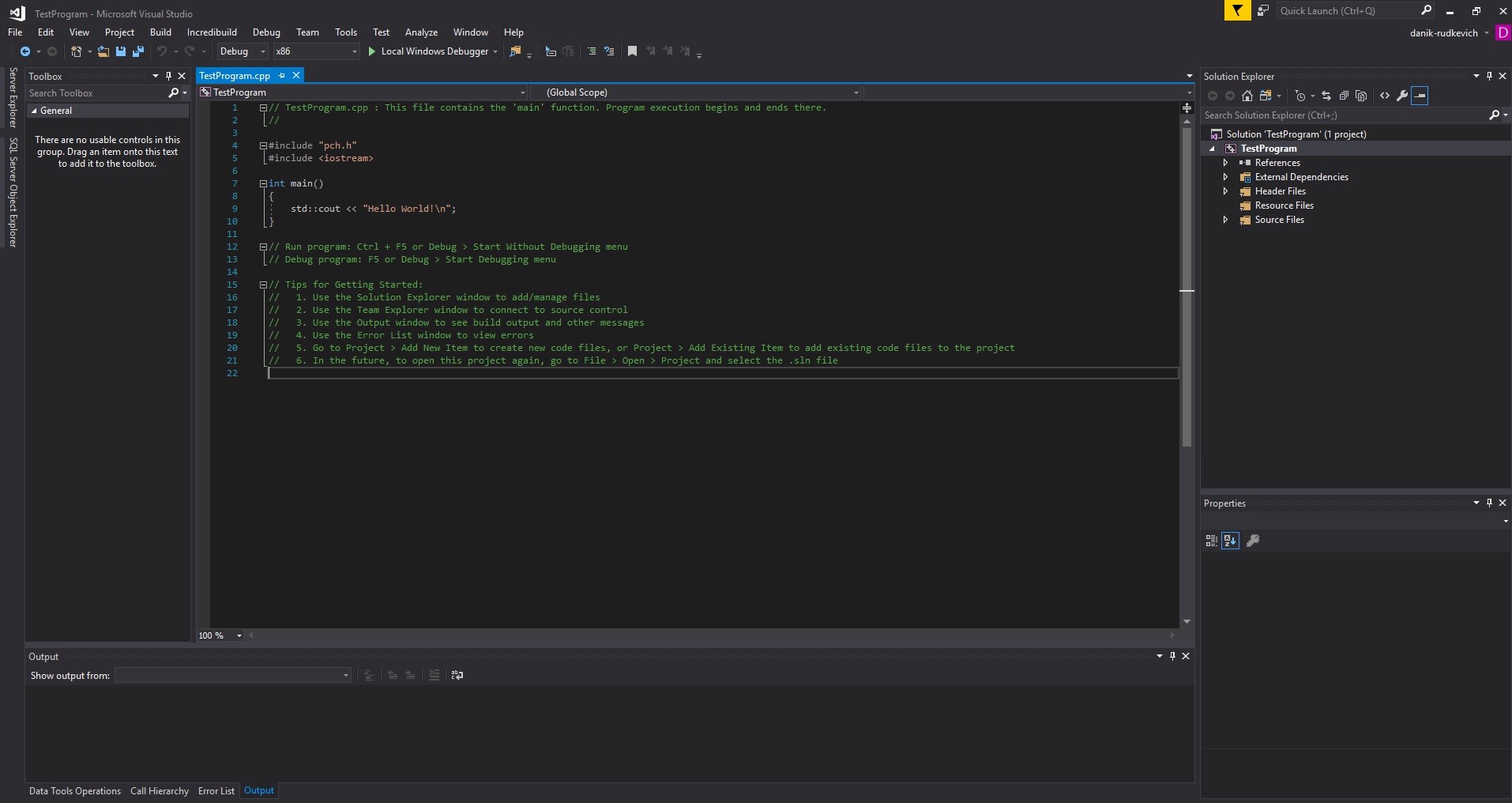


3) В строке Location укажите папку диска C:\, имя которой совпадает с вашей папкой (например, "C:\work\Ivanov") или воспользуйтесь кнопкой Browse для более удобного и быстрого указания пути где будет располагаться ваш проект. В строке Project Name введите "hello\_world\_console". По умолчанию Visual Studio сделает новую папку проекта C:\work\Ivanov\hello\_world\_console. В ней будут сохраняться все файлы, относящиеся к данному проекту.

4) Нажмите кнопку OK.

**2. Создание простейшей программы**

После выполнения действий указанных выше , у вас должен появится редактор кода с прописанным минимумом библиотек необходимых для компиляции программы и библиотекой потокового ввода и вывода <iostream> для вывода самой знаменитой фразы “Hello World!” и тем самым демонстрируя работоспособность программы.



Если по какой то причине данный код не сгенерировался сразу или же вы выбрали для работы пустой проект , то напечатайте данный код , чтобы проверить работоспособность вашего проекта.  
  
#include "pch.h"

#include <iostream>

int main()

{

std::cout << "Hello World!\n";

}

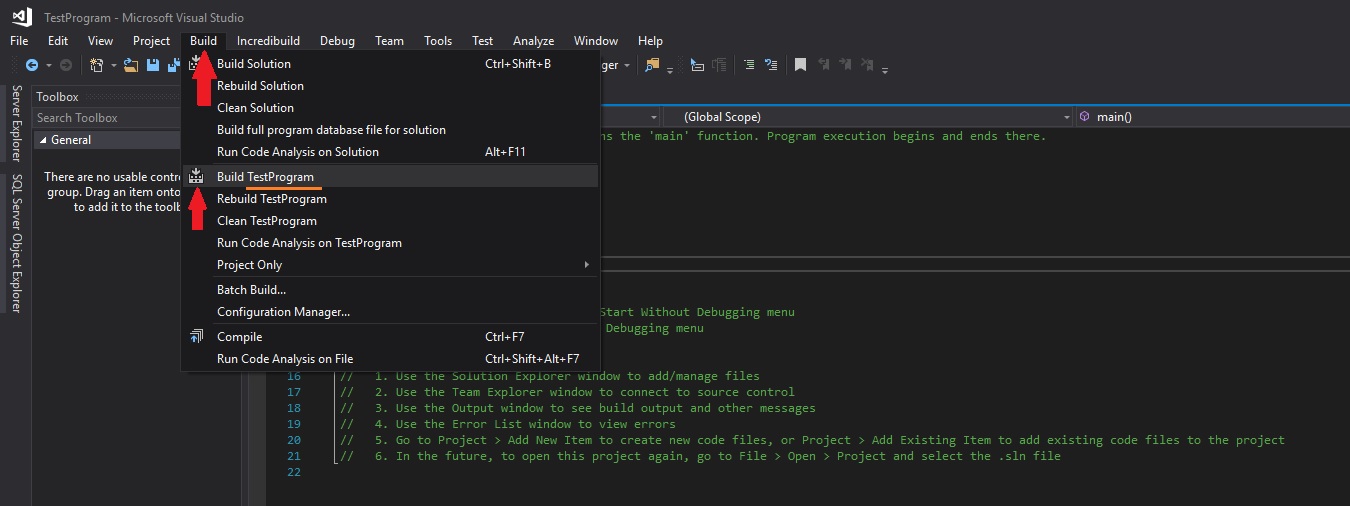
Если сообщение “Hello World!” вывелось, значит работает- не трогай!

Чтобы сохранить набранный текст на диске, выберите команду меню File→Save.Но ленивые могут просто запустить отладку программы нажав на клавишу f5 и после отладки (вне зависимости от успеха отладки), код сохранится.

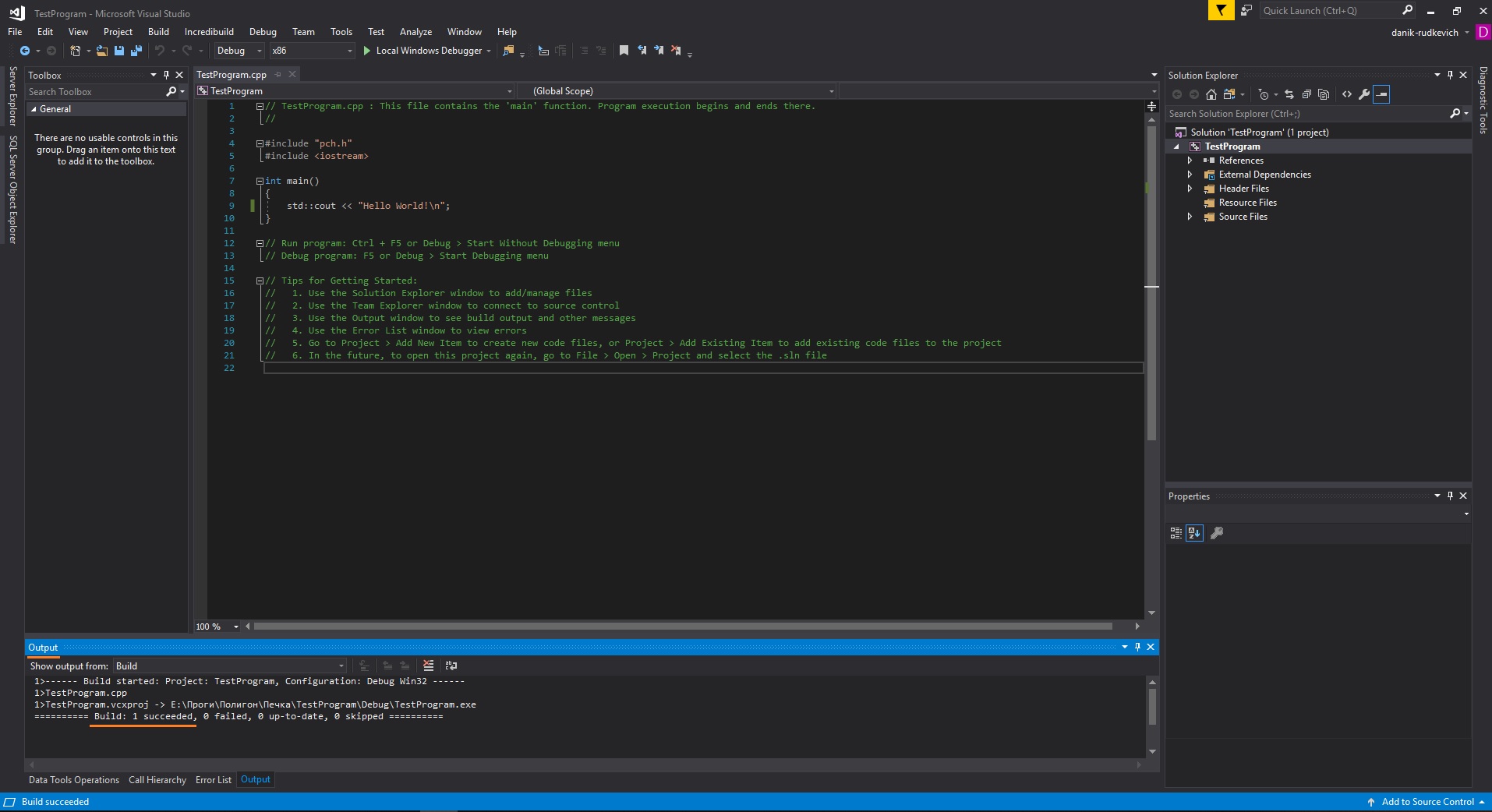
Сборка проекта

Результатом сборки проекта является исполняемый файл программы, который может работать независимо от Visual Studio.Этот исполняемый файл будет хранится в папке “ Debug” созданной в той же директиве , где хранится ваш проект.

Для сборки проекта выберите команду меню Build→” Название вашего проекта”.В данном слайде проект называется “TestProgram”.



В нашем примере проект содержит только один исходный файл. В процессе сборки он будет скомпилирован и скомпонован со стандартной библиотекой ввода/вывода. Информация о выполнении сборки от отображается в окне Output.



В нем выводятся сообщения, выдаваемые программами, работающими при сборке проекта: препроцессором, компилятором и компоновщиком. Среди этих сообщений могут встретиться сообщения об ошибках (errors) и предупреждения о возможных ошибках (warnings). Если таких сообщений не возникло, значит, сборка успешно завершена.

Если есть ошибки, их надо устранить и снова попытаться собрать проект!

Часть 2. Введение в тип данных КЛАСС.

ООП - методология, основанная на представлении программ в виде совокупности объектов, которые между собой взаимодействуют, причем каждый объект является реализацией конкретного класса. То есть основным элементом ООП является класс.

Класс — это тип данных, вводимый пользователем. Классы, как правило, организованы иерархически. Основное назначение класса - описание состава, основных свойств и поведения будущих объектов этого типа данных (введение в проекте некоей абстракции). При этом каждый класс имеет общедоступную часть - интерфейс, а также недоступную, скрытую от пользователя личную часть или реализацию, которая и представляет внутреннее строение будущих объектов данного типа. Таким образом, в ООП абстрагирование - процесс введения типа данных «класс», т.е. таких существенных характеристик некоторых будущих объектов, которые и будут отличать их от других видов объектов. Общий формат декларации типа данных «класс» следующий:

вид\_класса ID\_класса (идентификатор данного типа данных)

{ элементы-данные; // Далее - просто «данные» элементы-функции;

// Далее - просто «методы»

};

Рассмотрим краткую характеристику элементов, составляющих описание класса.

Атрибут «вид\_класса» Виды типа данных «класс»: union struct class

Атрибуты доступа Элементы, входящие в шаблон класса, могут иметь следующую степень защищенности:

public - глобальный, общедоступный атрибут доступа; в этом случае элементы общедоступны из любой функции проекта;

private - локальный, частный; элементы закрыты от общего доступа, и с ними могут работать только методы класса;

protected - защищенные элементы класса, к ним имеют доступ методы данного класса и методы классов, производных от данного класса. Как правило, закрытые элементы - это внутреннее строение объектов, которое нас не должно интересовать.

Все три вида класса имеют степень защиты элементов, которую компилятор установит по умолчанию.

Вид union - степень защиты public, управлять ею нельзя.

Вид struct - степень защиты public, управлять защитой можно.

Вид class - private, степенью защиты также можно управлять.

Для управления защитой элементов, входящих в состав класса, необходимо явно указать нужный атрибут, после которого добавить символ : (двоеточие).

Пример  
  
class X

{

Int a,b;//Поля a и b будут иметь доступ private по умолчанию

Public:

double a1,a2; //Поля a1 и a2 будут иметь доступ Public

Protected:

Char p,l; //Поля p и l будут иметь доступ Protected  
};

Существует два способа создания объектов с таким составом данных.

1) Между символами «}» и «;» помещается список идентификаторов объектов, и уже на этапе компиляции будут созданы объекты с указанным типом данных. При таком способе задания атрибут ID\_класса можно опускать.

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int a, b;

double c;

} m, \*t = &m;//Инициализируем объект m типа Test и указатель этого типа t который указывает на наш объект m

int main()

{

//Поля объекта m в данном пример не инициализированы так как в классе не объявлен конструктор для их инициализации

// Инициализировать поля можно вручную

t->a = 5;//Не забываем про то , что t - это указатель и сейчас мы даем значение 5 полю "a" для объекта m через косвенное обращение

m.b = 3;//Даем значение 3 полю "b" объекта m прямым обращением

cout<<t->a;//Косвенное обращение к полю "a" объекта m и вывод его значения

cout << endl << m.b;//прямое обращение к полю "b" объекта m и вывод его значения

}

2) Декларация объектов класса в любом месте программы по мере надобности: ID\_класса список ID объектов;

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int a, b;

double c;

};

int main()

{

//Поля переменных t и m в данном пример не инициализированы так как в классе не объявлен конструктор для их инициализации

Test X;//Инициализируем объект X

X.a = 4;//Даем значение полю a объекта X

Test \*Y;//Создаем указатель

Y = &X;//Указатель теперь указывает на место расположения объекта X в памяти

cout<<Y->a;//Косвенное обращение к полю a объекта X , так как Y хранит указатель , на адрес объекта X в памяти

cout<<endl<<X.a;//прямое обращение к полю b

}

Данные, входящие в состав класса, - любой допустимый вид данных языка C++, за исключением файлов.

Методы, входящие в состав класса, можно разделить на три группы.

1. Методы, определяющие операции над данными.

2. Функции-конструкторы, основная задача которых - создание и инициализация объектов данного класса разнообразными способами.

3. Функции для уничтожения объектов после их использования - деструкторы.

Внешние функции, не входящие в состав класса, могут работать только с элементами класса, имеющими атрибут доступа public.

Декларация методов класса возможна в двух формах:

- в форме определения, при которой приводится полный текст метода;

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int a, b;

double c;

void Show()//Полный текст метода написан внутри класс

{

cout << a<< b;

}

};

int main()

{

Test X;//Инициализируем объект X

X.a = 5;

X.b = 9;

X.c = 0;

X.Show();

}

- в форме описания (прототип метода).

В данном случае вне шаблона должно быть полное определение метода, при этом необходимо указать, к какому классу принадлежит данный метод. Для этого используется операция привязки «::». Назначение этой операции - привязать функцию к конкретному классу.

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int a, b;

double c;

void Show();//Инициализируем метод в классе

};

void Test::Show()//Пишем код метода Test:: - говорит о том , что данный код , должен выполнится при вызове функции Show() инициализированной в классе Test

{

cout << a << b;

}

void Show()//Мы можем создать метод с таким же названием и такой же сигнатурой , но для объекта код этого метода не выведется так как нет привязки к классу в отличии от случая описаного выше

{

cout << "User metods!";

}

int main()

{

Test X;//Инициализируем объект X

X.a = 5;

X.b = 9;

X.c = 0;

X.Show();//Output: 59

Show();//Output : User metods!

}

Использование этой операции вводит новое определение:

Полное квалификационное имя Запись X:: а означает: «Элемент **а** принадлежит классу **X**».

Запись X ::f1 () означает: «Функция f1 принадлежит классу X».

И в заключение рассмотрим некоторые приемы работы объектами и методами.

Пусть в процессе работы программы создан объект класса ID\_класса ID\_объекта; и пусть в составе класса есть метод с идентификатором ID\_метода. Тогда:

1. Прямой вызов метода ID\_объекта.ID\_метода; //- использована операция привязки «точка».

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int a, b;

double c;

void Show()//Полный текст метода написан внутри класс

{

cout << a << b;

}

};

int main()

{

Test X;//Инициализируем объект X

//прямой вызов полей

X.a = 5;

X.b = 9;

X.c = 0;

//Прямой вызов метода

X.Show();

}

2. Использование косвенной адресации Пусть объявлен указатель с типом ID\_класса \*ID\_указателя;

Теперь указатель типа ID\_класса можно устанавливать на существующие объекты этого класса: ID\_указателя = & ID\_объект; - указатель идентифицирован адресом объекта.

Косвенный вызов метода: ID\_указателя -> ID\_метода; - использована операция привязки«стрелка».

Пример:  
#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int a, b;

double c;

void Show()//Полный текст метода написан внутри класс

{

cout << a << b;

}

};

int main()

{

Test X;//Инициализируем объект X

Test \*Y;//Инициализируем указатель типа Test

Y = &X;//Теперь указатель Y указывает на адрес объекта X и теперь мы можем косвенно к нему обращаться

//Косвенный вызов полей

cout << Y->a << Y->b << Y->c;

//Косвенный вызов метода

Y->Show();

}

3. Создание объектов (безымянных) в динамической областипамяти ID\_класса \* ID\_указателя - new ID\_класса; Компилятор создает объект на этапе работы проекта. Косвенный вызов метода: ID\_указателя -> ID\_метода;

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int a, b;

double c;

void Show()//Полный текст метода написан внутри класс

{

cout << a << b;

}

};

int main()

{

Test \*Y = new Test();//Создаем динамический объект в памяти на расположение которого указывает указатель Y

//Косвенный вызов полей

cout << Y->a << Y->b << Y->c;

//Косвенный вызов метода

Y->Show();

}

Уровень 1:

Упражнение: Написать объектную программу, иллюстрирующую прямой и косвенный способы обращения к методам. Пользовательский класс должен содержать необходимые элементы - данные, метод установки их начальных значений, метод просмотра текущего состояния полей класса, метод, решающий поставленную задачу. Описание метода должно быть вне класса.

Варианты:  
  
1)Вы свободный фрилансер и на днях к вам обратилась компания «BestGameFromAss» , они просят вас помочь им (Разумеется не за бесплатно). Ваша задача : создать класс Player, в котором будет хранится HP, MagicPower , Stamina (все поля типа int) и написать метод , который будет выводить крутость игрока. Крутость игрока рассчитывается по формуле:

(HP \* MagicPower) / Stamina + (HP \* Stamina) / MagicPower – (Stamina\* MagicPower) / HP;

(Не забывайте про деление на нуль. В этом случае должно быть выведено сообщение об ошибке с указанием на поле которое было == 0)

2)Добро пожаловать в Научно-практический центр исследований «Explore all you want».

Вы младший научный сотрудник и сегодня вы изучаете … Плесень! Да, младшим научным сотрудником ещё мало что доверяют, но вы докажите свой профессионализм! Цель вашего исследования определить кол-во плесени через определенный промежуток времени. Вам известны: Кол-во начальной плесени, её прирост в день и кол-во дней, которое она будет размножаться с такой скоростью. Для того , что бы не пачкать бумагу на расчёты и тем самым сохранить природу , создайте класс Mold в котором будут 3 поля (типа int) : speed,day,kol и напишите метод , который вернет кол-во плесени через промежуток времени введенный в поле day;

3)Сегодня вы заступили на должность глав-врача больницы «Healthy like a bull». И первое что бы вы хотели узнать – это процент выздоравивших больных и симулянтов от общего числа пациентов. Чтобы все отделения могли быстро выдать вам правильный результат, не запутавшись в этих процентах и делениях, создайте класс с 3 полями (типа int) : convalescents, sick , simulators и метод который вернет кол-во выздоравивших , больных и симулянтов в % от общего числа пациентов. На всякий случай , общее кол-во пациентов (convalescents+ sick + simulators);

4)В банке «We are the most honest» где вы проходите стажировку хотят поставить новый автомат для оплаты каких-то услуг. Механизм автомата элементарен. Клиент вставляет свою карточку, выбирает услугу, которую он хочет оплатить и если у него хватает денег на карточке, то услуга оплачивается. Но есть одно, НО! За свои услуги автомат списывает комиссию с карточки в 5% от цены услуги (надо внимательней читать правило пользования автоматом которые приклеены позади автомата). Ваша задача: создать класс Machine в котором будет 3 поля (2 int residue- double): money, price, residue. Пользователь вводит значения для полей

money и price (residue изначально == 0), поле residue заполняется при выполнении метода которая посчитает остаток после оплаты операции (Не забываем про списываемые автоматом 5%).В случае успеха метод выводит сообщение об успехе, в случае нехватки денег, выводит сообщение о неудаче.

5)С сегодняшнего дня в ваше распоряжение переходит полицейский участок «liberty». И первое что вы хотели бы узнать – это процент простых, средних и сложных вызовов от общего числа вызовов. Чтобы сотрудники быстрее смогли дать вам эту важную информацию, создайте класс Call в котором будет хранится 3 поля (тип int) : Easy, Middling,Hard и метод который выведет кол-во простых, средних и тяжелых вызовов в % от общего числа вызовов. На всякий случай , общее кол-во пациентов (Easy + Middling + Hard);

6)Сегодня вы заступаете на должность Мэра славного города «Racun City».Вам срочно надо узнать уровень развития своего города. Так как город очень быстро растет , чтобы не производить сложные расчеты каждый раз после появления чего-то нового , вы записались на курсы «ООП за один день» и теперь можете создать программу которая сделает это за вас! Ваша задача: Создать класс City в котором будут 3 поля : ResidentialBuildings, IndustrialBuilding , UniqueBuildings и метод который будет выводить уровень развития города на основе этих полей.Формула для расчета уровня развития города:

( (ResidentialBuildings\* IndustrialBuilding) – ResidentialBuildings ) \* UniqueBuildings) / IndustrialBuilding;

(Не забывайте про деление на нуль. В этом случае должно быть выведено сообщение, что уровень развития города == 0)

7)Открыта новая фабрика по производству носков «Warm feet» и вы её новый директор! Уже прошел месяц со старта фабрики и вам очень хочется узнать сколько же вы заработали. Напишите класс Socks в котором будет 3 поля(int): produced, Sales , Price и метод который выведет прибыль от продаж и кол-во непроданного товара.

8)Вы устроились администратором в отель « flophouse».Это большой отель , вы даже не знаете сколько в нем этажей, вам лишь известно , что на последнем этаже комнат может быть меньше , чем на всех остальных этажах , так же вам известно общее кол-во комнат и кол-во комнат на каждом этаже (кроме последнего).Давайте узнаем , сколько же этажей в отеле , которым нам надо руководить(Лифт сломан , а подыматься на последний этаж и смотреть вам лень).Ваша задача: Написать класс Hotel в котором будет 2 поля(int): NumberOfRoomsOnTheFloor ,NumberOfRooms и метод который выведет кол-во комнат на последнем этаже и кол-во этажей в вашем отеле.

Уровень 2:

Упражнение: Написать объектную программу, иллюстрирующую прямой и косвенный способы обращения к методам. Пользовательский класс должен содержать необходимые элементы - данные, метод установки их начальных значений, метод просмотра текущего состояния полей класса, метод, решающий поставленную задачу. Описание метода должно быть вне класса.

Выполнение задания уровня 2 должно происходить в классах созданных для выполнения уровень 1 , весь функционал и поля уровень 1 должны быть сохранены!

Варианты:  
  
1)Вы успешно выполнили заказ и уже думаете куда поехать отдыхать на заработанные деньги , да вот только заказчик вспомнил , что персонаж ещё и имеет физический и магический урон и просит срочно добавить эти поля и вывести значения этого урона у персонажа с бонусом получаемым от Hp, MagicPower и Stamina.Заварив ещё одну чашечку кофе , вы приступили к работе. Ваша задача: Добавить в ранее написанный класс 2 поля (double) : MagicDamage , PhysicalDamage и добавить ещё один метод который будет выводить магический и физический урон по формулам:  
Магический урон: (MagicDamage\* MagicPower)- Stamina;

Физический урон: (PhysicalDamage \* Stamina + Hp ) – MagicPower;

2)Вы успешно написали программу и вычислив предполагаемое кол-во плесени через N дней , вы со спокойной душой ушли в отпуск, но вернувшись , вы обнаружили , что кол-во плесени не совпадает с тем значением , которое вам прогнозировала ваша программа. Ошибку в программе вы исключили сразу (Вы же лично её писали! В ней не может быть ошибок!).Выпив кофейку , вы поняли в чем дело. Каждый день помимо прибавлении плесени , шло и отмирание старой. Вычислив кол-во которое умирает в день , вы готовы доделать программу! Ваше задание :  
Добавить поле (int) : DeathSpeed и в методе которое должно выводить кол-во плесени через

промежуток времени day учтите эту переменную.

3)Вы получили статистику пациентов, а что насчет персонала? Сколько людей на вас вообще работает? Это надо срочно выяснить , да ещё и в процентах чтобы выводилось! Ваша задача:

Добавить 3 поля (int) : Doctor, Nurse, MedicalOrderly и метод который выведет кол-во сотрудников каждого типа в процентах от общего кол-ва сотрудников;

4)Вашему начальнику очень нравится работа автомата для оплаты каких-то услуг, теперь он хочет добавить туда новые опции. Теперь , пользователь должен иметь возможность добавить себе денег на карту взяв у банка деньги а потом в течении месяца вернуть ровно столько , сколько он указал в автомате (Что-то вроде обещанного платежа на телефоне).Вот только , начальник попросил оставить функцию изъятия 5% автоматом за выполнения операции на автомате.То есть,если пользователь возьмет 100 рублей у банка в долг, то автомат возьмет комиссию с этих денег 5% и перечислит пользователю 95 рублей , а вернуть пользователь должен будет 100 рублей. Со словами «Это не баг , это фича» вы пошли выполнять просьбу началства.Ваша задача: добавить поле (int) : Debt и добавить метод , который увеличит значение money на Debt с вычетом 5% комиссии и выведет старое значение money,значение money после пополнение и сумму долга перед банком;

5) Вы получили статистику вызовов, а что насчет персонала? Сколько людей на вас вообще работает? Это надо срочно выяснить, да ещё и в процентах чтобы выводилось! Ваша задача:

Добавить 3 поля (int) : policeman, SpecialForces, Detectiveи метод который выведет кол-во сотрудников каждого типа в процентах от общего кол-ва сотрудников;

6)Посмотрев на карту города , вы обнаружили , что у вас есть ещё и парки с магазинами , а это тоже надо учитывать при расчете уровня развития города. Завариваем кофе и приступаем к работе!Ваша задача : добавить 2 поля (int) : Park , Store .Новая формула для расчета уровня развития города:

( (ResidentialBuildings\* IndustrialBuilding \* Park + Store) – ResidentialBuildings ) \* (UniqueBuildings+ Park+ Store) ) / IndustrialBuilding;

(Не забывайте про деление на нуль. В этом случае должно быть выведено сообщение, что уровень развития города == 0)

7)Прибыль от продажи носков вас приятно удивила и вы решили расширяться. Долго думая , что бы ещё произвести , вы пришли к мысли , что чулки – это просто длинные носки и ваша фабрика вполне может их производить. Объяснив рабочим , что вы хотите , вы приступили за модификацию программы.Ваша задача : Добавить 3 поля(int): StockingsProduced, StockingsSales , StockingsPrice и в методе который выведет прибыль от продаж носков и кол-во непроданного товара вывести аналогичную информацию для чулков.

8)Вы смогли посчитать кол-во этажей в вашем отеле. Теперь посчитаем сколько прибыли вам приносит ваш отель. Узнав цену за одну комнату и кол-во жильцов , вы приступили к работе. Ваша задача: Добавить 2 поля(int): Price, tenant и метод который выведете прибыль отеля.

Уровень 3

Два бандита Джек и Фрэнк решили выйхать отдохнуть на природу. Пока жарился шашлык Джек и Фрэнк решили пострелять в банки из под кока-колы чтобы убить время.Они поставили банки на бревно(Не больше 15) первый начал стрелать справа , второй начал стрелять слева.В один момент они оба выстрелили в одну последнию банку из-за чего началась ссора. Джек утверждал что Фрэнк теперь должен отдать ему свой шампур шашлыка за то что тот лешил его удовольствия в простреливании последней банки.В ответ Фрэнк сказал что это Джек должен отдать ему свой шампур шашлыка по тем же причинам. В итоге они решили что Джек отдаст свой шампур с шашлыком Фрэнку , а Фрэнк Джек отдаст свой шампур с шашлыком Джеку.Теперь осталось только убрать банки , вот только из за ссоры никто не помнил сколько именно банок они поставили для срельбы.Каждый из них помнил только сколько банок он подсрелил сам.Давайте поможем Джеку и Френку узнать сколько банок у них было изначально и сколько банок не успел подстрелить Джек и сколько банок не успел подстрелить Френк.  
Ваша задача : Создать класс Relaxation в котором будет 2 поля Frank и Jack в которых будет хранится информация о том , сколько банок подсрелил каждый из них.Так же класс должен содержать метод который выведет сколько банок не успел прострелить Frank и сколько банок не успел прострилить Jack.  
Входные данные:

2 числа — количество банок, простреленных Фрэнком и Джеком соответственно.

Выходные данные:

3 числа — количество банок, не простреленных Фрэнком и Джеком соответственно и общее кол-во банок.

|  |  |
| --- | --- |
| Input | Output |
| 4 7 | 6 3 10 |

**Контрольные вопросы:**

1)Что является основным элементом ООП и расскажите, что входит в его состав.

2)Перечислите виды классов и укажите их различия.

3)Укажите способы защиты элементов входящих в состав класса.

4)Что такое операция привязки. Приведите пример(письменно!).

5) Дайте понятие полного квалификационного имени.

6)Почему этот код не работает?

#include "pch.h"

#include <iostream>

class Test

{

public:

void Show()

{

cout << "Hello world!";

}

};

using namespace std;

int main()

{

Test a;

a.Show();

}

7)Какие поля и функции будут доступны объекту данного класса?

class Test

{

int a, b;

double o;

void Show()

{

cout << "Hello world";

}

public:

int u, p;

void View()

{

cout << "Hello world 2";

}

protected:

double l, rt;

};

8)Что выведет данный код?

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

int a = 19, b = 32;

void Show()

{

cout <<endl<< "a="<<a<<endl<<"b="<<b;

}

};

int main()

{

Test a;

a.Show();

}

Лабораторная работа №2

Использование конструктора и деструктора. Работа с динамической памятью.

Для создания и инициализации объекта используется функция, получившая название Конструктор, которая определяет (контролирует), как создается и инициализируется объект.

После использования объектов их необходимо уничтожить.

Для этого предназначена функция - Деструктор, которая определяет, как и когда уничтожить объект.

Если этих методов явно в программе нет, компилятор при создании экземпляра класса вызовет стандартный конструктор, который и создаст объект после его декларации. После выхода из функции (из блока), в которой был создан объект, компилятор вызовет деструктор «по умолчанию», который его уничтожит. Если же объект создавался как глобальный, деструктор будет вызван по завершении работы программы.

Некоторые особенности конструктора и деструктора

1) Эти функции не имеют возвращаемого значения (даже void).

2) Они не наследуются производными классами, хотя и вызываются из них.

3) Нельзя работать с адресами этих функций.

4) К конструктору нельзя обратиться напрямую как к обычному методу.

5) Деструктор можно вызывать, используя его полное квалификационное имя.

6)Если деструктор или конструктор объявлены в поле private или protect, то они не смогут быть вызваны

Рассмотрим эти два метода более подробно.

Конструктор

Конструктор - функция-член класса, определяющая способ создания объекта. Имя конструктора должно совпадать с именем класса. Существует несколько форм конструктора. Наиболее часто используется конструктор с параметрами. Эти параметры используются для одновременной инициализации создаваемых объектов. Конструктор как функция может иметь умалчиваемые значения параметров. В программе может быть несколько конструкторов. Такие конструкторы часто называют конструкторы с перекрытием. Здесь работает механизм перегрузки функций.

Пример (Для лучшего понимания, советую запустить этот код с пошаговым выполнением при помощи клавиши f11):

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int a, b;

double c;

Test()//Конструктор без параметров

{

a = 5;

b = 6;

c = 7;

}

Test(int r, int t) //Конструктор с параметрами

{

a = r;

b = t;

c = 0;

}

void Show()

{

cout <<" "<< a << " " << b << " " << c;

}

};

int main()

{

Test A;//Вызовет конструктор без параметров

Test B(12,34);//Вызовет конструктор с параметрами

A.Show();//Output: 5 6 7

B.Show();//Output: 12 34 0

}

Конструктор - это специальная функция класса, поэтому допускается вторая форма его декларации: в виде прототипа и его определения вне пространства класса.

Пример:  
  
#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int a, b;

double c;

Test();//Конструктор без параметров

Test(int r, int t); //Конструктор с параметрами

void Show()

{

cout <<" "<< a << " " << b << " " << c;

}

};

Test::Test()//пишем код конструктора без параметров

{

a = 2;

b = 42;

c = 90;

}

Test::Test(int r, int t)//пишем код конструктора с данной сигнатурой

{

a = r;

b = t;

c = 23;

}

int main()

{

Test A;//Вызовет конструктор без параметров

Test B(12,34);//Вызовет конструктор с параметрами

A.Show();//Output: 2 42 90

B.Show();//Output: 12 34 23

}

Деструктор - специальная функция класса, которая отвечает за уничтожение объекта. Имя деструктора совпадает с именем класса, перед которым ставится символ «тильда» — ~.

Если декларирован class X {...}; то деструктор для него ~Х(){}

Пример:

class Test

{

~Test()//Деструктор класса Test

{

};

};

class ABCD

{

~ABCD()//Деструктор класса ABCD

{

};

};

То есть если элементы-данные класса имеют известный размер в байтах, например: int т; (2 байта), double x; (4 байта) и т.д. Компилятор при создании объектов выделит соответствующие участки памяти, и в этом случае деструктор не нужен, т.к. он будет иметь пустое определение. 28 Если же объект создается в динамической области памяти, например, с помощью операции захвата памяти new, то деструктор необходимо задавать явно, т.к. он должен уничтожить объект после его использования с помощью операции delete. Рассмотрим кратко эти две операции.

**Операции new и delete (C++)**

Управление программным размещением объектов в памяти, т.е. выделение памяти в процессе работы программы под переменные и массивы в языке C++, осуществляется с помощью операций new и delete. Операция new автоматически определяет размер выделяемой памяти для указанного типа, а также позволяет инициализировать вновь создаваемый объект.

Формат операций: ID\_указателя = new тип (значение); // Захват памяти (работа с динамическим объектом через косвенную адресацию - операция «\*»)

deleteID\_указателя; //Освобождение памяти после работы с объектом где ID\_указателя - идентификатор переменной-указателя на заданный тип;

тип - тип значений, для которых выделяется память;

значение - константа (необязательный атрибут), определяющая начальное значение динамической переменной.

Таким образом, операция new устанавливает указатель на участок свободной динамической оперативной памяти размером sizeof(num).

Если выделить память не удается, то возвращается нулевой указатель, что дает возможность проконтролировать процесс захвата участка динамической памяти. Объект существует до тех пор, пока память не будет освобождена при помощи операции delete (или до окончания работы программы). Следующий участок программы демонстрирует работу с целочисленной динамической переменной.

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

int \*p = new int(6);//Выделяем динамическую память под тип int и ложим туда значение 6

if (!p)//Проверка на случай не выделения памяти , в этом случае в указатель вернется null

{

cout << "Eror";

}

cout << \*p << " ";//Output: 6

\*p = 45;//Даем новое значение в нашей динамической памяти

cout << \*p;//Output: 45

delete p;//Освобождаем память

}

Указатели которые указывают на освобожденную память называются висячими указателями.

В нашем примере , после применения delete p; указатель p – стал висячим.

Внимание!!!

Оператор delete на самом деле ничего не удаляет. Он просто возвращает память, которая была выделена ранее, обратно в операционную систему. Затем операционная система может переназначить эту память другому приложению (или этому же снова).

Хотя может показаться, что мы удаляем переменную, но это не так! Переменная-указатель по-прежнему имеет ту же область видимости, что и раньше, и ей можно присвоить новое значение, как и любой другой переменной.

Вывод или повторное удаление такого указателя может привести к неожиданным результатам.

Спасибо за внимание!!!

Теперь можно видоизменить рассмотренный ранее пример, включив в него работу с динамической памятью, и увеличим все поля класса в 3 раза при помощи метода Run:

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int \*a, \*b;

double \*c;

Test();//Конструктор без параметров

Test(int r, int t); //Конструктор с параметрами

~Test();

void Show()

{

cout << " " << \*a << " " << \*b << " " << \*c;

}

void Run()

{

\*a \*= 3;

\*b \*= 3;

\*c \*= 3;

}

};

Test::Test()//пишем код конструктора без параметров

{

a = new int(2);

b = new int(42);

c = new double(90);

}

Test::Test(int r, int t)//пишем код конструктора с данной сигнатурой

{

a = new int(r);

b = new int(t);

c = new double(0);

}

Test::~Test()

{

delete(a, b, c);

}

int main()

{

Test A;//Вызовет конструктор без параметров

Test B(12, 34);//Вызовет конструктор с параметрами

A.Show();//Output: 2 42 90

B.Show();//Output: 12 34 0

A.Run();//Увеличиваем все поля объекта A в 3 раза

B.Run();//Увеличиваем все поля объекта B в 3 раза

//Выводим новые значения

A.Show();//Output: 6 126 270

B.Show();//Output: 36 102 0

}

Уровень 1:

Упражнение: Продемонстрируйте свое умение работать с конструкторами и деструкторам, а также умение работать с динамической памятью. За основу возьмите класс, созданный для решения предыдущей лабораторной работы (уровень 1 вашего варианта). Все поля, описанные в нем, сделать указателями. Добавьте в те классы конструктор с параметрами который будет служить для инициализации полей и конструктор без параметров в котором значения полям будут даваться такие, какие вы сами захотите. Так же создайте деструктор, который освободит всю память занятую полями. Описание конструкторов и деструктора должно быть выполнено за пределами класса. Так же добавьте функцию Set() , которая будет давать новые значения полям класса. Функция Set() так же должна быть описана за пределами класса.

Уровень 2:

Упражнение: За основу возьмите класс, созданный для решения предыдущей лабораторной работы (уровень 2 вашего варианта). И добавьте все то, что вас просят добавить в упражнение выше.

Уровень 3:  
В отделе работают 3 сотрудника, которые (вы не поверете) получают заработную плату. Проблема этих трёх сотрудников заключается в том , что они делают одинаковую работу , но получают разную зарплату.Давайте напишем программу которая выведет разницу между самой большой и маленькой зарплатой.

Ваша задача : Создать класс Salaryв котором будет 3 поля Salary1, Salary2 и Salary1 в которых будет хранится зарплата каждого сотрудника(Все поля являются указателями типа int).Заполнение полей происходит при помощи конструктора.Так же класс должен содержать метод который выведет разницу между самой большой и самой маленькой зарплатой.  
Входные данные:

3 числа обозначающие зарплату каждого сотрудника.

Выходные данные:

1 число которое является разностью между самой большой и самой маленькой зарплатой.

|  |  |
| --- | --- |
| Input | Output |
| 100 500 1000 | 900 |
| 36 11 20 | 25 |

**Контрольные вопросы:**

1) Что такое “конструктор”, какие его основные особенности?

2) Что такое “деструктор”, какие его основные особенности?

3) Когда наличие деструктора обязательно?

4) Назначение операций new и delete.

5) Сколько существуют динамические объекты?

6)Что такое висячий указатель?

7)Сколько конструкторов и деструкторов может быть в классе?

8)Какой будет выдан результат при выполнении данной программы?

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int m,n;

Test(int a,int b);

Test();

};

int main()

{

Test a(45, 89) , b;

cout << a.m << " " << a.n << endl << b.m << " " << b.n;

}

Test::Test(int a, int b)

{

m = a;

n = b;

}

Test::Test()

{

m = 14;

n = 12;

}

9)Найдите ошибку в данном фрагменте кода:

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int \*m,\*n;

Test(int a,int b);

Test();

~Test();//Создаем деструктор для конструктора Test();

~Test(int m, int n);//Создаем деструктор для конструктора Test(int a,int b);

};

int main()

{

Test a(45, 89) , b;

cout << a.m << " " << a.n << endl << b.m << " " << b.n;

}

Test::Test(int a, int b)

{

m = new int(a);

n = new int(b);

}

Test::Test()

{

m = new int(14);

}

Test::~Test()

{

delete m;

}

Test::~Test(int m, int n)

{

delete m;

delete n;

}

10)Какой будет выдан результат при выполнении данной программы?

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

int m,n;

Test(int a,int b);

Test();

};

int main()

{

Test a(45, 89) , b(34) , c;

cout << a.m << " " << a.n << endl << b.m << " " << b.n<<" "<<c.m << " " << c.n;

}

Test::Test(int a, int b)

{

m = a;

n = b;

}

Test::Test()

{

m = 14;

n = 12;

}

Лабораторная работа №3

Работа с динамической памятью. Одномерный динамический массив.

Как вы уже знаете, при стандартной декларации массивов компилятор воспользуется векторной организацией памяти и выделит фиксированный участок памяти, достаточный для хранения всех его элементов. Использование динамической памяти и списковой ее организации позволяет создавать массивы переменной длины. Помимо уже известных операций по захвату и освобождению динамической памяти new и delete для этих целей используют стандартные библиотечные функции, декларацию которых содержит заголовочный файл. Но все эти функции уже включены в библиотеку "pch.h".

Приведем сведения о наборе функций манипулирования памятью:

Sizeof(type) – функция которая возвращает кол-во байт необходим для хранения переменной типа, указанного в сигнатуре этой функции

void \*malloc (unsigned n); - выделение памяти для размещения блока размером n байт; возвращает указатель на распределенную область или NULL при неудаче;

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

//setlocale(LC\_CTYPE, "rus");//Если кто-то захочет использовать русский язык

double \*x; //Создаем указатель на будующий массив

int n = 5;//Создаем переменную которая будет указывать кол-во элементов массива

x = (double\*)malloc(n\*sizeof(double));//Выделяем необходимое кол-во байтов для 5 элементов типа double при помощи malloc(n\*sizeof(double))

/\*

Предположение:

(double\*) говорит о том , что каждая ячейка будет типа указателя double или то , что мы привязываем память к указателю double?

\*/

for (int i=0;i<5;i++)//Заполняем наш массив

{

x[i] = i+0.2;

}

for (int i = 0; i < 5; i++) //выводим наш массив

{

cout << endl<<x[i];

}

/\*

Output:

0.2

1.2

2.2

3.2

4.2

\*/

free(x);//Освобождаем память

}

void \*calloc (unsigned n, unsigned size); - выделение памяти для размещения n объектов размером size байт и заполнение полученной области нулями; возвращает указатель на захваченную область памяти или NULL при неудаче;

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

//setlocale(LC\_CTYPE, "rus");//Если кто-то захочет использовать русский язык

double \*x; //Создаем указатель на будующий массив

int n = 5;//Создаем переменную которая будет указывать кол-во элементов массива

x = (double\*)calloc(n, sizeof(double));//Выделяем 5 ячеек размером double при помощи calloc(n, sizeof(double))

/\*

Предположение:

(double\*) говорит о том , что каждая ячейка будет типа указателя double или то , что мы привязываем память к указателю double?

\*/

for (int i=0;i<5;i++)//Заполняем наш массив

{

x[i] = i+0.2;

}

for (int i = 0; i < 5; i++) //выводим наш массив

{

cout << endl<<x[i];

}

/\*

Output:

0.2

1.2

2.2

3.2

4.2

\*/

free(x);//Освобождаем память

}

/\*unsigned coreleft (void); - получение размера свободной памяти неиспользованной памяти в куче в байтах : \*/

void free(void \*b); - освобождение блока памяти, адресуемого указателем b;

Пример использования функции free() есть выше;

void \*realloc (void \*b, unsigne n); - изменение размера размещенного по адресу b блока на новое значение п и копирование (при необходимости) содержимого блока; возвращает указатель на перераспределенную область памяти или NULL при неудаче.

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

//setlocale(LC\_CTYPE, "rus");//Если кто-то захочет использовать русский язык

double \*x; //Создаем указатель на будующий массив

int n = 5;//Создаем переменную которая будет указывать кол-во элементов массива

x = (double\*)malloc(n\*sizeof(double));//Выделяем необходимое кол-во байтов для 5 элементов типа double при помощи malloc(n\*sizeof(double))

for (int i=0;i<5;i++)//Заполняем наш массив

{

x[i] = i+0.2;

}

x = (double\*)realloc(x, (n+2) \* sizeof(double)); //Заново выделяем блок памяти для x добавляя ему байт для хранения ещё 2 переменных при этом содержимое ста­рого блока копируется в новый блок и информация не теряется.

for (int i = 5; i < 7; i++) //добавляем новые значения элементам

{

x[i] = i + 0.2;

}

for (int i = 0; i < 7; i++) //выводим наш массив

{

//cout << endl << x[i];

}

/\*

Output:

0.2

1.2

2.2

3.2

4.2

5.2

6.2

\*/

free(x);//Освобождаем память

}

Общий формат операторов new и delete для работы с динамической памятью следующий:

type \*name; // Декларировали указатель на начало массива …

name = new type[size]; // Захватили память

delete [ ] name; // Освободили память где

type - тип элементов,

size - максимальное количество элементов массива name.

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

int \*n;//Указатель на массив

int lenght = 10;//Размер будующего массива

n = new int[lenght];//Выделяем память указателю n память для хранения элементов типа int кол-вом lenght

delete[] n;//Освобождаем память выделенную под массив

}

Рассмотрим пример создания динамического массива в классе:  
  
#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

//Указатели на массивы

int \*IntMass;

double \*DoubleMass;

int IntLenght, DoubleLenght;//Переменные для хранения длинны массивов

Test(int r, int t); //Конструктор с параметрами

~Test();

void Show();

};

Test::Test(int r, int t)//пишем код конструктора с данной сигнатурой

{

IntMass = new int[r];//Выделяем память при момощи new

IntLenght = r;//Сохраняем размер массива int

for (int i=0;i<r;i++)//Заполняем наш массив INT

{

cout <<endl<< "Enter Int element[" << i << "] : ";

cin >> IntMass[i];

}

DoubleMass = (double\*)calloc(t, sizeof(double));//Выделяем память при помощи calloc

DoubleLenght = t;//Сохраняем размер массива double

for (int i = 0; i < t; i++)//Заполняем наш массив DOUBLE

{

cout << endl << "Enter Double element[" << i << "] : ";

cin >> DoubleMass[i];

}

}

Test::~Test()

{

//Освобождаем память

delete[] IntMass;

free(DoubleMass);

}

void Test::Show()

{

cout << endl << "IntMass:\n";

//Вывод массива INT

for (int i=0;i<IntLenght;i++)

{

cout << endl << IntMass[i];

}

cout << endl << "DoubleMass:\n";

//Вывод массива DOUBLE

for (int i = 0; i < DoubleLenght; i++)

{

cout << endl << DoubleMass[i];

}

}

int main()

{

Test mass(5,4);//передаем размеры массивов в качестве параметров

mass.Show();//Выводим массивы

}

Уровень 1:

Упражнение: Пользовательский класс должен содержать конструктор с параметром для создания динамических массивов (оператор new или стандартная библиотечная функция calloc) и установки начальных значений элементов: (Размер массива передается как аргумент для конструктора),так же класс должен иметь метод для просмотра текущего состояния массива и деструктор который освободит память.

Варианты:  
  
1)На вашу почту пришел заказ от компании «Cool developer».Вас просят разработать инвентарь персонажа где будут хранится предметы. Так как игра в стадии пре альфа бета релиз гамма тест разработке, то названия у предметов нет , есть только ID предметов. Ваша задача: Создать в классе Inventory массив размером Size(Size- кол-во ячеек в инвентаре) и заполнить его рандомными значениями от 0 до 100000.После , реализовать метод Run() , который выведет кол-во элементов с нечетным ID.

2)После того , как вы показали своему начальству разработанную вами программу для наблюдения за плесенью , он решил , что программе платить не надо , а работу она делает не хуже вас , поэтому вас уволили. Вы не отчаялись и решили попробовать себя в астро-физике. Первым вашем заданием стало запись расстояний от Солнца до звезд, попадающих в область видимости телескопа «Hubble». Пачкать бумагу вам снова не захотелось, а верный ноутбук всегда под рукой и вы приступили к работе. Ваша задача:Создать класс Stars ,а в нем массив размером Size(Size- кол-во звезд которое наблюдает «Hubble») и заполнить его рандомными значениями от 0 до 999999.После,реализовать метод Run(),который выведет разность между самой дальней и самой близкой звездой.(Разумеется в космосе для оценки межзвездных расстояний используются парсеки).

3)Завтра в вашу больницу приезжает проверка которая будет смотреть на заполнение палат вашей больницы. Чтобы не водить их в каждую палату , вы решили написать программу , в которая будет хранится информация о кол-ве человек в каждой палате. Ваше задание:Создать класс Patients , в котором будет массив размером Size (Size-кол-во палат в больнице) и заполнить его рандомными значениями от 0 до 10. После, реализовать метод Run(),который выведет номера палат в которых кол-во пациентов > 5(Так как эта современная больница , то номера палат начинаются с 0 , а dто программисты которым говорили идти в самую первую палату , бродили по больнице и не могли найти палату с номером 0).

4)Ваш начальник хочет знать , сколько каждый клиент вошедший в его банк потратил денег. Желание начальника – закон. Вы приступаете к работе. Ваша задача:Создать класс Customers , а в нем массив размером Size(Size-кол-во клиентов посетивших банк за день) и кол-во денег которое потратил каждый клиент значение которого генерируется рандомно в диапазоне от 0 до 100000.После,реализовать метод Run() который выведет общее кол-во денег полученное банком от всех клиентом посетивших его за день так-же выведите номера клиентов , которые потратили больше 90000 (Начальник лично хочет выдать им карточки Vip-клиентов, в благодарность за свою новую Tesla Model X).

5)Намечаются чудесные выходные и вы бы не хотели, чтобы их что-нибудь испортило, к примеру, бунт в камерах. Бунт в камере может начаться, если в ней сидит> 8 человек. Для того , чтобы узнать сколько сидит заключенных в камере вы решили написать программу.Ваша задача:Создать класс Criminals , а в нем массив размером Size(Size-Кол-во камер) и заполнить его рандомными значениями от 0 до 10.После , реализовать метод Run() , который выведет номера камер в которых кол-во заключенных > 8(Так как недавно был день программиста и все камеры забиты исключительно представителями этой профессии , то вам пришлось переделать нумерацию камер начиная с 0 , иначе они отказывались задерживаться в камерах на положенные 15 суток).

6)Посмотрев в казну своего города , вы обнаружили , что пауки при помощи паутины сплели на весь потолок слово «Пустота». Подумав немного, вы решили узнать, куда делись все деньги бюджета путем составления списка с зарплатами всех работников мэрии. Ваша задача: Создать класс Workers , а в нем массив размером Size(Size-Кол-во сотрудников мэрии) и зарплата каждого сотрудника которая генерируется рандомно в диапазоне от 100 до 999999.После,реализовать метод Run() который урежет зарплату всем у кого она > 1000 , в 2 раза и выведет эти новые зарплаты сотрудников.

7) 23-е февраля закончилось и теперь ваша фабрика носков потеряла все свои заказы. Но вы не унываете , ибо скоро 8-е марта , а ваша фабрика теперь начинает выпускать букеты с тюльпанами! Но все хотят разное кол-во цветов в букете и чтобы не запутаться , вам нужна программа. Ваша задача: Создать класс Bouquet , а в нем массив размером Size(Size-Кол-во букетов) и кол-во цветов в каждом букете которое генерируется рандомно в диапазоне от 3 до 1001.После,реализовать метод Run() который добавит 1 цветок в каждый букет в котором будет четное число цветов(вас ещё бабушка учила , что нельзя дарить четное число цветов) и вывести новые значения цветов в каждом букете.

8)Для привлечения всех категорий граждан в ваш отель вы придумали новую опцию «Своя комната». Клиент говорит , сколько он готов отдать за комнату и исходя от предложенной им суммы , ему становятся доступны некие услуги отеля и мебель для обстановки комнаты. Теперь вам надо посчитать прибыль от такого нововведения. Ваша задача: Создать класс Hostel ,а в нем массив размером Size(Size-Кол-во сданных комнат в отеле) и заполненный ценой каждой комнаты которая генерируется рандомно в диапазоне от 100 до 8000.После , реализовать метод Run() который выведет прибыль от всех комнат.

Уровень 2:

Внимание!!!

Перед выполнением уровня 2, давай те посмотрим, как можно создавать массив объектов. Это не простая тема, поэтому советую запустить приложенный ниже пример и разобраться в том , как он работает и как это все выглядит в памяти.

Спасибо за внимание!!!

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

class MassElement // Класс объекты которого будут являться элементами массива

{

public:

double a, b;//Поля для хранения значений типа double

};

class Test //Класс объект которого будет хранить в себе массив объектов класса MassElement , его размер и методы для работы с этим массивом

{

public:

int Size;//Поля для хранения длинны массива

MassElement \*mass;//Поле для хранения нашего будущего масива объектов

Test(int lenght); //Конструктор с параметрами

~Test();//Деструктор

void Show();//Метод для просмотра элементов массива

};

Test::Test(int lenght)//пишем код конструктора с данной сигнатурой

{

mass = (MassElement\*)calloc(lenght, sizeof(MassElement));//Выделение памяти для массива объектов

this->Size = lenght;//созраняем размер нашего массива

for (int i = 0; i < lenght; i++)//Заполняем поля каждого элемента массива , которые в свою очередб являются объектами класса Test

{

cout << endl << "Enter 1 element[" << i << "] : "; cin >> mass[i].a;//Заполняем поле a объекта под номером i

cout << endl << "Enter 2 element[" << i << "] : "; cin >> mass[i].b;//Заполняем поле b объекта под номером i

}

}

Test::~Test()

{

//Освобождаем память

free(mass);

}

void Test::Show()

{

for (int i = 0; i < this->Size; i++)//Выводим поля каждого объекта массива

{

cout << endl << "1 element[" << i << "] : =" << mass[i].a;

cout << endl << "2 element[" << i << "] : =" << mass[i].b;

}

}

int main()

{

Test mass(5);//Создаем наш объект , который хранит в себе массив объектов

mass.Show();//Просматриваем наш массив объектов

}

А зачем мы заводили второй класс MassElement? Нельзя ли все сделать в одном классе?

Ответ: Можно!

#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

public:

double a, b;//Поля для хранения значений типа double

int Size;//Поля для хранения размера массива

Test \*mass;//Поле для хранения нашего будущего масива объектов

Test(int lenght); //Конструктор с параметрами

~Test();//Деструктор

void Show();//Метод для демонстрациивсех элементов массива

};

Test::Test(int lenght)//пишем код конструктора с данной сигнатурой

{

mass = (Test\*)calloc(lenght, sizeof(Test));//Выделение памяти для массива объектов

this->Size = lenght;//созраняем размер нашего массива

for (int i = 0; i < lenght; i++)//Заполняем поля каждого элемента массива , которые в свою очередб являются объектами класса Test

{

cout << endl << "Enter 1 element[" << i << "] : "; cin >> mass[i].a;//Заполняем поле a объекта под номером i

cout << endl << "Enter 2 element[" << i << "] : "; cin >> mass[i].b;//Заполняем поле b объекта под номером i

}

}

Test::~Test()

{

//Освобождаем память

free(mass);

}

void Test::Show()

{

for (int i = 0; i < this->Size; i++)//Выводим поля каждого объекта массива

{

cout << endl << "1 element[" << i << "] : =" << mass[i].a;

cout << endl << "2 element[" << i << "] : =" << mass[i].b;

}

}

int main()

{

Test mass(3);//Создаем наш объект , который хранит в себе массив объектов

mass.Show();//Просматриваем наш массив объектов

}

И данная реализация тоже сработает. Единственное к чему тут можно придраться – это выделение лишней памяти. Во втором случаи мы выделяем память элементу необходимую для хранения объекта класса Test , а в первом случае мы выделяем память для хранения объекта класса MassElement. Наш элемент должен хранить информацию только о переменной a и b,поле для хранения Size и указателя на массив которые прописаны в Test ему не нужны – это лишняя кража памяти! Поэтому первый пример будет экономнее в плане памяти. Но каким способом будет решена задача – решать только вам.

Упражнение: Пользовательский класс должен содержать конструктор с параметром для создания динамических массивов (операция new или стандартная библиотечная функция calloc) и установки начальных значений элементов: (Размер массива передается как аргумент для конструктора),так же класс должен иметь метод для просмотра текущего состояния массива и деструктор который освободит память.

Выполнение задания уровня 2 должно происходить в классах созданных для выполнения уровень 1.Для демонстрации своего умения работы с массивом объектов, для хранения информации используйте массив объектов в каждом элементе которого будут хранится поля необходимые в соответствии с вашим заданием. Внимательно ознакомьтесь с примером приведённым выше !

Варианты:

1)Как только заказчик увидел , что все ячейки заполнены какими то не понятными ему числами , он дал команду придумать каждому предмету название и дать массу для каждого предмета. Теперь вам нужно все переделывать с учетом новых данных. Ваша задача : В созданном ранее классе создать массив объектов этого класса и добавить поля : ID, weight и name. weight – генерируется рандомно в диапазоне от 1 до 100.Метод Run() должен вывести всю информацию о каждом объекте лежащем в инвентаре у которого 30<weight<90. Поля ID и name вводятся вручную. Размер массива передается в конструктор в качестве параметра(как и для задания уровень 1) и заполнение массива и информации о каждом объекте происходит в том же конструкторе.

2)Одного расстояния до других звезд не достаточно , надо добавить ещё больше данных чтобы начальнику было лень их заполнять и вас не уволили. Ваша задача: В созданном ранее классе создать массив объектов этого класса и добавить поля : distance, name и size. distance и size– генерируется рандомно в диапазоне от 1 до 100000.Метод Run() должен вывести всю информацию о каждом объекте лежащем в диапазоне 400< distance <90000. Поле name вводятся вручную. Размер массива передается в конструктор в качестве параметра(как и для задания уровень 1) и заполнение массива и информации о каждом объекте происходит в том же конструкторе.

3)Проверке понравилась ваша идея с программой, в которой все хранится, но они предпочли бы получить информацию о каждом пациенте, а не о палате. Давайте сделаем, то, что просят проверяющие (Нашей больнице нужен новый аппарат МРТ). Ваша задача: В созданном ранее классе создать массив объектов этого класса и добавить поля: HowLongWillBeInHospital, Name и WardNumber. HowLongWillBeInHospital – генерируется рандомно в диапазоне от 1 до 90.Метод Run() должен вывести всю информацию о каждом объекте у которого 20< HowLongWillBeInHospital <70. Поля Name и WardNumber вводятся вручную. Размер массива передается в конструктор в качестве параметра(как и для задания уровень 1) и заполнение массива и информации о каждом объекте происходит в том же конструкторе.

4)Вашему начальнику зачем-то надо знать ещё возраст и пол клиентов банка. Ну раз надо , значит будет. Приступаем! Ваше задание: В созданном ранее классе создать массив объектов этого класса и добавить поля: spending, Age и gender(это поле для удобства можно сделать типа bool , мужчины == 0 , девушки == 1). Age – генерируется рандомно в диапазоне от 16 до 90. spending – генерируется рандомно в диапазоне от 0 до 100000.Метод Run() должен вывести всю информацию о каждом объекте у которого 18< Age <30 и 25000<spending<100000 и gender==1. Полe gender вводится вручную. Размер массива передается в конструктор в качестве параметра(как и для задания уровень 1) и заполнение массива и информации о каждом объекте происходит в том же конструкторе.

5)Вы уже собирались домой , но вдруг услышали возмущенные крики программистов «ООП – рулит! Мы все, объекты! HTML – не язык программирования! ». Ну что ж, с последним вы согласны, а что делать с их заявлением «Мы все объекты»? «Придётся сделать их всех объектами», подумали вы и приступили к работе.Ваша задача: В созданном ранее классе создать массив объектов этого класса и добавить поля: experience, Name и Level\_In\_The\_Skyrim. experience – генерируется рандомно в диапазоне от 0 до 100. Level\_In\_The\_Skyrim - генерируется рандомно в диапазоне от 0 до 255.Метод Run() должен вывести всю информацию о каждом объекте у которого Level\_In\_The\_Skyrim>50 и experience>30 (Их начальник может и отпустит) .Поле Name вводится вручную. Размер массива передается в конструктор в качестве параметра(как и для задания уровень 1) и заполнение массива и информации о каждом объекте происходит в том же конструкторе.

6)Вы вроде бы урезали зарплату сотрудникам, а денег как не было так и нет. Подумав немного вы все же поняли «сотрудники та, подворовывают». Теперь надо узнать кто и сколько. Ваша задача: В созданном ранее классе создать массив объектов этого класса и добавить поля: stolen, Name и gender(это поле для удобства можно сделать типа bool , мужчины == 0 , девушки == 1). stolen – генерируется рандомно в диапазоне от 0 до 100000.Метод Run() должен вывести всю информацию о каждом объекте у которого stolen >10000 и gender==0 или stolen >20000 и gender==1 (Их надо будет пожалуй уволить) .Поле Name и gender вводится вручную. Размер массива передается в конструктор в качестве параметра(как и для задания уровень 1) и заполнение массива и информации о каждом объекте происходит в том же конструкторе.

7)Вы уже почти начали производство букетов, но вдруг вспомнили, что не знаете кому какой букет отправлять. Вам надо срочно записать адрес доставки каждого букета и определить цену и кол-во цветов в букете, чтобы в будущем быстро узнать свою прибыль. Ваша задача: В созданном ранее классе создать массив объектов этого класса и добавить поля: Adres , price , Kol.Kol-генерируется случайно в диапазоне от 1 до 1001.Price и Adres – вводится вручную. Размер массива передается в конструктор в качестве параметра(как и для задания уровень 1) и заполнение массива и информации о каждом объекте происходит в том же конструкторе.

8)Ваша идея увенчалась успехом , отбоя от желающих просто нет , но вы столкнулись с одной проблемой. Вы обнаружили , что один человек заказывает для себя абсолютно пустую комнату и платит за неё копейки , а живут в этой комнате 12 человек! Хитро, но вы хитрее.Надо проверить все комнаты и выселить неплательщиков.Ваша задача: : В созданном ранее классе создать массив объектов этого класса и добавить поля: NumberOfResidents, Price и NumberOfBeds. Price – генерируется рандомно в диапазоне от 100 до 100000.Метод Run() должен вывести всю информацию о каждом объекте у которого NumberOfResidents > NumberOfBeds (Их надо будет пожалуй выселить) .Поле NumberOfResidents и NumberOfBeds вводится вручную. Размер массива передается в конструктор в качестве параметра(как и для задания уровень 1) и заполнение массива и информации о каждом объекте происходит в том же конструкторе.

Уровень 3

Король 3/9 царства 1/19 государства решил вырубить некоторые деревья, растущие перед его дворцом. Деревья перед дворцом короля посажены в ряд, всего там растет n деревьев, расстояния между соседними деревьями одинаковы.

После вырубки перед дворцом должно остаться m деревьев, и расстояния между соседними деревьями должны быть одинаковыми. Вас наняли заняться вырубкой и вам надо сказать королю сколько существует способов вырубки деревьев.Для того чтобы не ошибится вы решили написать себе программу в помощь которая по заданным числам n и m определит, сколько существует способов вырубки некоторых из n деревьев так, чтобы после вырубки осталось m деревьев и соседние деревья находились на равном расстоянии друг от друга.

Ваша задача : Создать класс Tree котором будет 2 поля n,m в которых будут хранится кол-во деревьев и кол-во деревьев которое надо вырубить.Заполнение полей происходит при помощи конструктора.Так же класс должен содержать метод который выведет кол-во существующих способов вырубки деревьев.  
Входные данные:

Два целых числа n и m (0 ≤ m , n ≤ 1000).

Выходные данные:

Одно число - искомое число способов.

|  |  |
| --- | --- |
| Input | Output |
| 5 3 | 4 |

Пояснение к примеру

Если обозначить условно исходное расположение деревьев перед дворцом как «TTTTT», то возможные результаты после вырубки следующие:

«TTT..», «.TTT.», «..TTT», «T.T.T».

**Контрольные вопросы:**

1)Формат и назначение библиотечной функции malloc().

2) Формат и назначение библиотечной функции calloc().

3) Формат и назначение библиотечной функции free().

4) Как создать и уничтожить динамический одномерный массив при помощи операторов?

5) Как создать массив объектов класса?

6)Сколько памяти(в байтах) займет данный массив:

Int \*mass = new int[5];  
7) Сколько памяти(в байтах) займет данный массив:

Int \*mass = new int(5);

8)Сколько памяти (в байтах) займет данный массив:  
class Test

{

public:

int c=0,r=0;

double q = 0;

};

Test \*Mass = new Test[5];

Лабораторная работа №4.

Работа с динамической̆ памятью. Двумерный динамический массив. Дружественные функции.

**Создание двухмерного динамического массива.**

Учитывая тот факт, что имя двухмерного массива - это указатель на указатель int \*\*m; двухмерный динамический массив создается и уничтожается за два шага. Например, требуется создать целочисленный массив m[3][4]:

Пример 1:

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

//Создание двумерного массива

int \*\*mass = new int\*[3];//Создание массива указателей

for (int i=0;i<3;i++)

{

mass[i] = new int[4];//Создание еще одного массива каждому указателю

}

//Заполнение массива

for (int i=0;i<3;i++)

{

for (int j=0;j<4;j++)

{

mass[i][j] = i+j;

//или можно использовать запись \*(\*(mass + i) + j) = i+j;

}

}

//Вывод массива

for (int i = 0; i < 3; i++)

{

cout << endl;

for (int j = 0; j < 4; j++)

{

cout << " " << mass[i][j];

}

}

/\*

Outputs:

0 1 2 3

1 2 3 4

2 3 4 5

\*/

//Удаление двумерного массива

for (int i=0;i<3;i++)

{

delete[] mass[i];//Освобождение памяти занятой элементами

}

delete[] mass;//Освобождение памяти занятой под массив указателей

}

Пример 2(использование функций для выделения памяти):

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

//Создание двумерного массива

int \*\*mass = (int\*\*)calloc(3, sizeof(int\*));//Создание массива указателей

for (int i=0;i<3;i++)

{

mass[i] = (int\*)calloc(4, sizeof(int));//Создание еще одного массива для каждого указателя

}

//Заполнение массива

for (int i=0;i<3;i++)

{

for (int j=0;j<4;j++)

{

mass[i][j] = i+j;

//или можно использовать запись \*(\*(mass + i) + j) = i+j;

}

}

//Вывод массива

for (int i = 0; i < 3; i++)

{

cout << endl;

for (int j = 0; j < 4; j++)

{

cout << " " << mass[i][j];

}

}

/\*

Outputs:

0 1 2 3

1 2 3 4

2 3 4 5

\*/

//Удаление двумерного массива

for (int i=0;i<3;i++)

{

free(mass[i]);//Освобождение памяти занятой элементами

}

free(mass);//Освобождение памяти занятой под массив указателей

}

**Краткая характеристика функции-друга класса**

Механизм «функции-друга» класса обеспечивает возможность доступа к любым элементам класса тем функциям, которые не входят в состав этого класса. То есть функция-друг класса - это обычная функция пользователя, которая благодаря данному механизму имеет полные права доступа ко всем без исключения элементам класса, независимо от их степени защищенности. Чтобы обычная функция стала другом класса, в заголовке ее необходимо указать friend.

**Некоторые особенности:**

1. Функцию- друга класса необходимо декларировать в описании этого класса.

В форме описания (приводится прототип), а ее полное определение – вне шаблона класса, при этом не нужно указывать операцию привязки к данному классу, т.к. она не является членом данного класса;

2. При обращении к функции- другу не надо использовать операцию привязки (.) или (→).

3. Так как функция-друг - обычная функция, у нее отсутствует первый скрытый параметр this. Для обращения элементам класса из функции-друга используют или ссылку, или указатель, что и иллюстрирует следующий пример:

Пример:  
#include "pch.h"

#include <iostream>

using namespace std;

class Test

{

private:

int b;//защищенное поле

public:

int a;//Публичное поле

void Show();//Метод класса для просмотра

void Set(int a,int b);//Метод класса для установки значений полей

friend void Show(Test &);//Дружественная функция для просмотра полей

friend void Set(Test &,int,int);//Дружественная функция для установки значений полей

};

void Test::Show()//Реализация метода класса для просмотра полей

{

cout<<endl << a<<" " << b;

}

void Test::Set(int a, int b)//Реализация метода класс для установки новых значений

{

//Так как метод является частью класса , то ему доступен спец. указатель this

this->a = a;

this->b = b;

}

void Show(Test &obj)//Реализация дружественной функции для просмотра полей

{

//если бы эта функция не была бы указана в классе Test (Строка 15) как дружественная функция , то поле b вывести бы не удалось из-за нехватки доступа

cout<<endl << obj.a<<" " << obj.b;

}

void Set(Test &obj, int a, int b)//Реализация дружественной функции для заполнения полей

{

//если бы эта функция не была бы указана в классе Test (Строка 16) как дружественная функция , то поле b изменить бы не удалось из-за нехватки доступа

obj.a = a;

obj.b = b;

}

int main()

{

Test a;

a.Set(12, 23);//Задаем значения при помощи функции класса

a.Show();//Выводим значения полей при помощи метода класса

/\*

Output:

12 23

\*/

Set(a,44,55);//Задаем значения полей при помощи дружественной функции

Show(a);//Выводим значения полей при помощи дружественной функции

/\*

Output:

44 55

\*/

}

Отметим, что функции-друзья класса кроме ссылки или указателя на объекты класса могут иметь обычные параметры, например для инициализации данных объектов класса. Яркой демонстрацией такой функции является дружественная-функция Set(Test &obj, int a, int b),которая принимала в качестве аргументов ещё 2 переменные типа int , для установки новых значений полям.

И так, мы видим что результат выполнения методов класса и дружественных функций у нас одинаковый и отсюда возникает вполне логичный вопрос «А зачем нам вообще нужны эти дружественные функции»? В таком простом примере с всего лишь одним классом действительно эти функции нам не слишком нужны, а вот представьте проект где таких классов десятки! И вот чтобы не писать каждому классу свой метод для выполнения операции вывода полей к примеру, проще у всех у них указать одну общую дружественную функцию, а в ней уже вывести все поля переданных в неё объектов. Это существенно сократит код и в нем будет легче ориентироваться.

Уровень 1:

Упражнение: Пользовательский класс должен содержать конструктор с параметрами для создания динамических целочисленных массивов (операция new или стандартная библиотечная функция calloc): Размеры массива – число строк и столбцов передается в конструктор через параметры, заполнение массива происходит в конструкторе; Класс так же должен содержать Деструктор для освобождения занятой памяти;

Реализовать дружественные функции для:

1)Просмотра текущего состояния массива;

2)Установка новых значений элементов массива;

3)Выполнение поставленной задачи в соответствии с вариантом.

Варианты:

1)Вы очень любите игру «Heroes of Might and Magic» и поэтому решили написать свою версию этой замечательной игры. Разумеется написать игру полностью вы не успеете в связи с кучей других заказов, но прописать поле битвы за выходные вполне успеете! Поле битвы будет выглядеть как квадратная матрица размером NxN (N>1). В каждую ячейку вы вводите кол-во солдат стоящих на этой ячейке. Солдаты расположенные в ячейках выше главной диагонали – ваши , солдаты расположенные в ячейках ниже главной диагонали принадлежат противнику , солдаты расположенные на главной диагонали являются ничейными и вообще не учитываются при сражении. Ваша задача: Создать класс Arena в котором будет двумерный массив NxN и заполнить его случайными значениями в диапазоне от 0 до 1000 , после написать дружественную функцию Run которая выведет кол-во ваших солдат и кол-во солдат противника.

2)Вас попросили помыть линзы телескопа и вы решили выполнить задание добросовестно и помыть их изнутри тоже. Когда вы уже все сделали и были довольны работой , вы решили все таки загуглить «Как правильно мыть линзы телескопа». Прочитав о том как нужно правильно мыть линзы и осознав свой косяк , вы решили перевестись на другую кафедру. Так как вам в исследовательском центре ближайшее время лучше не показываться , то вы решили перевестись на кафедру растениеводства и записаться в добровольцы на недельную работу агрономом в поле. Приехав на место , вы увидели поле разделенное на участке очень похожее на матрицу размером NxM.На каждом участке растет разное кол-во растений и ваша задача посчитать кол-во всех растений. Задача ясна , давайте приступим! Ваша задача: Создать класс Garden в котором будет двумерный массив NxM и заполнить его случайными значениями в диапазоне от 0 до 100 , после написать дружественную функцию Run которая выведет кол-во всех растений на поле.

3)Работа глав-врача очень сложна и для того чтобы отдохнуть вы иногда выглядываете в окно чтобы полюбоваться на клумбу с цветами под окном. Вы уже давно заметили что клумба разделена на участки и очень похожа на матрицу размером NxM . Сегодня у вас незагруженный день и вы решили посчитать кол-во цветов по краю клумбы. Ваша задача: Создать класс LowerBed в котором будет двумерный массив размером NxM и заполнить его случайными значениями от 0 до 10 , после написать дружественную функцию Run() которая посчитает кол-во цветов по краям клумбы.

4)В вашем банке есть хранилище и ваш начальник спать не может не зная сколько в среднем хранится денег в ячейках и сколько ячеек в которых кол-во денег больше среднего значения. Вам надо ему помочь , ибо у сонного начальника сотрудники не получают премий! Хранилище давно напоминала вам матрицу размера MxN , а с матрицей вы работать умете. Ваша задача: Создать класс Storage в котором будет двумерный массив размером NxM и заполнить его случайными значениями от 0 до 100000 , после написать дружественную функцию Run() которая посчитает среднее кол-во денег во всех ячейках и кол-во ячеек в которых хранится больше среднего значения.

5)Вам нужно составить график патрулей по вашему городу , но для этого надо знать уровень преступности в каждом районе , чтобы знать сколько и куда надо отправить патрулей(Нет смысла посылать много патрулей в район , где живут пенсионеры и программисты , они все равно дома сидят и на улицу не выходят). Когда вы общались с заключенными программистами , они научили вас такой крутой штуке как матрица и вы решили разделить ваш город на равные участки чтобы получилась матрица размером NxM и у каждого участка будет свой уровень преступности. Ваша задача : Создать класс City в котором будет двумерный массив размером NxM и заполнить его случайными значениями от 0 до 100 , после написать дружественную функцию Run() которая выведет средний уровень преступности в городе и кол-во районов в которых уровень преступности превосходит среднее значение.

6)Вы решили запустить социальную программу благодаря которой каждая многодетная семья у которой больше 4 детей получала бы бесплатно квартиру в вашем городе. Но перед этим вам не помешало бы узнать сколько вообще жилых домов в вашем городе. На курсе «Высшая математика за 2 дня легко и с картинками» вам рассказывали про матрицу и вы решили разделить ваш город на одинаковое участки в результате чего у вас получилась матрица NxM. Ваша задача: Создать класс City в котором будет двумерный массив размером NxM и заполнить его случайными значениями от 0 до 1000 , после написать дружественную функцию Run() которая выведет среднее кол-во жилых помещений в городе и кол-во участков в которых кол-во жилых домов меньше среднего.

7) 8 марта приближалось, тюльпаны уже выращены и расфасованы по букетам , но тут вы поняли , что цветы без коробки конфет – это минус к вашей прибыли от этого прекрасного праздника! Вы опытный торговец и заказали срочную доставку пары тонн конфет и прямоугольных коробок для конфет на Aliexpress. Вот только когда они пришли , вы поняли что надо внимательней читать описание товара. Коробки которые к вам пришли имели слишком большое отделение для хранения конфеты и в каждую ячейку могло поместится несколько конфет. Подумав немного , вы решили сделать это фишкой ваших короб с конфетами типа : Никогда не знаешь сколько конфет внутри коробки . Своим рабочим вы сказали кидать в каждую ячейку разное кол-во конфет. Но проверка первой партии показала, что рабочие решили немого схитрить и иногда не ложили конфету в ячейку. Это никуда не годится и вам срочно требуется программа для проверки наполненности коробки конфетами. Ваша задача: Создать класс BoxOfCandies в котором будет двумерный массив размером NxM и заполнить его случайными значениями от 0 до 5 , после написать дружественную функцию Run() которая выведет номера ячеек в которых нет конфет.

8)В своем отеле вы решили открыть ресторан с названием «Матрица». Название пришло вам в голову из за того , что столики расположены на одинаковом расстояние друг от друга и схема расположения столов напоминает вам старую добрую матрицу размером NxM. У вас современный ресторан и у вас можно бронировать столик заранее , каждый стол вмещает в себя 5 человек. К вам очень скоро приедет большая группа туристов и вам надо срочно узнать , сколько столиков уже занято. Ваша задача : Создать класс Restaurant в котором будет двумерный массив размером NxM и заполнить его случайными значениями от -5 до 5 ( «-» будет означать , что столик забронирован но люди которые его бронировали ещё не пришли , а значит что за этот столик вы вполне можете посадить туристов, так же туристов можно посадить за столы на которых некто не сидит , то есть значение == 0), после написать дружественную функцию Run() которая выведет кол-во столиков за которые можно посадить туристов.

Уровень 2:

Внимание!!!

Перед выполнением уровня 2, давай те посмотрим, как можно создавать двумерный массив объектов. Тем кто ещё не разобрался с тем , как создавать одномерный массив объектов , просьба разобраться с этим перед выполнением данного задания.

Спасибо за внимание!!!

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

class MassElement // Класс объекты котороого будут элементами нашего двумерного массива

{

public:

double a, b;//Поля для хранения значений типа double

};

class Test

{

public:

int N, M;//Переменные для хранения размера массива

MassElement \*\*mass;//Поле для хранения нашего будущего масива объектов

Test(int N, int M); //Конструктор с параметрами

~Test();//Деструктор

void Show();//Метод для демонстрациивсех элементов массива

};

Test::Test(int N, int M)//пишем код для конструктора с данной сигнатурой

{

mass = (MassElement\*\*)calloc(N, sizeof(MassElement\*));//создаем массив каждый элемент которого будет ещё одним массивом

for (int i = 0; i < N; i++)

{

mass[i] = (MassElement\*)calloc(M, sizeof(MassElement));//Создание еще одного массива для каждого указателя

}

//сохраняем размер нашего массива

this->N = N;

this->M = M;

for (int i = 0; i < N; i++)//Заполняем поля каждого элемента массива , которые в свою очередь являются объектами класса MassElement

{

for (int j = 0; j < M; j++)

{

cout << endl << "Enter 1 element[" << i << "][" << j << "] : "; cin >> mass[i][j].a;//Заполняем поле a объекта под номером i

cout << endl << "Enter 2 element[" << i << "][" << j << "] : "; cin >> mass[i][j].b;//Заполняем поле b объекта под номером i

}

}

}

Test::~Test()

{

//Освобождаем память

for (int i = 0; i < N; i++)

{

free(mass[i]);//Освобождение памяти занятой элементами

}

free(mass);//Освобождение памяти занятой под массив указателей

}

void Test::Show()

{

for (int i = 0; i < N; i++)//Выводим поля каждого объекта массива

{

for (int j = 0; j < M; j++)

{

cout << endl << "1 element[" << i << "][" << j << "] : " << mass[i][j].a;

cout << endl << "2 element[" << i << "][" << j << "] : " << mass[i][j].b;

}

}

}

int main()

{

Test mass(3, 2);//Создаем наш объект , который хранит в себе массив объектов

mass.Show();//Просматриваем наш массив объектов

}

Упражнение: Пользовательский класс должен содержать конструктор с параметрами для создания динамических целочисленных массивов (операция new или стандартная библиотечная функция calloc): Размеры массива – число строк и столбцов передается в конструктор через параметры, заполнение массива происходит в конструкторе; Класс так же должен содержать Деструктор для освобождения занятой памяти;

Реализовать дружественные функции для:

1)Просмотра текущего состояния массива;

2)Установка новых значений элементов массива;

3)Выполнение поставленной задачи в соответствии с вариантом.

Выполнение задания уровня 2 должно происходить в классах созданных для выполнения уровень 1 .Для демонстрации своего умения работы с двумерным массивом объектов, для хранения информации создайте двумерный массив объектов в каждом элементе которого будут хранится поля необходимые в соответствии с вашим заданием. Внимательно ознакомьтесь с примером приведённым выше!

Варианты:

1)Поиграв пару часов в «Heroes of Might and Magic» вы вспомнили , что войска могут отличаться по уровню поэтому надо это тоже учесть. Ваша задача : В созданный ранее классe создать двумерный массив объектов класса в котором будут поля: Level1, Level2, Level3, Level4, Level5(тип int).Все поля заполняются ранодмно в диапазоне от 0 до 100. Дружественная функция Run() должна вывести кол-во ваших и вражеских войск : Level1, Level2, Level3, Level4, Level5.

2)Участки получились довольно большими и на каждом из них были несколько видов овощей. Давайте узнаем сколько овощей каждого вида растет на поле. Ваша задача : В созданный ранее классe создать двумерный массив объектов класса в котором будут поля: Potatoes, Carrots, Cabbage, Horseradish, Onions (тип int).Все поля заполняются ранодмно в диапазоне от 0 до 100. Дружественная функция Run() должна вывести кол-во участков на котором кол-во Potatoes> Carrots более в чем 2 раза и Horseradish>Onions.

3)После пристального просмотра вы поняли , что на клумбе растут все таки разные виды цветов и вы решили посчитать кол-во цветов каждого вида. Ваша задача : В созданном ранее классе создать двумерный массив объектов класса в котором будут поля: Tulips, Roses, Chrysanthemums, Snowdrops, Cornflowers(тип int). Все поля заполняются ранодмно в диапазоне от 0 до 20. Дружественная функция Run() должна вывести кол-во участков на которых Tulips, Roses или Snowdrops == 0.

4)Оказалось , что в ячейках лежат не только деньги , некоторые клиенты хранят в них ценности и ценные бумаги тоже. Вашего начальника не интересуют ценные бумаги ваших клиентов , его интересует их цена. Ваша задача : В созданном ранее классе создать двумерный массив объектов класса в котором будут поля: Money, Securities, Decorations, Stocks(тип int). Все поля заполняются рандомно в диапазоне от 0 до 100000. Дружественная функция Run() должна вывести среднее кол-во денег в ячейках(Для определения цены одной ячейки == Money+ Securities+Decorations+ Stocks) и кол-во ячеек у которых цена > среднего значения.

5)Из отдела аналитики недавно пришли результаты по разведке каждого участка вашего города и теперь вы знаете сколько и каких преступников на каждом участке.Давайте внесем эти данные.Ваша задача: В созданном ранее классе создать двумерный массив объектов этого класса и добавить поля: Thief, Recidivist, Mafia, Gangsters, Fans(тип int).Все поля заполняются рандомно Thief, Recidivist в диапазоне от 0 до 50 , Mafia от 0 до 5 , Gangsters от 0 до 20, Fans от 5 до 100. Дружественная функция Run() должна рейтинг опасности каждого участка. Рейтинг вычисляется по формуле:

(Thief+ Recidivist+ Mafia\*3+ Gangsters\*2+ Fans/4) / (Thief+ Recidivist+ Mafia+ Gangsters+ Fans)

6)Одной лишь информации о кол-ве жилых домов вам не достаточно. Вам надо так же знать кол-во свободных и занятых квартир в этих домах. Ваша задача : В созданном ранее классе создать двумерный массив объектов класса в котором будут поля: House, ApartmentBuilding, AvailableApartments, OccupiedApartments, TotalNumberOfApartments(тип int). Все поля заполняются рандомно House в диапазоне от 0 до 100 , ApartmentBuilding от 5 до 200, TotalNumberOfApartments = сгенерированное (ApartmentBuilding \* 60). OccupiedApartments в диапазоне от 0 до TotalNumberOfApartments. AvailableApartments = (TotalNumberOfApartments – OccupiedApartments). Дружественная функция Run() должна вывести общее число House, ApartmentBuilding, AvailableApartments, OccupiedApartments, TotalNumberOfApartments на всех участках.

7)Тех конфет которые вы заказали на Aliexpress все таки не хватило , а таких же вы найти не смогли поэтому вы заказали ещё 4 вида конфет и теперь вам надо знать сколько конфет какого вида положили в каждую ячейку. Ваша задача : В созданном ранее классе создать двумерный массив объектов класса в котором будут поля: KitKat, Mars, Galaxy, Cadbury, Toblerone (тип int). Все поля заполняются рандомно в диапазоне от 0 до 2. Дружественная функция Run() должна вывести все поля KitKat, Mars, Galaxy, Cadbury, Toblerone для каждой ячейки и кол-во ячеек у которых (KitKat=1 и Mars = 1) или (Galaxy = 0 и Cadbury=1).

8)Вы успешно приняли туристов , но пару часов спустя они начали кочевать между столиками и теперь за столами сидят жильцы отеля вперемешку с туристами. Ну , жалоб на поведение туристов ещё не было , а сейчас надо бы произвести перепись посетителей чтобы знать где и сколько кого сидит и найти столик который сделал самый дорогой заказ чтобы объявить этот столик победителем в конкурсе «А я потратил больше всех!». Ваша задача : В созданном ранее классе создать двумерный массив объектов класса в котором будут поля: Tourists, Guests, MoneySpent. Поля Tourists, Guests заполняются рандомно в диапазоне от 0 до 5 (Tourists+ Guests должно быть <=5), поле MoneySpent заполняется рандомно в диапазоне от 100 до 100000. . Дружественная функция Run() должна вывести общее кол-во туристов , общее кол-во постояльцев в ресторане и номер столика с самым большим MoneySpent.

Уровень 3   
Известный художник решил написать новый шедевр. После многих дней усердной работы он захотел исследовать свое творение. Художник вспомнил, что картина писалась следующим образом: сначала был взят белый холст, имеющий форму прямоугольника шириной w и высотой h. Затем художник нарисовал на этом холсте n прямоугольников со сторонами, параллельными сторонам холста и вершинами, расположенными в целочисленных координатах. Помогите художнику определить площадь незакрашенной части холста.

Ваша задача : Создать класс Picture в котором будет 3 поля w,h для хранения размера холста и поле n для хранения кол-ва прямоугольников.Заполнение полей происходит при помощи конструктора.Так же класс должен содержать метод который выведет площадь незакрашенной области.Заполнение n строк описывающих координаты прямоугольника можете сделать в методе для вывода площади незакрашенной поверхности или в конструкторе(На ваше усмотрение).  
Входные данные:

Два натуральных числа w и h (1 ≤ w, h ≤ 100). Во второй строке записано целое число n (0 ≤ n ≤ 5000) – количество прямоугольников. Следующие n строк содержат информацию о всех прямоугольниках. Каждая строка описывает один прямоугольник в виде четырех чисел x1, y1, x2, y2 , где (x1, y1) и (x2, y2) – координаты левого верхнего и правого нижнего угла прямоугольника соответственно.

Выходные данные:

Одно целое число – площадь незакрашенной части холста.

|  |  |
| --- | --- |
| Input | Output |
| 5 5  2  1 1 3 3  2 2 4 4 | 18 |
| 6 7  3  0 0 5 5  1 1 4 4  2 2 3 3 | 17 |

**Контрольные вопросы:**

1)Чем является имя двумерного массива?

2) Как косвенно обратиться к элементу двумерного массива?

3) Объясните понятие “функция-друг” класса?

4) Какие существуют способы обращения к элементам класса из функции-друга?

5) Как освободить память, занимаемую двумерным динамическим массивом?

6) «Функция-друг» имеет доступ ко всем полям класса ?

7)Сколько памяти займет данный двумерный массив?  
 int \*\*mass = new int\*[5];

for(int i=0;i<5;i++)

{

mass[i] = new int[5];

}

8)Дан двумерный массив:

1 2 3 4 5

3 4 5 6 7

9 0 3 2 1

Напишите алгоритм для нахождения суммы элементов главной диагонали.

Лабораторная работа №5

Работа с динамической строкой и перегрузка операций.

Напомним, что работа со строками в языке Си реализована путем использования одномерных массивов типа char, т.е. строка символов - это одномерный массив типа char, заканчивающийся нулевым байтом. Нулевой байт - это байт, каждый бит которого равен нулю, при этом для нулевого байта определена символьная константа ‘\0’ (признак окончания строки или нуль-терминатор). Поэтому, если строка должна состоять из k символов, то в описании массива необходимо указать размер k+1, а при ручном формировании строки в ее окончание нужно явно добавить признак ее окончания.

Операции над строками выполняются только через стандартные функции. Декларации функций для работы со строками размещены в файле string.h (Если вы используете Visual Studios , то string.h подключается автоматически при подключении "pch.h"). Перечислим некоторые из них, наиболее часто используемые.

1. Функция strcpy(S1, S2) - копирует содержимое строки S2 в строку S1.
2. Функция strcat(S1, S2) - присоединяет строку S2 к строке S1 и помещает ее в массив, где находилась строка S1, при этом строка S2 не изменяется. Нулевой байт, который завершал строку S1, заменяется первым символом строки S2.

Внимание!!!

Функции описанные выше на данный момент устарели и не используются , так как они были признаны не безопасными из-за не умения видеть границы из-за чего происходило переполнение буфера. В связи с этим были добавлены модифицированные аналоги данных функций [strcpy\_s(),strcat\_s() ]. Данные функции могут сами определить необходимый размер буфера , а так же (в случае необходимости) , пользователь сам может передать в качестве аргумента размер буфера.

Рассмотрим примеры с применением данных функций.

1)strcpy\_s():

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "rus");//Подключаем поддержку русских символов

char S1[] = "ООП - изи!";//Создаем наш масив char.Так как размер в скобках мы не указали,массив получится размером равным размера текста "ООП - изи!" + 1 для нуль-терминатора '\0'

char S2[40];//Создаем пустой массив на 40 симвалов

char S3[80];//Создаем пустой массив на 80 симвалов

strcpy\_s(S2,"Я - программист!");//Помещаем в массив S2 строку "Я - программист!".strcpy\_s() - автоматически добавит нуль-терминатора '\0'.При попытке поместить в S2 строку большего размера чем она может вместить, будет выдана ошибка!

strcpy\_s(S3,S2);//Помещаем в массив S3 строку хранящуюся в S2.

cout<<"S1= " << S1 << endl;//Output: "ООП - изи!"

cout << "S2= " << S2 << endl;//Output: "Я - программист!"

cout << "S3= " << S3 << endl;//Output: "Я - программист!"

}

2) strcat\_s():

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "rus");//Подключаем поддержку русских символов

char S1[] = "ООП - изи!";//Создаем наш масив char.Так как размер в скобках мы не указали,массив получится размером равным размера текста "ООП - изи!" + 1 для нуль-терминатора '\0'

char S2[40]="";//Создаем массив на 40 симвалов в котором в данном случае будет хранится только нуль-терминатор '\0' .

char S3[80];//Создаем массив на 80 симвалов

strcat\_s(S2,S1);//Объеденяем Содержимое строки S2 с содержимым строки S1.Результат будет хранится в S2

strcat\_s(S2," Я - программист!");//Объеденяем Содержимое строки S2 со строкой " Я - программист!".Результат будет хранится в S2

//strcat\_s(S3,S2);//Данная запись выдаст ошибку , так как массив S3 не содержит ничего , даже нуль-терминатора '\0'.Из-за этого функция не может определится с позицией в S3 с котрой надо начать запись содержимого S2

cout<<"S1= " << S1 << endl;//Output: "ООП - изи!"

cout << "S2= " << S2 << endl;//Output: "ООП - изи! Я - программист!"

}

Спасибо за внимание!!!

3) Функция strcmp(S1, S2) сравнивает строку S1 со строкой S2 и возвращает значение 0, если строки равны, т.е. содержит одно и то же число одинаковых символов, значение 1, если S1>S2 и значение -1 если S1<S2.

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "rus");//Подключаем поддержку русских символов

char S1[] = "ООП - изи!";//Создаем наш масив char.Так как размер в скобках мы не указали,массив получится размером равным размера текста "ООП - изи!" + 1 для нуль-терминатора '\0'

char S2[40]="ООП - изи!";//Создаем массив на 40 симвалов в котором в данном случае будет хранится строка "ООП - изи!" .

char S3[80] = "Я - программист";//Создаем массив на 80 симвалов в котором в данном случае будет хранится строка "Я - программист".

cout<<"S1==S2 выдаст : " << strcmp(S1,S2) << endl;//Output: 0

cout << "S3>S2 выдаст: " << strcmp(S3, S2) << endl;//Output: 1

cout << "S2<S3 выдаст: " << strcmp(S2, S3) << endl;//Output: -1

}

4)Функция strlen(S) возвращает длину строки, т.е. количество символов, начиная с нулевого и до нуль-терминатора, нулевой байт не учитывается.

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "rus");//Подключаем поддержку русских символов

char S1[] = "ООП - изи!";//Создаем наш масив char.Так как размер в скобках мы не указали,массив получится размером равным размера текста "ООП - изи!" + 1 для нуль-терминатора '\0'

char S2[40]="ООП - изи!";//Создаем массив на 40 симвалов в котором в данном случае будет хранится строка "ООП - изи!" .

char S3[80] = "Я - программист";//Создаем массив на 80 симвалов в котором в данном случае будет хранится строка "Я - программист".

cout<<"Lenght S1 = " << strlen(S1) << endl;//Output: 10

cout << "Lenght S2 = " << strlen(S2) << endl;//Output: 10

cout << "Lenght S3 = " << strlen(S3) << endl;//Output: 15

}

5) Функции преобразования строки S в число:

целое: int atoi(S);

длинное целое: long atol(S);

действительное: double atof(S);

при ошибке возвращает значение 0.

Пример:

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "rus");//Подключаем поддержку русских символов

char S1[] = "231";//Создаем наш масив char.Так как размер в скобках мы не указали,массив получится размером равным размера текста "231" + 1 для нуль-терминатора '\0'

char S2[40]="42.23";//Создаем массив на 40 симвалов в котором в данном случае будет хранится строка "42.23" .

char S3[80] = "45";//Создаем массив на 80 симвалов в котором в данном случае будет хранится строка "45".

int a = atoi(S1);

int b = atoi(S3);

double c = atof(S2);

if (a>b) // Output: S3>S1

{

cout << "S3>S1"<<endl;

}

if (c > a)// Output: S2>S3

{

cout << "S3>S2"<<endl;

}

else

{

cout << "S2>S3"<<endl;

}

if (atof(S2)<atof(S1))// Output: S1>S2

{

cout << "S1>S2"<<endl;

}

}

6) Функции преобразования числа V в строку S:

целое: itoa(int V, char S, int kod);

длинное целое: ltoa(long V, char S, int kod);

kod – это система счисления в которой мы хотим получить результат;

2<=kod<=36, для отрицательных чисел kod=10.

Внимание!!!

Функция itoa в данный момент так же устарела и мало где поддерживается. На смену ей пришёл её более безопасный аналог \_itoa\_s.

Пример:  
  
#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "rus");//Подключаем поддержку русских символов

int value;//Наше число

char string[6] = "";//Массив char в котором будет хранится наше число

cout << "Введите число: ";

cin >> value;

\_itoa\_s(value, string, 10);//Переводим value в десятеричную систему и помещаем в string

cout << "Введённое число в decimal: " << string << endl;

\_itoa\_s(value, string, 16);//Переводим value в шестнадцатеричную систему и помещаем в string

cout << "Введённое число в hexadecimal: " << string << endl;

\_itoa\_s(value, string, 2);//Переводим value в двоичную систему и помещаем в string

cout << "Введённое число в binary: " << string << endl;

system("pause");//Пауза

return 0;

}

Спасибо за внимание!!!

Помимо традиционной декларации строки, например char stroka[40]; можно создавать динамические строки.

Пример создания с использованием библиотечной функции:

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

char \*stroka;

int Size = 10;

stroka = (char\*)calloc(Size, sizeof(char)); // Выделяем память для хранения 10 элементов указателю stroka.Не забываем , что в реальности последний элемент должен быть '\0', поэтому stroka в себе сможет хранить 9 символов,всегда учитывайте '\0' в конце,он всегда должен занимать последний элемент.

//Для заполнения stroka теперь надо в функцию strcpy\_s добавить ещё один аргумент Size(размер нашей переменной stroka).

strcpy\_s(stroka,Size,"123456789"); // выполнится

cout << stroka;//Output: "123456789"

//strcpy\_s(stroka, Size, "1234567890"); // Ошибка.Нехватка памяти для размещения строки "1234567890" в stroka (Из-за того , что в конце должен стоять '\0'и символу '0'не хватает места .Не забываем про '\0' - это распространенная ошибка)

}

Можно воспользоваться и операторами new и delete для захвата и освобождения динамической памяти. Пример класса с конструктором для создания динамических строковых объектов и деструктором для их уничтожения после использования в программе.

#include "pch.h"

#include <iostream>

using namespace std;

class Stroka

{

public:

char \* Text;//Создаем наш указатель на строку

Stroka(const char \*s )//Создаем конструктор который принимает строку которую нам надо сохранить

{

Text = new char[strlen(s) + 1];//Мы предполагаем , что полученная строка s не будет иметь '\0' в конце, поэтому мы выделяем память для хранения строки s и добавляем +1 для записи '\0' в конец

strcpy\_s(Text, strlen(s) + 1, s);//Помещаем значение s в Text

}

~Stroka()//Деструктор

{

delete Text;//Очищаем память

}

void Print()//Метод для вывода содержимого поля Text

{

cout << Text << endl;//Вывод

}

};

int main()

{

Stroka t1("12345"), t2("6789");//Создаем 2 объекта и помещаем в них строки

t1.Print();//Output:"12345"

t2.Print();//Output:"6789"

}

Char – это конечно хорошо.Char – это основа основ! И мы должны уметь с ним работать. Но а тем, кто уже понял и научился работать с Char я советую изучить ещё шаблонный строковый класс String.Этот класс был создан специально чтобы облегчить нашу жизнь, коллеги! Давайте же рассмотрим малую часть того, что умеет наш String и вы поймете как именно он может облегчить вам жизнь.

Функции описанные выше нашему String не нужны. Все эти операции можно сделать при помощи операторов (Операторы – что это и с чем это едят? Не переживай! Все это будет рассмотрено и разобрано в следующей части).

Пример:  
  
#include "pch.h"

#include <iostream>

#include <string>//Подключаем наш класс string и функции для работы с классом string (to\_string - к примеру и тд.)

using namespace std;

int main()

{

string text1, text2="String - the Best!",text3 = "I'm a programmer";

//Данная запись заменяет нам strcpy. Оператор "=" в данном случае делает тоже самое , что и функция strcpy().Помещает значение text2 в text1

text1 = text2;

cout << text1<<endl;//Output: "String - the Best!"

//Данная запись заменяет нам сразу strcpy и strcat. Оператор "+" в данном случае делает тоже самое , что и функция strcat.Объеденяет значения text2 и text3 ,а потом оператор "=" помещает это объедененное значение в text1

text1 = text2 + text3;

cout << text1<<endl;//Output: String - the Best!I'm a programmer

//В классе String есть функция length() которая с удовольствием вернет вам длинну строки которая хранится в нашем объекте.Как вы уже дагодались - это замена функции strlen();

cout << text1.length()<<endl; //Output: 34

//А как просто можно сравнивать объекты String! strcmp() нам больше не нужна!

//При верности условия возвращается 1 иначе 0

if (text1>text2)

{

cout << "text1>text2 : "<< (text1 > text2)<<endl;//Output:1

}

if (text1==text1)

{

cout << "text1==text1 :" << (text1 == text1)<<endl;//Output:1

}

if (text2 > text1)

{

}

else

{

cout << "text2 > text1 : " << (text2 > text1)<<endl;//Output:0

}

// А вот для перевода string в число, функция atoi() и ей подобные понадобятся!

int a;

text1 = "123";

a = atoi(text1.c\_str()); // Метод c\_str() - возвращает массив char из text1 с которым уже может работать наша функция atoi.

cout <<"Number = "<< a<<endl;//Output:123

//А для перевода числа в string существует функция to\_string();

text1 = to\_string(7832.23);

cout <<"Number = "<< text1<<endl;//Output:7832.23

text1 = to\_string(a);

cout << "Number = " << text1 << endl;//Output:123

}

Как вы видите класс string облегчает работу со строками. Так же объект класса string можно считать динамическим так как он сам выделяет памяти столько , сколько ему нужно для хранения информации которую мы хотим в него поместить. Давайте рассмотрим пример с использованием string в классе.

Внимание!!!

String – это класс в котором уже прописан деструктор. Поэтому если вы используете string в своем классе , то создавать деструктор для освобождении памяти не нужно.String все сделает сам!

Спасибо за внимание!!!

Пример:  
#include "pch.h"

#include <iostream>

#include <string>//Подключаем наш класс string и функции для работы с классом string (to\_string - к примеру и тд.)

using namespace std;

class Stroka

{

public:

string info; //Создаем поле в котором будет хранится строка

Stroka(string s)//Создаем конструктор который будет принимать строку

{

info = s;//Помещаем строку полученную конструктором в поле info

};

void Print()//Метод для вывода поля info

{

cout << info << endl;//Выводим содержимое info

}

};

int main()

{

Stroka text1("12345"),text2("I know OOP!");//Создаем объекты и передаем строки в конструктор

text1.Print();//Output : "12345"

text2.Print();//Output : "I know OOP!"

}

Как мы видим результат будет точно таким же как и при использовании класса который был описан выше, только кода поменьше.

**Перегрузка операторов в C++ (краткая характеристика)**

В языке C++ наряду с обычными конструкциями операций над операндами со стандартными типами (арифметические, операции отношений, логические операции и т.д.) существует механизм перегрузки этих операторов, позволяющий манипулировать объектами классов пользователя, используя обычный синтаксис языка, т.е. привычную (удобную) запись операций. Если компилятор обнаружил, что в программе есть перегрузка операций для объектов конкретного типа, введенных пользователем, то, найдя эти операции, компилятор анализирует их. И, если

- операнды имеют встроенный тип данных, то будет заложена стандартная операция;

- операнды имеют тип данных пользователя, т.е. являются объектами класса, для которого операция была перегружена, то будет заложена перегруженная операция, запрограммированная пользователем.

Операции для конкретного класса перегружаются или методом этого класса, или функцией-другом для этого класса следующего вида:

тип operator@(список параметров с указанием их типа)

{

код, определяющий смысл указанной операции,

}

@ - символ перегружаемой операции.

Так как операция @ перегружается для объекта конкретного класса, а сам класс - это тип этого объекта, то тип - это идентификатор этого класса.

Отличия перегрузки операций при помощи метода класса или функции-друга следующие:

а) @а - унарная операция может быть перегружена методом без параметров или функцией-другом с одним параметром, т.к. у метода имеется первый скрытый от нас параметр (указатель this), который автоматически устанавливается на единственный операнд унарной операции, а функция-друг такого указателя не имеет, потому при перегрузке ее параметров, ее операнд и передают;

б) al @ а2 - бинарная операция перегружается методом класса с одним параметром или функцией-другом с двумя параметрами, т.к. при ее перегрузке методом на первый операнд устанавливается скрытый указатель (this), а второй передаем через параметр; при перегрузке функцией-другом первый операнд передается функцией через первый параметр, второй - через второй параметр.

А зачем нам вообще эта перегрузка операторов? Давайте представим , что мы создали класс Drobi в котором хранится 2 поля : n-для хранения числителя и d-для хранения знаменателя. Мы создали 2 объекта этого класса и положили в него значения:  
Drobi a(1,3);

Drobi b(2,4);

И теперь надо сложить первую дробь со второй. Первое что приходит на ум – это   
Drobi c = a +b;

Но есть проблема! Drobi – это наш пользовательский класс и компилятор не знает что для того чтобы сложить 1/3 и 2/4 надо сначала привести их к общему знаменателю , а только потом сложить числители. Именно при перегрузке оператора мы пишем определенный алгоритм который должен запускаться при использовании данного оператора с пользовательским типом данных.

В нашем случае когда компилятор увидит , что мы хотим сложить две дроби a +b , то он обратится к оператору данного класса где будет написан алгоритм который будет приводить дроби к общему знаменателю , складывать числители и возвращать правильно сложенные числитель и знаменатель.

**Есть три разных способа** [**перегрузки операторов**](https://ravesli.com/urok-130-vvedenie-v-peregruzku-operatorov/):

  -через дружественные функции;

  -через обычные функции;

  -через методы класса.

**Рассмотрим пример перегрузки через дружественные функции:**

#include "pch.h"

#include <iostream>

using namespace std;

class Drobi

{

private:

int n, d;//n -хранит числитель. d - хранит знаменатель

public:

Drobi(int n, int d)//Конструктор для заполнения числителя и знаменателя

{

this->n = n;

this->d = d;

}

void Print()//Вывод числителя и знаменателя

{

cout << "numerator = " << n << endl << "denominator = " << d << endl;

}

friend Drobi operator+(const Drobi &d1, const Drobi &d2);//Объявляем дружественную функцию

};

Drobi operator+(const Drobi & d1, const Drobi & d2)//Описываем нашу дружественную функцию.В качестве параметров в функцию передаём адреса наших объектов.d1 - соответсвует объекту стоящему слева от "+" d2 - соответствует объекту стоящему справа от "+"

{

return Drobi( (d1.n\*d2.d + d1.d\*d2.n) , d1.d\*d2.d ); //Приводим дроби к общему знаменателю , складываем числители , складываем знаменатели , возвращаем объект класса Drobi

}

int main()

{

Drobi a(1, 3),b(2,4); // Создаем 2 объекта класса Drobi и заполняем его поля

Drobi c = a + b;//объект "c" теперь указыввает на область памяти которая выделилась при выполнении operator+

c.Print();//Выводим значения объекта "c"

}

С оператором «+» вроде все понятно. А что означает объект "c" теперь указыввает на область памяти которая выделилась при выполнении operator+ ?

На самом деле до того как мы опишем оператор «=» для нашего класса наш объект будет просто указывать на область памяти которая стоить справа от «=».

К примеру , если мы в примере выше напишем

Drobi obj(1, 3),obj1(2,4);

obj = obj1;

То значение полей объекта «obj» не поменяются! Объект «obj» будет просто указывать на ту же область памяти где хранится наш объект obj 1(У этих переменных станет одинаковый адрес).В результате чего мы потеряем адрес по которому хранилась информация об объекте «obj»!

**Рассмотрим пример перегрузки оператора через обычную функцию:**

Операторы можно перегружать используя и обычные функции. Единственное в чем они уступают дружественным функциям это тем , что они не имеют доступ к полям и методам с модификатором доступа **private!** Поэтому перегрузки оператора через обычные функции используют только если этому оператору не нужны значения полей с модификатором доступа **private** или в классе предусмотрены методы которые смогут нам вернуть значение наших полей с доступом **private**.

В нашем классе нет таких методов которые помогут нам вернуть значение полей n и d , поэтому мы их добавим.  
  
#include "pch.h"

#include <iostream>

using namespace std;

class Drobi

{

private:

int n, d;//n -хранит числитель. d - хранит знаменатель

public:

Drobi(int n, int d)//Конструктор для заполнения числителя и знаменателя

{

this->n = n;

this->d = d;

}

void Print()//Вывод числителя и знаменателя

{

cout << "numerator = " << n << endl << "denominator = " << d << endl;

}

int ReturnN() //Функция которая возвращает нам значение поля n

{

return n;

}

int ReturnD() //Функция которая возвращает нам значение поля d

{

return d;

}

};

Drobi operator+( Drobi & d1, Drobi & d2)//Описываем нашу функцию.В качестве параметров в функцию передаём адреса наших объектов.d1 - соответсвует объекту стоящему слева от "+" d2 - соответствует объекту стоящему справа от "+"

{

return Drobi((d1.ReturnN() \* d2.ReturnD() + d1.ReturnD()\*d2.ReturnN()), d1.ReturnD()\*d2.ReturnD()); //Приводим дроби к общему знаменателю , складываем числители , складываем знаменатели , возвращаем объект класса Drobi

}

int main()

{

Drobi a(1, 3), b(2, 4); // Создаем 2 объекта класса Drobi и заполняем его поля

Drobi c = a + b;//объект "c" теперь указыввает на область памяти которая выделилась при выполнении operator+

c.Print();//Выводим значения объекта "c"

}  
  
Результат у нас получится тем же самым , что и при использовании дружественной функции. Единственное отличие – это способ получения значения полей n и d у наших объектов.

**Рассмотрим пример перегрузки оператора через метод класса:**

Перегрузка операторов через методы класса очень похожа на [**перегрузку операторов через дружественные функции**](https://ravesli.com/urok-131-peregruzka-operatorov-cherez-druzhestvennye-funktsii/). Но, при перегрузке оператора через метод класса, доступ к объекту для которого мы вызвали оператор можно будет получить при помощи спец указателя this.

Пример:  
#include "pch.h"

#include <iostream>

using namespace std;

class Drobi

{

private:

int n, d;//n -хранит числитель. d - хранит знаменатель

public:

Drobi(int n, int d)//Конструктор для заполнения числителя и знаменателя

{

this->n = n;

this->d = d;

}

void Print()//Вывод числителя и знаменателя

{

cout << "numerator = " << n << endl << "denominator = " << d << endl;

}

Drobi operator+(Drobi &d2)//Создаем метод для перегрузки оператора "+" в нашем классе

{

return Drobi((this->n\*d2.d + this->d\*d2.n), this->d\*d2.d); // this - указывает на объект который стоит слева от оператора "+"

};

Drobi operator+(int d2)//Создаем метод для перегрузки оператора "+" в нашем классе

{

return Drobi( this->n + (d2\*this->d) , this->d); // возвращаем нашу дробь с новыми значениями

};

};

int main()

{

Drobi obj(1, 3), obj1(2, 4); // Создаем 2 объекта класса Drobi и заполняем его поля

Drobi c = obj + obj1;//В данном случае объект "obj" вызовет перегрузку метода operator+ , а значит к полям "obj" можно будет обратится при помощи this

c.Print();//Выводим значения объекта "c"

//Реализуем ещё метод для прибавления целого числа к объекту нашего класса Drobi

Drobi obj2 = obj + 5;

obj2.Print();//Выводим значения объекта "obj2"

}

Давайте рассмотрим пример решения задачи при помощи перегрузки операторов.  
  
Задача: Ввести строку символов S1, признак окончания ввода строки - нажатие клавиши Enter. Программа должна содержать перегруженную операцию «=», использование которой скопирует S1 в S2 убрав от туда первые 2 и 2 последних символа:  
  
Решение:  
1)Использование массива Char:

#include "pch.h"

#include <iostream>

using namespace std;

class Stroka

{

public:

char Text[256];//Создаем наш массив char

void Print()//Метод для вывода содержимого поля Text

{

cout << Text << endl;//Вывод

}

void operator=(Stroka &str)//Перегружаем оператор = . str - объект стоящий справа от "="

{

for (int i = 2; i < strlen(str.Text)-2; i++)//заполняем массив Char объекта стоящего слева от "=".

{

Text[i-2] = str.Text[i];

}

Text[strlen(str.Text)-4] = '\0';//Добавляем '\0' для обозначение конца строки

}

};

int main()

{

Stroka t1, t2;//Создаем 2 объекта

gets\_s(t1.Text);//Помещаем строку введенную пользователем в поле Text объекта t1

t1.Print();//Выодим строку

t2 = t1;//Записываем в поле Text объекта t2 строку из объекта t1 без двух первых и последних символов при помощи перегрузки оператора "="

t2.Print();//Выводим строку

}

2)Использование String

#include "pch.h"

#include <iostream>

#include <string>

using namespace std;

class Stroka

{

public:

string Text;//Создаем наш объект string

void Print()//Метод для вывода содержимого поля Text

{

cout << Text << endl;//Вывод

}

void operator=(Stroka &str)//Перегружаем оператор = . str - объект стоящий справа от "="

{

for (int i=2;i<str.Text.length()-2;i++)//Идем по строке в объекте str.Обход начинаем сразу с третьего символа не доходя до конца строки 2 символа

{

Text += str.Text[i];//Добавляем в нашу строку Text i-ый символ из строки объекта str

}

}

};

int main()

{

Stroka t1, t2;//Создаем 2 объекта

getline(cin, t1.Text);//Помещаем строку введенную пользователем в поле Text объекта t1

cin.clear();//Очищаем поток для ввода

t1.Print();//Выодим строку

t2 = t1;//Записываем в поле Text объекта t2 строку из объекта t1 без двух первых и последних символов при помощи перегрузки оператора "="

t2.Print();//Выводим строку

}

Уровень 1:

Упражнение: Создайте класс в котором будет поле для хранения строки(Поле может быть типа char или string на ваше усмотрение).Реализовать метод вывода содержимого этого поля хранящего строку и реализовать перегрузку оператора «=» (Любым из 3 способов рассмотренных выше) чтобы результат удовлетворял условию задания вашего варианта.

Варианты:

1)Вас попросили разобраться с читерами в популярной MMORPG игре «Sword, Magic, Shield, Bow, Arrow and something else».Сервер постоянно возвращает какие-то непонятные строки типа : ”qqfdsdvn3u9r8329fjf239 n392hfnvowgn we 3wrfiodg” и тд. (Никто не понимает что означают эти строки возвращаемые сервером игры , но если работает – то не трогай). Но одно разработчики знают точно, если в строке возвращаемой сервером встречаются символы в верхнем регистре к примеру: «fdghjHJHKLdsjf FG F ddsgsdgwfg3 dsggsege» , то это значит , что кто то использует читы в игре.Вы опытный разработчик и знаете как выполнять такие задачи! Если символы в верхнем регистре – это читеры , то сервер просто должен перестать показывать эти символы в верхнем регистре и мы ему в этом поможем! Полные энтузиазмом вы приступили к работе!  
Ваша задача:Создать класс ServerAnswer в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая скопирует строку введенную пользователем в поле предназначенное для хранения строки в классе ServerAnswer удалив от туда все символы в верхнем регистре.

2)Ваша помощь агрономам была неоценима, но сбор урожая закончился и вам снова нечем заняться. Подумав немного вы решили перевестись на кафедру математики. На этой кафедре работают умнейшие люди в голове которых одни числа из-за чего общаться они тоже начали на языке цифр , а точнее вместо символа они используют его код из ASCII Table. К примеру фраза «Hello» у них будет записана вот так «72 101 108 108 111». Вы пока не выучили ASCII Table и не можете так быстро переводить предложения поэтому вы решили написать программу которая поможет вам в этом. Ваша задача: Создать класс MathematicalLanguage в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая поместит строку введенную пользователем в поле предназначенное для хранения строки в классе MathematicalLanguage заменив все символы введенной строки на её код в ASCII Table (Не забываем что у пробела тоже есть свой код).

3)Вчера вы прочитали прекрасный журнал «IT-сфера. Если нам лень что-то делать, мы напишем программу которая сделает это за нас!» в котором была интересная статья о том , что можно отсканировать лист с рукописным вводом в сканере и он может перевести этот рукописный ввод в редактируемые форматы текста с которой сможет работать компьютер. К вам сразу пришла идея о том , что такая программа не помешала бы в аптеках вашей больницы! Пришел , приложил рецепт к сканеру и фармацевт уже несет нужное вам лекарство. Однако программе будет сложно понять где закончился диагноз и началось название лекарства. Поэтому вы приказали всем врачам вашей больницы писать названия лекарств в фигурных скобках «{ }». Текст в этих скобках и будет являться названием лекарства. Осталось мелочь , создать программу которая будет выводить все эти названия заключенный между фигурными скобками.

Ваша задача: Создать класс Recipe в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая поместит из строки введенную пользователем в поле предназначенное для хранения строки в классе Recipe строки заключенные между фигурными скобками «{ }» (Не забываем учитывать то , что фигурных скобок может быть несколько и строки между другими фигурными скобками тоже надо вывести).

4)Вашему начальнику постоянно приходит куча счетов. И одним прекрасным днем ему это все надоело и он попросил написать вас программу которая из этой кучи текста выделит для него только числа. Задача ясна и вы приступили к работе! Ваша задача : Создать класс BillsForCommunal в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая поместит из строки введенную пользователем в поле предназначенное для хранения строки в классе BillsForCommunal все числа встречающиеся в введенной строке(Да,да знаю о чем вы подумали. А как же отрицательные и рациональные числа? Будем считать что числа будут только положительными и целыми. То есть после обработки строки «afsfefe77-3 9.2 dg3 -24» должно вернуть «77 3 9 2 3 24»).

5)В день программиста вы поймали очень много программистов и их было так много , что вы решили дать им самим заполнить бумаги на их арест. И вот , когда они уже отсидели своё и ушли на свой IT-фронт вы обнаружили что бумаги на их арест заполнены какими-то числами типа «72 101 108 108 111 32 102 114 111 109 32 116 104 101 32 97 117 116 104 111 114». Посидев немного и вспомнив с кем вы имели дело вы поняли что эти символы являются кодами для ASCII Table. Пожелав им побольше пропущенных “;” в коде вы приступили к написанию программы которая переведет эти числа в нормальные символы.

Ваша задача: Создать класс Report в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая переведёт числа из строки введенной пользователем в поле предназначенное для хранения строки в классе Report в символы из ASCII Table.

6)Вы потихоньку наводите порядок в своем городе и теперь вы решили выпустить пару важных указов! Вот только прочитав свой указ вас посетило чувство , что чего-то не хватает. Через пару минут вы поняли что вас смутило. Чтобы показать важность вашего указа надо чтобы все буквы в нем были в верхнем регистре! Перепечатать документы вы не успеете , но вот написать программу которая сделает это вы вполне успеете.

Ваша задача: Создать класс Document в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая скопирует строку введенную пользователем в поле предназначенное для хранения строки в классе Document переведя все символы в верхней регистр(В строке используются только буквы латинского алфавита).

7)Прочитав книгу «Криптография для взрослых» вы резко захотели заняться этим интересным делом. Вы поставили у себя в офисе новый , мощный компьютер с процессором Pentium P5 вы решили написать программку которая бы шифровала текст путем замены символа в вашей строке символом который стоит следующим в ASCII Table . К примеру при вводе «I am Batman» мы получим «J!bn!Cbunbo». Задача ясна , приступим!

Ваша задача : Создать класс Encrypt в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая скопирует строку введенную пользователем в поле предназначенное для хранения строки в классе Encrypt заменив символы исходной строки на символы стоящими следующими в ASCII Table.

8)Сегодня в отеле очень скучно и вы придумали себе интересное задание. Вы прошлись по отелю у просили у каждого встречного написать какую ни будь строку, а после вы определяли является ли эта строка Палиндромом. Но строки были такими длинными и их было так много , что вы решили написать программу которая проверит все эти строки и выведет вам только строки-Палиндромы.

Ваша задача: создать класс Palindrome в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая выведет слова Палиндромы из строки введенной пользователем в поле предназначенное для хранения строки в классе Palindrome (слова в строке разделяются одним или несколькими пробелами, все символы кроме пробела считаются частью слова) К примеру при вводе строки «saippuakivikauppias 123 22 3,3 yuy ;7; !!rr ?» должно вывести «saippuakivikauppias 22 3,3 yuy ;7; ?».

Уровень 2:

Упражнение: Создайте класс в котором будет поле для хранения строки(Поле может быть типа char или string на ваше усмотрение).Реализовать метод вывода содержимого этого поля хранящего строку и реализовать перегрузку оператора «=» (Любым из 3 способов рассмотренных выше) чтобы результат удовлетворял условию задания вашего варианта.

Варианты:

1)Какой-то умник сказал вашему боссу что простое удаление информации о читерах из логов не решит проблему , а только скроет её и теперь от вас требуют нормальное решение. Подумав немного вы решили применить самое страшное оружие из своего арсенала «Вычисление по ip». IP игрока вычисляется следующим образом : Сначала идет стандартная часть 192.168 , потом вставляется сумма цифр номера позиции нашего символа в верхнем регистре в строке, потом вставляется код этого символа в верхнем регистре который берётся из ASCII Table. К примеру при вводе «asdHM332 23 dfs sdfG» должно вернуть «192.168.3.72 192.168.4.77 192.168.10.71». И получив ip всех читеров вы сможете отдать их админу , а он их забанит. Задача ясна , приступаем. Ваша задача: Создать класс IPStorage в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая сгенерирует ip по методу описанному выше на основе символов в верхнеи регистре из строки введенной пользователем в поле предназначенное для хранения строки в классе IPStorage.

2)Вы уже неделю трудитесь вместе с математиками и отлично справляетесь со своими заданиями. За ваши успехи ваш руководитель пригласил вас в супер секретный клуб математиков «Пифагоровы штаны». Это очень элитный клуб в котором математики зачитывают первую тысячу знаков после запятой числа Пи под лунной, выступают со Stand Up типа «Пицца с радиусом ц и толщиной а имеет объем пи⋅ц⋅ц⋅а» и занимаются ещё многими интересными вещами. Вам очень хочется попасть в этот клуб но чтобы туда пройти на входе надо выполнить задание. Вам дается строка состоящая из слов(слова разделяются одним или несколькими пробелами) и вам надо разделить сумму кодов символов самого длинного слова на общее число символов в строке(без учета пробелов!). Так как вы ещё не можете так быстро считать , то вы решили написать программу которая вам в этом поможет. Ваша задача: Создать класс Result в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая вернет результат деления суммы кодов самого длинного слова на общее число символов строки(без учета пробелов!) из строки введённой пользователем в поле предназначенное для хранения строки в классе Result.

3)Вы закупили хорошие японские сканеры , вот только вы не учли что в Японии чтение идет справа на лево и поэтому у вас выводит название лекарств наоборот. Вам надо это исправить!

Ваша задача : Создать класс Drug в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая вернет строки заключенные между фигурными скобками «{ }» в перевернутом виде из строки введённой пользователем в поле предназначенное для хранения строки в классе Drug. К примеру при вводе «AEgsdgwf124 fdh{abcd45,}{ergrr}{gfdgfdg» должно вернуть «,54dcba rrgre».

4)Ваш начальник настолько обленился , что ему уже даже лень считать сумму чисел из его счетов и поэтому он попросил вас чтобы ваша программа сразу выводила сумму всех чисел которых она найдет в тексте.

Ваша задача: Создать класс TotalScore в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая вернет сумму всех чисел из строки введённой пользователем в поле предназначенное для хранения строки в классе TotalScore. К примеру при вводе «Afdee54 6 rg-6 PPe-5-4 4.3 3,70,3 011» должно вернуть «54+6-6-5-4+4.3+3.70+3+11=67».

5)За то что программисты так пошутили над вами заполнив документы на арест кодами из ASCII Table вы решили пошутить в ответ отправив всем им штраф в котором все будет записано тоже кодами из ASCII Table. Разумеется вы знаете что программисты быстро поймут что там написано поэтому вы решили все коды записать наоборот. Так как штрафов надо написать много , вы решили написать программу которая вам с этим поможет. К примеру при вводе строки «Learn assembler - salaga.» вам должно вернуть «67 101 79 411 011 23 79 511 511 101 901 89 801 101 411 23 54 23 511 79 801 79 301 79 64».

Ваша задача: Создать класс Joke в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая вернет перевернутые коды каждого символа строки введённой пользователем в поле предназначенное для хранения строки в классе Joke.

6)Недавно вы заключили договор с японцами о постройке новой лаборатории в которой будут создавать квантовый компьютер! Единственна проблема в том , что договор который вы составили им не подходит так как в Японии читают справа на лево и у них осталось очень мало бумаги в офисе поэтому полностью договор они распечатать не могут. Это очень выгодный договор поэтому вы не можете его потерять!

Ваша задача : Создать класс Document в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая вернет строку введённую пользователем перевёрнутой и удалит каждое нечетное слово(слова разделяются одним или несколькими пробелами , любой другой символ будет считаться частью слова) из этой строки в поле предназначенное для хранения строки в классе Document.К примеру при вводе «Herdsfg 89dsg,,sd ,,. ..?? 32fd» должно вернуть «??.. ds,,gsd98».

7)Когда вы закончили программу для шифровки текста путем замены символа следующим символом из ASCII Table вы вспомнили про некий «Шифр Цезаря». Покопавшись в интернете и разузнав об этом шифре побольше, вы поняли что ваши данные должны шифроваться именно так! Ваша задача: Создать класс Caesar в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая вернет зашифрованную при помощи шифра Цезаря строку введённую пользователем. Ключ будет равен сумме цифр кода первого символа строки в ASCII Table.К примеру при вводе «fqwert» должно вернуть «itzhuw»

8)Вам очень понравились слова Палиндромы и поэтому вы решили сделать программу которая бы превращала любое слово в Палиндром. Разумеется смысл у слова потеряется , но зато оно станет читаться с двух сторон одинаково!

Ваша задача : Создать класс Palindrome в котором будет поле для хранения строки,метод для просмотра этого поля , а также программа должна содержать перегруженную операцию «=»,которая все слова из строки введенной пользователем в поле предназначенное для хранения строки в классе Palindrome (слова в строке разделяются одним или несколькими пробелами, все символы кроме пробела считаются частью слова) превратит в Палиндромы. К примеру при вводе строки «101 LLK 89.98 12 qwert ++--» должно вывести «101 LLKLL 89.98 121 qwertrewq ++--++».

Уровень 3  
  
Знаете такую замечательную игру «Быки и коровы» ? Конечно знаете , все её знают. Но на всякий случай напомню вам правила игры. Компьютер генерирует 4-значное число с неповторяющимися цифрами. Игрок, пытается отгадать загаданное компьютером число вводя 4-значные числа. В ответ на введённое пользователем число компьютер сообщает в ответ, сколько цифр угадано без совпадения с их позициями в загаданом компьютере числе (то есть количество коров) и сколько угадано вплоть до позиции в загаданом компьютере числе (то есть количество быков).Теперь давайте напишем эту игру.

Ваша задача : Создать класс BullsAndCows в котором будет 1 поле number для хранения загаданного копмьютером числа.Заполнение полей происходит при помощи конструктора.Так же класс должен содержать метод который выведет подсказку на основе введённых вами данных (Введенное пользователем число передаётся в метод в качестве аргумента) или выведет поздравление в случае если вы отгадали загаданное число.  
Входные данные:

Четырёхзначное число.

Выходные данные:

Два числа - количество быков и коров.

Пример:

|  |  |
| --- | --- |
| Input | Output |
| 5671(Загаданное компьютером число) 7251(Введенное пользователем число) | 1 2 |
| 1234 (Загаданное компьютером число)  1234(Введенное пользователем число) | 4 0 (или победа) |
| 2034 (Загаданное компьютером число)  6234(Введенное пользователем число) | 2 1 |

**Контрольные вопросы:**

1) Как создать динамическую строку?

2) Что такое «перегрузка операций»?

3) Какие способы перегрузки операторов ты знаешь?

4) Какие отличия в перегрузке унарных и бинарных операциях?

5) Какие функции для работы со строками ты знаешь?

6) Что ты знаешь о классе String?

7) Почему при перегрузке унарного оператора через дружественные функции требуется в качестве аргумента передавать операнд, а при перегрузке через метод он не требуется?

8)Каков будет результат выполнения данного кода?

string text = "If you do not know the answer, you will get 2";

for(int i = text.length() - 1; i > -1; i--)

{

cout << text[i];

}

Лабораторная работа №6

Наследование и механизм виртуальных функции.

Наследование - такое соотношение между классами, когда один класс использует часть

другого класса, добавляя этому классу нечто свое и таким образом расширяя возможности этого

класса. При этом первый класс, который описывает наиболее общие свойства ряда объектов,

называется базовый, второй класс – производный.

Определение производного класса имеет следующий синтаксис:

class ID\_производного\_класса : ID\_базового\_класса

{

код\_производного\_класса

};Одна из особенностей порожденного класса - видимость унаследованных компонент базового класса со степенью защиты protected и public (атрибуты базового класса). Компоненты базового класса с ключевым словом private производному классу недоступны.

Производный класс может служить базовым классом для создания следующего, производного класса на более низком уровне иерархии классов. Наследование называется простым, если производный класс имеет одного родителя. В C++ наследование может быть множественным. Множественное наследование позволяет одному классу обладать свойствами двух и более родительских классов.

Видимые компоненты базового класса со степенью защиты protected и public в производном классе становятся private, а значит, не могут быть наследованы производными классами на более низких уровнях иерархии классов. Это связано с тем, что в заголовке производного класса перед ID базового класса по умолчанию компилятор указывает атрибут доступа private. Если же указать явно общедоступный атрибут

class ID\_производного\_класса : public ID\_базового\_класса

то унаследованные компоненты базового класса сохранят степень защиты в производном классе и могут быть унаследованы производным классам на следующем уровне иерархии.

Перед активизацией конструктора производного класса производится вызов конструктора базового класса. Если базовый класс является производным классом на более высоком уровне иерархии, то процесс вызова конструктора производится по своей цепи иерархии классов, пока не доберется до базового класса самого верхнего уровня иерархии. Таким образом, объекты базового класса всегда существуют в составе объектов производного класса. В противоположность этому деструктор производного класса вызывается перед вызовом деструктора базового класса. Это объясняется тем, что уничтожение объекта базового класса влечет за собой уничтожение и объекта производного класса.

Если конструктор базового класса имеет параметры для инициализации своих объектов, то конструктор производного класса обязан обеспечить передачу ему исходных данных. Для этой цели используется форма конструктора со списком инициализаторов, в который включаются идентификаторы конструкторов базового класса (явный вызов конструктора базового класса). При этом последние должны быть объявлены с атрибутом минимум protected.

Продемонстрируем данный механизм на конкретном примере:

#include "pch.h"

#include <iostream>

#include <string>

using namespace std;

class Basic //Базовый класс

{

public:

int BasicInt; // Поле типа public

Basic(int a) // Конструктор

{

BasicInt = a + 5;

};

protected:

int BasicProtectedInt; //Защищенное поле protected

private:

int BasicPrivateInt;//Защищенное поле private

};

class First:protected Basic //Наследуем с доступом protected

{

public:

int x;

First(int a,int b):Basic(a), x(b), y(b + 2), z(a + 2) //Передаем в конструктор Basic значение которое было введено в конструктор First и инициилизируем поля x,y,z

{

BasicProtectedInt = b;

//BasicPrivateInt;

};

private:

int y;

protected:

int z;

};

class Second:public Basic //Наследуем с доступом public

{

public:

int x;

Second(int a,int b):Basic(a),x(b),y(b+2),z(a+2)//Передаем в конструктор Basic значение которое было введено в конструктор Second и инициилизируем поля x,y,z

{

BasicProtectedInt = b;

//BasicPrivateInt;

};

private:

int y;

protected:

int z;

};

int main()

{

First obj(1,3);

//cout << obj.BasicInt; // Нет доступа к полю BasicInt , так как First наследуется с доступом protected из-за чего полю BasicInt присвоился доступ protected вместо public

Second obj2(1, 3);

cout << obj2.BasicInt;//Имеем доступ к полю BasicInt

}

Как правило, коды методов выносят из пространства класса. Тогда для предыдущего примера:

#include "pch.h"

#include <iostream>

#include <string>

using namespace std;

class Basic //Базовый класс

{

public:

Basic(int a);

int BasicInt; // Поле типа public

protected:

int BasicProtectedInt; //Защищенное поле protected

private:

int BasicPrivateInt;//Защищенное поле private

};

Basic::Basic(int a)

{

BasicInt = a + 5;

}

class First:protected Basic //Наследуем с доступом protected

{

public:

int x;

First(int a, int b);

private:

int y;

protected:

int z;

};

First::First(int a, int b) :Basic(a), x(b), y(b + 2), z(a + 2)

{

BasicProtectedInt = b;

}

class Second:public Basic //Наследуем с доступом public

{

public:

int x;

Second(int a,int b);

private:

int y;

protected:

int z;

};

Second::Second(int a, int b) :Basic(a), x(b), y(b + 2), z(a + 2)

{

BasicProtectedInt = b;

}

int main()

{

First obj(1,3);

//cout << obj.BasicInt; // Нет доступа к полю BasicInt , так как First наследуется с доступом protected из-за чего полю BasicInt присвоился доступ protected вместо public

Second obj2(1, 3);

cout << obj2.BasicInt;//Имеем доступ к полю BasicInt

}

**Использование косвенной адресации с установкой указателей на базовый класс.**

Механизм косвенной адресации рассмотрим на конкретном примере:

#include "pch.h"

#include <iostream>

using namespace std;

class Basic //Базовый класс

{

public:

Basic(int a);

int BasicInt; // Поле типа public

protected:

int BasicProtectedInt; //Защищенное поле protected

private:

int BasicPrivateInt;//Защищенное поле private

};

Basic::Basic(int a)

{

BasicInt = a + 5;

}

class First:protected Basic //Наследуем с доступом protected

{

public:

int x;

First(int a, int b);

private:

int y;

protected:

int z;

};

First::First(int a, int b) :Basic(a), x(b), y(b + 2), z(a + 2)

{

BasicProtectedInt = b;

}

class Second:public Basic //Наследуем с доступом public

{

public:

int x;

Second(int a,int b);

private:

int y;

protected:

int z;

};

Second::Second(int a, int b) :Basic(a), x(b), y(b + 2), z(a + 2)

{

BasicProtectedInt = b;

}

int main()

{

First f(1,2);// Конструктор класса First создает объект f

Second s(3,4);// Конструктор класса Second создает объект S

Basic \*b; // Указатель на базовый касс

b = &s;// Указатель Базового класса указывает на объект производного Second

//b = &f;// Указатель Базового класса не может указать на объект производного First из-за того , что First наследуется с доступом protected

//Хоть b сейчас и указывает на объект класса Second , без преобразования указателя С++ не позволит нам сейчас использовать поля объекта наследника.

//cout << b->x;//Не сработает!!!

int i = b->BasicInt; // Базовый класс виден напрямую

// Стиль языка С для явного преобразования типов с использованием круглых скобок:

//int j = ((Second\*)(b))->x;

//Использование кастов c++ для преобразования типов

// Введённые в С++ касты (приёмы преобразований типов) не дают программе собраться, если преобразование черезчур опасное. Эти ограничения повышают надёжность вашей программы (Советую поподробнее изучить данную тему для общего развития).

int j = static\_cast<Second\*>(b)->x;

// через переменные печатаем их текущее состояние

cout << " BasicInt = " << i << endl;

cout << " x = " << j << endl;

}

**Механизм виртуальных функции**

Механизм виртуальных функций - одна из основных концепций объектно- ориентированного программирования. Данный механизм предполагает использование идеи «один интерфейс — множество методов реализации». Эта идея заключается в том, что базовый класс обеспечивает для производных классов все элементы, которые эти классы могут использовать непосредственно, а также содержит набор функций, которые производные классы должны реализовать путем их переопределения (создание собственного кода функции в производном классе, которые позволяют решить поставленную перед производным классом задачу). Виртуальная функция - это функция, объявленная с ключевым словом virtual в базовом классе и переопределенная в одном или нескольких производных от этого классах. Обязательное требование: заголовок функции должен быть точно такой же, как в базовом классе (и имя, и список параметров должны быть одинаковые). Тогда при создании объекта или базового, или одного из производных классов компилятор определяет, какую из функций требуется вызвать, основываясь на типе (ID класса) объекта. Таким образом, компилятор на этапе компиляции, обнаружив виртуальные функции с одинаковыми заголовками, но с разными кодами, не конкретизирует, какой из виртуальных методов будет вызываться, а отводит для их адресов место в специальной таблице адресов виртуальных методов. И далее на этапе выполнения зарезервированное в таблице место инициализируется адресом соответствующей виртуальной функции конструктором того класса, объект которого создается в данный момент выполнения программы.

Такой процесс в C++ получил название «позднее связывание».

Как правило, на практике в базовом классе приводят только прототип виртуального метода, который определяет общий интерфейс, указывающий, с какими данными необходимо работать. А в производном классе приводят полные коды, которые определяют способы обработки указанных данных.

Если требование полного сохранения заголовка виртуального метода в производном классе нарушено, то компилятор, обнаружив это, механизм виртуальных функций проигнорирует и произведет на этапе компиляции его перегрузку.

Давайте решим задачу с использованием виртуальных функций чтобы лучше разобраться с ними.

Создадим базовый класс Weapons в котором будет хранится кол-во патрон в магазине для каждого производного от него класса и 3 класса оружия pistol, MachineGun и WaterGun.Каждое оружие должно стрелять по своему и у каждого оружие своё кол-во патронов в магазине. Давайте посмотрим как удобно решается такая задача с использованием виртуальных функций:

#include "pch.h"

#include <iostream>

using namespace std;

class Weapons //Базовый класс

{

public:

int AmmunitionStore;//Храним кол-во патронов в магазине

Weapons(int a);

virtual void ReloadingWeapons(); // Создаем прототип метода перезарядки оружия

virtual void Shoot();// Создаем прототип метода стрельбы оружия

};

Weapons::Weapons(int a) // Заряжаем оружие

{

AmmunitionStore = a;

}

void Weapons::ReloadingWeapons()

{

}

void Weapons::Shoot()

{

}

//Описываем пистолет

class pistol :public Weapons

{

public:

pistol(int a);

void ReloadingWeapons() override;

void Shoot() override;

};

pistol::pistol(int a):Weapons(a)

{

}

void pistol::ReloadingWeapons()//Описываем метода перезарядки оружия для pistol

{

cout << "\nReloadingWeapons\n";

AmmunitionStore = 8;//В пистолет ложим 8 патронов

}

void pistol::Shoot()//Описываем метода стрельбы оружия для pistol

{

if (AmmunitionStore==0)

{

cout << "AmmunitionStore empty!\n";

return;

}

for (int i=0;i< AmmunitionStore;i++)

{

cout << "BANG! ";

}

AmmunitionStore = 0;//Патроны закончились

cout << endl;

}

//Описываем пистолет end

//Описываем автомат

class MachineGun:public Weapons

{

public:

MachineGun(int a);

void ReloadingWeapons() override;

void Shoot() override;

};

MachineGun::MachineGun(int a) :Weapons(a)

{

}

void MachineGun::ReloadingWeapons()//Описываем метода перезарядки оружия для MachineGun

{

cout << "\nReloadingWeapons\n";

AmmunitionStore = 31;//В автомат ложим 31 патрон

}

void MachineGun::Shoot()//Описываем метода стрельбы оружия для MachineGun

{

if (AmmunitionStore == 0)

{

cout << "AmmunitionStore empty!\n";

return;

}

for (int i = 0; i < AmmunitionStore/3; i++)

{

cout << "TRATATA! ";

}

AmmunitionStore = 0;//Патроны закончились

cout << endl;

}

//Описываем автомат end

//Описываем водный пистолет

class WaterGun :public Weapons

{

public:

WaterGun(int a);

void ReloadingWeapons() override;

void Shoot() override;

};

WaterGun::WaterGun(int a) :Weapons(a)

{

}

void WaterGun::ReloadingWeapons()//Описываем метода перезарядки оружия для WaterGun

{

cout << "\nReloadingWeapons\n";

AmmunitionStore = 10;//В водный пистолет ложим воды на 10 выстрелов

}

void WaterGun::Shoot()//Описываем метода стрельбы оружия для WaterGun

{

if (AmmunitionStore == 0)

{

cout << "AmmunitionStore empty!\n";

return;

}

for (int i = 0; i < AmmunitionStore; i++)

{

cout << "Piu! ";

}

AmmunitionStore = 0;//Патроны закончились

cout << endl;

}

//Описываем водный пистолет end

int main()

{

int chose;

//Создаем объекты нашего оружия

WaterGun WG(5);

pistol P(3);

MachineGun MG(25);

Weapons \*arsenal = &P;//Создаем указатель базового класса который мы сможем направлять на каждый производный от него объект созданых выше.По стандарту поставим его на pistol

while (true)

{

cout << endl << "Chose Weapon: \n1:Pistol\n2:MachineGun\n3:WaterGun\nYor chose:"; cin >> chose;

switch (chose)

{

case 1:

{

arsenal = &P;//Берём наш пистолет

break;

}

case 2:

{

arsenal = &MG;//Берём наш автомат

break;

}

case 3:

{

arsenal = &WG;//Берём наш водный пистолет

break;

}

default:

cout << endl << "Invalid input!" << endl;

break;

}

//Выполняем операции с оружием на которое указывает arsenal

arsenal->Shoot();//Выстреливаем все патроны

arsenal->Shoot();//Получаем сообщение о нехватки патрон

arsenal->ReloadingWeapons();//Перезаряжаем оружие

arsenal->Shoot();//Выстреливаем все патроны

}

}

Упражнение: Программа должна содержать: базовый класс , включающий поля и виртуальные методы необходимые для выполнения задания в соответствии с вариантом и производные от базового классы с перезаписанными методами в соответствии с заданием выбранного варианта.

-Базовый класс должен содержать конструктор с параметрами для инициализации полей базового класса в динамической памяти.

- Деструктор для освобождения занятой памяти;

- виртуальные методы просмотра текущего состояния и переустановки значений полей базового класса в новое состояние.

Производные классы должны иметь:

- конструктор с параметрами и списком инициализаторов, передающий данные конструктору базового класса;

- переопределенные методы просмотра текущего состояния объектов и их переустановки в новое состояние.

Уровень 1

Варианты:

1)После того как вы успешно разобрались с читерами в игре «Sword, Magic, Shield, Bow, Arrow and something else» вас попросили помочь с разработкой нового патча обновления для этой игры.Это глобальный патч в котором планируют добавить очень многое : Новых монстров, героев,NPC,локации,оружие,заклинания,механику и даже новое название самой игры! Конкретно вы занимаетесь разработкой новых монстров. Так как монстров надо добавить очень много , то вы решили описать общую структуру монстра и создать 3 примера создания нового монстров на основе этой структуры , а дальше передать этот пример стажёрам и пусть они по аналогии создают новых монстров а вы в это время попьёте кофейку!

Ваша задача: Создать базовый класс Monsters в котором будут поля : HP, damage, armour и 2 виртуальных метода PrintInfo() и Scream(). Так же создать 3 производных класса от класса Monsters (Название классам придумайте самостоятельно) в которых будет происходить переопределение методов PrintInfo() и Scream(). Переопределённое PrintInfo() должно вывести поля HP, damage, armour умноженные на константное значение (На какое значение умножать решайте сами) , а метод Scream() должен вывести крик этого монстра (Как будет кричать ваш монстр придумайте сами) к примеру «Life for Nerzula!».

2)На кафедре математики вас попросили подготовить сообщение про число Эйлера и число ПИ для рассказа студентам. Во время подготовки сообщения вам пришла интересная мысль округлить Число ПИ (3,14…) и число Эйлера(2,718…) в результате чего вы пришли к выводу что Число ПИ == числу Эйлера. Рассказав об этом открытии на кафедре математики вас попросили немедленно покинуть кафедру и никогда на неё не возвращаться! Не зная куда податься вы решили пойти в администрацию центра чтобы они помогли вам найти новую кафедру где вы могли бы работать. Когда вы пришли в администрацию и рассказали вашу ситуацию вам предложили поработать на администрацию. Вашей задачей будет подсчет расходов на каждую кафедру. Подумав немного вы решили написать программу которая сама бы делала все необходимые расчеты.  
Ваша задача: Создать базовый класс Faculty в котором будут поля : NumberOfEmployees,

EmployeeSalary, Surcharge и 2 виртуальных метода PrintInfo() и CostsWithSurcharge(). Так же создать 3 производных класса от класса Faculty (Название классам придумайте самостоятельно) в которых будет происходить переопределение методов PrintInfo() и CostsWithSurcharge (). Переопределённое PrintInfo() должно вывести поля NumberOfEmployees, EmployeeSalary и расход на кафедру который равен NumberOfEmployees\* EmployeeSalary, а метод CostsWithSurcharge () должен вывести поля NumberOfEmployees,EmployeeSalary, Surcharge и расход на кафедру который равен NumberOfEmployees \* ( (EmployeeSalary + Surcharge) \* 0.13).

3)Вы уже целый месяц работаете в больнице глав врачом и уже многое успели сделать. Единственное что вы не успели – это выдать сотрудникам зарплаты! Пока они этого не заметили вам нужно срочно посчитать сколько и кому надо платить. Так как надо все это сделать быстро и времени считать столбиком у вас нет , то вы решили написать программу которая все посчитает за вас.  
Ваша задача: Создать базовый класс Employees в котором будут поля : HourlyPayment,

WorkedHours, Surcharge и 2 виртуальных метода PrintInfo() и CostsWithSurcharge(). Так же создать 3 производных класса от класса Employees (Название классам придумайте самостоятельно) в которых будет происходить переопределение методов PrintInfo() и CostsWithSurcharge (). Переопределённое PrintInfo() должно вывести поля HourlyPayment, WorkedHours и зарплату сотрудника которая равна HourlyPayment \* WorkedHours, а метод CostsWithSurcharge () должен вывести поля WorkedHours, HourlyPayment, Surcharge и зарплату сотрудника которая равна WorkedHours \* ( HourlyPayment + Surcharge \* 0.13).

4)Ваш начальник обожает инновационные технологии и на этот раз он попросил вас написать программу которая бы показывала сколько денег в месяц нужно будет платить человеку который берёт ипотеку в банке с учётом процента этого банка. Ваш начальник владеет несколькими банками и у каждого из них свой процент по ипотеке и это нужно учитывать!

Ваша задача: Создать базовый класс Bank в котором будут поля : AmountOfDebt,

NumberOfMonths и 2 виртуальных метода PrintInfo() и Debt(). Так же создать 3 производных класса от класса Bank (Название классам придумайте самостоятельно) в которых будет происходить переопределение методов PrintInfo() и Debt (). Переопределённое PrintInfo() должно вывести поля AmountOfDebt, NumberOfMonths, а метод Debt () должен вывести сколько в месяц должен будет платить человек именно этому банку.

Сколько в месяц должен будет платить человек вычисляется по формуле : (AmountOfDebt + AmountOfDebt \*[ (процент банка придумайте сами и вставьте сюда) /100] ) / NumberOfMonths ;

5)Скоро в ваш полицейский участок придет школьная экскурсия которой вы должны будите показать своё участок и рассказать про разные виды преступлений и что за эти преступления вам грозит. Так как экскурсий будет много , а вам ещё преступления раскрывать то вы решили поставить в холе интерактивное табло в котором бы дети просто выбирали бы тип преступника , а им выводило бы сколько лет заключения грозит этому преступнику. Осталось только запрограммировать это табло.  
Ваша задача: Создать базовый класс Criminals в котором будут поля : YearsImprisonment,

SeverityOfCrime, MonetaryPenaltyForCrime и 2 виртуальных метода PrintInfo() и HowManyDays(). Так же создать 3 производных класса от класса Criminals (Название классам придумайте самостоятельно) в которых будет происходить переопределение методов PrintInfo() и HowManyDays (). Переопределённое PrintInfo() должно вывести поля YearsImprisonment, SeverityOfCrime, MonetaryPenaltyForCrime, а метод HowManyDays () должен вывести кол-во дней которое надо отсидеть преступнику которое вычисляется по формуле (YearsImprisonment\*365)/2 если SeverityOfCrime<4 или по формуле YearsImprisonment\*365 если 4<=SeverityOfCrime<=10 или по формуле YearsImprisonment\*365 \* 2 во всех остальных случаях.

6) Вы хотите начать грандиозную стройку в городе но чтобы что-то построить за это что-то нужно сначала заплатить именно поэтому вы сегодня пригласили самых богатых людей своего города чтобы они инвестировали свои деньги в развитие города. Чтобы не спугнуть инвесторов одними лишь большими числами вы решили поставить интерактивное табло на которой бы выводилась красивая картинка будущего строения и расход на постройку одного такого строения. Осталось только запрограммировать это табло.

Ваша задача: Создать базовый класс Building в котором будут поля : Cost,MonthlyProfit и 2 виртуальных метода PrintInfo() и WhenWillPayOff (). Так же создать 3 производных класса от класса Building (Название классам придумайте самостоятельно) в которых будет происходить переопределение методов PrintInfo() и WhenWillPayOff (). Переопределённое PrintInfo() должно вывести поля Cost,MonthlyProfit, а метод WhenWillPayOff () должен вывести через сколько месяцев здание окупит свою стоимость. Вычисляется это по формуле : Cost/ MonthlyProfit.

7)Ваш завод по производству всякой всячины каждый день производит всё больше и больше товара что не может вас не радовать. Но вот только теперь вам нужно найти новые рынки сбыта для этого товара. Для привлечения внимания магазинов вы решили написать программу которая будет выдавать информацию о товарах производимых на вашей фабрике их кол-во и цену , а так же производить элементарные вычисления чтобы продавцам не тратить время на подсчёты столбиком.

Ваша задача: Создать базовый класс Product в котором будут поля : Cost, Quantity и 2 виртуальных метода PrintInfo() и PriceForBulkPurchases (). Так же создать 3 производных класса от класса Product (Название классам придумайте самостоятельно) в которых будет происходить переопределение методов PrintInfo() и PriceForBulkPurchases (). Переопределённое PrintInfo() должно вывести поля Cost, Quantity и общую сумма за весь товар равную Cost\*Quantity , а метод PriceForBulkPurchases () должен вывести сколько нужно заплатить при покупке всего товара этого типа оптом. Вычисляется это по формуле :

(Cost\*[(Процент скидки для данного товара придумайте сами и вставьте сюда)/100])\* Quantity.

8)Ваша блестящая идея «Своя комната» принесла вам много клиентов , но отель из-за низкой стоимости комнат начал нести убытки именно поэтому вам пришлось вернуть стандартные номера и фиксированные цены на них. Чтобы посетитель мог посмотреть информацию о комнате вы поставили в холле отеля интерактивное табло где изображены фото выбранной комнаты и информация о комнате. Осталось только запрограммировать это табло.

Ваша задача: Создать базовый класс Room в котором будут поля : Cost, NumberOfResidents, Convenience и 2 виртуальных метода PrintInfo() и HowMuchIs3Days (). Так же создать 3 производных класса от класса Room (Название классам придумайте самостоятельно) в которых будет происходить переопределение методов PrintInfo() и HowMuchIs3Days (). Переопределённое PrintInfo() должно вывести поля Cost, NumberOfResidents, Convenience ,а метод HowMuchIs3Days () должен вывести цену аренды данной комнаты за 3 дня которая вычисляется по формуле:

Cost\*3 если Convenience<3 или по формуле (Cost\*3)+( Cost\*0,15) если 3<Convenience<=5 или по формуле (Cost\*3+Cost\*0,15) \* NumberOfResidents во всех остальных случаях .

Уровень 2

Варианты:

1)Ваши стажеры добавили кучу новых монстров и теперь надо проверить сможет ли победить их наш игрок!  
Ваша задача: Добавить в ранее написанную программу класс Player в котором будут поля : HP, damage, armour. Поля заполняются при помощи конструктора. В классе Player должен быть так же описан метод WhoWin() принимающий в качестве аргумента указатель базового класса Monsters.Устанавливайте этот указатель на объекты ваших производных классов от Monsters и передавайте его в функцию WhoWin() которая должна вывести сообщение «Player win!» в случае если player оказался сильнее монстра и сообщение «Player lose!» если монстр оказался сильнее. Определить кто сильнее очень просто. Если HP(игрока)/(Урон монстра – броня игрока) > HP(Монстра)/ (Урон игрока – броня монстра) , то игрок победил, а в противном случае проиграл.

2)После того как вы показали общую сумму расходов ваш начальник попросил вас уменьшить кол-во сотрудников на кафедрах на 10% , а 50% зарплаты уволенных поровну распределить между оставшимися сотрудниками. Задача ясна , приступаем!  
Ваша задача: Добавить в ранее написанную программу класс Dismisses в котором будет описан метод Check() принимающий в качестве аргумента указатель базового класса Faculty.Устанавливайте этот указатель на объекты ваших производных классов от Faculty и передавайте его в функцию Check () которая должна переустановить значение NumberOfEmployees на (NumberOfEmployees - NumberOfEmployees\*0.1) , а значение EmployeeSalary на (EmployeeSalary + NumberOfEmployees(обновленное) / (NumberOfEmployees(не обновленное)\*0.1 \* EmployeeSalary) / 2 ).

3)Изучив информацию о сотрудниках вы увидели что у вас в больнице есть сотрудники которые работают намного больше ,чем другие сотрудники поэтому вы решили поощрить трудоголиков и наказать лентяев путем пересчета зарплат!  
Ваша задача: Ваша задача: Добавить в ранее написанную программу класс Recalculation в котором будет описан метод Check() принимающий в качестве аргумента указатель базового класса Employees.Устанавливайте этот указатель на объекты ваших производных классов от Employees и передавайте его в функцию Check () которая должна переустановить значение HourlyPayment на (HourlyPayment + HourlyPayment \*0.1) если WorkedHours>12, а в противном случае переустановить значение HourlyPayment на (HourlyPayment - HourlyPayment \*0.1).

4)Несмотря на большие проценты много людей взяли ипотеку в банках вашего начальника из-за чего в банке начали кончаться деньги ведь все только берут , а вернут только через годы. Чтобы не обанкротится ваш начальник принял решение уменьшить кол-во месяцев для выплаты ипотеки для тех , кто берёт не очень много денег у банка.  
Ваша задача: Добавить в ранее написанную программу класс Recalculation в котором будет описан метод Check() принимающий в качестве аргумента указатель базового класса Bank.Устанавливайте этот указатель на объекты ваших производных классов от Bank и передавайте его в функцию Check () которая должна переустановить значение NumberOfMonths на (NumberOfMonths - NumberOfMonths \*0.3) если AmountOfDebt <100000.

5)Только что вы получили новый уголовный кодекс вашего города в котором увеличили строки заключения преступников в зависимости от тяжести преступления и теперь вам нужно срочно изменить информацию о сроке заключения для каждого вида преступников.  
Ваша задача: Добавить в ранее написанную программу класс Recalculation в котором будет описан метод Check() принимающий в качестве аргумента указатель базового класса Criminals.Устанавливайте этот указатель на объекты ваших производных классов от Criminals и передавайте его в функцию Check () которая должна переустановить значение YearsImprisonment на (YearsImprisonment + (YearsImprisonment \*SeverityOfCrime/100) ).

6)На встрече с самыми богатыми людьми вашего города вы познакомились с хозяином заводов по производству строй материалов и он согласен снизить цену на стройматериалы (из-за чего цена постройки здания уменьшится на 20%) для зданий у которых будет высокий ежемесячный доход. Теперь вам нужно срочно произвести перерасчет зданий чтобы снизить их цену и тем самым привлечь ещё больше инвесторов!

Ваша задача : Добавить в ранее написанную программу класс Recalculation в котором будет описан метод Check() принимающий в качестве аргумента указатель базового класса Building.Устанавливайте этот указатель на объекты ваших производных классов от Building и передавайте его в функцию Check () которая должна переустановить значение Cost на (Cost / 5 \*4) если (MonthlyProfit\*12) > Cost(не обновленное).

7)Ваш завод просто бьет все рекорды производительности из-за чего вам надо понизить цену на товар которого у вас очень много чтобы поскорее его продать иначе у вашего склада просто снесёт крышу в прямом смысле этого слова!

Ваша задача : Добавить в ранее написанную программу класс Recalculation в котором будет описан метод Check() принимающий в качестве аргумента указатель базового класса Product.Устанавливайте этот указатель на объекты ваших производных классов от Product и передавайте его в функцию Check () которая должна переустановить значение Cost на (Cost / 5 \*4) если Quantity >1000.

8) Недавно ваш отель посетил известный критик отелей и дал вашему отелю 3.3 звезды! Так как 3.3 больше ем 3 , то и цены комнат тоже можно увеличить!

Ваша задача: Добавить в ранее написанную программу класс Recalculation в котором будет описан метод Check() принимающий в качестве аргумента указатель базового класса Room.Устанавливайте этот указатель на объекты ваших производных классов от Room и передавайте его в функцию Check () которая должна переустановить значение Cost на (Cost + (Cost \* Convenience /100) ).

Уровень 3

В конторе «Быки и коровы» подходит время подведения годового баланса. В бухгалтерию поступили сведения о том, что, согласно документам, суммарный расход составил а рублей, a суммарный приход – b рублей. Поскольку с реальным положением дел эти цифры все равно не имеют ничего общего, бухгалтер решил реализовать следующую свою идею. Как известно, при наборе чисел на компьютере люди часто вводят цифры в неправильном порядке. Поэтому бухгалтер хочет найти такой способ переставить цифры в числах a и b, чтобы в результате разность a-b (и, соответственно, количество денег, которые он положит к себе в карман), была максимальна, а в случае можно будет сослаться на ошибку секретаря. При этом нельзя забывать о знаке чисел и о том, что ноль не может быть первой цифрой числа, отличного от ноля. Напишите программу, которая поможет бухгалтеру.

Ваша задача : Создать класс Accountantв котором будет 2 поле a,b для хранения расхода и дохода.Заполнение полей происходит при помощи конструктора.Так же класс должен содержать метод который найтдёт такой способ переставить цифры в числах a и b, чтобы в результате разность a-b была максимальна.  
Входные данные:

Два целых числа a и b.

Выходные данные:

Одно целое число – наибольшую разность чисел

|  |  |
| --- | --- |
| Input | Output |
| 18  10 | 71 |
| 1  -23 | 33 |

**Контрольные вопросы:**

1)Что такое «наследование» и иерархия классов?

2) Какие элементы базового класса видны из производного? Как управлять степенью их защиты?

3) Какое наследование называют множественным?

4) Поясните механизм виртуальных функций.

5) Что такое чисто виртуальная функция?

6)Если указатель базового класса указывает на объект производного от него класса что нужно сделать чтобы получить доступ к полям и методам производного класса?

7)Какой порядок вызова имеют конструкторы и деструкторы при наследовании?

8)Какой будет результат выполнения данного кода?   
class A

{

public:

A()

{

cout << "Constructor A" << endl;

}

~A()

{

cout << "Destructor A" << endl;

}

};

class B:protected A

{

public:

B()

{

cout << "Constructor B" << endl;

}

~B()

{

cout << "Destructor B" << endl;

}

};

class C : B

{

public:

C()

{

cout << "Constructor C" << endl;

}

~C()

{

cout << "Destructor C" << endl;

}

};

int main()

{

C obj;

}

Лабораторная работа №7

Шаблоны классов.

Механизм шаблонов C++ - это средство построения обобщенных определений функций и классов, которые не зависят от используемых типов данных. Этот механизм позволяет сократить трудоемкость создания программ, т.к. повышает лаконичность текста, т.е. использование шаблонов избавляет от необходимости дублировать коды классов и функций для разных типов данных.

Компилятор по заданному типу аргументов на основе описания шаблона автоматически создает соответствующие экземпляры классов и функций, которые называются представители конкретных классов и функций.

Шаблоны класса в отличие от шаблона функции позволяют параметризовать, т.е. использовать в качестве параметров не только типы элементов данных, но и константы разных типов.

Синтаксис определения шаблона класса следующий:

**template обычное описание структуры класса;**

При этом список параметров шаблона не может быть пустым. Элементы в списке разделяются запятыми. Внутри пространства класса параметры шаблона должны быть хотя бы один раз упомянуты.

В список параметров могут входить два вида параметров:

1) типизированные параметры, начинающиеся со слов

class Идентификатор,

компилятор при создании экземпляра класса заменит его на конкретный тип данных;

2) нетипированные параметры шаблона:

Стандартный тип числовых данных Идентификатор

Таким образом, типированные параметры - это фиктивные имена типов данных, входящих в класс.

Нетипированные параметры - это поименованные типы числовых констант. При этом им при декларировании можно присваивать умалчиваемые значения.

Каждый параметр является локальным в рамках пространства класса.

В описании класса хотя бы один раз необходимо упомянуть ID типированных и нетипированных параметров, конкретные значения для которых будут переданы в момент создания объекта этого класса через список аргументов, который указывается через запятые в треугольных скобках сразу после ID класса:

**ID\_ класса ID\_объектов'**

ID\_объектов — идентификаторы объектов, которые создает данный класс (записанные через запятые, например а,b,с). При этом в списке аргументов каждому типированному параметру шаблона должен соответствовать известный конкретный тип данных, а каждому нетипированному параметру - константное выражение соответствуючего типа. Таким образом, между списком параметров шаблона и списком аргументов должно быть абсолютное соответствие по количеству, порядку их следования и типам. Если нетипированные параметры имеют умалчиваемые значения, их располагают в списке последними.

Давайте рассмотрим пример с применением шаблонов:

#include "pch.h"

#include <iostream>

using namespace std;

template<typename K> //Создаем шаблон у которого только один параметр - тип

class ArrayElement //Описываем класс для создания объектов будущего массива

{

public:

int a;//Поле типа int

double b;//Поле типа double

K c;//Поле с неизвестным типом данных

};

template<typename T,class AE,int MAXSIZE = 20> //Создаем шаблон у которого первые 2 параметра типизированные. 1 параметр - принимает стандартный тип данных, 2 параметр принимает пользовательский тип данных , 3 параметр прямое значение (нетипированный параметр ) . Если 3 параметр не будет указан при создании объекта , то MAXSIZE установится значение = 20.

class Array //Описываем класс для создания хранения массива неопределённого типа и переменной неопределённого типа.

{

public:

T k = 2.9; // Переменная неопределённого типа

AE \*Mass; // Указатель неопределённого типа

Array();//конструктор

void Print();//Метод для вывода массива

};

template<typename T, class AE, int MAXSIZE>

Array<T, AE, MAXSIZE>::Array()//Описываем конструктор

{

Mass = new AE[MAXSIZE]; //Выделяем память

for (int i = 0; i < MAXSIZE; i++) //Заполняем массив

{

Mass[i].a = i;

Mass[i].b = i;

Mass[i].c = k;

}

}

template<typename T, class AE, int MAXSIZE>

void Array<T, AE, MAXSIZE>::Print()//Описываем метод вывода

{

for (int i = 0; i < MAXSIZE; i++)

{

cout << Mass[i].a << endl;

cout << Mass[i].b << endl;

cout << Mass[i].c << endl;

cout << endl;

}

}

int main()

{

Array<int ,ArrayElement<int>, 15> obj; //Создаем объект obj в котором поле K c в классе ArrayElement будет типа int и поле T k в классе Array будет типа int, MAXSIZE будет = 15 ;

obj.Print();

Array<double, ArrayElement<double> > obj2;//Создаем объект obj в котором поле K c в классе ArrayElement будет типа double и поле T k в классе Array будет типа double, MAXSIZE будет = 20 так как мы не передали 3 параметр и установилось значение прописанное в template;

obj2.Print();

}

Упражнение: Создать шаблон класса, порождающего динамические одномерные массивы с элементами различных типов (вещественные, целочисленные, символьные и т.д.). Тип данных и результат являются параметрами по отношению к классу. Программа должна содержать: конструктор, деструктор, метод просмотра значений созданного массива, а также метод для решения задачи в соответствии с вариантом.

Уровень 1

Варианты:

1)Сегодня у вас выходной и вы решили заняться самообразованием! И тема сегодняшнего урока – шаблоны! Почитав о шаблонах в интернете вы решили закрепить материал выполнив небольшое задание.  
Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести среднее значение всех элементов массива. Если тип double , то нужно вывести сумму всех элементов массива. Если тип char , то нужно вывести значение первого и последнего элемента массива.

2)Поработав пару недель в администрации вам наскучило и вы решили найти новую работу. Так как вас отовсюду повыгоняли у вас осталась одна единственная надежда на Кафедру вычислительных методов и программирования! После собеседования выяснилось что вы не знаете что такое шаблон класса. Вам дали несколько часов на то чтобы исправить это недоразумение и выполнить тестовое задания для того чтобы вас зачислили на кафедру.  
Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести элементы массива находящиеся между максимальным и минимальным элементом. Если тип double , то нужно вывести разницу между максимальным и минимальным элементом массива. Если тип char , то нужно вывести код каждого символа в соответствии с ASCII Table.

3)Вы подали заявку на участие в конкурсе «Best Hospital in the world».Для участия в конкурсе вам требуется предоставить кол-во выздоровевших пациентов или расход больницы или счастливый символ больницы (Вот такие правила у конкурса. Сам в шоке). Так как каждый день эта информация была разной вам сразу пришла мысль о массивах для её хранения. А так как при обучении на врача вас так же обучили работе с шаблонами классов вы знаете как создать массив с неопределённым типом данных!  
Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести максимальный элемент массива. Если тип double , то нужно вывести минимальный элемент массива. Если тип char , то нужно просто вывести все элементы массива.

4)Ваш начальник попросил вас написать программу в которую можно было бы ввести 3 разных по типу массива , а она бы в зависимости от введённых данных выдала бы результат. За выполнение задания без лишних вопросов он обещал вам премию. Задача не сложная , поэтому вы согласились её выполнить.  
Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести нечётные элемент массива. Если тип double , то нужно вывести сумму элементов массива. Если тип char , то нужно просто вывести все элементы массива с конца.

5)За время своей работы вы собрали информацию о преступниках в каждом районе своего города и теперь вам нужно занести собранную вами информацию в компьютер. Если вводится последовательность целых чисел , то это говорит о кол-во преступников в каждом районе. Последовательность вещественных чисел говорит об уровне преступности в каждом районе, а последовательность символов говорит о кодовых кличках агентов под прикрытием в каждом районе. В зависимости от выбранного сотрудником типа данных ему должно выдать определённый результат для более продуктивной работы, именно поэтому вы изучили «шаблоны классов» на курсах повышения квалификации и знаете как решить данную задачу наиболее оптимальным способом.

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести сумму элементов массива. Если тип double , то нужно вывести максимальный элемент массива. Если тип char , то нужно вывести элемент с самым большим кодом в ASCII Table.

6)Ваш город растёт и теперь вам нужно заполнить информацию о каждом районе в вашем городе. Если вводится последовательность целых чисел , то это говорит о кол-во строений в каждом районе. Последовательность вещественных чисел говорит о сумме прибыли в каждом районе, а последовательность символов говорит о кодовом названии каждого района. В зависимости от от выбранного сотрудником типа данных ему должно выдать определённый результат . На курсах «ООП – за один день» который вы успешно прошли вам рассказывали про «шаблоны классов» поэтому вы знаете как решить данную задачу наиболее оптимальным способом.

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести среднее значение элементов массива. Если тип double , то нужно вывести (максимальный элемент массива – минимальный элемент массива). Если тип char , то нужно вывести элемент с самым маленьким кодом в ASCII Table.

7)Вы поставили на своём заводе супер компьютер который контролирует процесс производства и теперь вы хотите научить его принимать массив информации и исходя из типа полученного массива чтобы он производил какие ни будь действия. Вместе с компьютером вы получили инструкцию где есть глава про «шаблоны классов» и где подробно написано как реализовать то , что вы хотите. Вот только инструкцию вы выбросили сразу же как её увидели поэтому вам пришлось немного посидеть в интернете для обучения , но зато теперь вы знаете как выполнить задачу!

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести все элементы массива располагающиеся до максимального элемента. Если тип double , то нужно вывести сумму элементов между максимальным и минимальными элементами. Если тип char , то нужно поменять местами символ с максимальным кодом в ASCII Table с символом с минимальным кодом в ASCII Table.

8)Работать администратором очень трудно. Все время нужно помнить номера которые заняты , сумму которую должен этот номер ,а недавно сотрудники придумали символьную систему обозначения номеров и теперь у каждого номера есть свой символ. Вы не жалуетесь на память , но все же решили что пусть лучше компьютер помнит всю эту информацию , а вы будите у него просто спрашивать при необходимости.

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести все элементы массива располагающиеся после минимального элемента. Если тип double , то нужно вывести среднее значение элементов массива .Если тип char , то нужно вывести все элементы между символом с максимальным кодом в ASCII Table и символом с минимальным кодом в ASCII Table.

Уровень 2

Упражнение: Создать шаблон класса, порождающего динамические одномерные массивы с элементами различных типов (вещественные, целочисленные, символьные и т.д.). Тип данных и результат являются параметрами по отношению к классу. Программа должна содержать: конструктор, деструктор, метод просмотра значений созданного массива, а также метод для решения задачи в соответствии с вариантом.

Варианты:

1)Вы успешно выполнили предыдущее задание и готовы к задачке посложнее! Приступим!

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести среднее значение всех нечётных элементов массива. Если тип double , то нужно отсортировать и вывести массив в порядке возрастания. Если тип char , то нужно вывести значение нечётных по порядку элементов массива.

2)Вы выполнили тестовое задание и вас отправили к заведующему кафедры который сказал что задание слишком простое и решил его немного усложнить. Вы же не против?

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести все значения элементов массива с поставленными задом на перёд числами. То есть при вводе «12 3 4 56 78» вам должно вывести «21 3 4 65 87». Если тип double , то нужно вывести среднее значение чётных элементов массива .Если тип char , то нужно отсортировать этот массив и вывести его в порядке возрастания.

3)Вы отправили необходимые данные на конкурс , да вот только их правила снова изменились и теперь им нужны другие результаты.

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно вывести произведение всех элементов массивов заканчивающихся на 5 делённых на все элементы массива заканчивающихся на 3. Если тип double , то нужно округлить все значения и вывести элементы которые встречаются в массиве более 1 раза .Если тип char , то нужно отсортировать этот массив и вывести его в порядке убывания.

4)Вашему начальнику понравилась ваша программа но планы немного изменились и теперь ему надо чтобы программа выдавала немного другой результат.

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно прибавить значение текущего элемента прибавить к значению следующего. К примеру при вводе «1 2 3 4 5 6» должно вывести «1 3 6 10 15 21». Если тип double , то нужно вывести сумму самого большого и самого малого элемента массива .Если тип char , то нужно все символы код которых заканчивается на 5 заменить на пробел(код 32).

5)Прошла неделя и данные в вашей программе устарели так как вы уже переловили большую часть преступников. Так как старая программа больше не нужна, то вы решили написать новую так как «Повторение – мать учения!».

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно найти разность суммы всех четных элементов от суммы всех нечетных элементов. Если тип double , то нужно вывести разность самого большого и самого малого элемента массива .Если тип char , то нужно все символы код которых кратен 3 заменить на пробел(код 32).

6)Ваша программа успешно справляется с заданием , но вы решили написать ещё одну чуть посложнее чтобы закрепить материал.

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно найти сумму всех элементов кратным 5. Если тип double , то нужно вывести сумму элементов массива расположенных между минимальным и максимальным элементами массива .Если тип char , то нужно вывести первые три символа(с конца) с нечетным кодом.

7)Программа которую вы написали без проблем заработала на супер компьютере. Пользуясь возможностью вы решили ещё написать одну похожую программу.

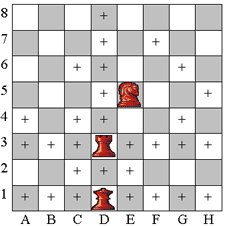
Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно найти сумму всех элементов первая цифра которых равна 3,4,7 или 8. Если тип double , то нужно поменять местами максимальный и минимальный элемент массива. Если тип char , то нужно вывести первые три символа(с конца) с чётным кодом.

8)Вам больше не нужно запоминать кучу информации так как она теперь хранится в компьютере и из-за этого у вас освободилось время которое вы решили потратить на создание похожей программы , но только чуть более сложной.

Ваша задача: Создать шаблон класса который в качестве параметра принимает тип (int,double или char) для класса Array.В классе должен быть массив (динамический или статический на ваш выбор) тип которого передаётся в шаблон в качестве параметра и конструктор в котором происходит заполнение массива необходимым типом данных. Если был передан тип int , то нужно найти сумму всех элементов последняя цифра которых равна 3,4,7 или 8. Если тип double , то нужно прибавить значение текущего элемента прибавить к значению следующего. К примеру при вводе «1.3 4 4.67 3.2 8.9» должно вывести «1.3 5.3 9.97 13.17 22.07». Если тип char , то нужно вывести первые три символа с чётным кодом заменить на пробел(Код 32).

Уровень 3

Устали от сложных задач? Ладно, давайте немного отдохнём и решим простую шахматную задачку.

На шахматной доске 8х8 расположены три фигуры: ферзь, ладья и конь. Требуется определить количество пустых полей доски, которые находятся под боем. Для простоты будем полагать, что фигуры могут «бить» через другие фигуры. Например, в рассмотренной справа ситуации будем считать, что ферзь бьет D5 через ладью.

Ваша задача : Создать класс Chess котором будет необходимое вам для решения данной задачи кол-во полей.Заполнение полей(Если они есть) происходит при помощи конструктора.Так же класс должен содержать метод который найдёт количество пустых полей, которые бьют указанные во входных данных фигуры.  
Входные данные:

Координаты расположения трех фигур: ферзя, ладьи и коня соответственно. Каждая координата состоит из одного английского символа (от A до H) и одной цифры (от 1 до 8).

Выходные данные:

количество пустых полей, которые бьют указанные во входных данных фигуры.

|  |  |
| --- | --- |
| Input | Output |
| D1 D3 E5 | 29 |

**Контрольные вопросы:**

1) В чем в C++ заключается механизм шаблонов классов и функций?

2) Каков общий формат шаблона класса?

3) Перечислите виды параметров шаблона класса.

4) Как создать конкретный экземпляр класса, используя шаблон класса?

5) Как определить метод вне пространства шаблона класса?

6)Чем отличается Шаблон класса от шаблона функции?

7)Какой будет результат выполнения данного кода?  
#include "pch.h"

#include <iostream>

using namespace std;

template<int SIZE = 20>

class Test

{

public:

int a = SIZE;

};

int main()

{

Test<5> obj;

cout << obj.a;

Test<> obj2;

cout << obj2.a;

}

8) Какой будет результат выполнения данного кода?

#include "pch.h"

#include <iostream>

using namespace std;

template<typename T>

class Test

{

public:

T a = 2.9;

};

int main()

{

Test<int> obj;

cout << obj.a<<endl;

Test<double> obj2;

cout << obj2.a;

}

Лабораторная работа №8

Обработка исключительных ситуаций (C++).

*Исключения* - возникновение непредвиденных ошибочных условий в момент работы программы. В то же время исключение - это более общее чем ошибка понятие, так как может возникать и тогда, когда в программе нет ошибок (например, сбой выделения памяти при создании объекта класса). В общем, исключение обозначает особую ситуацию, когда требуется безвозвратно переключить выполнение программы на блок обработки такой ситуации. Выявление особой ситуации производится только программным путём при помощи проверки нормального хода выполнения программы.

Средства обработки ошибочных ситуаций позволяют передать обработку исключений из кода, в котором возникло исключение, некоторому другому программному блоку, который выполнит в данном случае некоторые определенные действия. Таким образом, основная идея данного механизма состоит в том, что функция проекта, которая обнаружила непредвиденную ошибочную ситуацию, которую она не знает, как решить, генерирует сообщение об этом (бросок исключения). А система вызывает по этому сообщению программный модуль, который перехватит исключение и отреагирует на возникшее нештатное событие. Такой программный модуль называют «обработчик» или перехватчик исключительных ситуаций. И в случае возникновения исключения в его обработчик передаётся произвольное количество информации с контролем ее типа. Эта информация и является характеристикой возникшей нештатной ситуации.

Обработка исключений в C++ - это обработка с завершением. Это означает, что исключается невозможность возобновления выполнения программы в точке возникновения исключения.

Для обеспечения работы такого механизма были введены следующие ключевые слова:

**try - проба испытания;**

**catch - перехватить (обработать);**

**throw - бросать.**

Кратко рассмотрим их назначение.

Ключевое слово try открывает блок кода, в котором может произойти ошибка; это обычный составной оператор:

try

{

Код

};

Код содержит набор операций и операторов, который и будет контролироваться на возникновение ошибки. В него могут входить вызовы функции пользователя, которые компилятор также возьмет на контроль.

Среди данного набора операторов и операций обязательно указывают операцию броска исключения throw, которая имеет следующий формат:

**throw выражение',**

где выражение определяет тип информации, которая и описывает исключение (например конкретные типы данных).

Обработчик исключения catch перехватывает информацию:

catch ( тип и параметр)

{

Код

}

Через параметр обработчику передаются данные определенного типа, описывающие обрабатываемое исключение. Код определяет те действия, которые надо выполнить при возникновении данной конкретной ситуации. В C++ используют несколько форм обработчиков. Рассмотренный обработчик получил название параметризованный специализированный перехватчик.

Перехватчик должен следовать сразу же после блока контроля, т.е. между обработчиком и блоком контроля не должно быть ни одного оператора. При этом в одном блоке контроля можно вызывать исключения разных типов для различных ситуаций, поэтому обработчиков может быть несколько. В этом случае их необходимо расположить сразу же за контролирующим блоком последовательно друг за другом.

Кроме того, запрещены переходы как извне в обработчик, так и между обработчиками. Можно воспользоваться универсальным или абсолютным обработчиком: catch (...){ код} где (...) - означают способность данного перехватчика обрабатывать информацию любого типа. Такой обработчик располагают последним в пакете специализированных обработчиков. Тогда, если исключение не будет перехвачено специализированными обработчиками, то будет выполнен последний - универсальный.

Если не возникнет исключение, набор обработчиков будет обойден, т.е. проигнорирован.

Если же исключение было «брошено» при возникновении критической ситуации, то будет вызван конкретный перехватчик при совпадении его параметра с выражением в операторе броска, т.е. управление будет передано найденному обработчику. После выполнения кода вызванного обработчика управление передается оператору, который расположен за последним перехватчиком, или проект корректно завершает работу.

Существенное отличие вызова конкретного обработчика от вызова обычной функции заключается в следующем: при возникновении исключения и передаче управления определенному обработчику система осуществляет вызов всех деструкторов для всех объектов классов, которые были созданы с момента начала контроля и до возникновения исключительной ситуации, с целью их уничтожения.

Блоки try как составные блоки могут быть вложены друг в друга. В случае возникновения исключения в некотором текущем блоке, поиск обработчика последовательно продолжается в блоках, предшествующих уровней вложенности деструкторов. с продолжением вызова Пример обработки исключительных ситуаций. Функция Divide() возвращает частное от деления чисел, принимаемых в качестве аргументов (а, b), если делитель равен нулю (Ь = 0), то возникает исключительная ситуация.

#include "pch.h"

#include <iostream>

using namespace std;

double Divide(double, double);

int main()

{

double a, b, result;

cout << " Input a, b : " << endl;

cin >> a >> b;

try

{

result = Divide(a, b);

cout << " Nonnal Work " << endl;

cout << " a = " << a << ", b = " << b << endl;

cout << " Ansver : " << result << endl;

}

catch (double z) {

cout << " Division by zero... " << endl;

cout << " b = " << z << endl;

}

}

double Divide(double a1, double b1) {

if (b1 == 0) throw b1;

return a1 / b1;

}

Результаты выполнения программы, введя значения 1 и 2, получим:

Input a, b : 1 2

Normal Work

а=1, b = 2

Ansver : 0.5

А при вводе значений 1 и 0, получим:

Input a, b : 1 0

Division by zero...

b =0

Оператор throw сигнализирует об исключительном событии (попытку деления на ноль) и генерирует объект исключительной ситуации. В данном случае он находится внутри функции Divide(). Объект перехватывается обработчиком catch. Этот процесс, как уже известно, и называется вызовом исключительной ситуации. В рассмотренном примере исключительная ситуация имеет форму вещественной переменной - делителя.

Упражнение: Создать пользовательский класс в котором будут поля (название и кол-во полей будут даны в индивидуальном задании в соответствии с заданием) , конструктор для заполнения этих полей и метод выполняющий индивидуальное задание.

Уровень 1

Варианты:  
  
1)Компания «Baby» попросила вас разработать мини-игру для детей. Правила игры до безобразия просты: родитель вводит какое то число , оно высвечивается на экране , а ребёнок должен ввести это число.Так как игра детская , то родитель не может ввести числа меньше нуля и больше 100 при вводе числа выходящего за этот диапазон должно вывестись сообщение об ошибке и родителя должны снова попросить ввести число и так будет до тех пор , пока родитель не введёт число принадлежащее диапазону [0…100].После ввода числа за дело принимается ребёнок. При вводе неправильного числа ему выводится соответствующее сообщение и просят ввести число и так до тех пор , пока число не будет введено правильно.  
Ваша задача: Создать класс Game в котором будет поле number хранящее в себе число введённое родителем. Поле number заполняется при помощи конструктора с параметром. В классе так же должен быть метод Check() который будет проверять число введённое ребёнком и в случае если введено не то число вызывать ошибку.

2)Наконец то вы нашли кафедру где ваши знания по настоящему нужны и ценятся. Теперь чтобы завоевать доверия среди коллег вы попросили дать вам какое ни будь задание и такое задание нашлось! Вам нужно написать программу которая проверяет является ли число простым. В случае если число не простое должна быть вызвана ошибка.

Ваша задача: Создать класс PrimeNumber в котором будет поле number для хранения будущего простого числа. Поле number заполняется при помощи метода Check() которое проверяет является ли число простым и если да , то ложит это число в поле number , а если нет , то вызывает ошибку.

3)У продавца в аптеке сломался калькулятор и теперь ему сложно считать сдачу для клиентов. Давайте поможем фармацевту и напишем для него программу которая будет даже лучше простого калькулятора. Ведь помимо простого счёта наша программа и будет оповещать продавца в случае если клиент дал денег меньше чем нужно.  
Ваша задача: Создать класс Calculator в котором будет поля Price и MoneyPaid. Поля Price и MoneyPaid заполняются при помощи конструктора. Так же в классе должен быть метод Count() который выведет разность MoneyPaid – Price в случае если разность будет положительной.Если разность получилось отрицательной , то Count() должен вызвать ошибку.

4)Ваш начальник решил провести благотворительную акцию под названием «Помоги уставшему начальнику». Все деньги собранные благодаря этой акции пойдут на отпуск вашего начальника чтобы он мог наконец-таки отдохнуть. Вас же он попросил написать программу которая будет проверять сколько каждый сотрудник банка пожертвовал на благотворительность и в случае если он пожертвовал слишком мало , должно выдать ошибку!

Ваша задача: Создать класс Vacation в котором будет поле number. Поле number заполняется при помощи метода Check() которое проверяет является ли число > 1000 и если да , то ложит это число в поле number , а если нет , то вызывает ошибку.

5)Скоро будет день ВДВ и вам нужно повысить кол-во сотрудников в каждом районе. Но у вас нет времени на составление графика патрулирования поэтому вы назначили человека который может сделать это за вас. Вроде бы всё хорошо если бы не одно но , у этого сотрудника не очень с математикой , а кол-во людей отправленных на патруль должно быть кратно 13,17,11 или 7(Не спрашивайте почему, так надо). Как старший по званию вы не можете оставить младшего сотрудника в беде и поэтому решили создать для него программу которая будет проверять число на кратность этим числам , а в случае если введённое число не будет кратно 13,17,11 или 7 будет выдавать ошибку.

Ваша задача: Создать класс Patrol в котором будет поле number. Поле number заполняется при помощи метода Check() которое проверяет является ли число кратным 13,17,11 или 7 и если да , то ложит это число в поле number , а если нет , то вызывает ошибку.

6)Сегодня день утверждения городского бюджета , а это значит что к вам будет приходить куча людей и просить денег на то-сё. Так как людей очень много вы решили сэкономить своё время и написать программу в которую бы люди вводили кол-во необходимых им денег и если они просят слишком много , то программа должна вызывать ошибку и попросить их умерять аппетиты.

Ваша задача: Создать класс Budget в котором будет поле number. Поле number заполняется при помощи метода Check() которое проверяет является число < 1000 и если да , то ложит это число в поле number , а если нет , то вызывает ошибку.

7)Как вы помните вы поставили у себя на заводе супер компьютер который теперь отвечает за производство на фабрике. Сотруднику теперь просто достаточно выбрать товар который нужно произвести и его кол-во , вот только на фабрике завёлся шутник который вводит в компьютер отрицательные значения из-за чего компьютер начинает дымится, искриться и ругаться! Пока он окончательно не отключился вам нужно написать программу которая бы проверяла введённое в компьютер число.

Ваша задача: Создать класс Computer в котором будет поле number. Поле number заполняется при помощи метода Check() которое проверяет является число > 0 и если да , то ложит это число в поле number , а если нет , то вызывает ошибку.

8)В вашем отеле вы решили открыть комнату отдыха чтобы родители могли оставлять там детей пока они отдыхают на пляже. Ну и чтобы дети проводили время с пользой вы решили написать программу в которую бы дети вводили 2 числа , а программа бы возвращала частное суммы этих двух чисел на их разность.

Ваша задача: Создать класс Hotel в котором будет поле number1 и number2. Поле number1 и number2 заполняется при помощи метода Check() которое проверяет не возникает ли ошибки при делении суммы number1 и number2 на их разность и если number1 - number2 == 0 то вызывается ошибка деления на нуль.

Уровень 2

Варианты:

1) Компании «Baby» понравилась разработанная вами игра и они попросили написать вас новую игру. Игра называется «Угадай число». Правила игры очень просты: компьютер загадывает любое число в диапазоне [-50 , 50] . Пользователь вводит число и если введённое число не совпадает с загаданным , то вызывается ошибка. Если пользователь ввел число меньше загаданного , то выводится сообщение о том , что введённое число меньше загаданного. Если введённое число больше загаданного , то выводится сообщение о том , что введённое число больше загаданного. Если введенное число выходит за границы диапазона [-50 , 50] , то выводится сообщение о том , что данное число не могло быть загадано и диапазон в котором лежит загаданное число [-50 , 50].

Ваша задача: Создать класс Game в котором будет поле number хранящее в сгенерированное число. Поле number заполняется при помощи конструктора без параметров в котором происходит генерация случайного значения. В классе так же должен быть метод Check() который будет проверять число введённое пользователем на принадлежность к диапазону [-50 , 50] и на близость к загаданному числу.

2)Вы прекрасно справились с предыдущем заданием , получайте следующее!  
Ваша задача:Создать класс Palindromos в котором будет поле number для хранения будущего палиндрома. Поле number заполняется при помощи метода Check() которое проверяет является ли число палиндромом и если да , то ложит это число в поле number , а если нет , то вызывает ошибку.

3)В честь дня матиматики у вас проходит акция! Если |MoneyPaid – Price| является простым числом , то ошибка не вызывается. Вам нужно срочно переделать вашу программу под это условие.

Ваша задача: Создать класс Calculator в котором будет поля Price и MoneyPaid. Поля Price и MoneyPaid заполняются при помощи конструктора. Так же в классе должен быть метод Count() который выведет разность MoneyPaid – Price в случае если разность будет положительной или вывести |MoneyPaid – Price| если маодуль разности является простым числом.Если разность получилась отрицательной , то Count() должен вызвать ошибку.

4)После долгих раздумий ваш начальник решил не брать с собой мелочь в отпуск и поэтому попросил всех жертвовать суммы кратные 5. Чтобы сотрудники не забыли об этом нужно модифицировать программу.

Ваша задача: Создать класс Vacation в котором будет поле number. Поле number заполняется при помощи метода Check() которое проверяет является ли число > 1000 и кратно ли оно 5 и если да , то ложит это число в поле number , а если нет , то вызывает ошибку.

5)У вас в городе решили объеденить празднование дней всех видов войск в один день из-за чего в городе может начаться полный хаос.Чтобы предотвратить эти беспорядки кол-во людей отправленных в патруль должно быть не только кратно 13,17,11 или 7 , а ещё и сумма цифр этого числа должно быть кратно 2 или 3 (Не спрашивайте почему - это поможет 100%).

Ваша задача: Создать класс Patrol в котором будет поле number. Поле number заполняется при помощи метода Check() которое проверяет является ли число кратным 13,17,11 или 7 и сумма цифр этого числа кратно 2 или 3. И если да, то ложит это число в поле number , а если нет , то вызывает ошибку.

6)В честь дня математика вы решили все же утвердить буджеты которые выходят за рамки бюджета если сумма цифр запрашиваеммой суммы будет простым числом.  
Ваша задача: Создать класс Budget в котором будет поле number. Поле number заполняется при помощи метода Check() которое проверяет является число < 1000 и если да , то ложит это число в поле number , а если нет , то вызывает ошибку. А если число >1000 но сумма его цифр является простым числом , то всё равно ложим это число в number.

7)Ваша программа помогла компьютеру не взорваться от ввода оотрицательных чисел и теперь вы решили добавить ещё одно условие.Дело в том , что у вас 5 собирающих конвееров и при вводе числа не кратного 5 конвееры простаивают , а это не порядок. Именно поэтому вы решили исключить из ввода ещё и числа не кратные 5.

Ваша задача: Создать класс Computer в котором будет поле number. Поле number заполняется при помощи метода Check() которое проверяет является число > 0 и кратно ли оно 5 и если да , то ложит это число в поле number , а если нет , то вызывает ошибку.

8)Вы решили научить детей возведению чисел в квадрат и извлечению квадратного корня из числа. Так как извлекать корень из 5 для детей сложновато , вы решили написать программу в которую ребёнок бы вводил число , а она говорила бы можно ли извлеч из этого числа квадратный корень, в противном случае вызывала бы ошибку.

Ваша задача: Создать класс Hotel в котором будет поле number. Поле number заполняется при помощи метода Check() которое проверяет можно ли получить целое число при извлечении квадратного корня из number и если да , то ложит это число в поле number , а если нет , то вызывает ошибку.

Уровень 3

Студент «Здесь может быть ваше имя» являет собой классический пример лентяя. На занятия он практически не ходит, и только в конце семестра появляется в университете и сдает ”хвосты”. Его заветная мечта: найти такой день, когда можно будет сдать сразу все долги. У него есть расписание работы преподавателей, из которого точно известно, с какого и по какой день месяца каждый преподаватель ежедневно будет доступен.

Помогите «Здесь может быть ваше имя» написать программу, которая по расписанию будет определять, сможет ли «Здесь может быть ваше имя» сдать все долги за один день или нет.

Ваша задача : Создать класс Student в котором будет 1 поле N для хранения количество предметов, которые нужно сдать.Заполнение полей происходит при помощи конструктора.Так же класс должен содержать метод который выведите «YES», если возможно встретить всех преподавателей за один день, или «NO», если это сделать невозможно. Заполнение N строк, состоящих из двух чисел A и B, задающих отрезок работы очередного преподавателя можете сделать в методе для решения поставленной задачи или в конструкторе (На ваше усмотрение).

.  
Входные данные:

Натуральное число N – количество предметов, которые нужно сдать «Здесь может быть ваше имя» (N ≤ 100). Далее идет N строк, каждая из которых состоит из двух чисел A и B, задающих отрезок работы очередного преподавателя (1 ≤ A ≤ B ≤ 31).

Выходные данные:

Выведите «YES», если возможно встретить всех преподавателей за один день, или «NO», если это сделать невозможно.

|  |  |
| --- | --- |
| Input | Output |
| 1  1 2 | YES |
| 2  1 2  3 4 | NO |
| 3  1 8  3 5  4 9 | YES |

**Контрольные вопросы:**

1) Что называют исключением?

2) Что такое «блок с контролем»?

3) Дайте характеристику обработчику исключений. Какие бывают виды обработчиков?

4) Какие правила налагаются на соотношения между блоком контроля и обработчиками?

5) Чем отличается вызов обработчика от вызова обычной функции?

6)Что делает Оператор throw?

7)Какой будет результат выполнения данного кода?  
#include "pch.h"

#include <iostream>

using namespace std;

void Check(double b)

{

if (b == 0)

{

throw exception("Division by zero");

}

}

int main()

{

double a = 4.5, b = 5, c = 0;

try

{

Check(b);

cout << a / b<<endl;

Check(c);

cout << a / c;

}

catch (const exception &ex)

{

cout << endl << ex.what();

}

}

8)Сколько **catch** можно описать для одного **try?**

.

Работа окончена!!!  
Ура!