1. Пакуем в namespace
2. Const функции расставить. Хотя бы гетеры
3. Деструктор

Class myclass{

Public:

Myclass(int val): val{5} {

Val = 6; //constructor

}

Private:

Int val{7};

}

//Сначало val присвоится 7, потом 5, потом 6. В большинстве случаев 7 и 5 присваиваться не будут, т.к. компилятор оптимизирует и уберет лишние присваивания.

Operator<<(std::stream &out, const myclass& class1) { // перегрузка потока

Std::cout << “поле1: “ << class1.field1 << “ поле2: “ << class1.field2;

return out; }

Class s{

//нельзя менять arg (его значения, поля и т.п.)

но поля класса s с помощью my\_function изменить можно.

void my\_function(const MyType & arg); //применяется. чтобы избежать копирования

void my\_function(MyType & arg) const; //тут можно менять arg, но нельзя изменить поля класса s

}

const int\*t; // данные по указателю нельзя менять

int\* const t; // указатель нельзя менять

>- какие библиотеки используют?  
Нативные под разные ОС (WPF или WInUI для windows, Qt(KDE) или GTK(Gnome) под Linux Cocoa под MacOS  
Кросс-платформенные: Qt, GTK  
Кросс-языковые (часть отвечающая за UI на одном языке, а основная логика приложения на С++): Flutter, Electron, etc  
  
>- какие есть со свободной лицензией?  
Я так понимаю имеется в виду свободная для коммерческого использования?  
GTK выпускается под LGPL (там есть несколько нюансов, но если соблюдать условия LGPL, то можно выпускать окммерческое ПО без необходимости распространять его исходные коды)  
  
Flutter под BSD-3  
  
>- какие платные, но стоят не очень дорого с возможностью коммерческого использования?  
Qt. Но сейчас не продается для РФ  
Для WPF и WinUI нужно будет приобрести Visual Studio Prof, Для Cocoa кажется достаточно подписать на Apple Developer Account.

Вместе с конструктором move использовать noexept, иначе он вместо мув будет делать копи на всякий случай, чтоб откатить если произойдет исключение