پلاگین چیست ؟ پلاگین یک dynamic library برای گسترش برنامه است که در load run_time میشود. مشخصه اصلی یک پلاگین کلاس ابسترکتی به اسم interface است که تمام متد های آن از نوع pure virtual است
کاربرد پلاگین : ما برای گسترش یک برنامه و افزودن ابزار های مختلف به آن از پلاگین استفاده میکنیم که بیار آن را نوشته و آن را در برنامه های مختلف استفاده میکنیم
مقایسه پلاگین و لایبرری : qt plugin ها در shared library ذخیره میشن (DLL) که مزیتی نسبت به Qlibrary هایی که در shared library ذخیره میشن دارن :
۱:کیوتی plugin loader بررسی میکند که یک پلاگین با ورژن کیوتی برنامه ما مرتبط است یا نه
۲:به جای اینکه شمارو مجبور کنه که یک function c را به صورت دستی resolve کنید , دسترسی مستقیم به یک ابجکت component ریشه (()instance) را فراهم میکنه

نکات کلاس اینتر فیس
برای اینکه به query این امکان رو بدیم در run_time یک interface را در پلاگین پیاده سازی کند از Q_DECLARE_INTERFACE استفاده میکنیم
ماکرو ()Q_DECLARE_INTERFACE در interface دو ورودی میگیرد اولی اسم interface و دومی یک استرینگی است که اینتر فیس را به روش منحصر به فردی شناسایی میکند
(Java package name" syntax)

نکته: اگر بعداً اینترفیس ها را تغییر دهیم، باید از استرینک دیگری برای شناسایی اینترفیس جدید استفاده کنیم. در غیر این صورت، برنامه ممکن است از کار بیفتد. بنابراین ایده خوبی است که ورژن را در استرینگ اضافه کنید.

نکات کلاس پلاگین
در کلاس پلاگین, ماکرو Q_INTERFACE ضروریه به moc بگه که کلاس های base اینترفیس های پلاگین هستند
ماک moc: (meta_object_compiler)
ابزار ماک فایل های هدر c++ را میخواند اگر یک یا چند class declartion را پیدا کند که حاوی Q_Object macro باشد یک فایل سورس c++ حاوی کد meta_object برای آن کلاس ها تولید میکند همچنین کد متا آبجکت نیاز به مکانیسم signal_slot , run_time type info و ویژگی های dynamic سیستم را دارد
فایل های سورس c++ که توسط moc ساخته شدند باید کامپایل و به implementation of the class وصل بشن