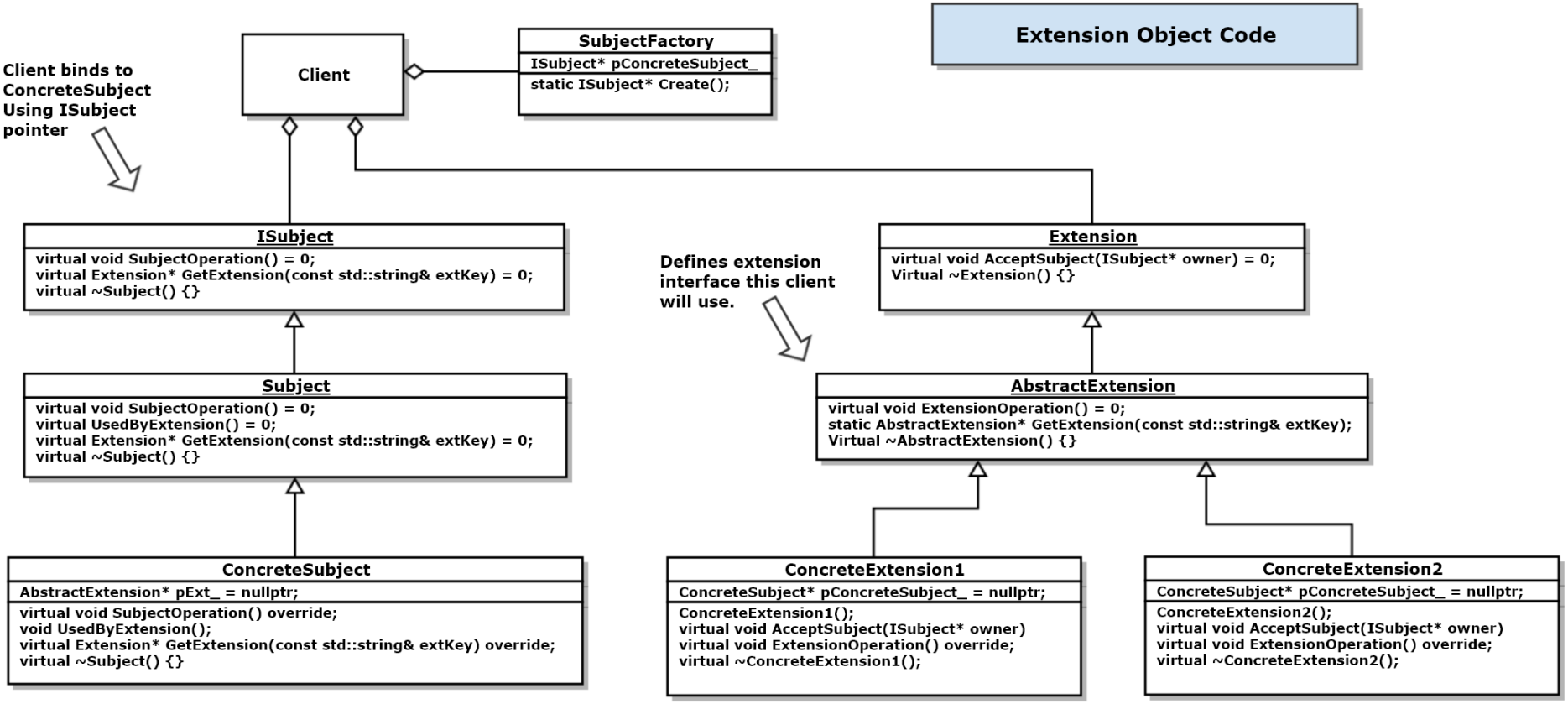
Extension Object Code
Client
SubjectFactory
ISubject* pConcreteSubject_
static ISubject* Create();
Client binds to ConcreteSubject Using ISubject pointer
ISubject
virtual void SubjectOperation() = 0;
virtual Extension* GetExtension(const std::string& extKey) = 0;
virtual ~Subject() {}
Extension
virtual void AcceptSubject(ISubject* owner) = 0;
Virtual ~Extension() {}
Defines extension interface this client will use.
Subject
virtual void SubjectOperation() = 0;
virtual UsedByExtension() = 0;
virtual Extension* GetExtension(const std::string& extKey) = 0;
virtual ~Subject() {}
AbstractExtension
virtual void ExtensionOperation() = 0;
static AbstractExtension* GetExtension(const std::string& extKey);
Virtual ~AbstractExtension() {}
ConcreteSubject
AbstractExtension* pExt_ = nullptr;
virtual void SubjectOperation() override;
void UsedByExtension();
virtual Extension* GetExtension(const std::string& extKey) override;
virtual ~Subject() {}
ConcreteExtension1
ConcreteSubject* pConcreteSubject_ = nullptr;
ConcreteExtension1();
virtual void AcceptSubject(ISubject* owner)
virtual void ExtensionOperation() override;
virtual ~ConcreteExtension1();
ConcreteExtension2
ConcreteSubject* pConcreteSubject_ = nullptr;
ConcreteExtension2();
virtual void AcceptSubject(ISubject* owner)
virtual void ExtensionOperation() override;
virtual ~ConcreteExtension2();