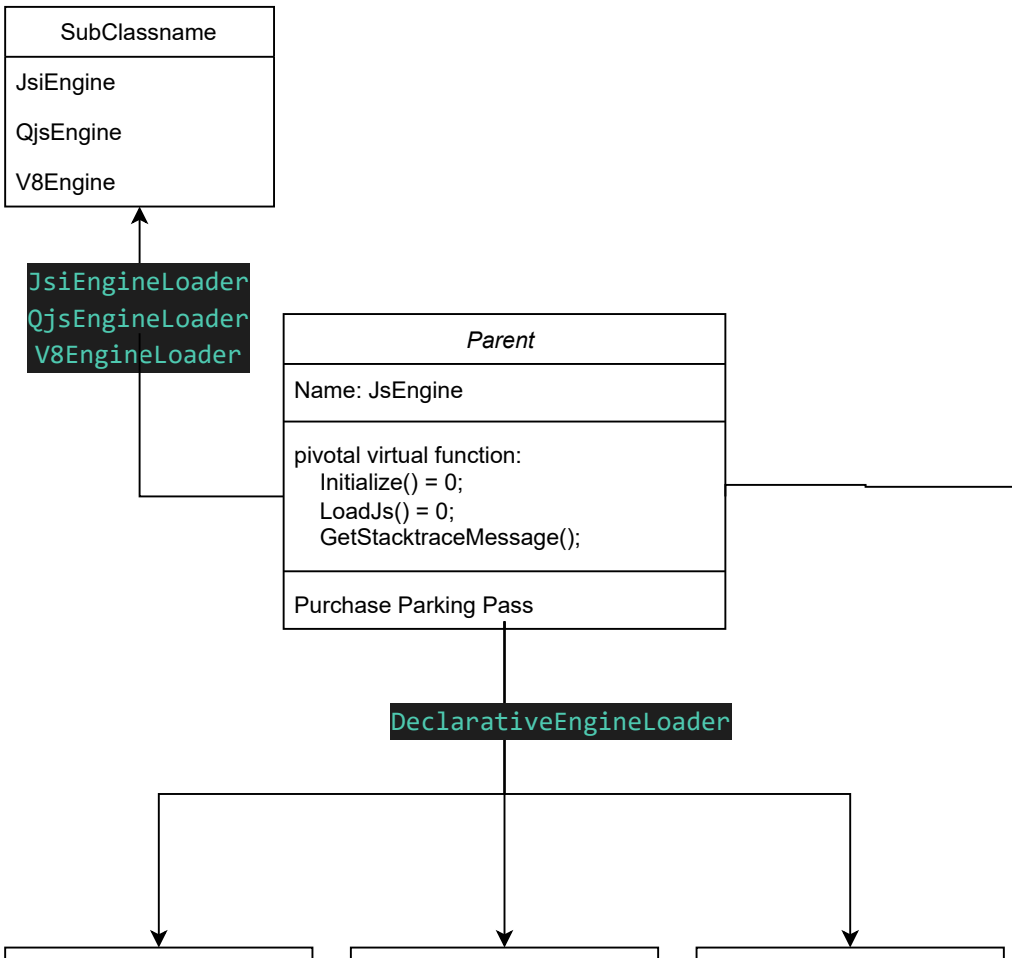
arkui_ace_engine



InitJsEnv
包含有与堆栈打印相关的关键
runtime_->RegisterUncaughtExceptionHandler(JsiE
注册出现崩溃等情况的监听回调

Function Description

Initialize(const RefPtr<FrontendDelegate>& delegate)

初始化事件，包含注册监听进程的堆栈日志，执行状态
建接口，全局事件监听接口等
和本次内容相关为，初始化监听的内容，关键代码：
engineInstance_->InitJsEnv(IsDebugVersion(), GetEx

LoadJs(const std::string& url, const RefPtr<JsAcePag
bool isMainPage)
捆绑运行进程时执行代码与源代码之间的映射

GetStacktraceMessage();

字面意义，提供给外部调用的堆栈打印接口，内部调用

piva

1:
RefPtr<JsEngine> DeclarativeEngineLoade
{
#ifdef USE_V8_ENGINE
return AceType::MakeRefPtr<V8Declarat
#endif

#ifdef USE_QUICKJS_ENGINE
return AceType::MakeRefPtr<QJSDeclara
#endif

#ifdef USE_ARK_ENGINE
return AceType::MakeRefPtr<JsiDeclarati
#endif
}

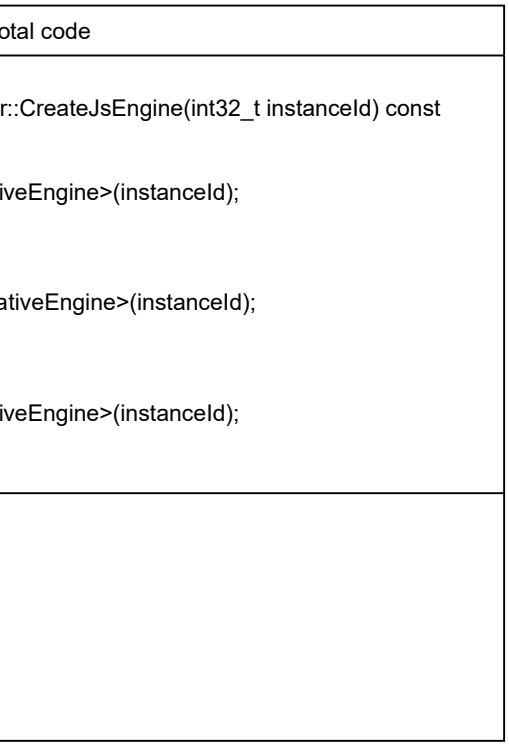
JsiBaseUtils

GenerateSummaryBody(std::shared_ptr<JsValue> error,
std::shared_ptr<JsRuntime> runtime)
异常状态（崩溃等）时的堆栈信息打印

TransSourceStack(RefPtr<JsAcePage> runningPage,
const std::string& rawStack)
正常的堆栈信息打印

JsiDumpSourceFile(const std::string& stackStr, const std::string& pageUrl,
const RefPtr<RevSourceMap>& pageMap,
const RefPtr<RevSourceMap>& appMap, const AceType *data)
根据映射打印


键代码：
BaseUtils::ReportJsErrorEvent);
)
态，任务发放线程创
xtraNativeObject());
ge>& page
用 TransSourceStack


otal code
r::CreateJsEngine(int32_t instanceId) const
iveEngine>(instanceId);
ativeEngine>(instanceId);
iveEngine>(instanceId);