

```

graph LR
    A[普通Java类] -- "?" --> B[beanDefinition]
    B -- "?" --> C[spring bean]

```

```

graph TD
    Start(SpringApplication.run()) --> PrepEnv[prepareEnvironment  
configureEnvironment  
printBanner]
    PrepEnv --> CreateCtx[createContext]
    CreateCtx --> PrepCtx[prepareContext]
    PrepCtx --> RefreshCtx[refreshContext]
    RefreshCtx --> AfterCtx[afterContext]
    AfterCtx --> StartListeners[listeners.started  
callRunners]
    RefreshCtx --> RefreshMethod[refresh()]
    RefreshMethod --> PrepRefresh[prepareRefresh]
    PrepRefresh --> ObtainFactory[obtainFreshBeanFactory  
prepareBeanFactory]
    ObtainFactory --> PostProcessFactory[postProcessBeanFactory]
    PostProcessFactory --> InvokePostProcessors[invokeBeanFactoryPostProcessors]
    InvokePostProcessors --> RegisterPostProcessors[registerBeanPostProcessors]
    RegisterPostProcessors --> InitMsgSrc[initMessageSource]
    InitMsgSrc --> InitAppEventM[initApplicationEventMulticaster]
    InitAppEventM --> OnRefresh[onRefresh]
    OnRefresh --> RegisterListeners[registerListeners]
    RegisterListeners --> FinishBeanFactoryInit[finishBeanFactoryInitialization]
    FinishBeanFactoryInit --> FinishRefresh[finishRefresh]
  
```

The flowchart illustrates the sequence of operations performed by `SpringApplication.refresh()`. It begins with `SpringApplication.run()`, which leads to `prepareEnvironment`, `configureEnvironment`, and `printBanner`. This is followed by `createContext`, `prepareContext`, and `refreshContext`. The `refreshContext` step then triggers the `refresh()` method. The `refresh()` method proceeds through `prepareRefresh`, `obtainFreshBeanFactory`, `prepareBeanFactory`, `postProcessBeanFactory`, `invokeBeanFactoryPostProcessors`, `registerBeanPostProcessors`, `initMessageSource`, `initApplicationEventMulticaster`, `onRefresh`, `registerListeners`, `finishBeanFactoryInitialization`, and finally `finishRefresh`.

```
PostProcessorRegistrationDelegate.  
    invokeBeanFactoryPostProcessors
```

