```
#pragma once
#include "Arp/System/Core/Arp.h"
#include "Arp/Plc/Commons/Esm/ProgramBase.hpp"
#include "Arp/System/Commons/Logging.h"
#include "BlinkComponent.hpp"

namespace Blink
{

using namespace Arp;
using namespace Arp::System::Commons::Diagnostics::Logging;
using namespace Arp::Plc::Commons::Esm;

//#program
//#component(Blink::BlinkComponent)
class BlinkProgram : public ProgramBase, private Loggable<BlinkProgram>
{
public: // typedefs

public: // construction/destruction
    BlinkProgram(Blink::BlinkComponent& blinkComponentArg, const String& name);
    BlinkProgram(const BlinkProgram& arg) = delete;
    virtual ~BlinkProgram() = default;

public: // operators
    BlinkProgram&  operator=(const BlinkProgram& arg) = delete;

public: // properties

private:

    int count;

public: // operations
    void    Execute() override;

public: /* Ports
           =====
           Ports are defined in the following way:
           //#port
           //#attributes(Input|Retain)
           //#name(NameOfPort)
           boolean portField;

           The attributes comment define the port attributes and is optional.
           The name comment defines the name of the port and is optional. Default is
the name of the field.
        */

    //#port
```

```
    //#attributes(Output)
    boolean blink;

private: // fields
    Blink::BlinkComponent& blinkComponent;


};

////////////////////////////////////////////////////////////////////////////
// inline methods of class ProgramBase
inline BlinkProgram::BlinkProgram(Blink::BlinkComponent& blinkComponentArg, const
String& name)
: ProgramBase(name)
, blinkComponent(blinkComponentArg)
{
}

} // end of namespace Blink
```