```
//
// Programmer:    Craig Stuart Sapp <craig@ccrma.stanford.edu>
// Creation Date: Fri May 12 08:47:02 PDT 2006
// Last Modified: Wed Jun 21 08:29:27 PDT 2006 (subclassed to MazurkaPlugin)
// Filename:      MzSpectrogramHost.h
// URL:           http://sv.mazurka.org.uk/include/MzSpectrogramHost.h
// Documentation: http://sv.mazurka.org.uk/MzSpectrogramHost
// Syntax:        ANSI99 C++; vamp 0.9 plugin
//
// Description:   Demonstration on how to parse host frequency data.
//

#ifndef _MZSPECTROGRAMHOST_H_INCLUDED
#define _MZSPECTROGRAMHOST_H_INCLUDED

#include "MazurkaPlugin.h"  // Mazurka plugin interface for Sonic Visualiser


class MzSpectrogramHost : public MazurkaPlugin {

   public:

   // plugin interface functions:

                    MzSpectrogramHost       (float samplerate);
      virtual      ~MzSpectrogramHost       ();

      // required polymorphic functions inherited from PluginBase:
      std::string   getName                 (void) const;
      std::string   getMaker                (void) const;
      std::string   getCopyright            (void) const;
      std::string   getDescription          (void) const;
      int           getPluginVersion        (void) const;

      // optional parameter interface functions:
      ParameterList getParameterDescriptors (void) const;

      // required polymorphic functions inherited from Plugin:
      InputDomain   getInputDomain          (void) const;
      OutputList    getOutputDescriptors    (void) const;
      bool          initialise              (size_t channels,
                                             size_t stepsize,
                                             size_t blocksize);
      FeatureSet    process                 (float **inputbufs,
                                             Vamp::RealTime timestamp);
      FeatureSet    getRemainingFeatures    (void);
      void          reset                   (void);

      // optional polymorphic functions from Plugin:
      // size_t     getPreferredStepSize    (void) const { return 0; }
      // size_t     getPreferredBlockSize   (void) const { return 0; }
      // size_t     getMinChannelCount      (void) const { return 1; }
      // size_t     getMaxChannelCount      (void) const { return 1; }

   // non-interface functions and variables:

   private:

      int    mz_minbin;      // minimum spectral bin to display
      int    mz_maxbin;      // maximum spectral bin to display

};

#endif // _MZSPECTROGRAMHOST_H_INCLUDED
```