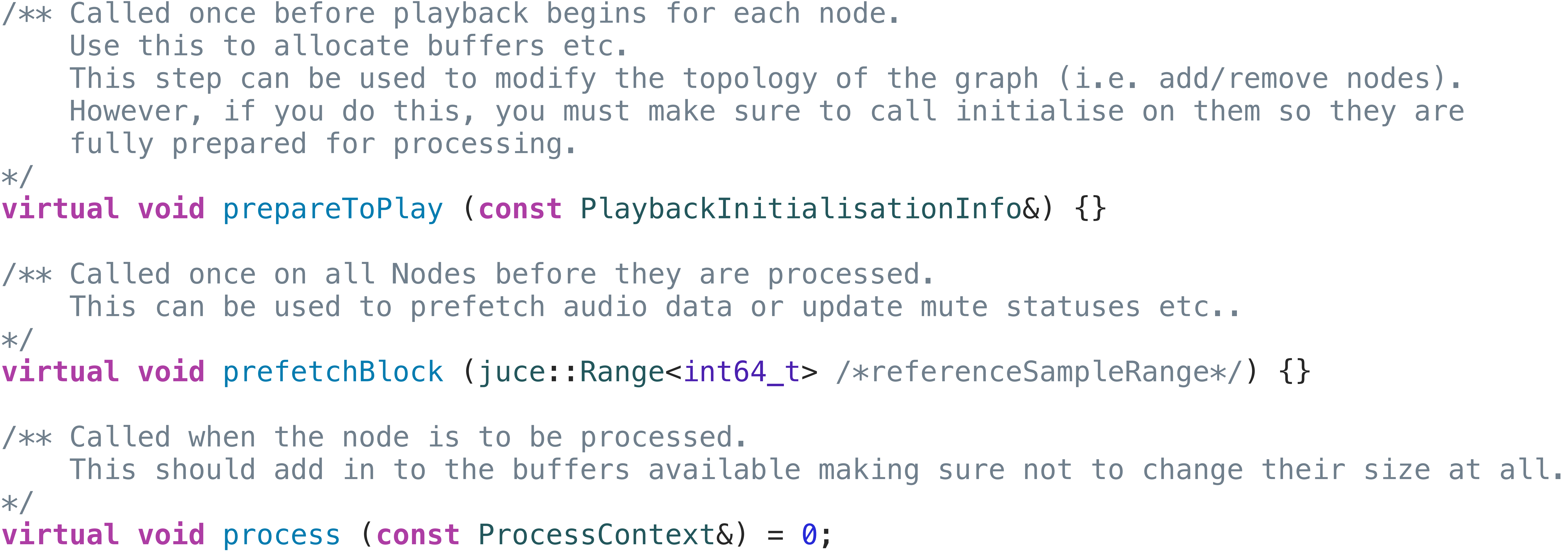
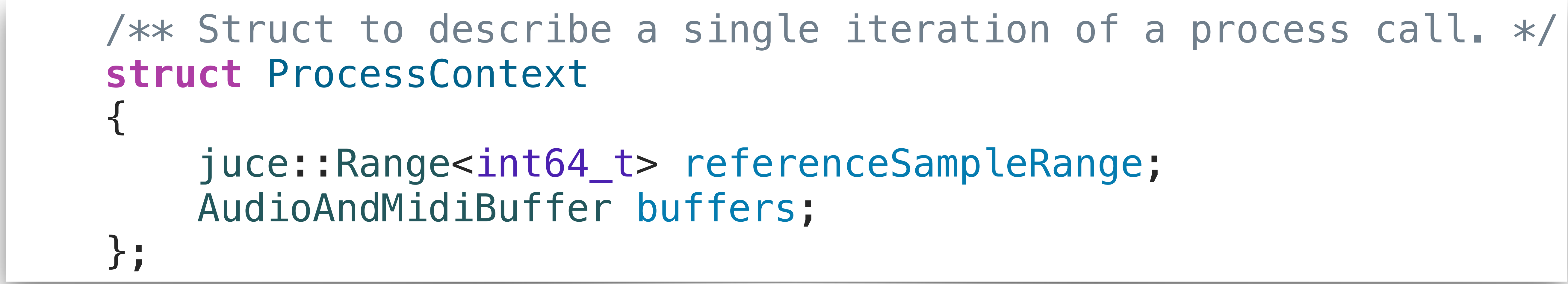
Node: Protected Virtual Methods



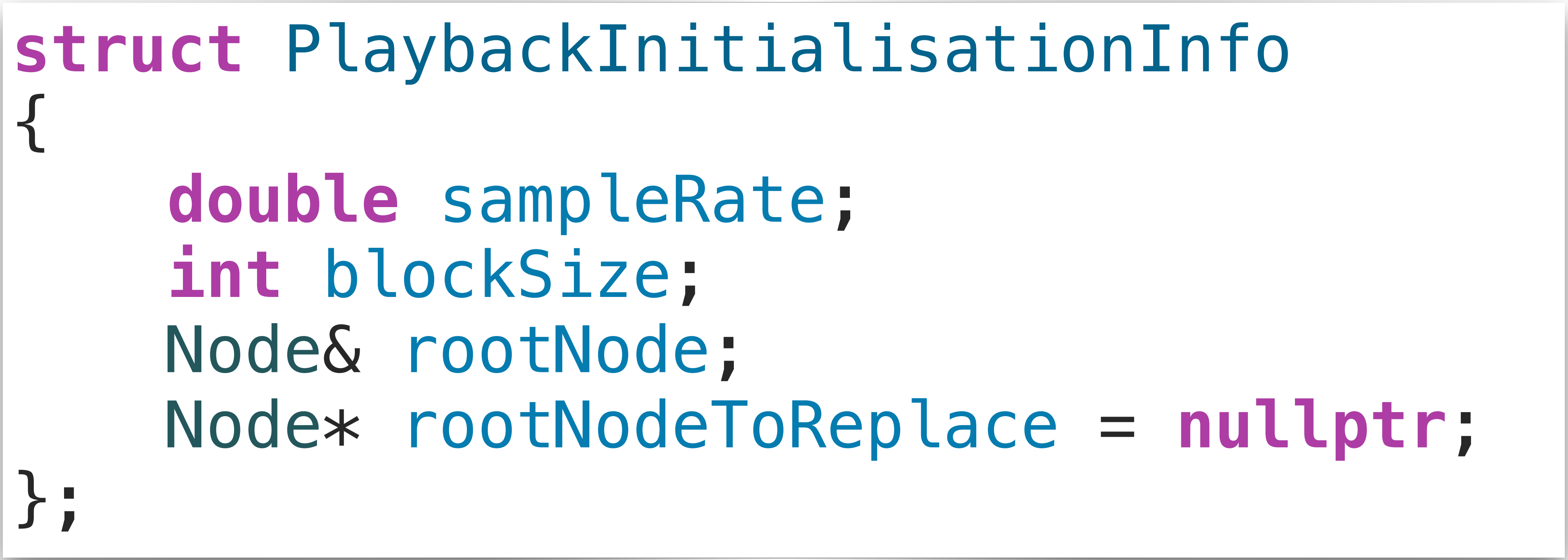
```
/** Called once before playback begins for each node.
    Use this to allocate buffers etc.
    This step can be used to modify the topology of the graph (i.e. add/remove nodes).
    However, if you do this, you must make sure to call initialise on them so they are
    fully prepared for processing.
*/
virtual void prepareToPlay (const PlaybackInitialisationInfo&) {}

/** Called once on all Nodes before they are processed.
    This can be used to prefetch audio data or update mute statuses etc..
*/
virtual void prefetchBlock (juce::Range<int64_t> /*referenceSampleRange*/) {}

/** Called when the node is to be processed.
    This should add in to the buffers available making sure not to change their size at all.
*/
virtual void process (const ProcessContext&) = 0;
```



```
    /** Struct to describe a single iteration of a process call. */
    struct ProcessContext
    {
        juce::Range<int64_t> referenceSampleRange;
        AudioAndMidiBuffer buffers;
    };
```



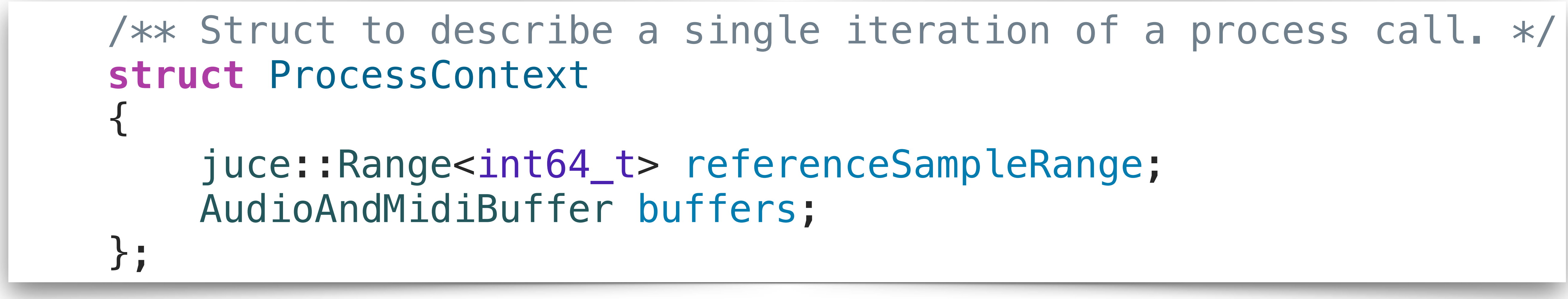
```cpp
struct PlaybackInitialisationInfo
{
    double sampleRate;
    int blockSize;
    Node& rootNode;
    Node* rootNodeToReplace = nullptr;
};
```

Node: Protected Virtual Methods

```
/** Called once before playback begins for each node.
    Use this to allocate buffers etc.
    This step can be used to modify the topology of the graph (i.e. add/remove nodes).
    However, if you do this, you must make sure to call initialise on them so they are
    fully prepared for processing.
*/
virtual void prepareToPlay (const PlaybackInitialisationInfo&) {}

/** Called once on all Nodes before they are processed.
    This can be used to prefetch audio data or update mute statuses etc..
*/
virtual void prefetchBlock (juce::Range<int64_t> /*referenceSampleRange*/) {}

/** Called when the node is to be processed.
    This should add in to the buffers available making sure not to change their size at all.
*/
virtual void process (const ProcessContext&) = 0;
```



```
/** Struct to describe a single iteration of a process call. */
struct ProcessContext
{
    juce::Range<int64_t> referenceSampleRange;
    AudioAndMidiBuffer buffers;
};
```

Node Summary