


```
MidiList& MidiClip::getSequenceLooped() const
{
    if (! isLooping())
        return getSequence();

    if (cachedLoopedSequence == nullptr)
        cachedLoopedSequence = createSequenceLooped (getSequence());

    return *cachedLoopedSequence;
}
```

```
class MidiClip : public Clip
{
private:
    mutable std::unique_ptr<MidiList> cachedLoopedSequence;
```



```

class MidiClip : public Clip
{
private:
    mutable std::unique_ptr<MidiList> cachedLoopedSequence;

MidiList& MidiClip::getSequenceLooped() const
{
    if (! isLooping())
        return getSequence();

    if (cachedLoopedSequence == nullptr)
        cachedLoopedSequence = createSequenceLooped (getSequence());

    return *cachedLoopedSequence;
}

```

Techniques for Optimisation

5. Reducing memory footprint