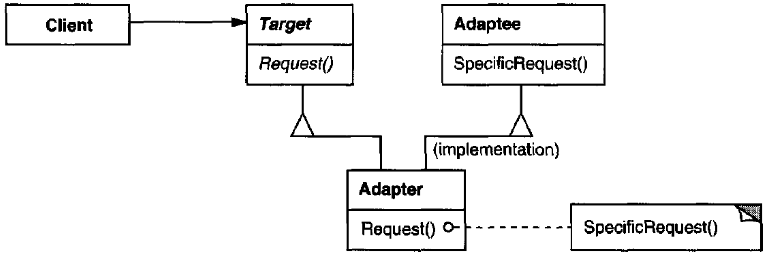
Design Patterns: Adapter

Glue all the things.

Jeremy Murphy

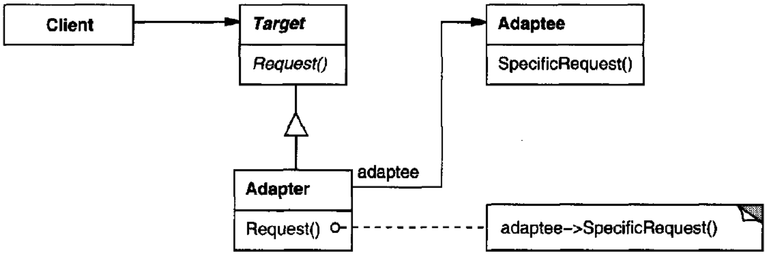
March 13, 2018

Class adapter



Adapter takes Target's interface and Adaptee's implementation.

Object adapter



Instead of inheriting from Adaptee, Adapter stores a reference to an instance. This instance can be reassigned.

Adaptation through partial specialization: Boost.Geometry

```
template <typename CoordinateType, std::size_t DimensionCount>
struct tag<boost::array<CoordinateType, DimensionCount> >
: detail::boost_array_tag<boost::is_arithmetic<CoordinateType>::value> {};


template <typename CoordinateType, std::size_t DimensionCount>
struct coordinate_type<boost::array<CoordinateType, DimensionCount> >
{
    typedef CoordinateType type;
};

template <typename CoordinateType, std::size_t DimensionCount, std::size_t Dimension>
struct access<boost::array<CoordinateType, DimensionCount>, Dimension>
{
    static inline CoordinateType get(boost::array<CoordinateType, DimensionCount> const& a)
    {
        return a[Dimension];
    }

    static inline void set(boost::array<CoordinateType, DimensionCount>& a,
                           CoordinateType const& value)
    {
        a[Dimension] = value;
    }

#define BOOST_GEOMETRY_REGISTER_BOOST_ARRAY_CS(CoordinateSystem)
namespace boost { namespace geometry { namespace traits {
    template <class T, std::size_t N>
    struct coordinate_system<boost::array<T, N> >
    {
        typedef CoordinateSystem type;
    };
}}}
```

Adaptation through ADL: Boost.Range

```
namespace Eigen {
  template <typename Derived>
  typename PlainObjectBase<Derived>::Scalar*
  begin(PlainObjectBase<Derived> &a)
  {
    return a.data();
  }

  template <typename Derived>
  typename PlainObjectBase<Derived>::Scalar*
  end(PlainObjectBase<Derived> &a)
  {
    return a.data() + a.size();
  }
}

namespace boost {
  template <typename T, int Rows, int Cols>
  struct mutable_range_iterator<Eigen::Matrix<T, Rows, Cols>> {
    using type = typename Eigen::Matrix<T, Rows, Cols>::Scalar*;
  };
}
```

Fix Eigen::Matrix's lack of standard container semantics.