pair.cm

```
/*
    ================================================================

    Copyright (c) 2012-2016 Seppo Laakko
    http://sourceforge.net/projects/cmajor/

    Distributed under the GNU General Public License, version 3 (GPLv3).
    (See accompanying LICENSE.txt or http://www.gnu.org/licenses/gpl.html
        )

    ================================================================
    */

// Copyright (c) 1994
// Hewlett-Packard Company
// Copyright (c) 1996
// Silicon Graphics Computer Systems, Inc.
// Copyright (c) 2009 Alexander Stepanov and Paul McJones

using System.Concepts;

namespace System
{
    public class Pair<T, U> where T is Semiregular and U is Semiregular
    {
        public Pair(): first(), second() {}
        public Pair(const T& first_, const U& second_): first(first_),
            second(second_) {}
        public Pair(T&& first_, U&& second_) : first(Rvalue(first_)),
            second(Rvalue(second_)) {}
        public T first;
        public U second;
    }

    public nothrow bool operator<<T, U>(const Pair<T, U>& left, const
        Pair<T, U>& right) where T is TotallyOrdered and U is
        TotallyOrdered
    {
        if (left.first < right.first) return true;
        else if (left.first > right.first) return false;
        else return left.second < right.second;
    }

    public nothrow bool operator==<T, U>(const Pair<T, U>& left, const
        Pair<T, U>& right) where T is Regular and U is Regular
    {
        return left.first == right.first && left.second == right.second;
    }

    public Pair<T, U> MakePair<T, U>(const T& first, const U& second)
        where T is Semiregular and U is Semiregular
```

```
    {
        return Pair<T, U>(first , second);
    }

    public class SelectFirst<T, U>: UnaryFun<Pair<T, U>, T> where T is
        Semiregular and U is Semiregular
    {
        public nothrow const T& operator()(const Pair<T, U>& p) const
        {
            return p.first;
        }
    }

    public class SelectSecond<T, U>: UnaryFun<Pair<T, U>, U> where T is
        Semiregular and U is Semiregular
    {
        public nothrow const U& operator()(const Pair<T, U>& p) const
        {
            return p.second;
        }
    }
}
```