

Objektorientierte Programmierung

Studiengang Medieninformatik

Hans-Werner Lang

Hochschule Flensburg

Vorlesung 9

10.05.2017

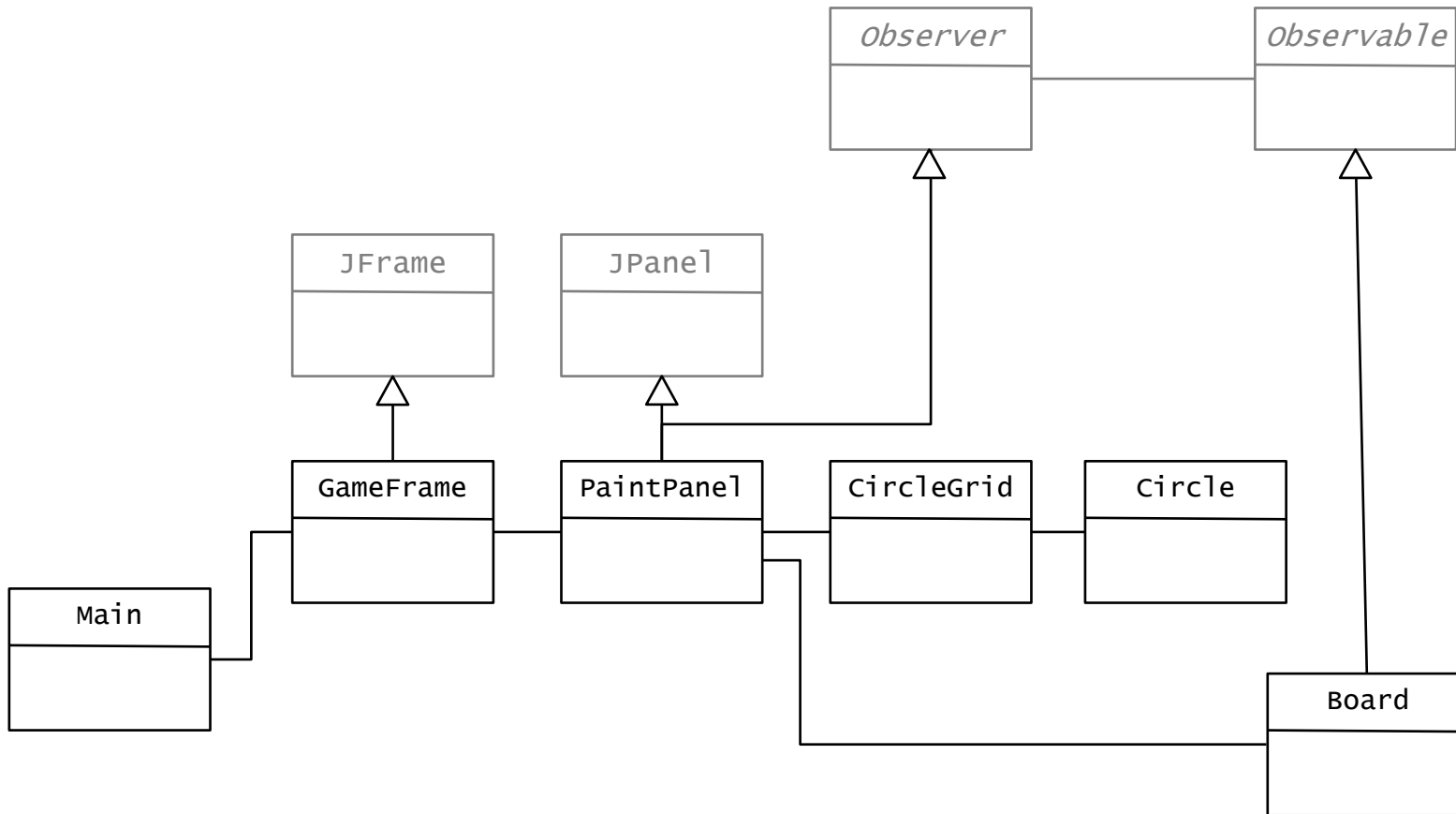
Heute:

- Stand der Entwicklung unseres Programms *Vier gewinnt*
- Weitere Entwicklung
- Probeklausur

Stand der Entwicklung

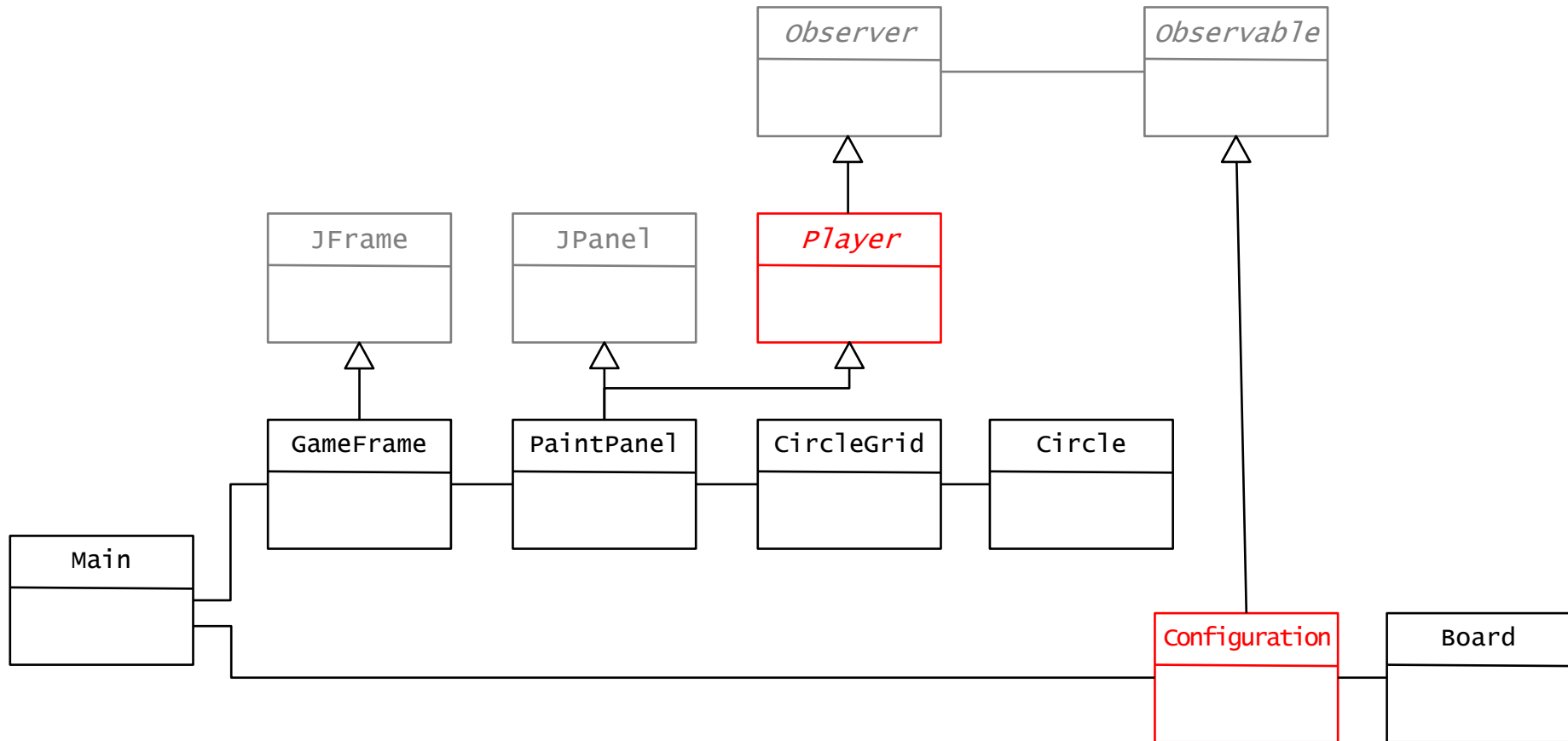
Programm...

Klassendiagramm *Vier Gewinn!*



Weitere Entwicklung

Klassendiagramm *Vier Gewinnt* (erweitert)



Interface Player

```
public interface Player extends Observer
{
    // Farbe des Spielers festlegen (0 oder 1):
    public void setColor(int k);

    // die Spielstellung festlegen, in der gespielt wird:
    public void setConfiguration(Configuration config);

    // einen Spielzug ausführen:
    public void makeMove(Move move);
}
```

Klasse Configuration

Die Klasse *Configuration* repräsentiert eine Spielstellung, gekennzeichnet durch

- die Belegung des Spielbretts,
- den Spieler, der am Zug ist, und
- die Anzahl der bereits gespielten Spielzüge.

Bei anderen Spielen können noch weitere Informationen hinzukommen, wie etwa bei Domino oder Scrabble die zur Verfügung stehenden Spielsteine oder jegliche sonstige Parameter, wie etwa bei Monopoly der Bestand an Straßen und an Spielgeld.


```

public class Configuration extends Observable
{
    public Board board;
    public int moves=0, turn=0;

    public static Configuration startConfiguration()
    {
        Configuration config=new Configuration();
        config.board=Board.emptyBoard();
        return config;
    }

    public Configuration copy()
    {
        Configuration config=new Configuration();
        config.board=this.board.copy();
        config.moves=moves;
        config.turn=turn;
        return config;
    }

    public void applyMove(Move move)
    {
        if (move.c==turn)
        {
            board.applyMove(move);
            moves++;
            turn=1-turn;
            this.setChanged();
            this.notifyObservers();
        }
    }
}

```

Probeklausur

[Probeklausur...](#)