



**Westfälische
Hochschule**

Gelsenkirchen Bocholt Recklinghausen

Vorlesung „Mensch-Computer-Interaktion“

MODEL/VIEW-PROGRAMMIERUNG

the Qt way

Prof. Dr. Tom Vierjahn

Visual Computing (<https://vc.w-hs.de>)

Fachbereich Wirtschaft und Informationstechnik – Campus Bocholt



Wintersemester 2020/21

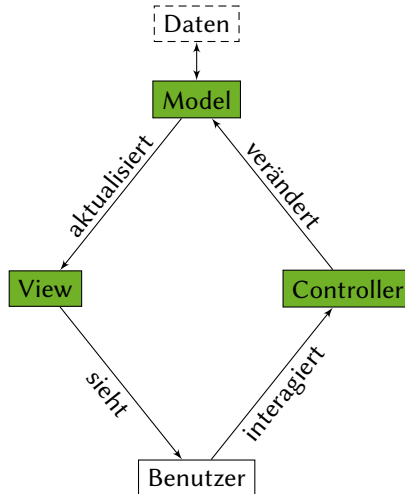


Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

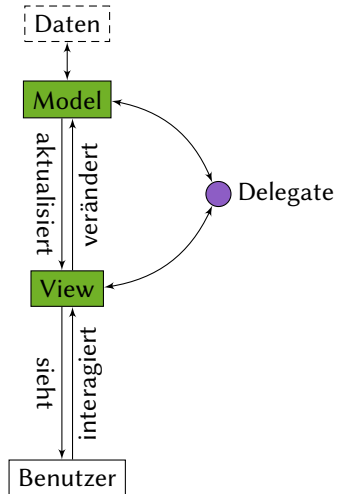
Model-View-Controller

der klassische Ansatz



Model-View

à la Qt





#	Family Name	Given Name	Position
64	Motte	Tyler	Center
26	Roussel	Antoine	Left Wing
31	McIntyre	Zane	Goalie
29	Sautner	Ashton	Defenseman
21	Eriksson	Loui	Left Wing
23	Edler	Alexander	Defenseman
20	Sutter	Brandon	Center
83	Beagle	Jay	Center
57	Myers	Tyler	Defenseman
25	Markstrom	Jacob	Goalie
4	Benn	Jordie	Defenseman
8	Tanev	Christopher	Defenseman

vancouver_canucks.csv:

```
64,"Motte","Tyler","Center"  
26,"Roussel","Antoine","Left Wing"  
31,"McIntyre","Zane","Goalie"  
29,"Sautner","Ashton","Defenseman"  
21,"Eriksson","Loui","Left Wing"  
23,"Edler","Alexander","Defenseman"  
20,"Sutter","Brandon","Center"  
83,"Beagle","Jay","Center"  
57,"Myers","Tyler","Defenseman"  
25,"Markstrom","Jacob","Goalie"  
4,"Benn","Jordie","Defenseman"  
8,"Tanev","Christopher","Defenseman"  
...
```

Datei öffnen:

```
QFile file(filename);  
file.open(QIODevice::ReadOnly);
```

Zeilen lesen:

```
while (!file.atEnd()) {  
    QByteArray line = file.readLine();  
    QList<QByteArray> cells = line.split(',');  
  
    // ...  
}
```

Modell-Klasse:

```
class MyModel : public QAbstractTableModel {
    Q_OBJECT
public:
    MyModel(QObject *parent = nullptr);

    // ...

};
```

Anzahl der Zeilen:

```
int MyModel::rowCount(const QModelIndex& parent) const { /* ... */ }
```

Anzahl der Spalten:

```
int MyModel::columnCount(const QModelIndex& parent) const { /* ... */ }
```

Zellinhalt:

```
QVariant MyModel::data(const QModelIndex &index, int role) const {  
    QVariant retval;  
    if (role == Qt::DisplayRole) {  
        switch (index.column()) { /* ... */ }  
    }  
    return retval;  
}
```

Zeilen-/Spaltentitel:

```
QVariant MyModel::headerData(int section, Qt::Orientation orientation,  
                             int role) const {  
    QVariant retval;  
    if (role == Qt::DisplayRole && orientation == Qt::Horizontal) {  
        switch (section) { /* ... */ }  
    }  
    if (role == Qt::DisplayRole && orientation == Qt::Vertical) {  
        switch (section) { /* ... */ }  
    }  
    return retval;  
}
```

main.cpp (Auszug):

```
int main(int argc, char **argv) {  
    QApplication a(argc, argv);  
  
    QTableView tableView;  
    MyModel model;  
    tableView.setModel(&model);  
  
    tableView.show();  
  
    return a.exec();  
}
```

- ▶ <https://doc.qt.io/qt-5/model-view-programming.html>
- ▶ <https://doc.qt.io/qt-5/modelview.html>

- ▶ Model-View-Controller vs. Model-View
- ▶ Qt's TableModel
- ▶ Integration

Prof. Dr. Tom Vierjahn

►  tom.vierjahn@w-hs.de

Visual Computing

►  <https://vc.w-hs.de>

►  VisualComputingWH

►  Visual Computing WH

►  @VisComputingWH

Westfälische Hochschule

Fachbereich Wirtschaft und Informationstechnik

Campus Bocholt



Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)