

Schönen Guten Tag Team,

die Idee zum Interface die ich mir überlegt haben sieht wie folgt aus:

Es gibt drei Klassen: Model, View und Controller. Alle haben die Jeweils anderen als Pointer als Inhalt der eigenen Klasse. Der Pointer ist notwendig da C++ beim setzen von Variablen immer Kopien macht und keine Referenzen wie C# oder Java übergibt. Wenn man einen Pointer nutzt, wird eine Speicheradresse von dem Objekt übergeben, welche wie der Name schon sagt, auf das Objekt im Speicher zeigt und die Klassen bzw. die Objekte so untereinander kommunizieren können. Die Festlegung der Objekte soll anfangs in der Main.cpp stattfinden und über entsprechende Setter-Methoden funktionieren.

Die Klasse Model soll die Methode „void SetzteNeueFrage()“ haben, die vom Controller bei Bedarf aufgerufen werden kann. Sie sollen auf die Variablen (unten in Grün) eine neue Frage schreiben, die dann von der View Gelsen werden kann.

```
class Model
{
public:
    string AktuelleFrage;
    string AktuelleAntwortA;
    string AktuelleAntwortB;
    string AktuelleAntwortC;
    string AktuelleAntwortD;
    chat AktuellerichtigeAntwort;

    void SetzteNeueFragen();
    SetView(View * input);
    SetController(Controller * input);

private:
    View * NameView;
    Controller * NameController;
};
```

Die View Klasse soll die Methoden „void LeseAktuelleFrage()“ und „void ModelKannGelsenWerden()“ haben. „LeseAktuelleFrage()“ soll die Aktuelle Frage aus dem Model lesen können und die „void ModelKannGelsenWerden()“ soll vom Controller genutzt werden, um die View zu informieren, dass es eine Neue Frage gibt.

```
class View
{
public:
    LeseAktuelleFrage()
    void LeseAktuelleFrage();

    SetController(Controller * input);
    SetModel(Model * input);

private:
    Controller * NameController;
    Model * NameModel;
}
```

Die Klasse Controller soll die Methoden „StartGame()“ und GetAntwort() haben. StartGame() soll die das Spiel starten. Und GetAntwort() soll die Antwort aus der View holen.

```
class Controller
{
public:
    StartGame();
    GetAntwort();
    SetView(View * input);
    SetModel(Model * input);
private:
    View * NameView;
    Model * NameModel;
}
```

So soll die Verknüpfung der Objekte in der Main.cpp aussehen:

```
Int main()
{
    Model nameModel = new Model();
    View nameView = new View();
    Model nameModel = new Model();
    nameModel->SetView(nameView);
    nameModel->SetController(nameController);
    nameView->SetModel(nameModel);
    nameView->SetController(nameController);
    nameController->SetView(nameView);
    nameController->SetModel(nameModel);

    nameController->StartGame();
}
```

Nochmal kurz zu Pointern falls ihr nicht genau wisst was das ist. Pointer sind letztendlich Speicheradressen die auf ein Objekt im Arbeitsspeicher zeigen. Wenn man sie einmal Initialisiert hat kann man mit Ihnen wie mit normalen Objekten arbeiten. Abgesehen von zwei Unterschieden. Sie werden anders Initialisiert:

Anstatt:                      Objekt nameDesObjekts;  
Schreibt man:                Objekt \* nameDesObjekts = new Objekt();

Und der Zugriff auf Variablen und Methoden von Pointer sieht anders aus anstatt eines Punktes nutzt man einen Pfeil.

Anstatt:                      Objekt.Variable  
Schreibt man:                Objekt->Variable

Ich wünsche viel Erfolg bei der weiteren Planung und hoffe, dass ich hiermit noch ein bisschen zu unserer Gruppenarbeit beitragen konnte.

Gruß Jan