

1.1 Fehlerbehandlung

Definition. **Ressourcenbelegung ist Initialisierung** (Resource Acquisition Is Initialization) ist eine Programmier Technik zur Verwaltung von Ressourcen. Dabei wird die Belegung von Ressourcen an den Konstruktoraufruf einer eigenen Klasse und die Freigabe der Betriebsmittel an dessen Destruktoraufruf gebunden. Dadurch wird durch Verlassen des Gültigkeitsbereichs die Ressource automatisch freigegeben.

Gegeben ist der Code für das Programm *Crypto*.

1.1.1 RAII

Schreibe eine Klasse "Datei" um die Ressource FILE zu kapseln.

Schreibe eine ähnliche Klasse "Puffer" um die dynamische Speicherverwaltung zu kapseln.

Ersetze im Code die Stellen, die dies bereits tun, durch Verwendung deiner Klassen. Passe, wenn nötig die Funktionssignaturen an.

1.1.2 Fehlercodes vs Ausnahmen

Nachdem wir die Ressourcenverwaltung sichergestellt haben, wollen wir den Code besser lesbar gestalten. Dazu soll der gute Pfad (funktionaler Teil) von dem Fehlerbehandlungspfad getrennt werden.

Ersetze Stellen wo es sinnvoll ist Fehlercodes gegen Ausnahmen auszutauschen (siehe Theorie).

Fange alle Ausnahmen an einer Stelle und behandle die Fehler dort (Ausgabe).

Erstelle dazu passende eigene Ausnahmeklassen, die von der Basisklasse

```
class MyBaseException : public std::exception {
public:
    using exception::exception;
    virtual void Print() = 0;
};
```

ableiten. Werfe diese in dem Fehlerfall, anstatt den Fehlercode zurückzugeben. Man kann oft die Ausgabeargumente nun zurückgeben, was den Code nochmal vereinfacht.

Passe Funktionssignaturen an, wenn nötig.

1.1.3 Fehlercodes vs Assertion

Ersetze Stellen wo es sinnvoll ist Fehlercodes gegen Assertion auszutauschen (siehe Theorie).

Passe Funktionssignaturen an, wenn möglich.

1.1.4 Zusammenfassung

Welche Schwierigkeiten sind beim Programmieren aufgetreten?

Wie hat sich der Code verbessert/verschlechtert?

Was sind die Kriterien, wenn man am besten Assertion, Ausnahmen bzw. Fehlercodes verwendet?