FH-OÖ Hagenberg/HSD
SDP3, WS 2012
Übung 3

Name(1): Abgabetermin: KW 46

Name(2): Punkte:

Übungsgruppe: korrigiert:

Geschätzter Aufwand in Mh: Effektiver Aufwand in Mh:

Beispiel 1 (24 Punkte) Verschlüsselung: Entwerfen Sie aus der nachfolgend gegebenen Spezifikation ein Klassendiagramm, instanzieren Sie dieses und implementieren Sie die Funktionalität entsprechend:

Die Firma High Speed Software Engineering (HSE) soll für die beiden Kunden Epcos und Nortel Networks ein Verschlüsselungssystem zur Verfügung stellen.

Es werden 2 Verschlüsselungsalgorithmen unterstützt: Caesar und RSA.

Die Algorithmen sollen zur Laufzeit austauschbar sein. Benützen Sie dafür ein entsprechendes Design Pattern. Da die beiden Kunden unterschiedliche Schnittstellen wünschen, verwenden Sie ein internes Interface und delegieren die Aufrufe der beiden Interfaces mit Hilfe eines geeigneten Design Pattern an die interne Schnittstelle.

Die interne Schnittstelle (Auszug):

```
1 bool ReadFile(std::string const & fileName);
2 bool WriteFile(std::string const & fileName);
3 void Encrypt();
4 void Decrypt();
```

Schnittstelle von Epcos:

```
1 virtual void EncryptRSA(std::string const & fileName) = 0;
2 virtual void DecryptRSA(std::string const & fileName) = 0;
```

Schnittstelle von Nortel Networks:

```
enum TEncoding {
   eRSA,
   eCaesar
};
virtual void Encipher(TEncoding enc, std::string const & fileName) = 0;
virtual void Decipher(TEncoding enc, std::string const & fileName) = 0;
```

Für `fileName` gilt in allen Fällen vereinfachend:
Bei `fileName` = `Message.txt`:
`Message.txt` ist der Klartext
`Message.txt.Caesar` ist die Caesar-verschlüsselte Datei
`Message.txt.RSA` ist die RSA-verschlüsselte Datei

Die jeweiligen Verschlüsselungs-Parameter (key bei Caesar; n,e,d bei RSA) können intern festgelegt werden. Sie brauchen nicht vom Kunden konfigurierbar sein. Wählen Sie für key selbst einen beliebigen Wert. Benützen Sie bei RSA:

$e = 7, d = 23$

Schreiben Sie einen Testtreiber, der verschiedene Nachrichtendateien im ASCII-Format vom Dateisystem einliest und ver- bzw. entschlüsselt. Verwenden Sie dazu jeweils das Interface von Epcos und auch jenes von Nortel. Geben Sie die Ergebnisse (soweit druckbar) aus!

Treffen Sie für alle unzureichenden Angaben sinnvolle Annahmen. Verfassen Sie weiters eine Systemdokumentation (Funktionalität, Klassendiagramm, Schnittstellen der beteiligten Klassen, etc)!

Allgemeine Hinweise: Legen Sie bei der Erstellung Ihrer Übung großen Wert auf eine **saubere Strukturierung** und auf eine **sorgfältige Ausarbeitung!** Dokumentieren Sie alle Schnittstellen und versehen Sie Ihre Algorithmen an entscheidenden Stellen ausführlich mit Kommentaren! Testen Sie ihre Implementierungen ausführlich! Geben Sie den **Testoutput** mit ab!