Budapesti Műszaki- és
Gazdaságtudományi Egyetem

Programozás alapjai 2.

Házi Feladat

Ujhelyi Bence

V2VU9O

2023. május 8.

Tartalomjegyzék

1. Specifikáció

1.1. Feladatleírás

Titkosító osztály

Készítsen titkosító osztályt!

Az osztály legyen képes tetszőleges hosszúságú szöveg kódolt formátumú tárolására. A kódoláshoz olyan egyszerű műveletet használjon, ami nem függ a kódolandó szöveg hosszától (pl. kizáró vagy). Valósítsa meg a szokásos string műveleteket! Az osztályt úgy tervezze meg, hogy az örökléssel könnyen felhasználható legyen, az algoritmus könnyen cserélhető legyen! Valósítsa meg az összes értelmes műveletet operátor átdefiniálással (overload), de nem kell ragaszkodni az összes operátor átdefiniálásához! Legyen az osztálynak iterátora is! Specifikáljon egy egyszerű tesztfeladatot, amiben fel tudja használni az elkészített adatszerkezetet! A tesztprogramot külön modulként fordított programmal oldja meg!

A megoldáshoz ne használjon STL tárolót!

1.2. Feladatspecifikáció

A feladat egy titkosító osztály elkészítése. A feladatot az Objektum Orientált paradigmák felhasználásának fényében valósítom meg. Az osztály elkészítésekor különös figyelmet szentelek a program továbbfejlesztésének lehetőségire, hogy később, akármilyen szövegtitkosító algoritmussal bővíthető legyen.

A feladat nem írja elő, hogy milyen műveleteket kell az osztálynak megvalósítania, de mivel egy szöveg titkosításáról beszélünk, ezért alapvetően minden stringgel megvalósítható műveletre képesnek kell lennie. Szűrni kell a hibás indexelést, ezért ebben az esetben `std::out_of_range` kivételt kell dobnia. Ellenőriznie kell továbbá, hogy a string null pointerre mutat-e.

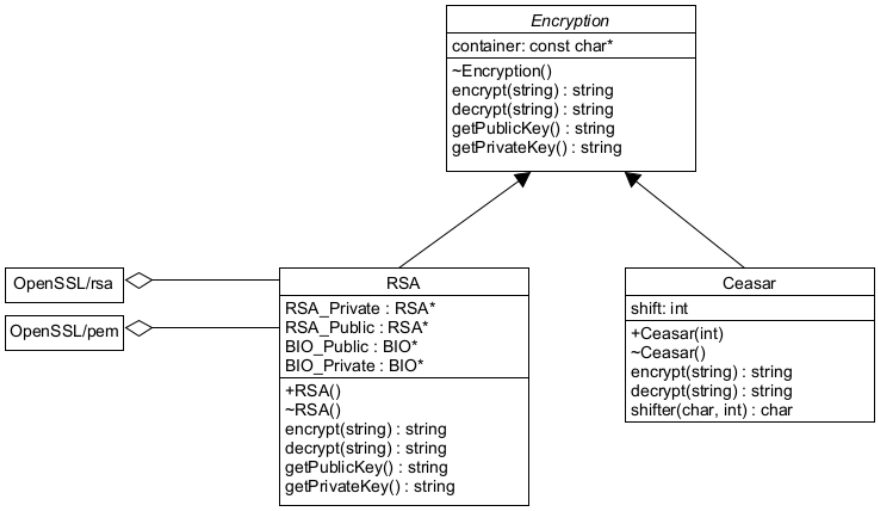
A program bemenetként két paramétert kap, az egyik maga a szöveg (string), a másik pedig a titkosítás típusa. Kimenetként a választott titkosítással kódolt szöveget kapjuk vissza. Ugyanennek visszafelé is kell működnie, tehát képesnek kell lennie dekódolni a már titkosított szöveget.

A program működését, külön teszt programmal tervezem ellenőrizni. A teszt programnak ellenőriznie kell minden előfordulható hibát a bemeneten és a kimeneten egyaránt.

2. Terv

2.1. Objektum modell

A feladat megvalósításához egy absztrakt titkosító osztály szükséges. Ebből az ős osztályból lesznek leszármaztatva a különböző titkosítások osztályai. Az osztály tesztelésére kétfajta titkosító algoritmust tervezek használni, az egyik a nyílt kulcsú RSA, a másik pedig egy egyszerűbb, úgynevezett ceasar algoritmus (ABC eltolása).



osztálydiagram

2.2. Algoritmusok

2.2.1. Encryption osztály algoritmusai

Az Encryption osztály kizárólag virtuális függvényeket tartalmaz, mivel ez egy absztrakt osztály.

Függvények:

- encrypt
- decrypt
- getPublicKey
- getPrivateKey
- Összeadó operátor
- Értékadó operátor
- Destruktor

2.2.2. RSA osztály algoritmusai

Az osztály változói:

Változó	Leírás
RSAPublic	RSA publikus kulcsa
RSAPrivate	RSA privát kulcsa
BIOPublic	Publikus kulcs kiolvasásához
BIOPrivate	Privát kulcs kiolvasásához

Konstruktor: Az OpenSSL könyvtárat felhasználva létrehozza a titkosítási kulcsokat.

Destruktor: Felszabadítja az összes dinamikusan foglalt memóriát.

encrypt: Túlterheli az ősosztály encrypt függvényét, és a titkosító kulcsokat felhasználva titkosítja a beérkező karaktersorozatot.

decrypt: A titkosító kulcsokat felhasználva visszaállítja a titkosított üzenet eredeti formáját.

getPrivateKey: Visszaadja a privát titkosító kulcsot egy karaktersorozatban.

getPublicKey: Visszaadja a publikus titkosító kulcsot egy karaktersorozatban.

2.2.3. Ceasar osztály algoritmusai

Az osztály változói:

Változó	Leírás
shift	Egész szám, amellyel eltoljuk a karaktereket.

Konstruktor: Beállítja 'shift' értékét, amely megadja, hogy mennyivel legyen eltolva az ABC.

Destruktor: Felszabadítja a dinamikusan foglalt memóriát.

encrypt: Az algoritmust felhasználva titkosítja a karaktersorozatot.

decrypt: A karaktereket "visszatolja" eredeti értékükre, így visszakapva az eredeti karaktersorozatot.

shifter: Egy karakter titkosított (shiftelt) megfelelőjét adja vissza.