

SOFTWAREVÝ PROJEKT

CAESAROVA ŠIFRA

STSW (Specifikace testování SW)

Jana Michnová

(michnjan@fel.cvut.cz)

Testování

Program Caesarova šifra může obsahovat spoustu interních chyb, pokud se na tyto chyby během vývoje nemyslí. Proto v kódu existuje několik ověřovacích podmínek, které jim zamezují.

Nejviditelnější chyby mohou nastat v okamžiku, kdy uživatel, pracující s programem, zadá na dotaz programu, hodnotu, se kterou neumí pracovat.

Tyto stavy se nacházejí v každé části programu, jelikož je sestaven tak, aby co nejvíce komunikoval s uživatelem a aby se sám uživatel podílel na výsledku. Tudíž je to pokaždé, kdy se zadává velikost kroku, nebo uvádí cesta k souboru do, nebo z něhož se načítá (ukládá) výsledný řetězec.

V těchto případech se kontroluje zadávaná hodnota pomocí bloku **Try{...}Catch(...){...}**, ve kterém se odchytávají výjimečné stavy, jako například, že uživatel zadá místo číselné volby v menu, volbu znakovou. V takovém případě nám program naši chybu ohlásí a očekává správný formát volby.

Další stav, ve kterém je nutno dávat si pozor na zadaná data je tehdy, kdy se program dotazuje uživatele na cestu k souboru. V takovém případě se ověřuje, jestli soubor již existuje nebo ne. Když existuje, nabídne nám možnost přepsání souboru nebo zadání nového pojmenování. To vše se opět kontroluje v bloku **Try{...}Catch(...){...}**.

Uživatel může cestu k souboru, ať už určenému na uložení nebo na načítání dat, zadat svůj název jak bez koncovky „.txt“, nebo s ní. „.txt“ se přidává automaticky za každý zadaný název a to z toho důvodu, že by bylo rozlišování koncovek velice složité. S formáty jinými než „.txt“ program Caesarova šifra pracovat neumí, jelikož pro ně není určen ani navržen.

Ukázka kódu zachycení vyjímky (v okamžiku existence zadaného názvu souboru):

```

if (cesta.exists() == true) {
    System.out.println("Soubor již existuje!");
    System.out.println("Prejete si sifrovany text do tohoto souboru připsat? ANO/NE");
    String vyber = new Scanner(System.in).next();
    String odpoved = vyber.toLowerCase();
    if (odpoved.equals("ano")) {
        try {
            FileWriter fw = new FileWriter(cesta, true);
            BufferedWriter bw = new BufferedWriter(fw);
            bw.write(sifrovaneZnaky);
            bw.newLine();
            bw.close();
            fw.close();
        } catch (FileNotFoundException e) {
            System.out.println("Soubor nebyl nalezen!");
        } catch (IOException e1) {
            System.out.println("Chyba při zápisu!!!");
        }
        System.out.println("Sifrovany text byl ulozen do souboru " + cesta);
    } else {
        System.out.println("Zadejte nový název souboru:");
        cesta = new File(new Scanner(System.in).next() + ".txt");
        if (cesta.exists() == true) {
            System.out.println("Soubor opet existuje! Akce byla ukoncena!");
            ZasifrujNactiZKlavesnice();
        } else {
            try {
                FileWriter fw = new FileWriter(cesta, true);
                BufferedWriter bw = new BufferedWriter(fw);
                bw.write(sifrovaneZnaky);
                bw.newLine();
                bw.close();
                fw.close();
            } catch (FileNotFoundException e) {
                System.out.println("Soubor nebyl nalezen!");
            } catch (IOException e1) {
                System.out.println("Chyba při zápisu!!!");
            }
            System.out.println("Sifrovany text byl ulozen do souboru " + cesta);
        }
    }
} else {
    try {
        FileWriter fw = new FileWriter(cesta, true);
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write(sifrovaneZnaky);
        bw.newLine();
        bw.close();
        fw.close();
    } catch (FileNotFoundException e) {
        System.out.println("Soubor nebyl nalezen!");
    } catch (IOException e1) {
        System.out.println("Chyba při zápisu!!!");
    }
    System.out.println("Sifrovany text byl ulozen do souboru " + cesta);
}
ZasifrujNactiZKlavesnice();
}

```