

Plagioinnin tunnistaminen lähdekielisisistä ohjelmista

Plagiointi- ja tutkimuseetiikka
–seminaari 30.09.2003

Kirsti Ala-Mutka
TTY/Ohjelmistotekniikka

Sisältö

- Plagiointi ohjelmointikursseilla
- Tyypillisiä ulkoasumuutoksia
- Plakki-työkalu
- Plagioinnin tarkistus kursseilla
- Työkalun käyttökokemuksia

Plagiointi ohjelmointikursseilla

- Plagioinnin määritelmä ohjelmointiharjoitustöissä TTY:lla on ”elektronisessa muodossa tapahtuva ohjelmakoodin siirto”
- Plagiointi on helppoa, koska ohjelmat ovat tekstimuotoisia tiedostoja
- Kiinnijäämisen riski suurissa opiskelijaryhmissä on usein pieni
- On mahdollista tehdä näennäisiä ulkoasumuutoksia vaikeuttamaan plagioinnin tunnistusta

Tyypillisiä ulkoasumuutoksia

- Merkkijonojen muuttaminen
- Kommenttien muuttaminen
- Erilaiset nimeämiskäytännöt
- Ohjelman osien järjestyksen vaihtaminen
- Lausekkeiden kirjoitusasun muokkaus

```
// Jakolaskun toteuttava funktio
int jaa(const int jaettava,
        const int jakaja)
{
    float tulos = 0;
    // tarkistetaan ettei jakaja ole 0
    if (jakaja == 0){
        cout << "Jakaja ei saa olla 0";
    } else {
        tulos = jaettava/jakaja;
        cout << "Tulos on " << tulos;
    }
    return EXIT_SUCCESS;
}
```

```
// assignment no 1
int divide ( int a, int b ){
    if ( b == 0 )
        cout << "Illegal divisor"; // no div
    else {
        float tmp = 0;
        tmp = a/b; // do div
        cout << "Result: " << tmp;
    }
    return 0;
}
```

Plakki-työkalu

- Toteutettu UNIX-ympäristöön C/C++-kielisille ohjelmille
- Pelkistää ohjelmien ulkoasuyksityiskohdat ennen käsittelyä
- Vertaa annettuja ohjelmia pareittain ja tuottaa samankaltaisuusprosenttilistan, esim.

100% / 96% : pekka.cc maija.cc

90% / 93% : ville.cc pekka.cc

85% / 90% : ville.cc maija.cc

75% / 80% : kalle.cc maija.cc

...

- Tuloksen tulkinnassa huomioitava ohjelman koko ja annetun ohjeistuksen määrä
-

```

// Ohjelma, joka muuntaa kirjainjonoja Caesar-salauksella
#include <string.h>
#include <iostream.h>

// Määritellään vakioiksi käytettävät aakkostot ja kirjainmäärä
const char* alphabets = "abcdefghijklmnopqrstuvwxyzääö";
const char* capitals = "ABCDEFGHIJKLMNPOQRSTUVWXYZÄÖ";
const int amount = 28;

// Apufunktio joka laskee annettua merkkiä vastaavan merkin
char caesar(int, int);
char caesar(int charno, int key)
{
    charno = (charno+key) % 29;
    return charno;
}

// Pääohjelma
int main()
{
    int key=0;
    char ch=' ';
    int index=0;

    // Kysytään käyttäjältä salausavain
    cout << "Anna merkkijonon muunnoksessa käytettävä avain";
    cin >> key;

    // Luetaan ja koodataan viesti
    while (cin.get(ch)){
        if (ch==' ' || ch=='\n' || ch=='\t'){
            cout << ch ;
        } else {
            while(index<=amount){
                if(alphabets[index]==ch){
                    cout << alphabets[caesar(index, key)];
                    break;
                }
                if(capitals[index]==ch){
                    cout << alphabets[caesar(index, key)];
                    break;
                }
            }
            // Kirjainta ei löytynyt aakkostoista
            if (index==amount+1){
                cerr << "Tuntematon kirjain!" << endl;
            }
            index++;
        }
    }
}
return 0;
}

```

```

#include <iostream.h>
#include <string.h>

int kirjainmaara = 28; // aakkoston pituus
char* aakkosto = "abcdefghijklmnopqrstuvwxyzääö"; // pienet
char* isoakkosto = "ABCDEFGHIJKLMNPOQRSTUVWXYZÄÖ"; //isot

char caesar( int merkkintro, int avain );

int main() {
    int avain = 0;
    cout << "Anna salausavain: "; // kysytään avain
    cin >> avain; // luetaan avain

    char merkki = ' ';
    while ( cin.get(merkki) ) { // luetaan syötettä
        if ( merkki == '\n' || merkki == '\t' || merkki == ' ' )
            cout << merkki ; // tulostetaan suoraan

        else {
            int monesko = 0;

            while ( monesko<=kirjainmaara ) {
                if( isoakkosto[monesko] == merkki ) { // iso
                    cout << aakkosto[ caesar(monesko, avain) ];
                    break;
                }
                if( aakkosto[monesko] == merkki ) { // pieni
                    cout << aakkosto[ caesar(monesko, avain) ];
                    break;
                }
                if ( monesko == kirjainmaara + 1 )//tunnistamaton
                    cerr << "Tunnistamaton poistettu." << endl;
                monesko++;
            }
        }
    }
    return EXIT_SUCCESS;
}

char caesar(int merkkintro, int avain) {
    merkkintro = ( merkkintro + avain ) % 29;
    return merkkintro; // palautetaan muunnettu merkki
}

```


Plagioinnin tarkistus kursseilla

- Harjoitustöitä vertaillaan palautusmääräajan päätyttyä
 - Vertailua voidaan tehdä myös aiempien vuosien töihin
- Työkalu toimii hälytyskellona, opettaja tarkastaa tapaukset henkilökohtaisesti
 - Tutkii tarkemmin esim. lausekkeiden järjestystä, työn versiohistoriaa, näkymättömiä merkkejä
 - Epäilty opiskelijat kutsutaan haastatteluun
- Plagioitua työtä ei hyväksytä suoritukseksi, myös tietoisesti plagiointiin työnsä luovuttaneen opiskelijan kurssisuoritus keskeytyy

Työkalun käyttökokemuksia

- Plakin käyttöönotto selvästi auttanut plagioinnin tunnistamista
 - Ensimmäisillä käyttöönottokerroilla plagiointitapauksia löytyi hyvin runsaasti
 - Edelleenkin harjoitustöistä paljastuu aina muutamia kopiointitapauksia
- Työkalu ei ole antanut väärää hälytyksiä
- Lisäkehityskohteita löytyy monikäyttöisyyden, tehokkuuden ja kopioiden tunnistamisen suhteen
- Työkalu on ollut hyödyllinen ja tarpeellinen apuväline ohjelmointikurssien opettajille

Kiitoksia mielenkiinnosta!

Yhteystiedot:

Kirsti Ala-Mutka

TTY/Ohjelmistotekniikka

kirsti.ala-mutka@tut.fi