

Mobiilijärjestelmien ohjelmointi
Loppukoe 5.5.03

Nimi: _____ Opiskelijanro: _____

Harjoitustyön suorituslukukausi: _____

Voit jatkaa vastaustasi paperin kääntöpuolelle. Merkitse tämä kuitenkin nuolella (tms) tehtävän yhteyteen.

1. Vastaa lyhyesti seuraaviin kysymyksiin.

a) Mitä tarkoittaa Symbian OS:n käyttämä ansahaarniska-mekanismi (trap harness)?
Anna pieni esimerkki. (2pts)

b) Mitä tarkoittaa konfiguraatio Java-mobiiliympäristön yhteydessä? Miten se näkyy ohjelmoijalle? (2pts)

c) Mobiililaitteisiin voidaan tehdä hyödyllisiä sovelluksia, joiden toiminta ei suoranaisesti edellytä verkon käyttöä. Mainitse syitä, joiden vuoksi sellaisenkin sovelluksen on järkevää turvautua verkon käyttöön? (2pts)

2. Valitse oikea vaihtoehto ja perustele vastauksesi (1+1pts).

a) Mobiililaitteessa toimivan Symbian-ohjelman toteutuksessa keon käyttöön suunnittelu on keskeisempää kuin pinon käytön suunnittelu. (2pts) Tosi Epätosi

b) Virtuaalikoneen käyttö mobiilisovelluksissa yksinkertaistaa tietoturvan toteuttamista. (2pts) Tosi Epätosi

c) Välimuistin käytön optimointiin ei mobiililaitteiden ohjelmoinnissa tyypillisesti tarvitse kiinnittää huomiota. Sitä ei ole tai sen erilaiset käyttötavat vaikuttavat tehokkuuteen vain marginaalisesti. (2pts) Tosi Epätosi

3. Selitä lyhyesti (yhdellä lauseella) seuraavat termit.

a) Sulautetut osoittimet (2pts)

b) Etäevaluointi (2pts)

c) KVM (2pts)

4. Mitä Symbian-ohjelmointikäytäntöihin (tai yleiseen C++-ohjelmointiin) liittyviä virheitä löydät oheisesta koodinpätkästä? Mitä ongelmia ne voivat aiheuttaa? (6pts)

Määrittelyt:

```
enum TMode { EEat, ESleep, EWait };

class CMyResource {
public:
    CMyResource(CScreenShot & aParent);
    ConstructL();
    ~CMyResource();
    ShowEatingItemL();
    ShowSleepingItemL();
    ShowWaitingItemL();
private:
    ...
};
```

Annettu ohjelma:

```
void Behave( TMode aMode, CScreenShot& myWindow)
{ // Show the activity indicated by the parameter in main window.
    CMyResource *res;
    res = new CMyResource(myWindow);
    res->ConstructL();
    CleanupStack::PushL(res);
    switch ( aMode ) {
        case EEat:
            res->ShowEatingItemL();
            break;
        case ESleep:
            res->ShowSleepingItemL();
            break;
        case EWait:
            res->ShowWaitingItemL();
            break;
        default:
            break;
    }
    CleanupStack::Pop(); // res
    delete res;

    MakeImagePermanentL(myWindow);
}
```

