

Eksamensopgave i RD2-SWC

20. August 2018

Alle hjælpemidler, som ikke kræver brug af internet, er tilladt.

Aflevering

Der skal afleveres én PDF-fil med opgavebesvarelsen i C++-delen og en (PDF)-fil med opgavebesvarelsen i Softwareudviklingsdelen. Desuden skal al kildekode afleveres i en zip-fil. En komplet *aflevering indeholder altså 3 filer*.

Til hver C++ opgave er der en test-kode, som skal eksekveres og et screendump skal indsættes i afleveringen. Hvis programmet ikke kan compile, så indsættes i stedet fejlmeddelelsen fra compileren.

Enhver form for kopier/indsæt (copy/paste) fra tidligere opgaver anses som eksamenssnyd.

Opgavesættet består af 6 sider, 1 forside, 4 sider med C++ opgaver og 1 side med softwareudviklingsopgaven.

Opgave 1 (1 time, 25 point)

I denne opgave skal vi lave en klasse til at håndtere varer i en indkøbskurv. Til dette skal der oprettes en klasse "CartItem". CartItem skal have følgende member variable:

```
std::string _item_no;
int _quantity;
double _price;
```

samt følgende constructors og member funktioner:

```
CartItem();
CartItem(const std::string& item_no, int quantity);

void setItemNo(const std::string& item_no);
void setQuantity(int qty);
void setPrice(double price);

const std::string& getItemNo() const;
int getQuantity() const;

double getPrice() const;
double getTotalPrice() const;
```

Default constructor skal initialisere member variable fornuftigt, og constructoren med parametre skal sætte de respektive membervariable. Set og get-metoder skal implementeres. Funktionen getTotalPrice() skal udregne den totale pris for varerne af denne type.

Derudover skal der oprettes en klasse "Cart" som kan håndtere tilføjelse og fjernelse af elementer fra indkøbskurven. Cart skal implementere constructors og member funktionerne:

```
Cart();
Cart(const Cart& cart);

void addItem(const CartItem& item);
void removeItem(const std::string& item_no);

double getTotalAmount() const;
std::vector<CartItem> getItems() const;
```

Default og copy constructor skal implementeres, addItem skal tilføje et element til indkøbskurven og removeItem skal fjerne et element ud fra "item_no". Hvis det forsøges at tilføje et element med samme "item_no" skal antallet af elementer opdateres, således at antallet med pågældende item_no blot forøges. Eksempel vare "1a25b17" ønskes forøget med 5 når der allerede er lagt 3 i indkøbskurven. Nu er der altså 8 stk. i indkøbskurven for denne vare.

Memberfunktionen removeItem skal fjerne et element ud fra en streng, som angiver varenummeret (item no). Funktionen getTotalAmount skal beregne den totale pris på alle varerne i indkøbskurven, og getItems skal returnere en vector med alle items i indkøbskurven (HINT: brug en vector som member-variabel til at holde alle CartItems).

Includes: string, vector

Se test-kode på næste side.

Koden skal testes med eksempelkoden nedenfor og der skal laves et **screendump** af output fra program, som medtages i eksamensbesvarelsen.

```
{
    CartItem ci1("271845", 15);
    ci1.setPrice(20);
    CartItem ci2, ci3, ci4, ci5;
    ci2.setItemNo("183416");
    ci2.setQuantity(15);
    ci2.setPrice(15.251);
    ci4.setItemNo("12345");
    ci5.setItemNo("18no2p1");
    ci5.setQuantity(10);
    CartItem ci6("18no2p2", 19);
    CartItem ci7(ci2);
    ci7.setQuantity(3);

    Cart c;
    c.addItem(ci1);
    c.addItem(ci2);
    c.addItem(ci3);
    c.addItem(ci4);
    c.addItem(ci5);
    c.addItem(ci6);
    c.addItem(ci7);
    c.removeItem("12345");

    std::vector<CartItem> cis = c.getItems();

    std::cout.width(8);
    std::cout << "Antal" << " ";
    std::cout.width(10); std::cout << "Varenr." << " ";
    std::cout.width(8); std::cout << "Pris" << " ";
    std::cout.width(12); std::cout << "Total" << std::endl;
    std::cout.setf(std::ios::fixed);
    std::cout.setf(std::ios::showpoint);
    std::cout.precision(2);

    for (unsigned int i = 0; i < cis.size(); ++i) {
        std::cout.width(8); std::cout << cis[i].getQuantity() << " ";
        std::cout.width(10); std::cout << cis[i].getItemNo() << " ";
        std::cout.width(8); std::cout << cis[i].getPrice() << " ";
        std::cout.width(12); std::cout << cis[i].getTotalPrice() << std::endl;
    }

    std::cout << std::endl << "Total amount " << c.getTotalAmount() << std::endl
    << std::endl;
}
```

Opgave 2 (1 time, 25 point)

I denne opgave skal vi kigge på omkostninger i en produktion. En log-fil fra et produktionsapparat ser således ud:

```
2012ad,15#15EUR0.01
17625ac,20#10EUR0.25
2182af,17#20EUR1.25
8121ad,17#35EUR2.5
51221adf,21#32EUR2.1
```

Betydningen af ovenstående er for hver linje: først tal angiver ID, de følgende bogstaver angiver variant og det efterfølgende tal en størrelse. Tallet efter # angiver styktal og EUR efterfulgt af et tal er stykprisen i euro.

Dvs. ovenstående eksempelinput kan også opstilles i følgende tabel:

ID	Variant	Størrelse	Antal	Pris i EUR
2012	ad	15	15	0.01
17625	ac	20	10	0.25
2182	af	17	20	1.25
8121	ad	17	35	2.5
51221	adf	21	32	2.1

I denne opgave skal vi indlæse data fra en fil vedrørende en produktion. Opgaven går ud på at registrere og beregne den totale omkostning på et produktionsapparat.

For at løse opgaven skal der oprettes en klasse til håndtering af en produktionsomkostning som kaldes "ExpenseItem" samt en klasse til håndtering af alle omkostningerne "ProductionCalculator".

Det er frit for opgaveløseren hvordan ExpenseItem klassen implmenteres, men kvaliteten af det implementerede indgår i bedømmelsen af opgaven. Selve ProductionCalculator skal kunne indlæse filen vha. funktionen `void readData(std::istream& is)`, som skal indlæse data indtil der ikke er mere input. Derudover skal der være en standard constructor, en funktion til at beregne den totale produktionsomkostning `double totalProductionCost()` og endelig en friend funktion for insertion operatoren (<<). Output fra insertion operator skal med ovenstående test-input give følgende output

ID	Variant	Size	No	Price
2012	ad	15	15	0.01
17625	ac	20	10	0.25
2182	af	17	20	1.25
8121	ad	17	35	2.50
51221	adf	21	32	2.10

Bemærk højrejustering af output i tabellignende format samt antal cifre på pris.

Det kan antages, at der altid står EUR foran styk-prisen.

Includes: string, iostream

Se test-kode på næste side.

Koden skal testes med eksempelkodeen nedenfor og der skal laves et **screendump** af output fra program, som medtages i eksamensbesvarelsen.

Includes: sstream

```
{
    ProductionCalculator pc;
    std::stringstream
ss("2012ad,15#15EUR0.01\n17625ac,20#10EUR0.25\n2182af,17#20EUR1.25\n8121ad,17#35EUR2.
5\n51221adf,21#32EUR2.1\n24221acf,11#16EUR21.1");
    pc.readData(ss);

    std::cout << pc << std::endl;
    std::cout << "Total production cost: " << pc.totalProductionCost() <<
std::endl;
}
```