

Tentamen i TDP004

Objektorienterad Programmering

- Datum: 2007-12-19
- Tid: 14-18
- Plats: Teoretisk del: BL31 i B-huset..
Praktisk del: PC6-PC7 i E-huset.
- Jour: Per-Magnus Olsson, tel 285607
Jourhavande kommer att besöka skrivsalarna varje timme under skrivtiden.
- Hjälpmedel: Teoretisk del: Inga.
Praktisk del: En bok om C++.
- Betygsättning: Max antal poäng: 48 med 24 poäng vardera på teori och praktikdel.
- | Poäng | Betyg |
|-------|----------|
| 41-48 | 5 |
| 33-40 | 4 |
| 25-32 | 3 |
| 0-24 | U |
- Anvisningar: Skriv svaret på varje uppgift på ett separat blad.
Uppgifterna är inte ordnade i svårighetsgrad.

TDP004 Objektorienterad Programmering

Praktisk del

1. Skriv en funktion som tar en `std::list` som inparameter. Du kan förutsätta att listan endast innehåller element av typ `int`. Gå igenom listan framifrån. Efter att en operation (se nedan) har utförts på ett element ska resultatet läggas till i en `std::vector` som benämns `result`. Beroende på vilket värde ett element har ska olika operationer utföras: om elementet är > 0 ska elementet kvadreras och resultatet läggas till sist i vektorn. Om elementet är 0 ska vektorn sorteras så att det minsta elementet hamnar först och sedan ska hela vektorn skrivas ut på skärmen. Elementet med värdet 0 ska inte läggas till i vektorn. Om elementet är < 0 ska behandlingen av listan avbrytas. Slutligen ska vektorn `result` returneras. (8p)

2. Definiera en klasshierarki för att hantera olika klasser av grillar. Klassen `Grill` ska vara en abstrakt basklass. Alla grillar ska ha en modellbeteckning, vilket är en `std::string`. Modellbeteckningen anges när en instans av någon slags grill skapas. Det finns två typer av grillar, *gasolgrillar* och *kolgrillar*. Alla sorters grillar ska spara bränslet den använder samt vilket sorts grill det är.

En gasolgrill har ett visst bränsle och ett visst antal brännare. Om antalet brännare inte anges ska det ges värdet `2`. Om bränslet inte anges är det "Propan". En gasolgrill ska kunna svara på frågan om den använder ett visst bränsle, vilket kommer i form av en `std::string`. Denna funktions signatur ska vara

`bool UsesFuel(const std::string& aSuggestedFuel) const.`

En annan funktion som bara ska finnas för gasolgrillar ska returnera antalet brännare och dess signatur ska vara

`int GetNumberOfBurners() const.`

En kolgrill har alltid bränslet kol.

Följande funktioner ska finnas för alla grillar:

`GetName` ska returnera modellbeteckningen.

`GetInformation` ska returnera modellbeteckning och bränsle för en grill.

Returvärden och inparametrar ska väljas på lämpligt vis till dessa funktioner. (10p)

3. Skriv om följande procedurrella kod för en stackliknande datastruktur till att vara objektorienterad. Du behöver inte göra felkontroller i koden och kan anta att alla anrop till externa funktioner fungerar. (6p)

```
Node* PopSpecialStack(std::vector<Node*>& aStack)
{
    Node* node = aStack.back();
    aStack.pop_back();
    return node;
}

void PushSpecialStack(std::vector<Node*>& aStack, Node* aNode)
{
    aStack.push_back(aNode);
}

Node* TopSpecialStack(std::vector<Node*>& aStack)
{
    return aStack.back();
}

bool SpecialStackIsEmpty(std::vector<Node*>& aStack) const
{
    return aStack.empty();
}

bool ElementIsOnSpecialStack (std::vector<Node*>& aStack, Node* aNode) const
{
    std::vector<Node*>::const_iterator i;
    for( i = aStack.begin(); i != aStack.end(); i++)
    {
        if( aNode == (*i) )
        {
            return true;
        }
    }
    return false;
}
```