

Tenta programmering 1



Hemtenta två

Start 10:50 fredag den 10/2 2023

Inlämning 23:59 FREDGEN den 17/2 23:59

Lämna in hemtenta i uppgiften Hemtenta två

INSTRUKTIONER

- All kod skall vara kommenterad i ett separat worddokument.
- Följ namngivningsreglerna och var konsekvent.
- Klasser och metoder skall användas.
- Följ instruktionerna och läs igenom noga vad jag frågar efter.
- Är frågan oklar, fråga!!!
- Det finns inga ursäkter för att inte lämna in i tid, ex datorn gick sönder, allt mitt arbete försvann, hunden åt upp min dator osv. TA BACKUPP!
- Om du lämnar in fel fil eller tomma filer, räknas det som din inlämning.
- Skapa ett projekt och döp ditt projekt till hemTenta2DittNamn. Lämna in tentan i Teams inlämning Hemtenta två. Zippa mappen och lämna in.
- I de uppgifter, där ni skall planera, skall dessa lämnas i wordfil. Skriv namn inne i filen!
- I era planeringar skall alla de punkter som vi gått igenom finnas. Dessa punkter återfinns i de två övningar som ni har arbetat med.

Fråga 1

Planeringsfråga

I denna uppgift skall du enbart planera ditt program.

Det program som du skall planera är schack. I din planering skall GUI planeras.

<https://help.gnome.org/users/gnome-chess/stable/chess-rules.html.sv>

De punkter som skall vara med i din planering är dessa:

1. Identifiera vad som ska göras och vilken information som ska behandlas

A) Vad skall programmet göra?

B) Vad finns på nätet?

C) Olika funktioner i programmet

D) Olika kodexempel

2. Bryt upp problemet i mindre delar och börja skriva psuedokod, med finger eller text eller en kombination

.

Fråga 2 och 3

I fråga 2&3 skall dessa punkter skrivas ner och lämnas in i ett separat dokument.

1. Identifiera vad som ska göras och vilken information som ska behandlas

A) Vad skall programmet göra?

B) Vad finns på nätet?

C) Olika funktioner i programmet

D) Olika kodexempel

2. Bryt upp problemet i mindre delar och börja skriva halvkod

3. Tills alla delar av projektet är kodade:

A) Koda en mindre del så att den fungerar med andra fungerande delar

B) Vid behov skriv om halvkod och kod så att programmet blir bättre

C) Skriv dokumentation och kommentarer för fungerande kod

D) Dokumentera vad som finns kvar att göra.

4. Gå igenom koden och se till att onödig kod, onödiga kommentarer och onödig dokumentation tas bort.

5. Om tid finns gå igenom koden och se om den går att optimera. Glöm inte arbetsgång med länkar och kodavsnitt

Fråga 2

Planering och kodning med GUI

Skapa ett program där användaren kan beställa mat och dryck. Använd lämplig teknik och förklara dina val.

Gästen skal kunna välja mellan 4 olika rätter, 4 olika drycker och 4 olika efterrätter.

Huvudrätt:

Pizza 75 kr, Vegetarisk biff 100 kr, Soppa 60 kr, Biff 165 kr

Dryck:

Vatten 45 kr, Coca-cola 55 kr, Öl 95 kr, Vin 100 kr

Efterrätt:

Paj 65 kr, Glass 55 kr, Choklad 30 kr, Tårta 65 kr

Del 1

Det som gästen väljer skall hamna i gästens beställningslista. Välj lämplig teknik.

Del 2

Summan av vad gästen har beställt skall visas på gästens beställningslista.

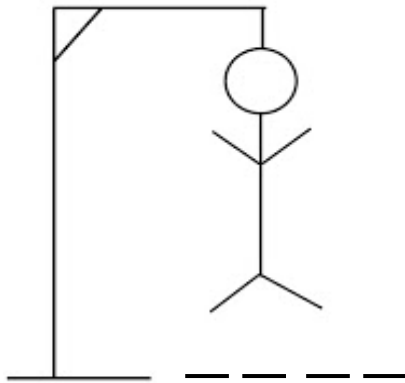
Gästen skall kunna ta bort och lägga till innan gästen klickar på beställknappen. Gästens beställning skall därefter printas ut med valda rätter och hela summan.

Fråga 3

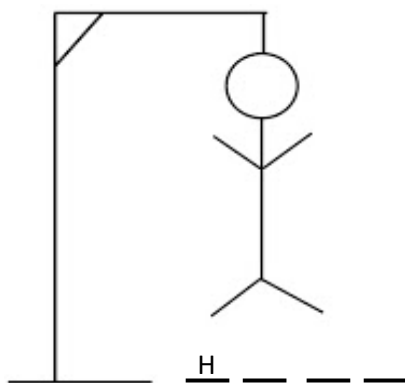
Planering och kodning med GUI

Rädda gubbe- motsats till att hänga gubbe. Gubben skall finnas hel när spelet skall börjar. För varje fel gissning, skall ex ett ben försvinna. Ordet som de skall gissa fram är hund.

1.

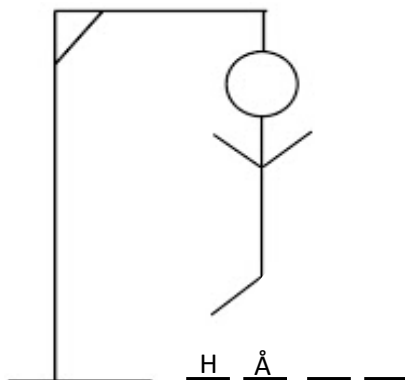


2.



Rätt gissat, alla delar är kvar av gubben

3.



Fel gissat, benet försvinner

E-nivå:

Skapa spelplan

Valfritt antal bokstäver

Enkel grafik men fullt fungerande

Slumpa ett ord utifrån en valfri lagringsplats

C-nivå

Ovan krav

Mer komplex grafik

Bokstäver spelaren har gissat på, skall placeras på spelplanen

Två spelare, varav en väljer det ord som skall gissas på.

Spelaren skall även kunna välja hur många bokstäver som ordet skall innehålla. Spelaren har 30 sekunder på sig att välja en bokstav i varje spelomgång

A-nivå: Hur bra är din kod?

1. Följ: DRY – Don't Repeat Yourself- inte WET -Write Everything Twice

2. Utvecklingsbarhet?

3. Lätt att underhålla och lätta att identifiera problem

4. Ha ett genomtänkt Mall-tänkande. Kan jag återanvända min kod till andra program?

I fall ni behöver lite avkoppling:

<https://www.goodreads.com/quotes/tag/programming>

<https://codeslaw.com/insight/10-jokes-only-programmers-will-find-funny-1110518>

https://www.youtube.com/watch?v=bzZEH_5Ouls

<https://youtu.be/1O6Qstncpnc>

