TENTAMEN
För kursen
Forsättningskurs i Programmering
TIG054

DATUM:	2020-03-18
TID:	08:00-16:00
PLATS:	Online: Egen dator
Kursansvar:	Alan B. Carlson
Förfrågningar:	031-786 2786 (mellan 08:00-09:00)
Max poäng:	60
Betygsskala:	G 40 p

OBS!!!!
Det går inte att lämna in något i efterhand!
Se till att du laddar upp dina filer i god tid!

1) Träd (6p)
Antag följande klass Tree::

```
class Tree:
    def __init__(self,data):
        self.data = data
        self.left = None
        self.right = None


    def __str__(self):
        return str(self.data)
```

Förklara hur följande kod fungerar. Visa också hur trädstrukturen som byggs upp ser ut.

```
from tree import Tree
def insert(node,data,n):
    if node == None:
        node = Tree(data)
        return node
    if (n % 2) == 0:
        node.left = insert(node.left,data,n)
        return node
    node.right = insert(node.right,data,n)
    return node

def write(node):
    if node == None:
        return
    write(node.right)
    print(node.data,end=' ')
    write(node.left)

root = None
for i in range (10):
    root = insert(root,i*2,i)

print()
write(root)
print()
```

2) Rekursion (8p)
Förklara hur följande kod fungerar och varför det som skrivs ut skrivs ut. Visa även "anropsträdet" dvs hur funktionerna kallar på varandra.

```
def funkis (str):
    if str:
        print(str)
        no_funkis (str[1:])

def no_funkis (str):
    if str:
        funkis (str[1:])
        print(str)

funkis("Hello")
print()
no_funkis("Hello")
```

3) Turtle grafik (3+1+2p)
a.) Skriv ett turtleprogram som ritar en rektangel med början på koordinat(55,55) och fortsätta programmet med att rita en kvadrat med början på koordinat(-75,-75).
b.) Var hamnar rektangeln respektive kvadraten på skärmen (tänk på att 0,0 är mitten av skärmen).
c.) Beskriv några användningsområden för Turtle grafik.

4) SQL (2+2+2+4p)
Antag följande relationer (tabeller):

Player

Name	Shoe	Racket	Country
Agassi	Nike	Wilson	USA
Federer	LA Gear	Fischer	Switzerland
Bjorkman	Puma	Slazenger	Sweden
Roddick	Keds	Spalding	USA
Coria	Adidas	Wilson	Argentina
Jeltsin	Stolichnaya	Koskenkorva	Russia
Nadal	Reebock	Prince	Spain
Hewitt	New Balance	Wilson	Australia

Tour

Tname	Name	Place
Båstad	Federer	32
Båstad	Bjorkman	1
Båstad	Nadal	2
Key Biscane	Agassi	4
Key Biscane	Roddick	4
Key Biscane	Coria	16
Key Biscane	Nadal	1
Key Biscane	Hewitt	2
Moscow	Roddick	4
Moscow	Nadal	4
Moscow	Newitt	1
Queens	Federer	4
Queens	Coria	4
Queens	Nadal	2
Queens	Hewitt	1

Svara på frågor (a-c) genom att skriva en (interaktiv) SQL-sats:
a.) Ta fram spelare som spelar med ett racket från Fischer?
b.) Vilka spelare spelar med ett Adidas racket och har vunnit en turnering (placerade sig som etta)?
c.) Vilka spelare har inte deltagit i en turnering?
d.) Antag följande pythonkod *(där tennis.dbase innehåller tabellerna ovan)*:

```
import sqlite3
conn = sqlite3.connect('tennis.dbase')
cursor = conn.cursor()
cursor.execute('''SELECT Name, Shoe from Player  where
                  Player.Country = 'USA' ''')

all_rows = cursor.fetchall()
```

Färdigställa pythonprogrammet genom att skriva kod som printar ut resultatet av utsökningen i en snygg tabell (med vänsterjusterade kolumner). Glöm inte att inkludera kolumnnamnen.

5) Grafik, generellt (1+1+2+2p)
a.) Vad innebär "cross-platform"?
b.) Vad är ett "native" UI?
c) Beskriv för- och nackdelarna med ett grafikbibliotek som genererar "native UI".
d) Beskriv i vilka situationer man kan gynnas av att *inte* ha ett "native" UI. Vilka typer av applikation hade det passat för?

6) Databaser, generellt (2+6p)

a) Beskriv fördelarna med en relationsdatabas jämfört med dokumentbaserade system (där varje dokument är en "entry" i databasen).
b) I en relationsdatabas används relationerna one-to-one, one-to-many samt many-to-many. Vilken av dessa relationer passar bäst för tabellerna (Ge en kort motivering av varje svar.):
   i) Författare - Böcker
   ii) Användare - Personnummer
   iii) Användare - Mailadress

7) Felhantering (2p)
Om någonting går fel i ett Pythonprogram under körning så kan det orsaka s.k. exceptions som stoppar programmet. Exempel på detta är när man försöker dividera med noll, eller utföra beräkningar på en variabel som innehåller annat än siffror. Hur använder man Pythons inbyggda mekanism för att fånga upp sådana här exceptions och förhindra att programmet kraschar? Skriv ut syntaxen och förklara hur det fungerar.

8.) Objektorientering (4+4+2+2+2p)
Skapa en klass som beskriver en produkt i en webbshop. Anta att klassen finns lagrad i en fil kallad product.py. Klassens instansvariabler är:
- name (produktnamn)
- stock (lagersaldo)
- price

a.) På hemsidan där produkten ska användas vill man förutom namnet på produkten även lista lagersaldo och pris. Skriv en funktion i klassen som skriver ut dessa egenskaper.
b.) Skriv funktioner i klassen som låter dig uppdatera lagersaldo och pris (en funktion för varje).
c.) Använd klassen för att skapa ett produktobjekt med några initiala hittepåvärden.
d.) Använd funktionerna du skrivit för att ändra lagersaldo och pris hos ditt nyskapade objekt.
e) Vilka fördelar finns det med att använda en objektorienterad design av sin applikation?