Skilaverkefni 4 gildir 20%
Lykilsmatsþáttur

Setjum verkefnið upp í valmynd líkt og hér fyrir neðan. Valmynd uppsetning kóða, nöfn klasa og falla og athugasemdir í kóða gilda til einkunnar **(20%)**.

1. Hringur
2. Bankareikningur
3. Spil
4. Hætta

Liður 1 (20%)

Hannið forrit sem hefur klasann Hringur.

- Klasa sem nefnist **`Hringur`**, hann hefur einn eiginleika – radius sem við gefum gildi í smiðnum.
    - Útfærðu smiðinn `__init__`
    - Útfærðu str fallið `__str__`

  Klasinn Hringur hefur tvö föll sem eru reiknuð út frá radius. Föllin eiga að hafa return þar sem þau skila því sem þau reiknuðu með einum aukastaf.
    - Fall til að reikna ummál

      ```
      def ummal(self):
      ```
    - Fall til að reikna út flatarmál hrings

      ```
      def flatarmal(self):
      ```

Inn í þessum lið gerið þið aðra valmynd sem leyfir notanda að setja inn radíus fyrir tilvikið og síðan að velja milli þess að reikna ummál eða flatarmál og prenta það snyrtilega á skjáinn.

Liður 2 (20%)

Hannið forrit sem hefur klasann **`Eigandi`** og á að halda utan um innistæðu á reikningi eigandans. Gerið fimm tilvik / instance af þessum klasa og setjið í lista, þið veljið nöfn og innistæðu

- Klasinn nefnist Eigandi, hann hefur tvo eiginleika / attribute: nafn og innistæða sem við gefum gildi í smiðnum

```
def __init__(self,nafn,innistaeda=0)
```

- Útfærðu einnig string fallið

```
def __str__(self)
```

Klasinn Eigandi hefur síðan föllin.

1. Eitt til að taka út af reikningnum, hér þarf að passa að ekki sé tekið út meira en notandinn á. Það gerum við í fallinu sjálfu

```
def uttekt(self,upphaed):
```

2. Eitt til að leggja inn á reikninginn.

```
def innlogn(self,upphaed):
```

Utan klasans eru síðan eftirfarandi föll:

1. Eyða reikning eftir nafni. Þetta fall á ekki að vera í klasanum.

```
def eyda(listi,nafn):
```

2. Skoða alla reikninga. Prentar út upplýsingar um alla reikninga. Þetta fall á ekki að vera í klasanum

```
def prenta_reikninga(listi):
```

3. Fallið finnur og skilar tilviki klasans með hæstu innistæðuna á reikningnum. Þetta fall á ekki að vera í klasanum.

```
def hæsta(listi):
```

Í lið 2 gerið þið valmynd sem leyfir notanda að velja nafn og innistæðu fyrir tilvikið/instance og bjóðið síðan upp á að gera þær aðgerðir sem hægt er í valmyndinni.

Liður 3 (40%)

Hannið forrit sem hefur klasann `Spil` hann hefur eiginleikana/attributes tegund og númer

Tegund er (hjarta, spaði, tígull, lauf)

Númer er nr. spilsins frá 1-13 (gosi nr 11, drotting nr 12 og kóngur nr 13)

Klasinn þarf að hafa föllin

- `def __init__(...)`
- `def __str__(...)`

Leiðbeiningar

- Búið til öll 52 spilin (52 tilvik/instances af klasanum) og setjið þau í listann **spilastokkur**.
- Stokkið **spilastokkinn** ykkar (listann með öllum spilunum)
- Gefið notanda 25 spil og tölvu 25 spil.
- Látið tölvu velja random tromp (hjarta,spaði, tígull eða lauf), sama trompið er síðan í öllum umferðum spilsins
- Notandi og tölva setja út til skiptis. **Notandi byrjar alltaf.**
- Látið notanda og tölvu setja út aftasta spilið í stokknum sínum og finnið út hvor vinnur slaginn.
- Látið notanda og tölvu spila og teljið hvað marga slagi hvor fær
  - Í hverri umferð á að prentast út hvaða spil bæði notandi og tölva hafa og hver vinnur
  - Haldið áfram þar til öll spil á hendi er búin! Þá þarf að tilkynna hvort notandi eða tölva vann og hvað hvor um sig fékk marga slagi.

Reglur:

Tromp vinnur allar hinar sortirnar, ef báðir eru með tromp vinnur sá með hærra trompið.

Dæmi í spili þar sem hjarta er tromp :

Ef notandi lætur út tígulþrist og tölva lætur út spaðakóng þá vinnur notandi því tígull er úti -

Ef tölva lætur út spaðadrottingu og notandi lætur út hjartatvist vinnur notandi því hjarta er tromp

Sá sem setur út ræður sem sagt alltaf litnum nema þegar notað er tromp.