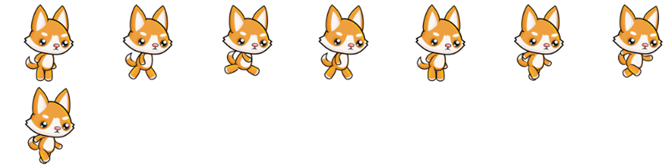
33. Час Инф. 7. разред - **Анимације** - девета недеља учења на даљину

Вероватно већ знаш да цртани филм настаје тако што се на екрану брзо смењују сличице, при чему је свака следећа слика веома слична претходној (ликови на слици су само мало померени у односу на претходни положај).

На пример, од наредних осам сличица лика у различитим положајима:



настаје анимација трчања.

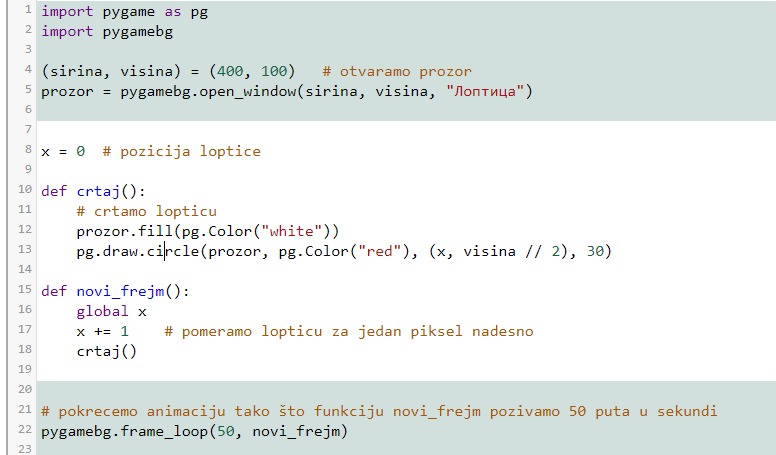
Анимације подразумевају брзу промену слике на екрану (на пример, 20 пута у секунди), обично у правилним временским интервалима (на пример, на сваких 50 милисекунди). Свака тако кратко приказана слика назива се оквир или фрејм анимације (енгл. frame). У овом делу приручника видећемо како можемо направити програме у којима се приказују неке анимације.

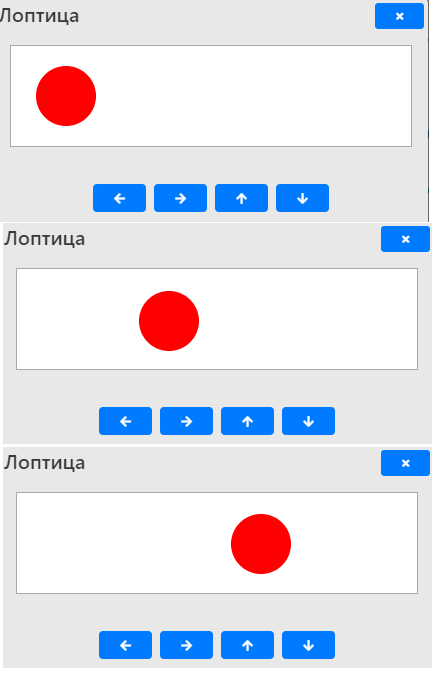
У програмима које смо до сада сретали слика се није мењала током извршавања и цртање смо вршили само једном, пре главне петље програма у којој смо чекали да корисник искључи прозор (у програмима заснованим на библиотеци PyGameBg, та петља се остварује позивом функције pygamebg.wait\_loop).

У програмима са анимацијом цртање ћемо вршити обично унутар тела те главне петље или, још боље, у засебној функцији коју ћемо на том месту позивати (у програмима заснованим на библиотеци PyGameBg, таква петља се остварује позивом функције pygamebg.frame\_loop).

Кретање лоптице

Прикажимо један пример анимације организоване на начин који смо управо описали. Написаћемо програм који анимира лоптицу која се креће са леве ка десној ивици екрана.



.....

* Потребно је да дефинишемо променљиве које ће описивати стање објеката који се анимирају. У нашем примеру то је једна црвена лоптица и пошто се она креће хоризонтално, по средини екрана довољно је да памтимо само њену координату x (то може, на пример, бити координата њеног центра, а могла би бити и, на пример, координата горњег левог темена квадрата описаног око ње). Пошто лоптица своје кретање започиње на левом крају екрана, променљиву x ћемо иницијализовати на нулу.
* Функција crtaj се сада реализује веома једноставно. У њој бојимо позадину екрана у бело и затим исцртавамо лоптицу коришћењем вредности њеног положаја x.
* При преласку на сваки нови фрејм потребно је да лоптицу померимо мало (на пример, за 1 пиксел) удесно. Дакле, у функцији novi\_frejm потребно је само да увећамо вредност променљиве x за 1. Пошто се мења вредност променљиве x која је глобална, у функцији novi\_frejm морамо променљиву x означимо помоћу кључне речи global. Након померања лоптице позивамо функцију crtaj.

Општи облик програма са анимацијама

Видели смо неколико примера и можеш приметити да смо у свима њима анимације остваривали по истом принципу, који ћеш ти примењивати и у наредним задацима.

1. Потребно је да дефинишеш глобалне променљиве којима се представљају подаци о ликовима и објектима на сцени (ти ће се подаци мењати током анимације).

2. Потребно је да дефинишеш функцију crtaj која коришћењем тих података црта сцену. У тој функцији нећемо вршити никакву промену података.

3. Потребно је да дефинишеш функцију novi\_frejm која ажурира податке о ликовима и објектима на сцени (при чему све променљиве којима се у тој функцији мења вредност морају на њеном почетку бити експлицитно означене као глобалне коришћењем кључне речи global). Након ажурирања променљивих, потребно је да позовеш функцију crtaj, да би се промењена сцена исцртала.

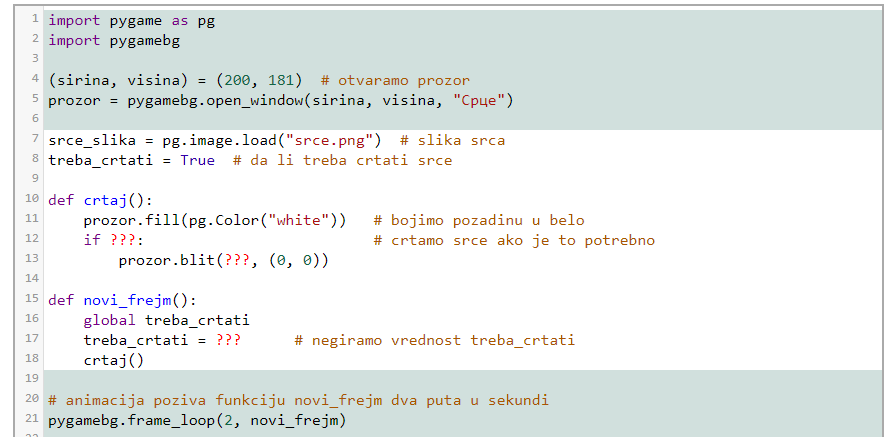
Срце које куца

Напиши програм који приказује анимацију срца које куца. Срце можеш приказати коришћењем слике srce.png.

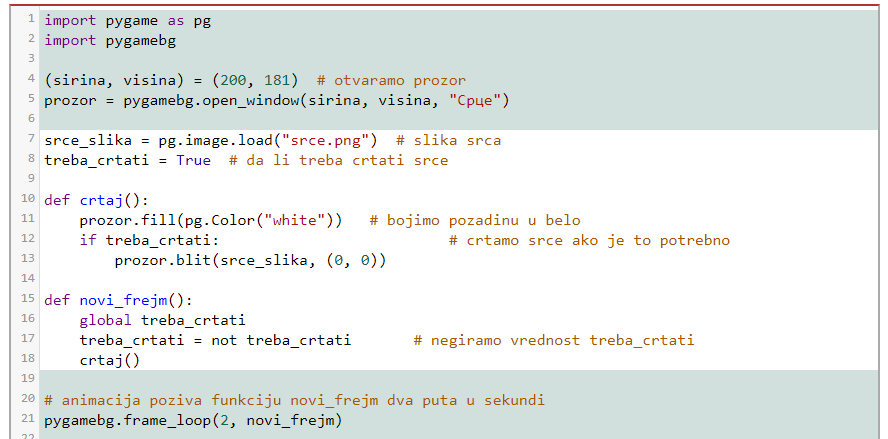
[](https://petlja.org/biblioteka/r/lekcije/_images/srce.png)

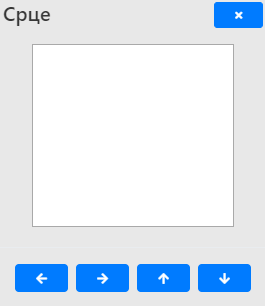
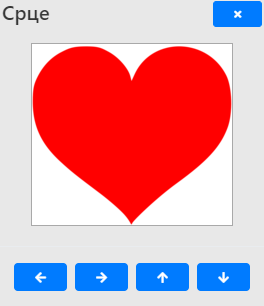
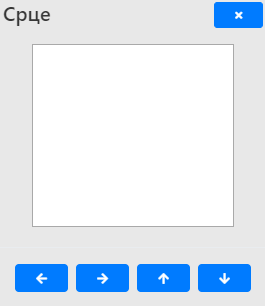
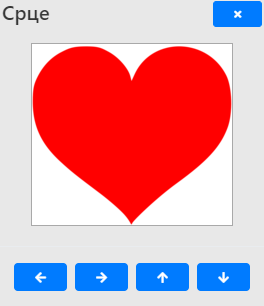
* Једини податак који се мења од фрејма до фрејма је то да ли треба или не треба приказати срце. Можемо увести логичку променљиву treba\_crtati.
* У функцији crtaj слику ћемо приказивати само ако променљива treba\_crtati има вредност тачно (тј. True).
* У функцији novi\_frejm мењаћемо вредност променљиве treba\_crtati. Ако јој је тренутно вредност True, тада треба да се промени на False, а ако јој је тренутно вредност False, тада треба да се промени на True. Најлакши начин да се то уради је да текућу вредност негирамо помоћу оператора not (наравно, можемо употребити и гранање).

Покушај да на основу овога допуниш започети програмски кôд.



Једно од могућих ресења је:



Смајлић и тужић

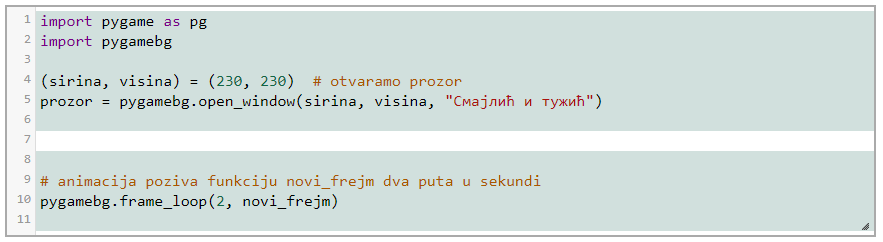
Микица често мења расположење. Час је срећна, час је тужна. Напиши програм који приказује слике смајлића и тужића које се наизменично смењују. Можеш употребити слике smajlic.png и tuzic.png.

[](https://petlja.org/biblioteka/r/lekcije/_images/smajlic.png) [](https://petlja.org/biblioteka/r/lekcije/_images/tuzic.png)

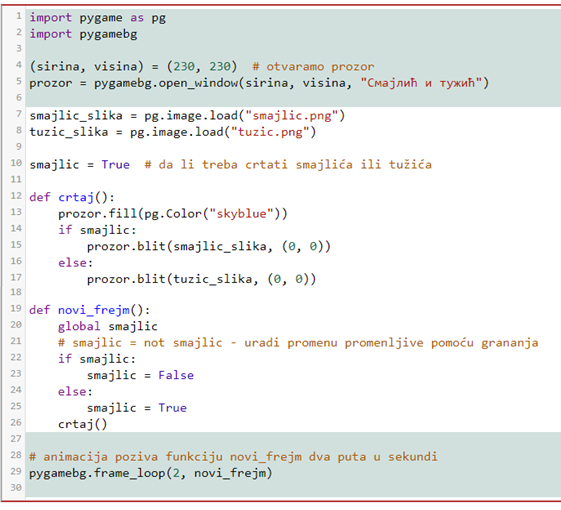
Овај задатак је поново веома сличан претходном.

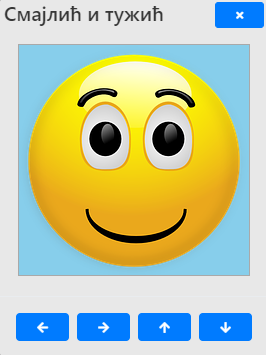
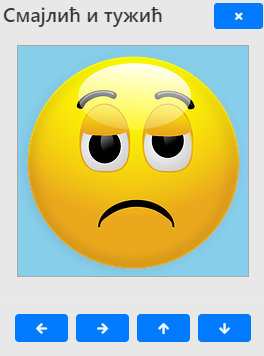
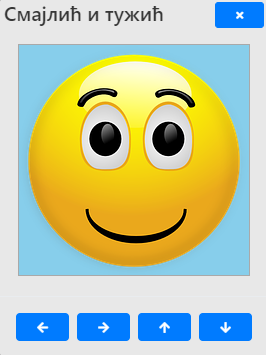
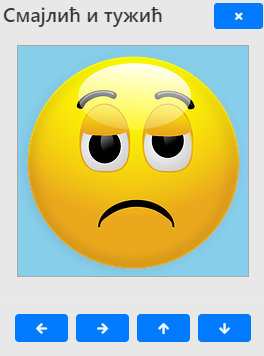
* Логичка променљива smajlic ће одређивати стање програма.
* Ако је у функцији crtaj вредност те променљиве True, приказаћемо слику смајлића, а ако је вредност False, приказаћемо слику тужића.
* У функцији novi\_frejm негираћемо вредност променљиве smajlic.

Покушај да на основу овога потпуно самостално напишеш програм. Ако не успеш, онда потражи помоћ и допуни започети програмски кôд.



**Једно од могућих решења је:**



Питања за утврђивање градива:

