
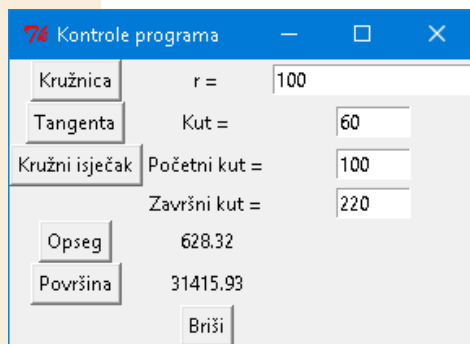


Tijekom osnovne škole upotrebljavali ste različite digitalne obrazovne sadržaje. Svaki od njih pisan je u nekom programskom jeziku. Kako izraditi takav sadržaj? Osim poznavanja programiranja treba dobro poznavati nastavni sadržaj za koji izrađujete digitalni obrazovni sadržaj.

U ovom poglavlju izradit ćete vlastiti digitalni obrazovni sadržaj o kružnici. Sastavit ćete program koji crta kružnicu polumjera R , računa opseg i površinu kruga te crta tangentu, kružni isječak i kružni odsječak

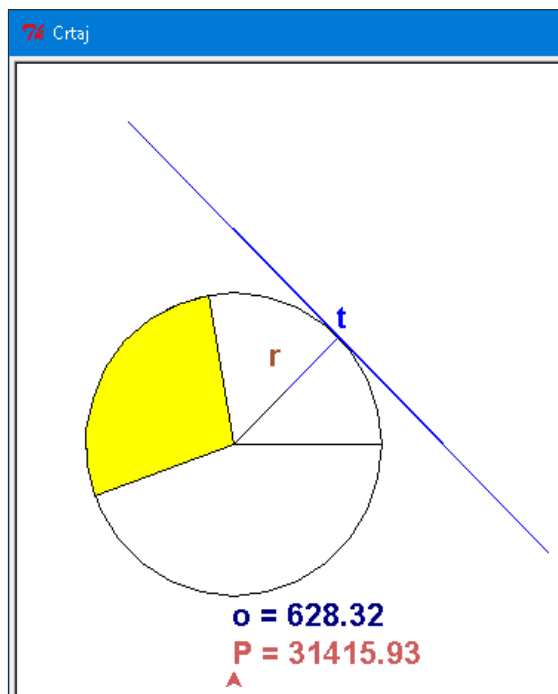
Razmislite o kojoj biste još temi mogli sastaviti digitalni nastavni sadržaj slično kao što ćemo pokazati u ovom poglavlju.

Pokrenite gotov program  **86Bkružnica.py**, da se upoznate s dijelovima korisničkog sučelja (engl. *User Interface*) i načinom rada aplikacije.



Slika 27. Kontrola programa

Primijetit ćete da su se otvorila dva prozora: osim prozora u kojem crta kornjača (slika 28.) pojavio se drugi prozor s grafičkim kontrolama (slika 27.). Grafičke kontrole već ste upotrebljavali u drugim programima i alatima, ali ne u Pythonu.



Slika 28. Crtanje kružnice, tangente, kružnog isječka i računanje opsega i površine kruga

Isprobajte kako radi program. U bijele okvire (to su okviri za unos podataka) možete upisivati brojeve, a lijevo pokraj njih su gumbi. Klikom na pojedini gumb nacrtat će se ili izračunati:

Kružnica – upišite duljinu polumjera i kornjača će nacrtati kružnicu

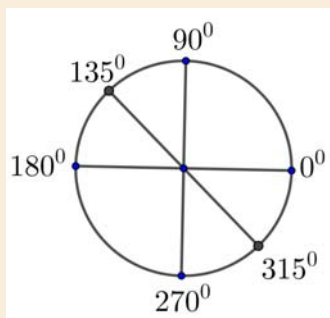
Tangenta – svaku točku na kružnici možete zadati njezinim kutom, upišite kut i u toj točki crta se tangenta

Kružni isječak – upišite početni i završni kut i nacrtat će se kružni isječak obojen žutom bojom

Opseg – računa opseg kruga

Površina – računa površinu kruga.

Za izradu ovakva složenijeg programa najprije treba razraditi algoritam i podijeliti program na manje dijelove. Prvo ćemo sastaviti potprograme koji



Slika 29. Svaka točka na kružnici zadana je kutom između 0° i 360°

crtaju kružnicu, tangentu i kružni isječak. Zatim ćemo napisati potprogram koji računaju opseg i površinu.

Prije pisanja potprograma morate znati koje ulazne vrijednosti trebate. Osim toga morate dobro poznavati gradivo koje želite prikazati programom.

1. Kružnica je zadana duljinom polumjera r . Potprogram `kružnica(r)` nacrtat će kružnicu i njezin polumjer koji spaja središte kružnice i točku označenu s 0^0 na slici 29.
2. Da bismo nacrtali tangentu u bilo kojoj točki kružnice, trebamo upisati veličinu kuta k . Potprogram `tangenta(r, k)` treba nacrtati tangentu. Tangenta je pravac koji dodiruje kružnicu u jednoj točki i okomit je na polumjer u toj točki. Zato treba istaknuti i polumjer u toj točki. U matematici je pravac beskonačna ravna crta. Treba istražiti koliku crtu treba nacrtati kornjača da se na crtežu lako uoči da smo nacrtali pravac.
3. Kružni isječak dio je kruga omeđen dvama polumjerima i kružnim lukom. Da bismo nacrtali ta dva polumjera, moramo znati u kojim se točkama kružnice nalaze polumjeri. Znači, trebamo upisati još dva broja, $kut1$ i $kut2$, pa sastaviti potprogram `kružni_isječak(r, kut1, kut2)` koji crta kružni isječak. Budući da je kružni isječak dio kruga, treba ga i obojiti.
4. Opseg kruga računa se formulom $o = 2\pi r$. U modulu `math` postoji konstanta `pi` s čak 15 decimala. Želimo li smanjiti broj decimala, u ispisu trebamo upotrijebiti naredbu `round`.
5. Površina kruga računa se formulom $P = r^2\pi$.

U Pythonu postoji naredba `circle` za crtanje kružnice. Npr. `circle(100)` nacrtat će kružnicu polumjera 100. Uočite da se kružnica crta ulijevo od trenutnog položaja kornjače. Nakon crtanja kružnice kornjača ostaje u položaju koji smo označili s 0^0 . Pomaknemo li je ulijevo za 100, bit će točno u središtu kružnice. Dopunimo li program naredbama `lt(90); fd(100); rt(90)`, kornjača će biti u središtu kružnice.

1) Kružnica

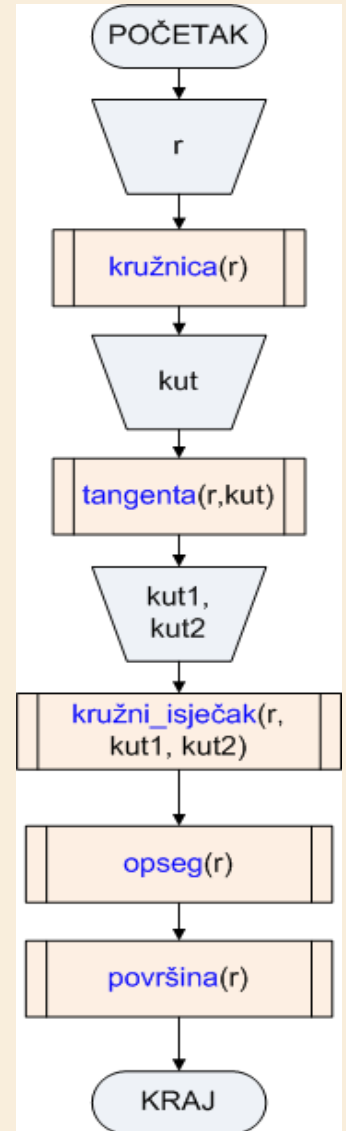
Možemo započeti pisati program koji smo gore opisali i sastaviti potprogram `kružnica(r)`. U glavnom programu upisat ćemo duljinu

Program za crtanje kružnice polumjera 100:

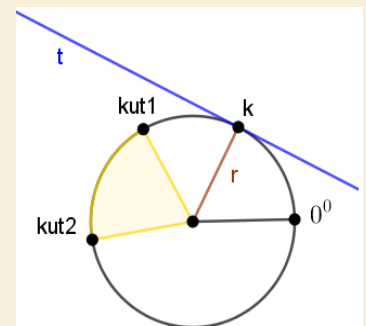
```
from turtle import *
lt(90)
circle(100)
```

```
from turtle import *
def kružnica(r):
    global x
    circle(r)
    lt(90); fd(r)
    rt(90)
    x = pos()

title('Crtaj')
st(); lt(90);
r = textinput('Duljina polumjera', 'r = ')
r = int(r)
kružnica(r)
```



Slika 30. Dijagram toka



Slika 31. Shema kružnice i ulaznih vrijednosti

`pos()` – vraća trenutni položaj kornjače

`setpos(100, -200)` – postavlja kornjaču u točku s koordinatama (100, -200) i ostavlja trag ako je pero spušteno

globalne varijable – varijable koje vrijede u čitavom programu

lokalne varijable – varijable koje vrijede samo u potprogramu u kojem se definiraju. Ulažne vrijednosti potprograma lokalne su varijable

`write` – ispisuje tekst na mjestu gdje se nalazi pero, dodatno se mogu odrediti parametri fonta

polumjera kružnice. U potprogramu se crta kružnica polumjera r i polumjer u toj točki.

Program će se sastojati od više potprograma pa je važno na kraju svakog potprograma vratiti kornjaču u središte kružnice. To je najlakše napraviti naredbom `pos()`, koja vraća trenutni položaj kornjače. Naredba `x = pos()` pamti položaj kornjače u varijabli x . Želimo li varijablu x upotrijebiti i u drugim potprogramima koje ćemo napisati, treba zadati da je x globalna varijabla pa je na početku potprograma naredba `global x`.

Spremite program i isprobajte crta li kružnicu i jedan njezin polumjer.

2) Tangenta

Dopunimo glavni program naredbama za upisivanje kuta pod kojim se crta tangenta i pozivom potprograma `tangenta(r, k)` koji crta tangentu.

```
k = textinput('Kut', 'k = ')
k = int(k)
tangenta(r, k)
```

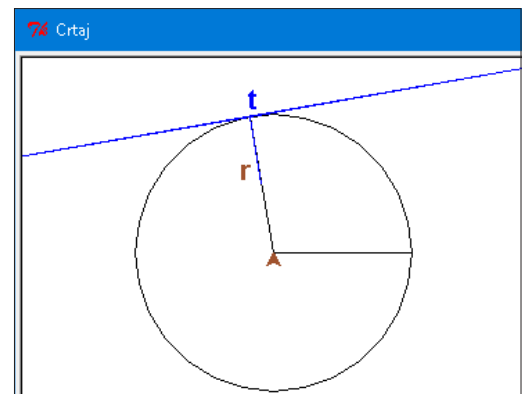
Počnimo pisati potprogram za crtanje tangente. Kornjaču treba okrenuti udesno za $90^\circ - k$, zatim joj promijeniti boju u plavu i nacrtati tangentu. Istražite koja vam duljina

```
def tangenta(r, k):
    rt(90-k); fd(r); lt(90)
    color('blue')
    fd(2*r); bk(4*r); fd(2*r)
    color('black')
    setpos(x); seth(90)
```

tangente najbolje odgovara. Svakako je važno zapamtiti da duljina tangente ovisi o duljini polumjera. U našem primjeru tangenta je duga $4r$. Vratite boju kornjače u crnu, a kornjaču vratite u središte kružnice. Naredba `setpos` ne mijenja kut kornjače. Okrenuti kornjaču prema gore najlakše je naredbom `seth` (ili `setheading`). Općenito, `seth(kut)` okreće kornjaču u smjeru kut , gdje je kut upravo ono što smo opisali na slici 29.

Ostaje napisati t ispod tangente i r ispod polumjera. Naredba `write` omogućuje pisanje kornjačom: `write('t')` ispisat će t na mjestu gdje se na-

```
def tangenta(r, k):
    rt(90-k); fd(r); lt(90)
    color('blue')
    fd(2*r); bk(4*r); fd(2*r)
    write('t', font = ('Arial', 16, 'bold'))
    rt(90); bk(r/2); pu()
    lt(90); fd(15); pd()
    color('Sienna')
    write('r', font = ('Arial', 16, 'bold'))
    pu(); setpos(x); seth(90); pd()
```



Slika 32. Tangenta na kružnicu pod kutom od 100°

lazi kornjača, a želimo li zadati vrstu pisma, veličinu i oblik, to pišemo iza, opisujući parametar font:

```
write('t', font=( 'Arial', 16, 'bold' ))
```

Za ispis *r* pomičemo kornjaču na sredinu polumjera i malo je odmičemo od crte.

3) Kružni isječak

Dopunimo glavni program s naredbama za upisivanje početnog i završnog kuta kružnog isječka. Zatim pozovemo potprogram kružni_isjecak.

Napišite sada potprogram za crtanje kružnog isječka. Ulazne vrijednosti potprograma moraju biti duljina polumjera *r* te početni i završni kut kružnog isječka: *kut1* i *kut2*. Kružni isječak treba ispuniti žutom bojom. Zato na početku potprograma zadajte boju kornjače naredbom `color('black', 'yellow')`, gdje je 'black' boja crtanja, a 'yellow' boja ispunje. Da bi kornjača ispunila isječak, treba joj to najaviti: naredba `begin_fill()` piše se na početku, zatim se nacrtat isječak i napiše `end_fill()`. Kornjača će ispuniti dio koji je nacrtala između `begin_fill()` i `end_fill()`.

Kornjača se nalazi u središtu kružnice. Okrećemo je u smjeru *kut1* `seth(kut1)`, crtamo polumjer `fd(r)` i okrećemo je u smjeru kružnice `lt(90)`. Sada treba nacrtati kružni luk. Dodamo li naredbi `circle` *kut* kao drugu ulaznu vrijednost, nacrtat će kružni luk. Npr. `circle(100, 60)` crta kružni luk duljine polumjera 100 od 0° do 60°.

Znači, `circle(r1, kut2-kut1)` crta kružni luk od *kut1* do *kut2*.

Na kraju crtamo i drugi polumjer `lt(90); fd(r)` i završavamo ispunu: `end_fill()`.

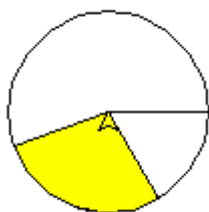
Provjerite radi li potprogram ispravno za sve vrijednosti *kut1* i *kut2*. Promotrite slike od 33. do 40.

```
kut1 = textinput('Početni kut', 'kut1 = ')
kut1 = int(kut1)
kut2 = textinput('Završni kut', 'kut2 = ')
kut2 = int(kut2)
kružni_isjecak(r, kut1, kut2)
```

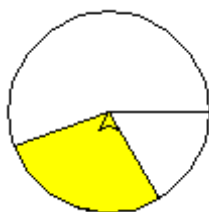
boja kornjače – `color` (boja crte, boja ispunje)

```
def kružni_isjecak(r, kut1, kut2):
    color('black', 'yellow')
    begin_fill()
    seth(kut1);
    fd(r); lt(90)
    circle(r1, kut2-kut1)
    lt(90); fd(r)
    end_fill()
    seth(90)
```

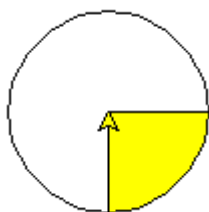
ispuna bojom
početak ispunje: `begin_fill()`,
kraj ispunje: `end_fill()`



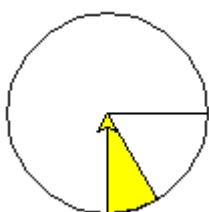
Slika 33. Kružni isječak za $r = 50$, $kut1 = 200$, $kut2 = 300$



Slika 34. Kružni isječak za $r = 50$, $kut1 = 300$, $kut2 = 200$



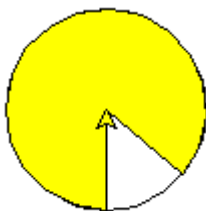
Slika 35. Kružni isječak za $r = 50$, $kut1 = -90$, $kut2 = 0$



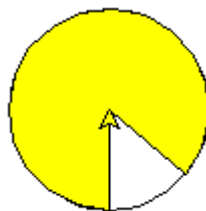
Slika 36. Kružni isječak za $r = 50$, $kut1 = -60$, $kut2 = -90$



Slika 37. Kružni isječak za $r = 50$, $kut1 = 90$, $kut2 = 400$



Slika 38. Kružni isječak za $r = 50$, $kut1 = -90$, $kut2 = -400$



Slika 39. Kružni isječak za $r = 50$, $kut1 = -90$, $kut2 = -400$



Slika 40. Kružni isječak za $r = 50$, $kut1 = 60$, $kut2 = 440$

Isprobajte u prozoru *Python Shell*:

```
>>> 3<4 or 0==5-5
True
>>> 3>4 or 0==5-5
True
>>> 3==4 or 5==5-5
False
```

Logička funkcija **or** daje **True** ako je barem jedan uvjet istinit, a **False** ako su oba uvjeta lažna.

```
def kružni_isjecak(r, kut1, kut2):
    while kut1<0 or kut1>=360:
        kut1 = textinput('Početni kut', 'kut1 = ')
        kut1 = int(kut1)
    while kut2<0 or kut2>=360:
        kut2 = textinput('Završni kut', 'kut2 = ')
        kut2 = int(kut2)
    color('black', 'yellow')
    ...
```

Zadate li $0 < kut2 < kut1 < 360$, program će ipak nacrtati ispravan kružni isječak (slike 34, 38 i 39).

Za $kut1 < 0$ i $kut2 < 0$ program radi ispravno, ali to ćete učiti u srednoj školi. Zato je bolje tražiti da se upišu $kut1 \geq 0$ i $kut2 \geq 0$.

Svaka točka na kružnici može se zadati kutom od 0° do 360° , dapače za 0° i 360° dobije se ista točka na kružnici. Ako je kut veći od 360° , onda se luk „namata“ na kružnicu. Tako se za 40° dobije ista točka kao za 400° jer je $400 \% 360 = 40$. Ako upišete $kut1 = 60$, $kut2 = 440$, vjerojatno želite da se nacrtaju kružni isječak od 60° do 80° jer je $440 \% 360 = 80$. Ali, pogledajte sliku 40. Ispunjen je čitav krug. Zato potprogram `kružni_isjecak` treba izmijeniti tako da za $kut1$ i $kut2$ vrijedi: $0 \leq kut1 < 360$ i $0 \leq kut2 < 360$.

Ako su u glavnom programu upisane vrijednosti za $kut1$ i $kut2$ koje ne zadovoljavaju gornje uvjete, u potprogramu se traži ponovni upis $kut1$ sve dok je $kut1 < 0$ ili $kut1 \geq 360$. Isto tako i za $kut2$.

Upotrijebit ćete petlju `while` i logičku funkciju `or`.

4) Opseg kruga

Želimo li upotrijebiti konstantu π za broj π , na početku glavnog programa treba unijeti modul `math` i kasnije pozvati potprogram `opseg(r)`.

```
from math import *
.....
opseg(r)
```

U potprogramu treba izračunati opseg. Neka se opseg ispiše ispod kružnice. Budući da je kornjača u središtu kružnice, treba je pomaknuti unazad za $25+r$, tako će neovisno o r opseg biti ispisan 25 točaka ispod kružnice. Slično kao u potprogramu tangenta, opseg ispisujemo naredbom `write`, samo što sada osim znakova `'o = '` treba ispisati i vrijednost varijable `o` – zato je prvo treba pretvoriti u string `str(o)` pa dodati na `'o = '`. Evo potprograma:

```
def opseg(r):
    o = 2*r*pi
    pu(); bk(25+r); pd()
    color('navy')
    write('o = '+str(o),font=('Arial',16,'bold'))
```

5) Površina kruga

Ostaje još na sličan način napraviti potprogram površina(r) za računanje površine kruga. Na kraj glavnog programa dodajemo poziv potprograma:

```
površina(r)
```

Potprogram za računanje površine kruga:

Primijetite da su se opseg i površina ispisali s puno decimala:

```
def površina(r):
    P = r*r*pi
    pu(); bk(25); pd()
    color('indianred')
    write('P = '+str(P),font=('Arial',16,'bold'))
```

Konstanta pi zadana je s 15 decimala. Želite li to izbjeći, treba upotrijebiti naredbu round. Isprobajte u prozoru *Python Shell*:

```
>>> from math import *
>>> round
(45.454545, 2)
45.45
>>> round
(45.4555555, 3)
45.456
```


round – zaokružuje prvu ulaznu vrijednost na broj decimala zadan drugom ulaznom vrijednosti. Izmijenite ovako naredbe za računanje o i P želite li točnost na dvije decimale:


```
o = round(2*r*pi, 2)
```

```
P = round(r*r*pi, 2)
```

 Čitav program nalazi se u datoteci: **86Bvje14.py**.

Dopunite program tako da crta i kružni odsječak. Kružni odsječak dio je kruga omeđen tetivom i kružnim lukom. Sastavite potprogram kružni_odsječak.

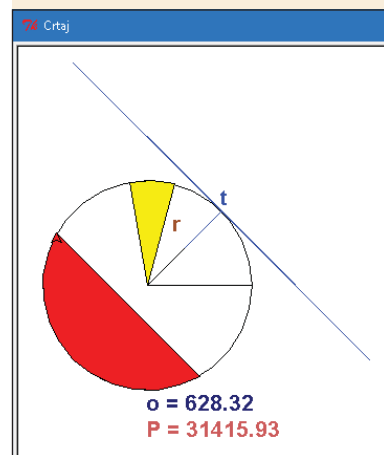
 Rješenje u datoteci: **86Bvje15.py**.

 Kako to sve napraviti i grafičkim kontrolama, možete pročitati u datoteci **86Bgraf_kontrole.pdf**.

Nakon što ste završili svoj digitalni sadržaj o kružnici, program možete ponuditi drugim učenicima da njime crtaju tangente, kružne isječke i računaju opseg i površinu kruga. Da bi se programom mogao služiti drugi korisnik, on ne mora znati kako ste vi sastavili sve potprograme i koje ste naredbe

```
o = 628.3185307179587
P = 31415.926535897932
```

Slika 41. Ispis opsega i površine za r = 50



Slika 42. Kružni odsječak crvene boje

Na stranici informatika.azoo.hr/kategorija/3/Razvoj-sofтверa pogledajte što sve treba sadržavati dokumentacija programa u kategoriji Razvoj softvera.

upotrijebili, već samo što program omogućuje i što korisnik programa treba upisati da bi dobio željeni rezultat. Zato programeri nakon što sastave program i testiraju ga pišu dokumentaciju programa koja sadržava sljedeće dijelove:

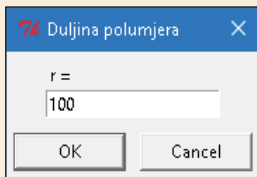
- **Uvod** – na početku treba opisati ideju programa
- **Detaljan opis rada** – u ovome dijelu treba pokazati kako funkcionira rješenje i koje su mu osobine te slikama ilustrirati svaki pojedini funkcionalni dio. Ako programsko rješenje zahtijeva određene ulazne podatke, posebice ako postoje ograničenja u vezi s njihovim vrijednostima, to treba navesti u ovome dijelu (npr. koliko smije biti najveći/najmanji r i slično)
- **Tehničke informacije** – u zadnjem dijelu treba opisati kako je rješenje napravljeno, koje su tehnologije upotrijebljene i kakva je konfiguracija sustava potrebna za primjenu ponuđenog rješenja.

Za program koji smo izradili u datoteci **86Bvje14.py** možete napisati:

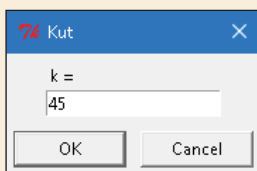
- **Uvod:**
- Program crta kružnicu, tangentu i polumjer u diralištu tangente, kružni isječak i računa opseg i površinu kruga.

- **Detaljan opis rada:**

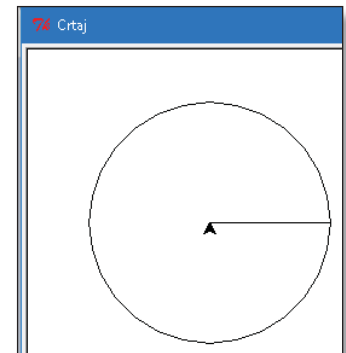
1. Otvorite i pokrenite program **86Bvje14.py** pritiskom na tipku F5.
2. U dijaloški okvir Duljina polumjera upišite duljinu polumjera kružnice i kliknite OK. Nacrta se kružnica zadanog polumjera.
3. U dijaloški okvir *Kut* upišite kut tangente i kliknite OK. Nacrta se tangenta i polumjer u diralištu.



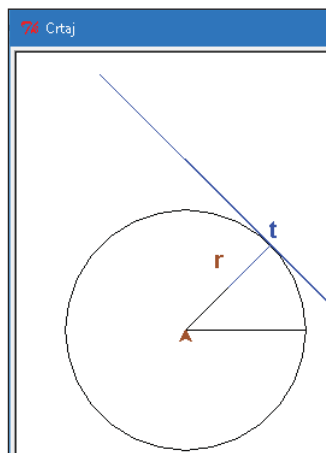
Slika 43. Upis duljine polumjera



Slika 45. Upis kuta tangente



Slika 44. Kružnica

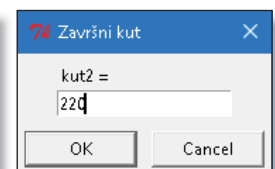


Slika 46. Tangenta i polumjer u diralištu

4. U dijaloški okvir *Početni kut* upišite početni kut kružnog isječka *kut1* i kliknite OK, a u dijaloški okvir *Završni kut* upišite završni kut kružnog isječka *kut2* i kliknite OK. Ulazne vrijednosti *kut1* i *kut2* trebaju biti veće ili jednake 0 i manje od 360. Inače će



Slika 47. Upis početnog kuta kružnog isječka



Slika 48. Upis završnog kuta kružnog isječka

program tražiti ponovno upisivanje sve dok se ne upišu brojevi koji zadovoljavaju: $0 < kut1 < 360$ i $0 < kut2 < 360$. Nakon upisanih podataka nacrtat će se kružni isječak određen upisanim kutovima.

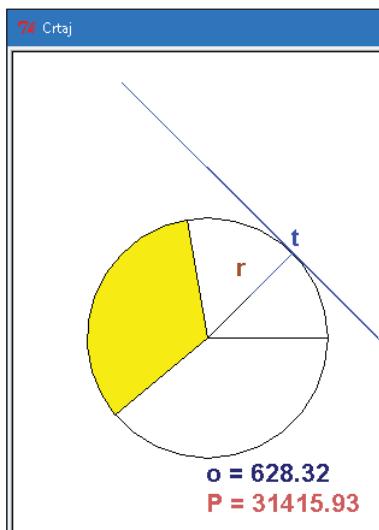
I, na kraju, ispod kružnice ispisat će se veličine opsega i površine kruga.

- Postoje ograničenja duljine polumjera. S jedne strane, za jako male vrijednosti polumjera r (npr. $r < 10$) nacrtat će se tako mala kružnica da će korisnik teško uočiti ostale dijelove crteža. S druge strane, za $170 < r < 350$ kornjačin prozor treba maksimizirati da bi se u njemu vidjela čitava kružnica, a za $r > 350$ neće se nacrtati cijela kružnica. Ali, granične vrijednosti za r ovise i o razlučivosti i veličini monitora na kojem radite, pa se mogu razlikovati od ovdje navedenih.

- **Tehničke informacije:**

Opisati hardverske i softverske, tj. systemske preduvjete za ispravan rad programa. Na primjer:

Program je napravljen u programskom jeziku Python, verzija 3.3. Prije pokretanja programa na računalu morate imati instaliran Python 3.3. Instalaciju Pythona možete preuzeti s poveznice <https://www.python.org/downloads/windows>.



Slika 49. Nacrtan kružni isječak i ispisani opseg i površina kruga