*LANACOIntroduction to C#*

UVODNI KURS

# **Lab3** Datum:24.2.2017.god *.*

a) Napisati klasu *Tacka* u C#-u:

1. Klasa treba da sadrži sledeće promenljive/polja:

* *x* (cijeli broj > 0 u metrima) – vidljivo na nivou klase
* *y* (cijeli broj > 0 u metrima) – nije vidljivo na nivou objekta

1. Za sve promenljive kreirati odgovarajuće property-e (voditi računa o prosleđenim vrijednostima):

* *X*
* *Y*
* *X\_km* (vraća i postavlja vrijednost u kilometrima)
* *Y\_km* (postavlja vrijednost u kilometrima)

1. Kreirati dva konstruktora:

* Jedan kome se prosleđuju svi parametri.
* Defaultni koji postavlja početne vrijednosti na 1000m i 1000m za x i y respektivno.

1. Napisati dvije metode:

* *Ispis()* – vraća string u sledećem formatu: X: <x u m> m; Y: <y u m> m

Npr.: X: 150 m; Y: 175 m

* *IzracunajUdaljenostOdKoordinatnogPocetka* – ova funkcija računa udaljenost od koordinatnog početka po formuli: 𝑈𝑑𝑎𝑙𝑗𝑒𝑛𝑜𝑠𝑡=

1. Predefinisati operator explicitne konverzije u *float* (i ostale operatore ako je potrebno) tako da vraća udaljenost od koordinatnog početka.
2. Predefinisati operator poređenja > tako da poredi dvije tačke naspram udaljenosti od koordinatnog početka.

b) Napisati klasu *Putanja*:

1. Klasa treba da sadrži sledeće promenljive/polja:

* *tačke* (niz tačaka) – vidljivo na nivou objekta
* *naziv* (neprazan string) – vidljivo i na nivou objekta i na nivou klase

1. Za sve promenljive kreirati odgovarajuće property-e (voditi računa o prosleđenim vrijednostima):

* *NizTacaka* (*Read* *Only*)
* *Naziv*

1. Kreirati konstruktor kome se prosleđuju svi parametri.
2. Napisati dvije metode:

* *Ispis*() – vraća string u sledećem formatu: <rbr tacke>. X: <x u m> m; Y: <y u m> m<novi red>

Npr.: 1. X: 150 m; Y: 175 m

2. X: 300 m; Y: 175 m

**…**

* *IzracunajUdaljenostOdKoordinatnogPocetka* – ova funkcija računa sumu udaljenosti svih tačaka od koordinatnog početka.

1. Predefinisati operator implicitne konverzije u *float* (i ostale operatore ako je potrebno) tako da vraća ukupnu udaljenost od koordinatnog početka.

c) U *Main* metodi omogućiti korisniku da unese jednu putanju sa proizvoljnim brojem tačaka (unosa tačaka prekinuti kada se unese -1 i -1 za x i y).

d) Pronaći i ispisati najbližu i najudaljeniju tačku od koordinatnog početka.

\* *Napomena*: koristiti predefinisane operatore.