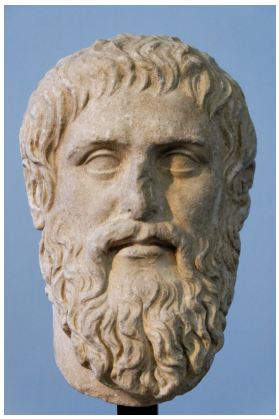


# Logika

```
string kaVeikti;

if (mokiCpp)
{
        kaVeikti = “Mokytis naujų dalykų”;
} else {
        kaVeikti = “Skaityt storas knygas”;
}
```

## Svarbiausi loginiai operatoriai

- **&&** AND ir
- **||** OR arba irba*
- **==** EQUALS lygu
- **!** NOT ne
- **!=** NOT EQUAL nelygu
    - Taip pat **> < >= <=** ir t.t.

## Priskyrimo operatoriai

- Priskyrimas **a = 5;**
- Matematinės operacijos **b = a + 3 / 4 * (5 - 6);**
    - **a += 2;** **b -=1;** **c *= 3;** **d /= 4;**

## Kintamųjų tipai

- **int** 32 bitų sveikieji skaičiai
- **float, double** 32 ir 64 bitų racionalieji skaičiai (su kableliu)
- **bool** loginis kintamasis. true / false
- **string** raidžių kratinys “žodis”
- **T[]** masyvas. Pvz. **int[] skaiciai = {1, 2, 3};**
    - Gali būti sudaryti iš bet kokio kito tipo kintamųjų
    - **T[][]** dvi dimensijos, **T[][][]** trys dimensijos ir t.t.
        - **int[][] aaa = new int[2][];**
          **aaa[0] = new[] { 1, 2 };**
          **aaa[1] = new[] { 3, 4 };**

* http://rokiskis.popo.lt/2012/09/16/arba-kalbainiu-nekompetencija-irba-tinginyste/

# For ciklas

Naudojamas kai reikia kartoti tą pačią operaciją n kartų.

```
int pasikartojimuKiekis = 5;
int faktorialas = 1;
for (int a = 0; i < pasikartojimuKiekis; i++)
{
        faktorialas *= a + 1;
}
```

Pavyzdys su 2 dimensijų masyvu:

```
int[][] arr;
for (int i = 0; i < arr.Length; i++)
{
        for (int j = 0; j < arr[i].Length; j++)
        {
                Console.Write(arr[i][j]);
        }
}
```

# While

Naudojamas kai iš anksto nežinoma kada užsibaigs ciklas.

```
while (true)
{
        If (ArJauSustoti())
        {
                Console.WriteLine(“valio”);
                break; // sustojam
        }
        Console.WriteLine(“dar palaukiam”);
        // anksti dar
}
```

Break yra ciklo terminatorius. Jis naudojamas norint išeiti iš bet kokio ciklo. Tačiau reikia prisiminti, kad jei esame dvigubame (ar dar gilesniame) cikle, mes išeisime tik iš vieno ciklo.

# Metodai

Kiekvienas metodas turi atitikti tam tikrą šabloną kuris deklaruoja kaip metodas elgsis ir kaip jis bus pasiekiamas. Paanlizuokime Main metodą iš praeitos pamokos:

**public static void Main(string[] args)**

**public** - metodas yra atviras kitoms klasėms. Atvirkštinis variantas yra private, tada metodas būtų pasiekiamas tik klasės viduje.

**static** - metodas priklauso pačiai klasei, o ne jos instancijai. Kadangi mes pradedame nuo statinio metodo, jis ir gali matyti tik kitą statinį kontekstą, nebent jo viduje mes susikuriame kitos klasės instanciją (apie tai bus gerokai daugiau informacijos vėliau). Dabar nėra būtina suprasti kas tai yra ir kodėl tai naudojama, tiesiog reikia turėti omenyje, kad ateityje šio operatoriaus reikės vis mažiau.

**void** - metodas negražins nieko. T.y. mes nesitikime gauti visiškai jokio rezultato, nors metodo vidus ir gali pakeisti programos būseną. Norint gražinti kažkokį tipą nurodome jo pavadinimą pvz. **int KiekManMetu()**, **string KoksManoVardas()**. Nors kartais labai norisi, bet gražinti galima tik vieną reikšmę. Norint gražinti kelias reekšmes rekomenduojama tam naudoti objektus.

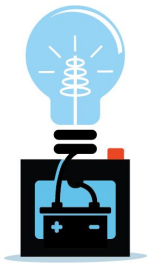
**Main** - yra tiesiog pavadinimas. Tačiau jis yra privalomas kaip C# programos pradžios dalis. Savo metodus galite vadinti kaip norite, tačiau rekomenduojama laikytis CamelCase principo, kur žodžiai rašomi be tarpų ir pradedami didžiosiomis raidėmis.

**(string[] args)** - įvesties argumentai. Šiuo atveju tai yra stringų masyvas. Atskiri parametrai skiriami kableliais. Pvz. **(string vardas, string pavarde, int amzius)**.

# Laivų mūšis

Klonuokite mūsų repozitoriją ir pabandykite paleisti programą. Ji yra nepilna. Jūsų užduotis yra panaudoti tai kas paminėta aukščiau* ir parašyti šaudymo logiką.
Rekomenduojame pradėti nuo paprastų ir galbūt kvailų sprendimų ir juos palaipsniui tobulinti.

*Jei norite generuoti atsitiktinius skaičius, tai yra daroma taip:
**Random generator = new Random();**
**generator.Next(pradineRiba, galutineRiba);**



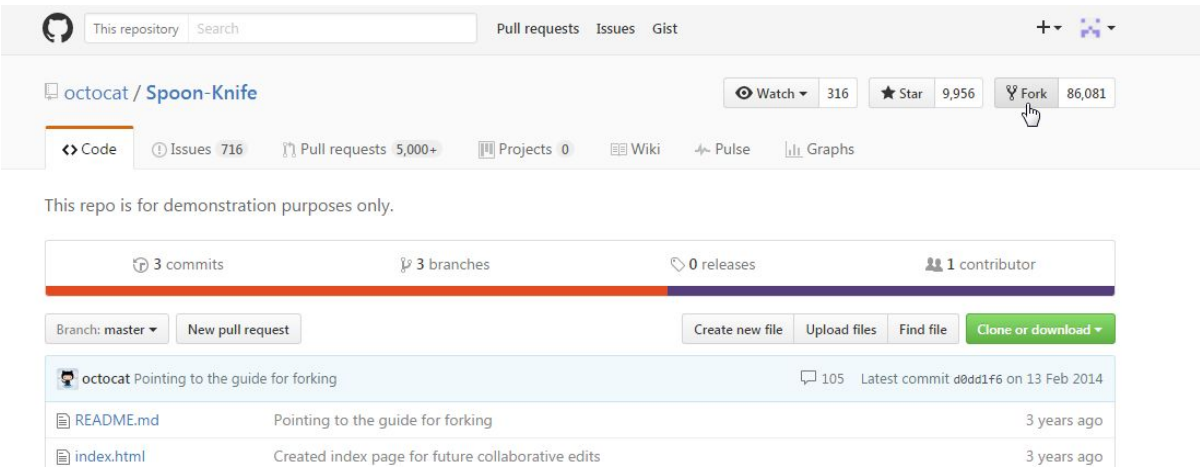
## Ką reiškia “fork a repository”?

“Forkindami” pasiimate kodą iš kažkieno kito repozitorijos ir nusikopijuojate į savo repozitoriją. Taip galite eksperimentuoti su savo turima kodo kopija, nekeisdami originalaus kodo iš svetimos repozitorijos.

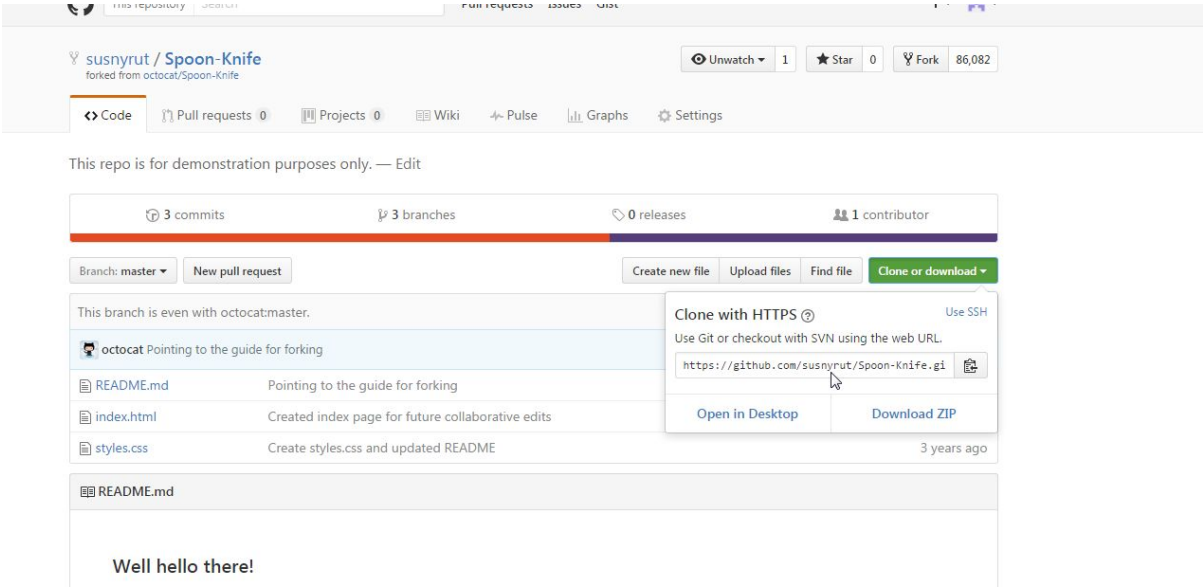
# Kaip nusiforkinti repositoriją?

Komandas darbui su Git galima rašyti komandinėje eilutėje arba naudotis programa Git extensions. Kaip geriau? Priklauso nuo tavęs. Mes mokysime naudotis Git extensions nes tai yra paprasčiau pradedantiesiems.

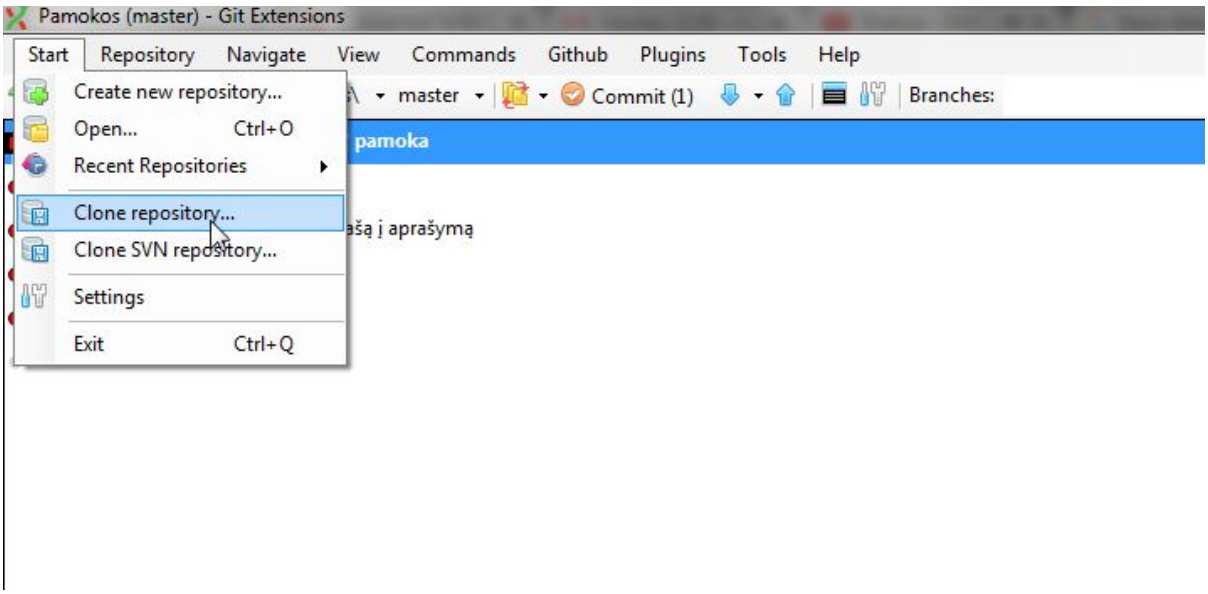
1. Einam https://github.com/susnyrut/Pamokos (paveikslėlyje kitas adresas, bet veiksmai bus tie patys). Spauskit “Fork”, iššokusiam lange pasirinkit savo account’ą ir palaukit.

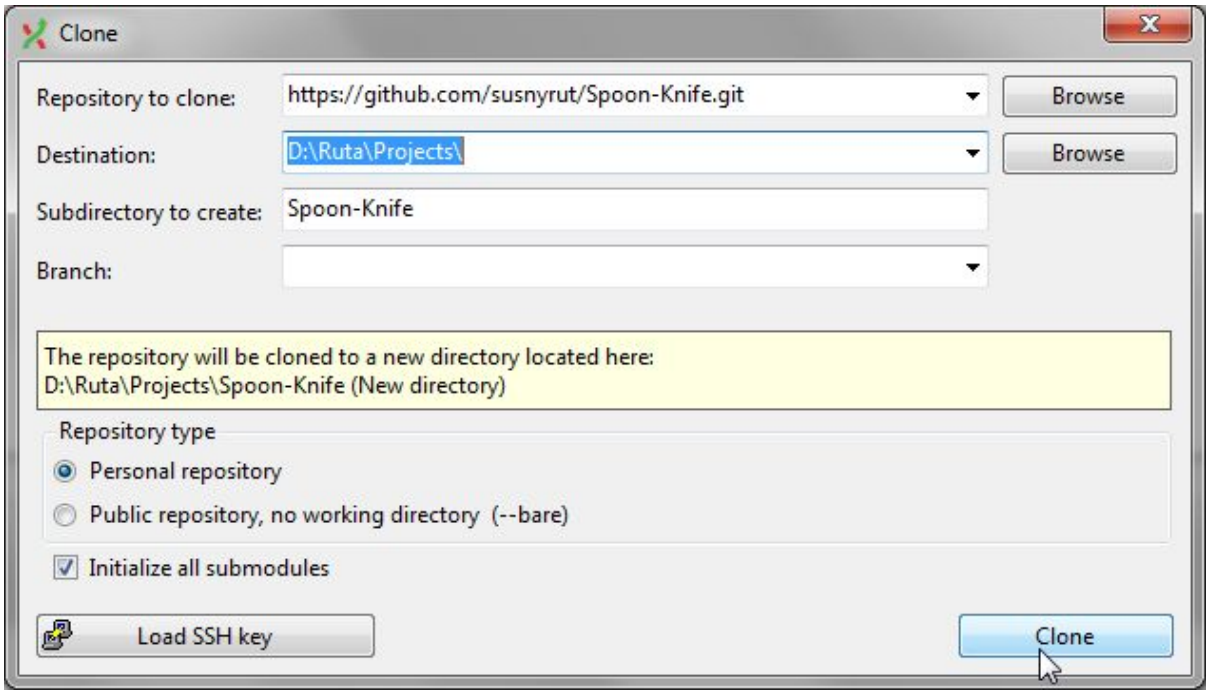


2. Dabar repozitorija pamokos atsidūrė tavo paskyroje ir turėtum ją matyti naršyklėje. Nusikopijuojam repozitorijos adresą:




3. Atsidarom Git extensions ir klonuojam repozitoriją į savo kompiuterį, kad galėtume su kodu dirbti lokaliai:






## Angliškas aprašymas:

https://help.github.com/articles/fork-a-repo/