VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA

**Lola kodo generavimas naudojant Blockly**

**Lola Code Generation Using Blockly**

Kursinis darbas

Atliko: 4 kurso 1 grupės studentas

Marius Pozniakovas (parašas)

Darbo vadovas: Irus Grinis (parašas)

Vilnius – 2021

**Turinys**

[Įvadas 3](#__RefHeading___Toc3955_2905141138)

[1 1dalykas 4](#__RefHeading___Toc3957_2905141138)

[1.1 Super-rezoliuciniai generatyviniai priešiški tinklai (SRGAN) 5](#__RefHeading___Toc3959_2905141138)

[1.2 Generatyviniai priešiški stilių tinklai (StyleGAN, StyleGAN2) 7](#__RefHeading___Toc3961_2905141138)

[1.3 Cikliniai generatyviniai priešiški tinklai (CycleGAN) 9](#__RefHeading___Toc3963_2905141138)

[2 Veidų ir eskizų generavimas 11](#__RefHeading___Toc3965_2905141138)

[2.1 Veido generavimas 12](#__RefHeading___Toc3967_2905141138)

[2.2 Eskizo generavimas 13](#__RefHeading___Toc3969_2905141138)

[2.3 Sukurto veido generavimo algoritmo analizė 14](#__RefHeading___Toc3971_2905141138)

[2.4 Veidų ir eskizų generavimo apibendrinimas 16](#__RefHeading___Toc3973_2905141138)

[3 Veido eskizo generavimas naudojant GAN technologijas 17](#__RefHeading___Toc3975_2905141138)

[3.1 Naudojamos technologijos 17](#__RefHeading___Toc3977_2905141138)

[3.2 Duomenų pasirinkimas 18](#__RefHeading___Toc3979_2905141138)

[3.3 Architektūros pasirinkimas ir apmokymo metodas 19](#__RefHeading___Toc3981_2905141138)

[4 Veido eskizo generavimo bandymas 21](#__RefHeading___Toc3983_2905141138)

[Išvados 24](#__RefHeading___Toc3985_2905141138)

[Literatūros sąrašas 25](#__RefHeading___Toc3987_2905141138)

# Įvadas

Dažnas savamokslis programuotojas savo kelią pradeda nuo aukšto lygio - interpretuojamų (angl. *high-level*) programavimo kalbų (*Python, JavaScript, iš dalies C#*), praleisdami (angl. *low-level)* žemo lygio - asembliuojamas ar kompiliuojamas (*C, Assembly)* kalbas. To priežastis yra ta, kad aukšto lygio kalbos yra lengviau suprantamos, skaitomos, internete lengviau rasti pagalbos, o žemo lygio kalbos viską atlieka be įmantrių jau sukurtų (ir programuotojui labai patogių naudojimui) funkcijų, tačiau taip bendruoju atveju pagreitėja programos veikimo laikas. Programuojant žemo lygio kalbomis pramokstama dirbti su programos atmintimi, geriau suprantamas pats procesorius ir aukšto lygio kalbos pasidaro daug lengvesnės.

Aukšto ir žemo lygio programavimo kalbos galėtų būti sulygintos su mašina, turinčia skirtingo tipo pavarų dėžes. Norint saugiai naudotis automobiliu, kuris turi mechaninę pavarų dėžę (kuri šiuo atveju atitinka žemo lygio programavimo kalbą) reikia daug daugiau žinių, tačiau patyręs vairuotojas turės žymiai didesnę mašinos kontrolę, o mažiau patirties turintis vairuotojas gali labai lengvai ir greitai apgadinti pavarų dėžę.

Tiesa, mokymosi resursų ir programų nėra daug, todėl pradžia gali būti sunki. Būtent iš to ir išplaukia šio **darbo temo aktualumas –** žemesnio lygio kalbų mokymosi resursų trūkumas. Šiuo kursiniu darbu analizuosime žemo lygio (angl. *low-level*) techninės įrangos aprašymo kalbą (angl. *HDL)* *Lola* ir jos palengvinimą norintiems ją išmokti.

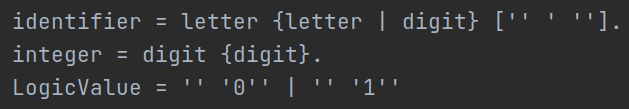
**Darbo tikslas** – išanalizuoti ir teoriškai techininės įrangos aprašymo kalbos *Lola* generavimą panaudojantvieną iš tinkamiausių įrankių vizualiam programavimui (angl. *visual programming)* – *Blockly.* Tikslui pasiekti formuojami **uždaviniai:**

* Teoriškai išanalizuoti ir aprašyti:
  + techninės įrangos aprašymo kalbą *Lola*
  + *JavaScript* bibliotekos įrankį *Blockly*
  + *Lola* kodo generavimą panaudojant *Blockly*

# **Lola**

* 1. HDL
  2. Lola
     1. Identifikatoriai (angl. *identifiers)*, sveikieji skaičiai (angl. *integers)*, loginės reikšmės (angl. *logical values*) ir komentarai (angl. *comments*)

*Lola* kalba naudoja identifikatorius aprašyti konstantas, kintamuosius ir tipus:



***paveiksliukas***

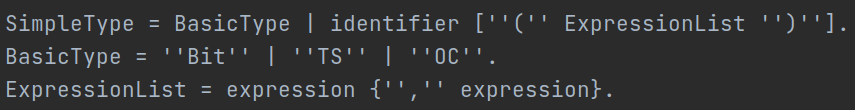
Komentarai yra simboliai atskiriami skliaustu su žvaigždute. Jie gali būti pridėti bet kurioje kodo vietoje.



***paveiksliukas***

* + 1. Elementarūs ir paprasti tipai (angl. *basic and simple types*)

Visi kintamieji *Lola* kalboje turi tipą*.* Tipas gali būti paprastasis *(basic)*, jau priskirtas (*predefined*) arba struktūrizuotastipas *(structured type)* jau apibrėžtas programiniame kode. Paprastas tipas gali būti *BIT, TS* arba *OC*, kurie skiriasi tik jų **rules govening assignment**. (TS atvaizduoja **tri-state bus**, OC atvaizduoja **open-collector bus)**. Paprastojo tipo (*basic*) kintamieji turi loginę reikšmę ir yra aprašomi ‘0 arba ‘1.

 ***paveiksliukas***

* + 1. Masyvai ir jų tipai (angl. *array types)*

# **Blockly**

# **Lola kodo generavimas su Blockly**

Lola

pav. 1 *test image*

# Išvados

Isvados

isvados

isvados

# Literatūros sąrašas

[test18] Author Name. How three french students used borrowed code to put the first AI portrait in Christie‘s <https://www.theverge.com/2018/10/23/18013190/ai-art-portrait-auction-christies-belamy-obvious-robbie-barrat-gans>, 2018