Rīgas 34. vidusskola

Piekļuves nosacījums Programmēšana II kursā

Evolūcijas un dabiskas izlases virtuāla simulācija.

Darba autore: 12.a klases skolniece

Anastasija Voropajeva

Darba vadītāja: Pedagogs Programmēšana II

Mg. Math., Mg. Paed. Marina Juzova

Mārupe, 2024

Saturs

[Ievads 3](#_Toc164708979)

[1. Problēmas izpēte un analīze 3](#_Toc164708980)

[2. Programmatūras prasību specifikācija 3](#_Toc164708981)

[2.1 Programmēšanas Valoda 3](#_Toc164708982)

[2.2 Uzdevumi: 4](#_Toc164708983)

[2.3 Tehniskās detaļas: 4](#_Toc164708984)

[2.4 Mērķauditorija: 4](#_Toc164708985)

[2.5 Specifikācija: 4](#_Toc164708986)

[2.5.1 Ievaddati: 4](#_Toc164708987)

[2.5.2 Ievaddatu tips, mērvienības, iespējamās vērtības: 5](#_Toc164708988)

[2.5.3 Ievaddatu iesniegšanas veids: 5](#_Toc164708989)

[2.5.4 Rezultātu iegūšanai: 6](#_Toc164708990)

[2.5.6 Izvaddatu tips, mērvienības, iespējamās vērtības: 7](#_Toc164708991)

[2.5.7 Piemēri: 7](#_Toc164708992)

[3. Programmatūras izstrādes plāns 9](#_Toc164708993)

[4. Programmatūras kods, kas veidots ievērojot labās prakses principus. 9](#_Toc164708994)

[5. Atkļūdošanas un akcepttestēšanas pārskats 12](#_Toc164708995)

[5.1.Vienībtestēšana: 12](#_Toc164708996)

[5.2.Akcepttestēšana 12](#_Toc164708997)

[6. Lietotāja ceļvedis 13](#_Toc164708998)

[7. Piemērotās licences pamatojums 14](#_Toc164708999)

# Ievads

Evolūcijas Simulācijas Programmatūra ir instruments, kas paredzēts evolūcijas procesu simulēšanai virtuālā vidē. Šajā dokumentācijā ir sniegts pārskats par programmatūru, tās funkcionalitāti, instalācijas instrukcijām un lietotāja vadlīnijām.

# 1. Problēmas izpēte un analīze

Evolucionārās bioloģijas pētījumi bieži vien prasa sarežģītu evolucionāro scenāriju simulēšanu, lai saprastu ģenētiskās variācijas, dabisko atlasi un pielāgošanos dinamiku. Ikgadēju eksperimentu veikšanas tradicionālajās reālās pasaulēs ir laika un resursu patēriņš. Tādēļ ir nepieciešama programmatūra, kas var efektīvi un precīzi simulēt evolucionāros procesus.

Lai noteiktu, kas nepieciešams manai evolūcijas simulācijai es izmantoju tādas izpētes metodes, kā literatūras pārskats un pilotprojekti. Pirmkārt, es veicu plašu literatūras pārskatu, kas ietver pētījumu par esošajām evolūcijas simulācijām un to izmantošanu zinātniskos un industriālos projektos. Šis pētījums palīdzēja man iegūt pamatzināšanas par evolūcijas procesiem un identificēt esošus risinājumus. Turklāt es arī izmantoju pilotprojektus, lai pārbaudītu savas idejas un pieejas reālā vidē. Šie pilotprojekti ļāva man novērtēt ideju efektivitāti un atbilstību maniem mērķiem, nodrošinot pamatu galīgās programmas izstrādei.

# 2. Programmatūras prasību specifikācija

## 2.1 Programmēšanas valoda

Programmatūra ir izstrādāta, izmantojot Python, augstākā līmeņa programmēšanas valodu, kas pazīstama ar tās vienkāršību un lasāmību. Python ir objektorientētā valoda, kas automātiski dzēš norādes uz izlietoto kodu, atbrīvojot atmiņu. Bezmaksas un atvērtā pirmkoda programmatūra, kurā var brīvi izmantot bibliotēkas, kuras nemitīgi pieaug. Nodrošina ērtu datu bāzes apstrādi, piemēram, ar SQLite.

## 2.2 Uzdevumi:

1. Izpētīt zinātniskus avotus par evolūciju un dabisko atlasi
2. Izstrādāt algoritmus ģenētiskas variācijas un mutācijas simulēšanai.
3. Implementēt dabiskās atlases mehānismus un vides faktoru modeļus.
4. Nodrošināt vizualizācijas rīkus, kas atspoguļo evolūcijas dinamiku.
5. Pievienot iespēju pielāgot simulācijas parametrus un veikt personalizētus eksperimentus.
6. Nodrošināt datu glabāšanu un analīzi, ieskaitot rezultātu apkopošanu un atskaišu ģenerēšanu.

## 2.3 Tehniskās detaļas:

* API, dizaini: tkinter
* Datu bāze: SQLite
* Platforma: Visual Studio Code

## 2.4 Mērķauditorija:

* Bioloģijas Pētnieki: Programma būs noderīgs rīks bioloģijas pētniekiem, kuri vēlas simulēt un analizēt dažādus evolūcijas scenārijus, lai saprastu ģenētiskas variācijas un dabiskās atlases dinamiku.
* Izglītības Iestādes: Skolotāji un studenti varēs izmantot programmu, lai vizualizētu evolūcijas procesus un veiktu eksperimentus, kas palīdz izprast evolūcijas principus un mehānismus.
* Programmētāji un Izglītības Entuziasti: Programma sniegs iespēju iepazīties ar evolūcijas algoritmiem un simulācijas tehnoloģijām, kā arī pielāgot un paplašināt tās funkcionalitāti, veidojot savus eksperimentus un pielāgojot algoritmus.

## 2.5 Specifikācija:

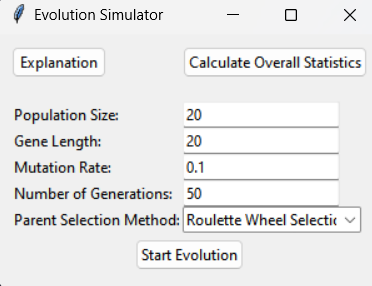
### 2.5.1 Ievaddati:

1. **Population Size (Populācijas izmērs):** Nosaka radību skaitu katrā paaudzē. Šo vērtību var mainīt, lai pielāgotu simulāciju atbilstoši vēlamajai radību daudzumam paaudzē.
2. **Gene Length (Gēnu garums):** Noteikts katras radības ģenētiskā koda garums. Jo garāks gēns, jo vairāk iespēju dažādībai un pielāgošanās potenciālam.
3. **Mutation Rate (Mutācijas ātrums):** Nosaka varbūtību, ka kāds gēns mutēsies (mainīsies) reprodukcijas laikā. Jo augstāka mutācijas ātruma vērtība, jo lielāka iespēja uzģenerēt jaunas ģenētiskas variācijas.
4. **Number of Generations (Paaudžu skaits):** Noteikts kopējais paaudžu skaits, kurā norisināsies simulācija. Jo vairāk paaudžu, jo ilgāka ir simulācija un vairāk iespēju evolūcijas procesam izpausties.
5. **Parent Selection Method (Vecāku atlases metode):** Ļauj izvēlēties starp dažādām metodēm vecāku radību izvēlēšanai reprodukcijas procesā. Pieejamās metodes ir Ruletes griešana un Turnīra atlase.

### 2.5.2 Ievaddatu tips, mērvienības, iespējamās vērtības:

1. **Population Size (Populācijas izmērs):**
   * Ievaddatu tips: vesels skaitlis (int)
   * Mērvienības: Radību skaits
   * Iespējamās vērtības: Pozitīvi veseli skaitļi
2. **Gene Length (Gēnu garums):**
   * Ievaddatu tips: vesels skaitlis (int)
   * Mērvienības: Gēnu garuma vienības
   * Iespējamās vērtības: Pozitīvi veseli skaitļi
3. **Mutation Rate (Mutācijas ātrums):**
   * Ievaddatu tips: Decimālskaitlis (float)
   * Mērvienības: Procenti vai decimālskaitļi
   * Iespējamās vērtības: Decimālskaitļi intervālā no 0 līdz 1 vai procenti no 0% līdz 100%.
4. **Number of Generations (Paaudžu skaits):**
   * Ievaddatu tips: vesels skaitlis (int)
   * Mērvienības: Paaudzes skaits
   * Iespējamās vērtības: Pozitīvi veseli skaitļi

### 2.5.3 Ievaddatu iesniegšanas veids:

1. **Ievades lauki**: lietotājiem ir iespēja ievadīt dažādus skaitliskus datus, piemēram, populācijas lielumu, ģenētisko garumu, mutācijas ātrumu un ģenerāciju skaitu, izmantojot tekstveida ievades laukus.
2. **Izvēles rūtiņas**: lietotāji var izvēlēties no dažādiem pieejamajiem iespēju variantiem, piemēram, vecāku atlases metodi - "Roulette Wheel Selection" vai "Tournament Selection".

### 2.5.4 Rezultātu iegūšanai:

Sākumā lietotājs ievada vairākus parametrus, piemēram, populācijas lielumu, ģenētisko garumu, mutācijas ātrumu un ģenerāciju skaitu, izmantojot programmā esošos ievades laukus un izvēles rūtiņas. Kad lietotājs ir ievadījis visus nepieciešamos parametrus un noklikšķinājis uz pogas "Start Evolution", programma sāk evolūcijas simulāciju, pamatojoties uz norādītajiem ievades datiem.

Simulācijas laikā programma ģenerē sākotnējo populāciju, izmantojot nejauši ģenerētas ģenētiskās vērtības katram radījumam. Pēc tam, izmantojot izvēlēto vecāku atlases metodi, kā arī krustojumu un mutāciju operācijas, tiek radītas jaunas paaudzes radījumi, kuri tiek novērtēti pēc to piemērotības, izmantojot noteiktu “fitness” funkciju.

Katrā paaudzē tiek atlasīti labākie radījumi, kuri tiek uzglabāti un turpmāk izmantoti evolūcijas procesā. Tas tiek atkārtots, līdz sasniedz maksimālo ģenerāciju skaitu, kuru norādījis lietotājs.

2.5.5 Atbilde izvade:

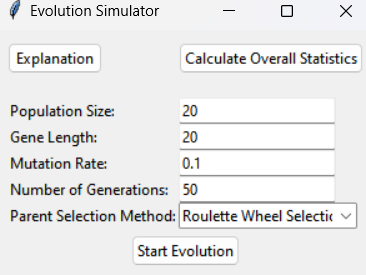
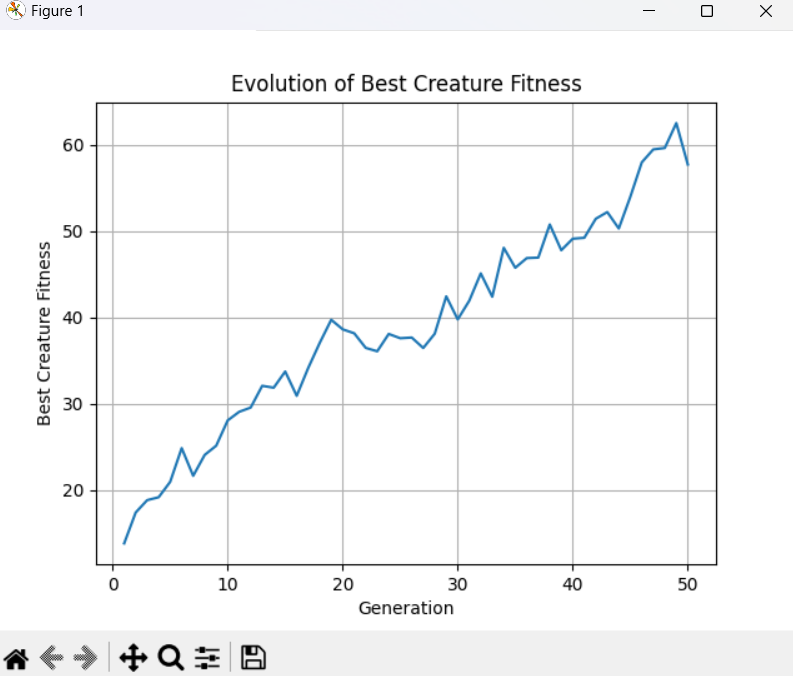
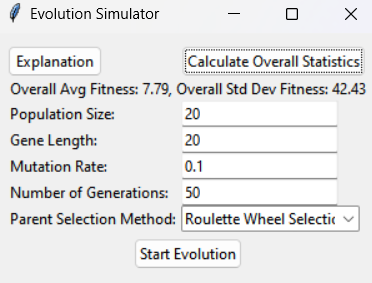
Kad ir pabeigts evolūcijas process, programma attēlo rezultātus, piemēram, labāko radījumu, tā “fitness” vērtību, kā arī citus statistikas datus, kā arī attēlo grafiku, kurā redzama labāko radījumu “fitness” vērtības attīstība pēc paaudzēm.

Tādējādi programma ļauj lietotājiem dinamiski iegūt izvades datus, kas atspoguļo evolūcijas procesa rezultātus, pamatojoties uz norādītajiem ievades datiem un veikto simulāciju.

### 2.5.6 Izvaddatu tips, mērvienības, iespējamās vērtības:

1. **Labākā radība un tās fitness vērtība:**
   * Ievaddatu tips: Teksts (string) un decimālskaitlis (float)
   * Mērvienības: Nav noteiktas
   * Iespējamās vērtības: Labākās radības apraksts (piemēram, svars un citi svarīgi parametri) un tās fitness vērtība, kas norāda, cik labi radība ir pielāgota konkrētajam uzdevumam. Fitness vērtība parasti ir decimālskaitlis, kur lielāka vērtība nozīmē labāku pielāgošanos.
2. **Grafiks ar labāko radību fitness attīstību:**
   * Ievaddatu tips: Grafika objekts
   * Mērvienības: Nav noteiktas
   * Iespējamās vērtības: Grafiks parāda labākās radības fitness attīstību pār paaudzēm vai citu laika periodu. X ass parasti attēlo paaudzes vai laika periodus, savukārt Y ass attēlo fitness vērtības.
3. **Vidējā fitness vērtība pār paaudzēm:**
   * Ievaddatu tips: Decimālskaitlis (float)
   * Mērvienības: Nav noteiktas
   * Iespējamās vērtības: Vidējā fitness vērtība parasti ir decimālskaitlis, kas norāda uz visu paaudžu vai laika periodu vidējo pielāgošanos. Tas var būt noderīgs, lai novērtētu kopējo uzdevuma risināšanas progresu.

### 2.5.7 Piemēri:

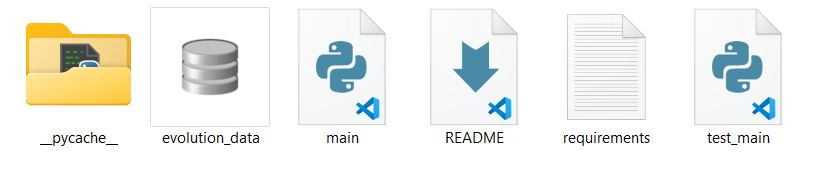


# 3. Programmatūras izstrādes plāns

Simulācijas izstrāde tika veikta, izmantojot “Agile” modeļa iteratīvo pieeju. Sākot ar plānošanu un analīzi, es identificēju mērķus un prioritizēju funkcionalitāti. Izstrādes laikā es pakāpeniski paplašināju risinājumu, veicot regulāru testēšanu un novērtējot rezultātus. Ar regulāru atsauksmju saņemšanu un pielāgošanos es nodrošinājām, ka gala produkts atbilst klienta vajadzībām. Uzņemšanas un uzturēšanas posmā es nodrošināju nepieciešamo dokumentāciju un atbalstu.

# Programmatūras kods, kas veidots ievērojot labās prakses principus.

Manas simulācijas programmatūras kods ir veidots, ievērojot labās prakses principus, kas ietver:

1. **Atdalītas funkcijas:** Programmatūras kods ir sadalīts mazākās funkcijās vai moduļos, kas veicina kodu saprotamību, atkārtotu izmantojamību un uzturēšanu.
2. **Komentāri:** Kods ir labi dokumentēts ar komentāriem, kas skaidri izskaidro katru funkciju, procesu vai sarežģītu loģiku, kas palīdz saprast kodu citiem izstrādātājiem un uzlabo kopējo uzturēšanas procesu.
3. **Readme fails:** Projekta sakņu direktorijā ir "README" fails, kurā ir sniegta vispārīga informācija par projektu, tā mērķiem, struktūru un to, kā to palaist vai iestatīt.
4. **Requirements fails:** Projekta sakņu direktorijā ir "requirements.txt" fails, kas uzskaita visus nepieciešamos bibliotēku un atkarību nosacījumus, lai nodrošinātu, ka projekts var tikt uzstādīts un palaists pareizi.
5. **Vienkāršība un saprotamība:** Kods ir rakstīts, ņemot vērā vienkāršību un saprotamību, izmantojot skaidrus nosaukumus, pietiekami ilgus komentārus un loģisku kodu struktūru. Sk. attēls 6.
6. **Tiek izmantotas relācijas datubāze:**

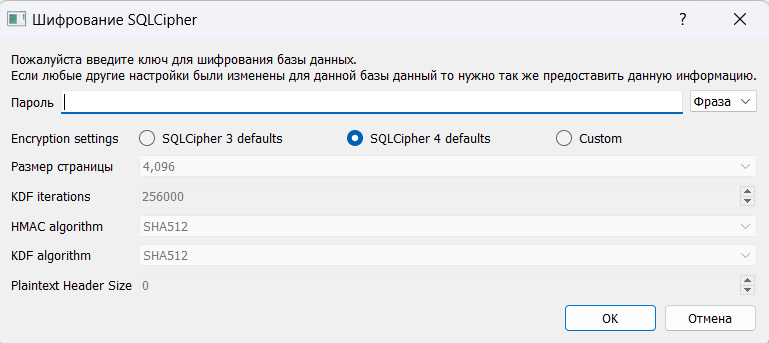
Ar relācijas tipu viens pret vienu starp laukiem “generation” vieglākai datu apstrādei un statistikai.

Tabula nosaukums: evolution\_results

|  |  |  |
| --- | --- | --- |
| Lauka nosaukums | Datu tips | Piemērs |
| generation | INTEGER | 1 |
| best\_creauture | TEXT | {'weights\_input\_hidden': array([[-2.28946095, -0.14526671, 0.44230048, 0.64240046, 0.65375795],  [ 1.00992632, -1.15355424, 0.29569425, 1.82307774, -0.29805377],  [-0.58257672, -0.65245443, 0.85460946, -0.59702246, -0.64270104]]), 'weights\_hidden\_output': array([[ 0.8910665 ],  [-0.70532673],  [ 0.51924879],  [-0.39768956],  [-0.7298999 ]])} |
| fitness | REAL | -1.06192454389851 |

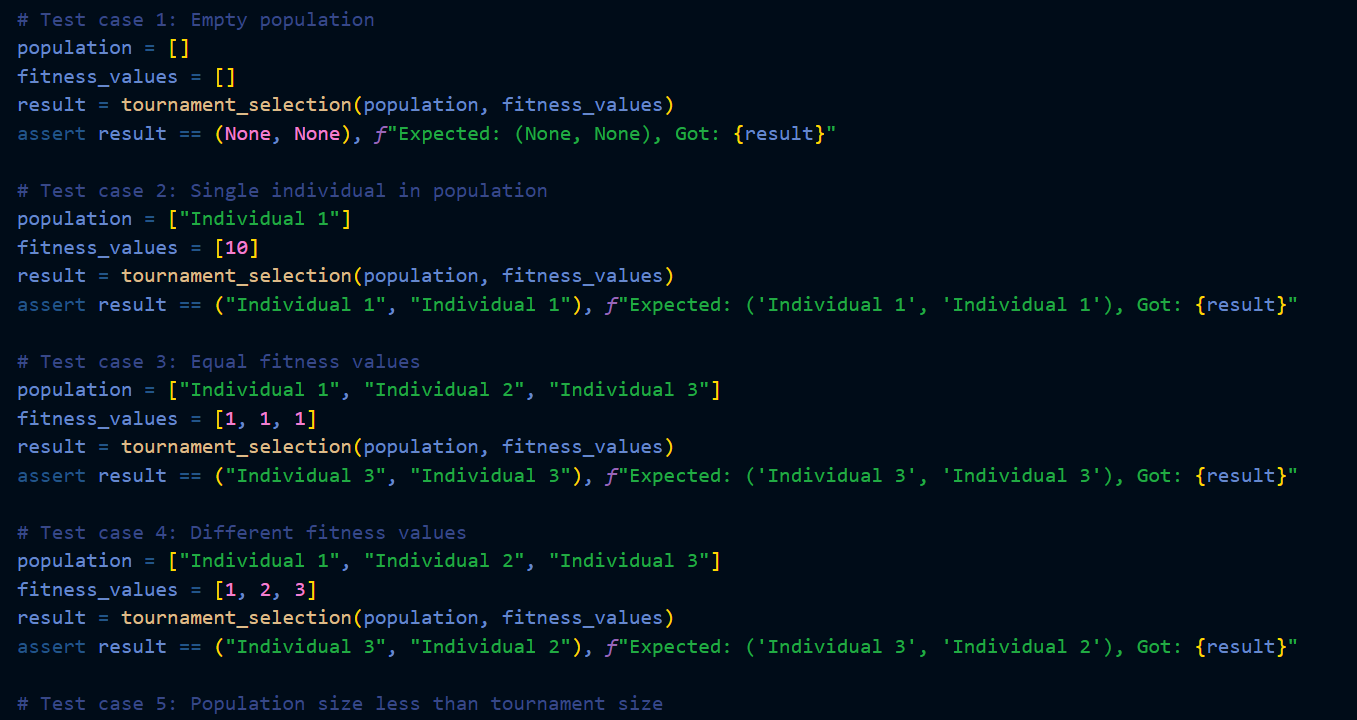
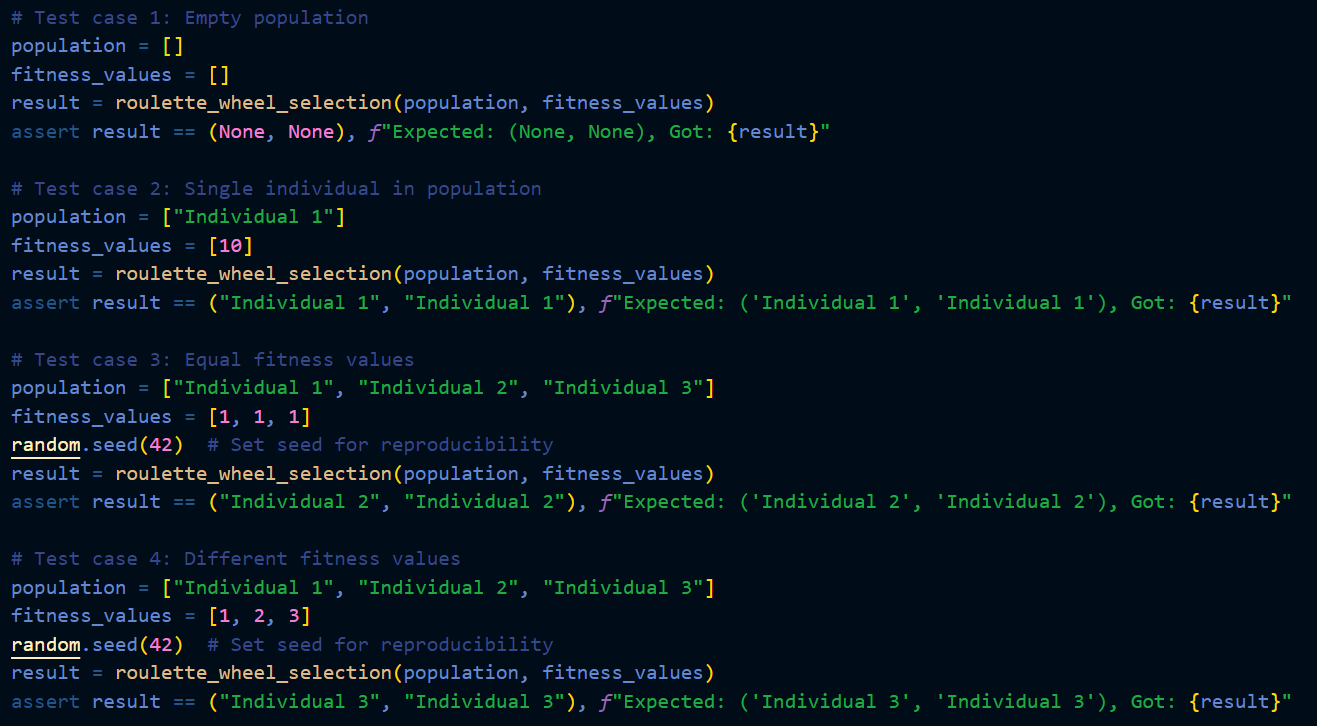
Tabulas nosaukums: evolution\_results

|  |  |  |
| --- | --- | --- |
| Lauka nosaukums | Datu tips | Piemērs |
| generation | INTEGER | 1 |
| avg\_fitness | REAL | 0.772087002046385 |
| max\_fitness | REAL | 6.27633495490171 |

1. Datu bāzes aizsardzībai tiek pielietoti kriptogrāfijas paņēmieni. Datu bāzes piekļuvei tiek pieprasīta parole, nodrošinot datu drošību. Piemēram uz doto brīdi parole ir “321”.

# 5. Atkļūdošanas un akcepttestēšanas pārskats

## 5.1.Vienībtestēšana:

Es izmantoju vienībtestus, lai pārbaudītu katru funkciju un metodi, nodrošinot, ka kods darbojas pareizi un atbilst paredzētajai funkcionalitātei. Vienībtesti tiek izpildīti automātiski un palīdz atklāt kļūdas agrīnā posmā, ļaujot mums veikt nepieciešamos labojumus un uzlabot kodu kvalitāti un uzticamību.

## 5.2.Akcepttestēšana

1. **Galveno funkcionalitāšu pārbaude**: Sākotnēji es veicu rūpīgu pārbaudi, lai nodrošinātu, ka galvenās funkcijas, piemēram, ģenerēšanas process, krustojums, mutācija un vecāku atlase, darbojas korekti un atbilst specifikācijām. Nekādas kļūdas netika atrastas.
2. **Scenāriju un datu testēšana**: Veicot testēšanu, es izmantoju dažādus scenārijus un reālistiskus datus, lai pārliecinātos, ka simulācija darbojas pareizi dažādos apstākļos un ar dažādiem ievades datiem.
3. **Izņēmuma gadījumu apstrāde**: Lai nodrošinātu drošu un uzticamu darbību, es veicu testēšanu arī ar izņēmuma gadījumiem un kļūdām, lai pārliecinātos, ka simulācija pareizi reaģē uz neparedzētiem apstākļiem un korekti apstrādā kļūdas.
4. **Lietotāja pieredzes novērtēšana**: Es arī pievērsu uzmanību lietotāja pieredzei, pārbaudot, vai lietotāja saskarne ir intuitīva un viegli lietojama, nodrošinot efektīvu un patīkamu lietošanas pieredzi.

# 6. Lietotāja ceļvedis

1. **Ievads**
   1. Šī programmatūra ir izstrādāta, lai simulētu evolūcijas procesu, kurā populācija no dažādiem indivīdiem tiek pakāpeniski uzlabota, izmantojot ģenētiskā algoritma principus.
2. **Programmatūras apraksts**:
   1. **Pieslēgšanās un pārlūki:** Šī programma darbojas lokālajā vidē un netiek piekļauta caur pārlūkiem. Tas darbojas kā atsevišķa aplikācija bez pieslēgšanās prasības vai pārlūka atbalsta.
   2. **Lietotāja klase:** Kad programma ir palaista, lietotājam tiek piedāvāta iespēja izvēlēties evolūcijas parametrus, sākot no populācijas lieluma un ģenētiskā koda garuma līdz mutācijas līmenim un paaudžu skaitam. Lietotājs ir brīvs mainīt visus parametrus uz viņam vajadzīgām vērtībām.
3. **Ieviešanas plāns, uzturēšanas plāns, lietotāju mērķauditorijas specifika**:
   1. **Ieviešanas plāns:** Programmatūra tiks izvietota lokālajā vidē, un lietotāji var to izmantot, startējot attiecīgo failu vai izpildot koda skriptu.
   2. **Uzturēšanas plāns:** Plānots regulāri atjaunināt programmatūru, lai uzlabotu tās veiktspēju un paplašinātu funkcionalitāti, balstoties uz saņemto atgriezenisko saiti un jaunajām izstrādātajām idejām.
   3. **Lietotāju mērķauditorijas specifika:** Programmatūra ir domāta gan izglītojošiem nolūkiem, lai saprastu ģenētisko algoritmu darbību, gan arī izstrādātājiem un pētniekiem, kas vēlas pētīt un izmēģināt evolūcijas procesus un to ietekmi uz populācijām.

# Piemērotās licences pamatojums

Šī simulācija tiek izplatīta ar MIT licenci. Šī licence nodrošina elastību un brīvību izmantot un izplatīt programmatūru, ieskaitot izmaiņas un pielāgojumus, gan komerciālos, gan nekomerciālos nolūkos, piešķirot izstrādātājam plašas tiesības. Tādējādi, lietotāji var brīvi izmantot, kopēt, modificēt un izplatīt programmatūru, saglabājot autortiesības un atbildību.

Šī izvēle tika veikta, ņemot vērā vēlamo elastību, atvērtību un sadarbību, lai nodrošinātu, ka programmatūra ir pieejama pēc iespējas plašākai auditorijai, neierobežojot tās izmantošanas iespējas un veicinot atvērtību un kopīgo attīstību.