Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №3.3**

**з курсу: «Інтелектуальні вбудовані системи»**

Виконав:

студент групи ІП-84

Сапбиєв М. Н.

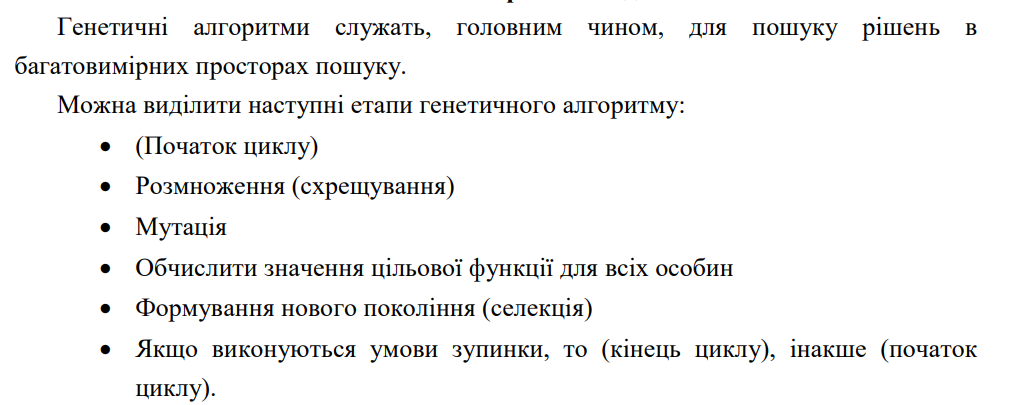
Залікова книжка №8419

Перевірив:

Регіда П. Г.

Київ 2020 р.

**Теоретичні дані:**

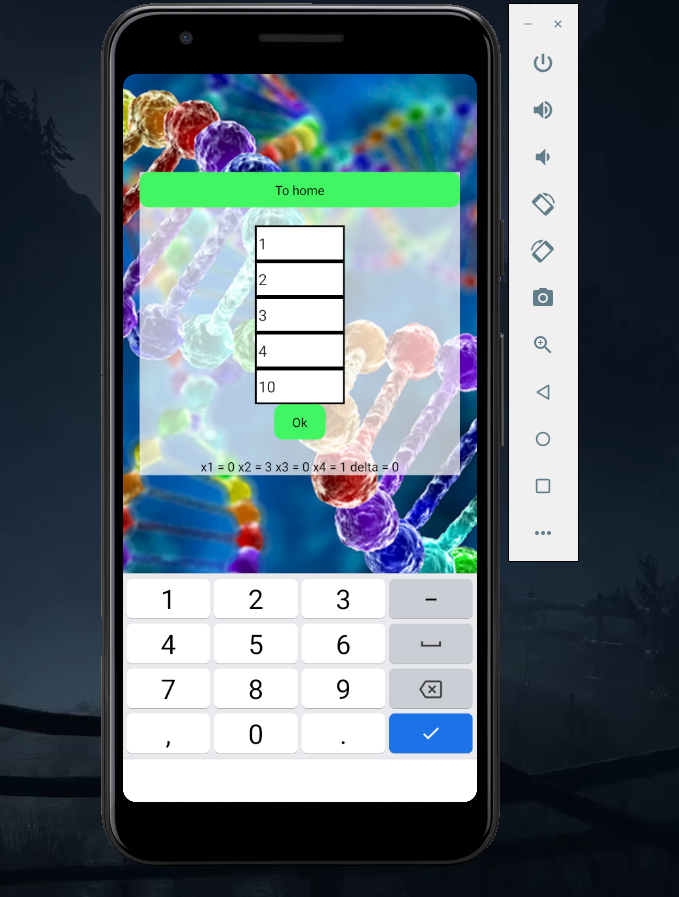


**Код:**

import React, *{*FC, useState*}* from "react";  
import *{* Text,  
 View,  
 ViewStyle,  
 StyleSheet,  
 ***Dimensions***,  
 TouchableOpacity,  
 ImageBackground,  
 TextInput, TextStyle  
*}* from "react-native";  
import *{*NavigationType*}* from "../App";  
import *{*parts*}* from "../lab3.1/parts";  
import *{*gen*}* from "./gen";  
  
interface Styles *{* mainContainer: ViewStyle,  
 button : ViewStyle,  
 whiteContainer : ViewStyle  
 labContainer: ViewStyle,  
 input: TextStyle  
 ok: ViewStyle,  
*}*const stl = StyleSheet.create*<*Styles*>({* mainContainer: *{* flex: 1,  
 justifyContent: "center",  
 alignItems: "center"  
 *}*,  
 button: *{* backgroundColor: '#42f563',  
 marginBottom: 20,  
 paddingHorizontal : 150,  
 padding: 10,  
 borderRadius: 10  
 *}*,  
 whiteContainer : *{* backgroundColor: 'rgba*(*255,255,255,0.7*)*',  
 *}*,  
 labContainer: *{* justifyContent: "center",  
 alignItems: "center"  
 *}*,  
 input: *{* fontSize: 17,  
 backgroundColor: 'white',  
 borderWidth: 2,  
 borderColor: 'black',  
 padding: 4,  
 width: 100  
 *}*,ok: *{* backgroundColor: '#42f563',  
 marginBottom: 20,  
 paddingHorizontal: 20,  
 padding: 10,  
 borderRadius: 10  
 *}*,  
  
*})*export const Lab33: FC*<{* setNavigation: React.Dispatch*<*React.SetStateAction*<*NavigationType*>> }>* = *({*setNavigation*})* => *{* const *[*a, changeA*]* = useState*(*''*)* const *[*b, changeB*]* = useState*(*''*)* const *[*c, changeC*]* = useState*(*''*)* const *[*d, changeD*]* = useState*(*''*)* const *[*y, changeY*]* = useState*(*''*)* const *[*result, changeResult*]* = useState*(*''*)* return *<*ImageBackground style=*{*stl.mainContainer*}* source=*{{*uri: 'https://uchitel.pro/wp-content/uploads/2018/07/content\_geny2\_1\_\_econet\_ru-790x480.jpg'*}}>  
 <*View style=*{*stl.whiteContainer*}>  
 <*TouchableOpacity style=*{*stl.button*}* onPress=*{()* => setNavigation*(*'Home'*)}>  
 <*Text*>*To home*</*Text*>  
 </*TouchableOpacity*>  
 <*View style=*{*stl.labContainer*}>  
 <*View*>  
 <*TextInput style=*{*stl.input*}* keyboardType=*{*'numeric'*}* placeholder=*{*'a'*}* value=*{*a*}* onChangeText=*{*changeA*}/>  
 <*TextInput style=*{*stl.input*}* keyboardType=*{*'numeric'*}* placeholder=*{*'b'*}* value=*{*b*}* onChangeText=*{*changeB*}/>  
 <*TextInput style=*{*stl.input*}* keyboardType=*{*'numeric'*}* placeholder=*{*'c'*}* value=*{*c*}* onChangeText=*{*changeC*}/>  
 <*TextInput style=*{*stl.input*}* keyboardType=*{*'numeric'*}* placeholder=*{*'d'*}* value=*{*d*}* onChangeText=*{*changeD*}/>  
 <*TextInput style=*{*stl.input*}* keyboardType=*{*'numeric'*}* placeholder=*{*'y'*}* value=*{*y*}* onChangeText=*{*changeY*}/>  
 </*View*>  
 <*TouchableOpacity style=*{*stl.ok*}* onPress=*{()* => changeResult*(*gen*(****Number****(*a*)*,***Number****(*b*)*,***Number****(*c*)*,***Number****(*d*)*,***Number****(*y*)))}>  
 <*Text*>*Ok*</*Text*>  
 </*TouchableOpacity*>  
 <*Text*>{*result*}</*Text*>  
 </*View*>  
 </*View*>  
 </*ImageBackground*>  
  
}*

export const gen: *(*a: number, b: number, c: number, d: number, y: number*)* => string = *(*a, b, c, d, y*)* => *{* const population: number = 40  
 const max\_gen\_value = 5  
 const time: number = 4000  
 let gens: Array*<*Array*<*number*>>* = *[]* let deltas: Array*<*number*>* = *[]* let interests: Array*<*number*>* = *[]* const start: number = new ***Date****()*.getTime*()* for *(*let index: number = 0; index < population; index++*) {* gens*[*index*]* = *[]* gens*[*index*][*0*]* = ***Math***.floor*(****Math***.random*()* \* max\_gen\_value*)* gens*[*index*][*1*]* = ***Math***.floor*(****Math***.random*()* \* max\_gen\_value*)* gens*[*index*][*2*]* = ***Math***.floor*(****Math***.random*()* \* max\_gen\_value*)* gens*[*index*][*3*]* = ***Math***.floor*(****Math***.random*()* \* max\_gen\_value*)  
 }* for *(*let index: number = 0; index < population; index++*) {* let result: number = a \* gens*[*index*][*0*]* + b \* gens*[*index*][*1*]* + c \* gens*[*index*][*2*]* + d \* gens*[*index*][*3*]*;  
 deltas*[*index*]* = ***Math***.abs*(*result - y*)* if *(*!deltas*[*index*])* return 'x1 = ' + gens*[*index*][*0*]* + ' x2 = ' + gens*[*index*][*1*]* + ' x3 = ' + gens*[*index*][*2*]* + ' x4 = ' + gens*[*index*][*3*]* + ' delta = ' + deltas*[*index*]  
 }* let end: number = new ***Date****()*.getTime*()* while *(*end - start < time*) {* let all\_interests: number = 0  
 for *(*let index = 0; index < population; index++*)* all\_interests += 1 / deltas*[*index*]* for *(*let index = 0; index < population; index++*)* interests*[*index*]* = *(*1 / deltas*[*index*])* / all\_interests  
  
 let all\_chances: Array*<*number*>* = *[]* for *(*let index = 0; index < population; index++*)* all\_chances*[*index*]* = ***Math***.floor*(****Math***.random*()* \* deltas*[*index*]* \* 100*)* let live\_gen\_delta: number = -***Infinity*** let father\_index: number = 0  
 for *(*let index = 0; index < population; index++*) {* if *(*live\_gen\_delta < all\_chances*[*index*]) {* live\_gen\_delta = all\_chances*[*index*]* father\_index = index  
 *}  
 }* all\_chances = *[]* for *(*let index = 0; index < population; index++*)* all\_chances*[*index*]* = ***Math***.floor*(****Math***.random*()* \* deltas*[*index*]* \* 100*)* live\_gen\_delta = -***Infinity*** let mother\_index: number = 0  
 for *(*let index = 0; index < population; index++*) {* if *(*live\_gen\_delta < all\_chances*[*index*]) {* live\_gen\_delta = all\_chances*[*index*]* mother\_index = index  
 *}  
 }* let children\_gens: Array*<*Array*<*number*>>* = *[]* for *(*let index = 0; index < population; index++*) {* children\_gens*[*index*]* = *[]* children\_gens*[*index*][*0*]* = gens*[*father\_index*][*0*]* children\_gens*[*index*][*1*]* = gens*[*father\_index*][*1*]* children\_gens*[*index*][*2*]* = gens*[*mother\_index*][*2*]* children\_gens*[*index*][*3*]* = gens*[*mother\_index*][*3*]  
 }* gens = children\_gens  
  
 for *(*let index: number = 0; index < population; index++*) {* let result: number = a \* gens*[*index*][*0*]* + b \* gens*[*index*][*1*]* + c \* gens*[*index*][*2*]* + d \* gens*[*index*][*3*]*;  
 deltas*[*index*]* = ***Math***.abs*(*result - y*)* if *(*!deltas*[*index*])* return 'x1 = ' + gens*[*index*][*0*]* + ' x2 = ' + gens*[*index*][*1*]* + ' x3 = ' + gens*[*index*][*2*]* + ' x4 = ' + gens*[*index*][*3*]* + ' delta = ' + deltas*[*index*]  
 }* end = new ***Date****()*.getTime*()  
 }* let smallDelta: number = ***Infinity*** let deltaIndex: number = 0  
 for *(*let index = 0; index < population; index++*) {* if *(*deltas*[*index*]* < smallDelta*) {* smallDelta = deltas*[*index*]* deltaIndex = index  
 *}  
 }* return 'x1 = ' + gens*[*deltaIndex*][*0*]* + ' x2 = ' + gens*[*deltaIndex*][*1*]* + ' x3 = ' + gens*[*deltaIndex*][*2*]* + ' x4 = ' + gens*[*deltaIndex*][*3*]* + ' delta = ' + deltas*[*deltaIndex*]  
}*

**Результати виконання:**

****

**Висновки:**

В ході виконання лабораторної роботи ми ознайомилися з принципами реалізації генетичного алгоритму, вивчили та дослідили особливості даного алгоритму з використанням засобів моделювання і сучасних програмних оболонок.