Київський національний торговельно-економічний університет Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Візуалізація алгоритмів роботи машини Т'юрінга»

Студента 2м курсу, 6 групи, спеціальності 121 «Інженерія програмного забезпечення» спеціалізації «Інженерія програмного забезпечення»

Чернієнко Андрія Ігоровича

підпис студента

Науковий керівник старший викладач кафедри інженерії програмного забезпечення та кібербезпеки

Гарант освітньої програми доктор технічних наук, професор кафедри інженерії програмного забезпечення та кібербезпеки Десятко Альона Миколаївна

KRITEN

підпис керівника

Криворучко Олена Володимирівна

підпис гаранта

КИЇВ - 2020

Київський національний торговельно-економічний університет

Факультет	інформац	ійних т	технологій
Кафедра	інженерії і	програ	много забезпечення та кібербезпеки
Освітній ст	упінь 🔼	TE	магістр
Спеціальні	сть	121	«Інженерія програмного забезпечення»

Затверджую Зав. кафедри інженерії програмного забезпечення та кібербезпеки Криворучко О. В. 8 листопада 2019 р.

Завдання на випускний кваліфікаційний проект студентові Чернієнко Андрію Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту <u>«Візуалізація роботи</u> алгоритмів машини Т'юрінга»

Затверджена наказом ректора від <u>"13" грудня 2019 р. № 4304</u>

2. Строк здачі студентом закінченого проекту <u>01 грудня 2020 р.</u>

3. Цільова установка та вихідні дані до проекту

<u>Мета проекту</u> створення программи Машина Т'юрінга для візуалізації роботи її алгоритмів

Об'єкт дослідження процес розробки програмного забезпечення

Предмет дослідження розробка розробка прграмного забезпечення для візуалізації роботи алгоритмів машини Т'юрінга.

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант	Підпис, дата					
FIK	(прізвище, ініціали)	Завдання видав	Завдання прийняв				
TEX	NTEXAVIE	A MUEL	NUXKINU				
TE!	KM TE KH T	ENHITE	CUTEEN TE				

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом) ВСТУП

РОЗДІЛ 1. ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Принципи роботи машини Т'юринга

1.2. Алгоритм складання

1.3. Варіанти реалізацій машини Т'юринга

1.3.1. Машина Т'юринга, що працює на півбескінечній стрічці

1.3.2. Двомірні машини Т'юринга

1.3.3. Імовірнісна машина Т'юринга

1.3.4. Універсальна машина Т'юринга

1.4. Огляд систем-аналогів

1.4.1. Машина Поста

1.4.2. «Algo2000»

1.5. Діаграма об'єктів предметної області

1.6. Постановка задачі

1.7. Висновок до розділу 1

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ

2.1. Структурна схема системи

2.2. Специфікація системи

2.2.1. Функціональна специфікація

2.2.2. Специфікація якості

2.2.3. Перелік виняткових ситуацій

2.3. Розробка прототипу інтерфейсу користувача системи

2.4. Розробка і опис алгоритмів обробки даних

2.5. Вибір і обгрунтування комплексу програмних засобів

2.5.1 Вибір мови програмування і середовища розробки

2.5.2 Вибір операційної системи

2.5.3 Вибір середовища програмування

2.6. Висновоки до розділу 2

РОЗДІЛ З. РЕАЛІЗАЦІЯ СИСТЕМИ

3.1. Розробка і опис інтерфейсу користувача

3.1.1. Розробка і опис користувальницького меню

3.1.2. Робота з алгоритмом

3.1.3. Налаштування параметрів

3.1.4. Моделювання роботи МТ

3.2. Реалізація класів

3.3. Діаграма модулів

3.4. Вибір і обгрунтування комплексу технічних засобів

3.4.1. Розрахунок обсягу займаної пам'яті

3.4.2. Мінімальні вимоги, що пред'являються до системи

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк вико про	нання етапів екту
	I KN ME KN ME KH ME CH	за планом	фактично
1	LATE A 2E A EL	3	4
1.	Вибір теми випускної кваліфікаційної роботи	20.09.2019	20.09.2019
2.	Розробка та затвердження завдання на проект магістра	13.12.2019	13.12.2019
3.	Вступ та перелік літературних джерел	24.01.2020	24.01.2020
4.	Наукова стаття	01.09.2020	01.09.2020
5.	Технічне завдання	28.02.2020	28.02.2020
6.	Розділ 1. Опис і аналіз предметної області	25.06.2020	25.06.2020
7.	Розділ 2. Проектування системи	07.09.2020	07.09.2020
8.	Розділ 3. Реалізація системи	19.10.2020	19.10.2020
9.	Програма та методика тестування	21.10.2020	21.10.2020
10.	Керівництво користувача	23.10.2020	23.10.2020
11.	Висновки та пропозиції	30.10.2020	30.10.2020
12.	Здача випускного кваліфікаційного проекту на кафедру (перша перевірка)	05.11.2020	05.11.2020
13.	Підготовка автореферату та презентації доповіді	05.11.2020	05.11.2020
14.	Попередній захист випускного	25.11.2020-	25.11.2020
	кваліфікаційного проекту	27.11.2020	-27.11.2020
15.	Зовнішнє рецензування випускної кваліфікаційної роботи	01.12.2020	01.12.2020
16.	Підготовка до публічного захисту випускної	10.12.2020-	10.12.2020-
	кваліфікаційної роботи	11.12.2020	11.12.2020

6. Календарний план виконання проекту

7. Дата видачі завдання <u>«8» листопада 2019 р.</u>

8. Науковий керівник випускного кваліфікаційного проекту <u>Десятко А. М.</u> (прізвище, ініціали, підпис)
9. Гарант освітньої програми <u>Криворучко О. В.</u> (прізвище, ініціали, підпис)
10. Завдання прийняв до виконання студент <u>Чернієнко А. І.</u> (прізвище, ініціали, підпис) 11. Відгук керівника випускного кваліфікаційного проекту

								-	A	
				V V						
1.	· · · · ·				-				<u> </u>	
			V K	1.1						
1						NV		10		
10.			XX							1
	XX		V S						1 1 1	1 1 1
A VA			111					NU		
	V.	VI					ILV /	N 1	1	
	1. 1		5				1.11		15-15-	
					1 1 1 .					
	111	XXX	NY	111	NU				(1).	10.
							XX	1 \L\		
							1- 11			
4										
	1	NY I	6.1-	121	1	IV	N N		141-	
	1 March			~ 1						
						·	KI		171	11
	ILV.			N		ALV.				
					IV I			N V		
			XX							
			INV	N.	TTV -	1 KL	11.	TY Y		
		C' IN								1
			111		XX	1				
<u> </u>	L KL	1V	UXV.	See.	22			N' i	1V	N X
VA			N. 1		L X		11.20			
AX.	·							10		· · · ·
-										
							1 6 1			
-	· · · ·		141						NY	
	14 1			1		6	X IX	1		1
		113	1 V	1	XX	NV	- C - J - J			

Науковий керівник випускного кваліфікаційного проекту

	(підпис, дата	N KINE KINGE N
Відмітка про попередній зах	хист	XNV.TF	Десятко А. М
TENK TENK			(ШБ, підпис, дата)
12. Висновок про випускни	й квал	іфікаційни	ій проект
			ALE KULTE KULTE
Випускний кваліфікаційний	і прое	кт студента	а <u>Чернієнко А. І.</u>
RY RAIER RAIE	A	HI-KI	(прізвище, ініціали)
може бути допущена до зах	исту е	екзаменацій	йній комісії.
Гарант освітньої програми	E	KITE	Криворучко О. В.
			(прізвище, ініціали, підпис)
Завідувач кафедри	T'E	KINI	Криворучко О. В.
			(підпис, прізвище, ініціали)
	20	n	

АНОТАЦІЯ

Відповідно до мети робота присвячена розробці программи для ПК що має весь доступний функціонал машини Т'юрінга з візуалізацією алгоритмив що дозволить покроково контролювати алгоритм.

В результаті порівняльного аналізу аналогічних рішень визначено вагомі недоліки подібних програм що були усунені в ході розробки.Розробка виконана у середовищі розробки MS Visual Studio, а також база даних – в SQLite. Обрана мова програмування – С#. Окремі елементи. Збирання проекту виконувалося за допомогою інструмента стандартного інструменту в VS.

Готовий програмний комплекс «Машина Тьюрінга» було успішно протестовано відповідно до функціональних вимог.

Ключові слова: Visual Studio, С#, машина Т'юрінга, RAD, алгоритм.

ABSTRACT

According to the purpose, the work is devoted to the development of a program for a PC that has all the available functionality of a Turing machine with visualization of algorithms that will allow step-by-step control of the algorithm.

As a result of the comparative analysis of similar decisions significant shortcomings of similar programs which have been eliminated during development are defined. The development is performed in the development environment of MS Visual Studio, as well as the database - in SQLite. The selected programming language is C #. Individual elements. The assembly of the project was performed using the standard tool in VS.

The finished software package "Turing Machine" was successfully tested in accordance with the functional requirements.

Keywords: Visual Studio, C #, Turing machine, RAD, algorithm.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- ЖЦ життєвий цикл
- ПЗ програмне забезпечення
- МТ машина Т'юринга
- ОС операційна система
- ОП оперативна пам'ять, об'єм ОЗП

L'H	NUTE	TUN X	EE.	КНТЕУ 1.	21 06-22.1	MP	J Kt	
Зм. Аркул	ш № докум.	Підпис	Дата	KHTENKY				
Зав. каф.	Криворучко О.В.	EN	19.10.20	Візуалізація роботи	Стадія	Арку	Аркушів	
Керівник	Десятко А.М.	2K	19.10.20	алгоритмів машини	ПС	2	55	
Гарант	Криворучко О.В.	1 FEY	19.10.20	Т'юрінга	Факультет	інформ	аційних	
Розробив	Чернієнко А.І.	HI	19.10.20	Перелік умовних	mex	нологій		
TE?	TEEY	TEEY	ITE	11	скорочень	2м курс, 6 група		

3MICT

вступ				N N N N	1 K all		3
розділ і	I. ОПИС I АН.	АЛІЗ ПІ	редмет	НОЇ ОБЛАСТІ			3
1.1. При	инцип роботи м	ашини Т	Г'юринга	<u> </u>			3
1.2. Алг	соритм складан	ня	<u> </u>				8
1.3. Bap	іанти реалізаці	ї машині	и Т'юрин	га		1	10
1.3.1 M	ашина Т'юрин	га, що пр	рацює на	півбескінечній стрічці			10
1.3.2 Д	вомірні машин	и Т'юри	нга		r'ill'i	1K	12
1.3.3 IN	иовірнісна маш	ина Т'ю	оинга	I FE MILTE		ZE V	14
1.3.4 У	ніверсальна ма	и шина Т'і	юринга		JE K		14
1.4. Огл	иял систем-анал	югів	IE.	KH ZE CHI	TE MA	TEI	14
1.4.1 M	ашина Поста		TE	W EN M		NV)	
1.5 Л	іаграма об'єкті	в прелме	тної обла	асті	1 ZEN		
1.6 П	остановка зала	чі	Nº TE	E LANGE	40.27	L'H	
1.7. Вис	сновки до розді	лу 1	1Kr.	TE'KH'TE'	KH TF		
РОЗЛІЛ 2	2. ПРОЕКТУВ	ання (СИСТЕМ	П			
21 0	труктурна схем	иа систем	ли	TE KA TE	E KH	TE	22
$2.1 \ c$	пецифікація си	стеми	E A	MITE KN T	E N		23
2.2 C	ункціонацьна с	пенифік	anig	KI TE KN	TE K		24
2.2.1 \$	Специфікація	якості	ація	KN TE KN	JE V	N	25
223	Перелік винят	кових си	ารงอบเห	STR. UEIK	T. TE	KM	25
2.2.5 2.3 Post	обка прототип	у інтерф	ейсу кор	истувача системи	NTE	2	26
2.5.1 05r	робка і опис ап	у штерф горитмів	епеу корі в обробки	а ланих	K	C V	29
2.1 T 05	fin i offinyetye	ння ком	nnekov ni	пограмних засобів	KANT	E,	32
2.5 Dire	Вибір мови пг		auug i cer	рограмних засобнь		TE	32
2.5.1 2.5.2 B	ибір операційн	ої систел		сдовища розрооки	EN	TE	33
2.5.2 D	Вибір сереция			110	JIE K		
2.3.3.	виогр середов	ища про	грамуван	кп	JTE V		
	а обла пізані	лу 2 9 сист	теми		TTE.	I K I	
2 1 Вар				novo			
2.1.103		терфеису	/ користу	вача			
5.1.1 PC	озроока гопис і	користув	альницы	Кого меню			
J.E.	NULTE	iun	TEN.	LITEN 1	21.06.22	MD	
Зм. Аркуш	№ докум.	Підпис	Дата	КПІЕУ І.	21 00-22.1	VIP	
Зав. каф.	Криворучко О.В.	EN	24.01.20	Візуалізація роботи	Стадія	Арку	Аркушіе
Керівник	Десятко А.М.	EK.	24.01.20	алгоритмів машини	Зміст	3	55
Гарант	Криворучко О.В.	TEY	24.01.20	Т'юрінга	Факультет	інформ	аційних
Розробив	Чернієнко А.І.	HIE	24.01.20	Зміст	тех 2м кур	нологій эс, 6 груг	na

3.1.3 Налаштування параметрів 4 3.1.4 Моделювання роботи МТ 4 3.2 Реалізація класів 4 3.3 Діаграма модулів 4 3.4 Вибір і обгрунтування комплексу технічних засобів 4 3.4.1 Розрахунок обсягу займаної пам'яті 4 3.4.2 Мінімальні вимоги, що пред'являються до системи 4 3.5. Висновки до розділу 3 4 ВИСНОВКИ ТА ПРОПОЗИЦІЇ. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ. ГЕХНІЧНЕ ЗАВЛАННЯ	3.1.2 Робота з алгоритмом	
3.1.4 Моделювання роботи МТ 4 3.2 Реалізація класів 4 3.3 Діаграма модулів 4 3.4 Вибір і обгрунтування комплексу технічних засобів 4 3.4.1 Розрахунок обсягу займаної пам'яті 4 3.4.2 Мінімальні вимоги, що пред'являються до системи 4 3.5. Висновки до розділу 3 4 ВИСНОВКИ ТА ПРОПОЗИЦІЇ СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.	3.1.3 Налаштування параметрів	
3.2 Реалізація класів	3.1.4 Моделювання роботи МТ	
3.3 Діаграма модулів	3.2 Реалізація класів	
3.4 Вибір і обгрунтування комплексу технічних засобів	3.3 Діаграма модулів	
3.4.1 Розрахунок обсягу займаної пам'яті 4 3.4.2 Мінімальні вимоги, що пред'являються до системи 4 3.5. Висновки до розділу 3 4 ВИСНОВКИ ТА ПРОПОЗИЦІЇ 4 СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ. 4 ГЕХНІЧНЕ ЗАВЛАННЯ 4	3.4 Вибір і обгрунтування комплексу технічних засобів	
3.4.2 Мінімальні вимоги, що пред'являються до системи	3.4.1 Розрахунок обсягу займаної пам'яті	
3.5. Висновки до розділу 3	3.4.2 Мінімальні вимоги, що пред'являються до системи	51
ВИСНОВКИ ТА ПРОПОЗИЦІЇ СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ГЕХНІЧНЕ ЗАВЛАННЯ	3.5. Висновки до розділу 3	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ГЕХНІЧНЕ ЗАВЛАННЯ	ВИСНОВКИ ТА ПРОПОЗИЦІЇ	
ГЕХНІЧНЕ ЗАВЛАННЯ	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
	ГЕХНІЧНЕ ЗАВДАННЯ	

H	EX	NUTE	TUN	FEY.	KHTEY L	21 06-22.1	MP	
Зм.	Аркуш	№ докум.	Підпис	Дата	KHIEKK	L'EK.	114	
Зав.	каф.	Криворучко О.В.	EN	24.01.20	Візуалізація роботи	Стадія	Арку	Аркушів
Кер	івник	Десятко А.М.	2K	24.01.20	алгоритмів машини	Зміст	3	55
Гара	ант	Криворучко О.В.	1 FEY	24.01.20	Т'юрінга	Факультет	інформ	аційних
Розр	обив	Чернієнко А.І.	HI	24.01.20		mex	нологій	
K	3.	THEY	TE	11	Зміст	2м кур	ос, 6 груг	na

ВСТУП

Актуальність роботи. У 1935 р з'явилось таке положення: властивості, виявлені у деяких, точних, певних класів обчислюваних теоретико-числових функцій, які вивчалися Черчем і Кліні в 1932-1935 роках, наполегливо виказували думки, що цей клас, може охоплювати всі функції, які відповідно з нашим інтуїтивним уявленням можна розглядати, як обчислювані. За цих обставин Черч висунув тезу (опубліковану в 1936 р), що всі функції, які інтуїтивно ми можемо розглядати, як обчислювані, або, кажучи його словами, як «ефективно обчислювані», є загальнорекурсивними.

Трохи пізніше, але вже незалежно, з'явилася стаття Т'юринга (1936), в якій був введений ще один точно визначений клас інтуїтивно обчислюваних функцій, «функції які обчислюванні по Т'юрингу». Незабаром Т'юрингом в 1937 році було показано, що його обчислюванні функції - це те ж саме, що λ-визначні функції, і, отже, те ж саме, що і загальнорекурсивні функції [1].

Логічним було питання: чи можна для будь-якого математичного твердження вказати кінцеву послідовність інструкцій, які могли б виконуватися механічно, одна за одною, людиною або обчислювальним пристроєм, і в підсумку з'ясувати, істинне це твердження або хибне? Як математичній моделі для опису таких алгоритмів було запропоновано абстрактно обчислювальний пристрій, який згодом було названо машина Т'юринга (МТ).

МТ - дискретно обчислювальний пристрій, який змінює свої характеристики в певні моменти часу. Хоча і доведено, що поставлена

HE	NULTE	AU	TE	КНТЕУ 12	21 06-22.1	MP	
Зм. Аркуи	№ докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.	EN	19.10.20	Візуалізація роботи	Стадія	Арку	Аркушів
Керівник	Десятко А.М.	2K	19.10.20	алгоритмів машини	В	2	55
Гарант	Криворучко О.В.	1 FEY	19.10.20	Т'юрінга	Факультет	інформ	аційних
Розробив	Чернієнко А.І.	HI	19.10.20	TEXPERIEN	тех	снологій	
TE I	THEY	TE	11	Вступ	2м кур	ос, 6 гру	na

проблема в загальному випадку не має розв'язку, застосування МТ вийшло далеко за межі первісної постановки завдання. Хоча МТ не стала реально діючим пристроєм, але вона до теперішнього часу постійно використовується в якості основної моделі для з'ясування суті таких понять, як «обчислювальний процес», «алгоритм», а також для з'ясування зв'язку між алгоритмом і обчислювальними машинами [2].

Так як процес складання алгоритму вручну займає багато часу, потрібно автоматизувати цей процес. У зв'язку з цим, під час лабораторного практикуму авторам необхідно розробити систему, що моделює роботу МТ, яка буде складати алгоритми, перевіряти їх працездатність і візуалізувати свою роботу.

При розробці будуть використані методологія структурного підходу до проектування, яка побудована на декомпозиції (розбитті) системи на автоматизуючі функції, і методологія ООАП (об'єктноорієнтований аналіз і проектування), в основі якої лежить об'єктна декомпозиція, і також технологія RAD (з англ. RapidApplicationDevelopment - швидка розробка додатків), яка гарна в першу чергу для відносно невеликих проектів, що розробляються для конкретного замовника, так званого замовного ПО [3].

Основні принципи технології RAD:

розробка додатків ітераціями;

необов'язковість повного завершення робіт на кожному
 етапі ЖЦ;

обов'язкове залучення користувачів на етапі розробки;

тестування і розвиток проекту одночасно з розробкою;

- правильне керівництво розробкою, чітке планування контроль виконання робіт.

1	JU I	TEN) TE		DIES MILEY FILES	Аркуш
4	X	T.KR	XHI	4	КНТЕУ 121 06-22.МР	1
Зм.	Аркуш	№ докум	Підпис	Дата	R'ITE KITE KATE	1 SH

Мета роботи є полягає у дослідженні впливу машини Тюрінга на розвиток технологій в умовах сучасності.

Для досягнення поставленої мети було вирішено наступні задачі:

розглянути принципи роботи машини Т'юринга;

визначити алгоритм складання;

визначити варіанти реалізацій машини Т'юринга;

дослідити машину Т'юринга, що працює на півбескінечній стрічці;

визначити двомірні машини Т'юринга;

розглянути імовірнісну машину Т'юринга;

проаналізувати універсальну машину Т'юринга;

здійснити розробку і опис алгоритмів обробки даних;

провести моделювання роботи МТ.

Об'єкт дослідження: інформаційні технології.

Предмет дослідження дослідження впливу машини Тюрінга на розвиток технологій.

Методи дослідження, що були використані у роботі: аналіз, абстрагування, порівняння, графічний метод і моделювання.

Наукова новизна дослідження полягає в удосконаленні алгоритмів роботи машини Т'юрінга.

Практичне значення дослідження: готове альтернативне рішення алгоритмів роботи машини Т'юрінга обраною мовою програмування С #.

7	KY.	EKIK	THE K	K	A THE KANTE W	ILITE
1	U	,TE'.1	JAF	7	L'EN MILEN MILEN	Аркуш
Y	Y	TEKP	xH1.	14	КНТЕУ 121 06-22.МР	AN I
Зм.	Аркуш	№ докум	Підпис	Дата	RITE KITE KTTE	44

РОЗДІЛ 1

ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Предметна область - область діяльності в реальному світі, яка містить елементи системи, процеси їх взаємодії між собою, інформаційні потоки, різні функціональні підсистеми, а також зовнішні, по відношенню до досліджуваної системі процеси, події і явища [4].

1.1. Принцип роботи машини Т'юринга

Машина Т'юринга - абстрактний виконавець (абстрактна обчислювальна машина). Була запропонована Аланом Т'юрингом в 1936 році для формалізації поняття «алгоритм».

Машина Т'юринга є розширенням кінцевого автомата і, згідно з тезою Черча-Т'юринга, здатна імітувати всіх виконавців (за допомогою задання правил переходу), будь-яким чином реалізувати процес покрокового обчислення, в якому досить елементарний кожен крок обчислення.

Тобто, всякий інтуїтивний алгоритм може бути реалізований за допомогою деякої машини Т'юринга.

До складу машини Т'юринга входить необмежена в обидві сторони стрічка (можливі машини Т'юринга, які мають кілька нескінченних стрічок), розділена на осередки, і керуючий пристрій (також називається головкою запису-читання (ГЗЧ)), який здатний перебувати в одному з безлічі станів. Число можливих станів керуючого пристрою кінцеве і точно задане (на місці, вправо, вліво). На рисунку 1.1 зображена структура МТ.

HE	NUTE	TUP I	TE	КНТЕУ 12	121 06-22.MP		
Зм. Аркуг	и № докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.	EA	25.06.20	Візуалізація роботи	Стадія	Арку	Аркушів
Керівник	Десятко А.М. Криворучко О.В.	EK	25.06.20	алгоритмів машини	P1	6	55
Гарант			25.06.20	Т'юрінга	Факультет інформаційних		
Розробив	Чернієнко А.І.	HI	25.06.20	Опис і аналіз	mex	нологій	
TET	NEEY	ITE	14	предметної області	2м курс, 6 група		



Рис. 1.1. Структура МТ

Джерело [8]

Керуючий пристрій може переміщатися вліво і вправо по стрічці, читати і записувати в осередки символи деякого кінцевого алфавіту.

Кінцевий алфавіт - кінцева безліч атомарних символів будь-якої формальної мови [5]. Прикладом кінцевого алфавіту є ноти, цифри. Виділяється особливий порожній символ, що заповнює всі клітини стрічки, крім тих з них (кінцевого числа), на яких записані вхідні дані.

Керуючий пристрій працює згідно з правилами переходу, які представляють алгоритм, який реалізований машиною Т'юринга.

Кожне правило переходу наказує машині, в залежності від поточного стану і спостережуваного в поточній клітці символу, записати в цю клітку новий символ, перейти в новий стан і переміститися на одну клітку вліво або вправо.

Деякі стани машини Т'юринга можуть бути позначені як термінальні, і перехід в будь-який з них означає кінець роботи, зупинку алгоритму (Рис. 1.2):

	5	EN	A KHI	- KR	(HI	KRUTE KRUTE KRU	TEX
	11		11, 13		.17	E TE KITE KI	Аркуш
5		(H)	KNUK	TEX	NV.	КНТЕУ 121 06-22.МР	V TE
2	Зм.	Аркуш	№ докум	Підпис	Дата		CHILE

1	q 0	q1	q ₂	q ₃	q 4	95	q 6	q 7	q 8
1	q ₀ 1→q ₀ 1R	q ₁ 1→q ₂ aR	$q_21\rightarrow q_21L$	$q_3 1 \rightarrow q_4 a R$	q ₄ 1→q ₄ 1R	TE	ZKI	q ₇ 1→q ₂ aR	KRY
×	$q_0 \times \rightarrow q_1 \times R$		$q_2 \times \rightarrow q_3 \times L$	I K'L	$q_4 \times \rightarrow q_4 \times R$	Krill	$q_6 \times \rightarrow q_7 \times R$	TUT	q ₈ ×→q ₉ ×N
-	EN UN	ALLE I	$q_2 = \rightarrow q_2 = L$	EEK	q ₄ =→q ₄ =R	KA	E.	$q_7 = \rightarrow q_8 = L$	N C
a		JU Z	q₂a→q₂aL	q₃a→q₃aL	q₄a→q₄aR	- Kh	q ₆ a→q ₆ 1R	q ₇ a→q ₇ aR	q ₈ a→q ₈ 1L
*	q₀*→q₀*R	HTE	ZKR	q ₃ *→q ₆ *R	q ₄ *→q ₅ 1R	E V	MAE	'LA'	
1	U.S.	K'II	K	TE	Khi	$q_5 \rightarrow q_2^*L$	CH T	EVY	TE

Рис. 1.2. Приклад реалізації правил переходу

Джерело [11]

Алфавітом є рядки (1, х, =, а, *,), а станами є стовпці (q0 - q8). На перетині рядка і стовпця розташовується команда. Як приклад, розберемо МТ, яка в даний момент знаходиться на символі «1» і в стані q3. Для даного прикладу команда «q4 а R» здійснює перехід в стан q4, замінює поточний символ на «а», а ГЧЗ рушає вправо.

Машина Т'юринга називається детермінованою, якщо кожна комбінація стану і стрічкового символу в таблиці відповідає одному правилу. Якщо існує пара «стрічковий символ - стан», для якої існує 2 і більше команд, така машина Т'юринга називається недетермінованою.

1.2. Алгоритм складання

Розглянемо алгоритм складання двох чисел, заданих на стрічці з унарною системою числення для машини Т'юринга. Алгоритм приймає 2 числа, розділених символом роздільника ('*'), і записує результат їх

2	10	L' KI'	21.	N.	K K K K K	TE L
11				1.	E TE KITE K	Аркуш
	(H)	KR	TEK	10	КНТЕУ 121 06-22.МР	8
Зм.	Аркуш	№ докум	Підпис	Дата	TENKT ENKLEY	

складання на стрічці і відразу за цими числами, відокремлюючи його символом '='.

Початковий стан машини Т'юринга - q0, а головка оглядає перший символ першого числа.

Кінцевий стан - qz, при цьому головка повертається до вихідної позиції.

Так, щоб обчислити 3 + 2, слід записати в якості вихідних даних 11 * 111 результатом повинно бути 11111. Програма перед своїм завершенням повинна відновити вихідні дані на стрічці (Рис. 1.3):

1KI	90	q_1	<i>q</i> ₂	<i>q</i> ₃	94	$q_5 = q_z$
1V	0 R q1	1 R q ₁	1 R q ₂	1 L q3		1 S q _z
0	C'ITEJ	Kr. TE	KM	0 R q ₀	1 L q4	SHIL
*	* R q0	* R q1	ET KAN	*Lq3	* L q4	NN
J=	$=Lq_4$	$= \mathbf{R} q_2$	(F. Kr	$=Lq_3$	KH T	EVH
X	KANT	$= \mathbf{R} q_2$	1L q3	NTE	$\lambda R q_5$	11.13

Рис. 1.3. Програма складання двох чисел в унарній системі для машини Т'юринга

Джерело [3]

Програма знаходить черговий символ '1' доданків, замінює його на символ "0" (своєрідна позначка, що цей символ вже оброблений) і записує черговий символ '1' в результат. Потім голівка повертається до початку в пошуках символу "0".

Якщо є ще символи '1' до символу '=', то повторюємо попередній крок, в іншому випадку - повертаємо голівку в початковий стан, відновлюючи на стрічці вхідні дані (замінюємо'0' на '1')

Траса обчислень виглядає наступним чином:

$\langle \rangle$	1 F	11. 12	E.C.Y	17	EN RITE RITE K	Аркуш
	(1)	KANG	TEX	10	КНТЕУ 121 06-22.МР	N act
Зм.	Аркуш	№ докум	Підпис	Дата		

1.3. Варіанти реалізації машини Т'юринга

Модель машини Т'юринга також може мати розширення. Можна розглядати машини Т'юринга з довільним числом стрічок і багатовимірними стрічками з різними обмеженнями. Однак всі ці машини є повними по Т'юрингу і моделюються звичайною машиною Т'юринга.

1.3.1 Машина Т'юринга, що працює на півбескінечній стрічці

Як приклад такої відомості розглянемо наступну теорему: для будьякої машини Т'юринга існує еквівалентна машина Т'юринга, що працює на півбескінечній стрічці (тобто на стрічці, яка нескінченна в одну сторону).

Розглянемо доказ, наведеним Ю. Г. Карповим в книзі «Теорія автоматів» [6]. Доказ цієї теореми конструктивний, тобто ми дамо алгоритм, за яким, для будь-якої машини Т'юринга може бути побудована еквівалентна машина Т'юринга з оголошеною властивістю.

По-перше, довільно пронумеруємо осередки робочої стрічки МТ, тобто визначимо нове розташування інформації на стрічці (Рис. 1.4):



Рис. 1.4. Нумеровані осередки робочої стрічки МТ

Джерело [8]

Потім перенумеруємо осередки і будемо вважати, що символ «*» не міститься в словнику МТ (Рис. 1.5):

11		111. 53		. 17	EN MITEN KITENK	Аркуш
	(H)	KNUK	TEX	10	КНТЕУ 121 06-22.МР	10
Зм.	Аркуш	№ докум	Підпис	Дата		10



Рис. 1.5. Осередки робочої стрічки МТ після перетворення Джерело [8]

Нарешті, змінимо машину Т'юринга, подвоївши число її станів, і змінимо зрушення голівки зчитування-запису так, щоб в одній групі станів робота машини була б еквівалентна її роботі в замальованій зоні, а в іншій групі станів машина працювала б так, як вихідна машина працює в незамальованій зоні. Якщо при роботі МТ зустрінеться символ '*', значить голівка зчитування-запису досягла межі зони (Рис. 1.6).



Початковий стан нової машини Т'юринга встановлюється в одній або іншій зоні в залежності від того, в якій частині вихідної стрічки розташовувалася голівка зчитування-запису у вихідній конфігурації. Вочевидь, що зліва від обмежуючих маркерів «*» стрічка в еквівалентній машині Т'юринга не використовується [7].

1.3.2 Двомірні машини Т'юринга

Мураха Ленгтон - двомірний клітинний автомат з дуже простими правилами, винайдений Крісом Ленгтон. Мураху можна також вважати двомірною машиною Т'юринга з 2 символами і 4 станами.

Правила

Розглянемо нескінченну площину, розбиту на клітини, пофарбованою деяким чином в чорний і білий колір. Нехай в одній з клітин знаходиться «мураха», який на кожному кроці може рухатися в одному з чотирьох напрямків в клітинки, сусідні по стороні. Мураха рухається згідно з такими правилами:

На чорному квадраті - повернути на 90 ° вліво, змінити колір квадрата на білий, зробити крок вперед на наступну клітинку.

На білому квадраті - повернути на 90 ° вправо, змінити колір квадрата на чорний, зробити крок вперед на наступну клітинку.

Ці прості правила викликають досить складну поведінку: після деякого періоду досить випадкового руху мураха, мабуть, починає будувати дорогу з 104 кроків, яка повторюється нескінченно, незалежно від початкового розмалювання поля. Це наводить на думку, що «магістральна» поведінка є стабільним аттрактором мурашки Ленгтон.

	5	6	C. Kr.	cr'i	SKI	C KI HI KN JI KR	TE'L
	11				1.17	E TE KITE KI	Аркуш
2		(H)	XAU	TEX	10	КНТЕУ 121 06-22.МР	12
0	Зм.	Аркуш	№ докум	Підпис	Дата	TENKNIEN KHUEK	12

Мураха Ленгтон також може бути описаний, як клітинний автомат в якому майже все поле пофарбовано в чорно-білий колір, а клітка з «мурахою» має один з восьми різних кольорів, що кодують відповідно всі можливі комбінації чорного / білого кольору клітин і напрямків руху мурахи. На рисунку 1.7 зображений хаотичне зростання мурашки Ленгтон.

Існує просте розширення мурашки Ленгтон, в якому використовується більше двох кольорів клітинок. Кольори змінюються циклічно. Для таких мурах існує також проста форма назви: для кожного наступного кольору використовується буква L або R (Л і П), в залежності від того, повертає мураха направо або наліво. Таким чином, мураха Ленгтон - це мураха RL.

Деякі з цих узагальнених мурах Ленгтон малюють візерунки, які стають все більш симетричними. Один з простих прикладів - мураха RLLR. Одна достатня умова цього полягає в тому, що ім'я мурашки, що розглядається як циклічний список, складається з послідовних пар повторюваних букв LL або RR (циклічність списку означає, що остання буква може повторюватись з першої) [8].



Рис. 1.7. Мураха Ленгтон

X	Джерело [9]										
11		111. 53		1.17	EN MITE KITE K	Аркуш					
	(H)	KR	TEX	20	КНТЕУ 121 06-22.МР	13					
Зм.	Аркуш	№ докум	Підпис	Дата	TENKNIEN KHIEK						

1.3.3 Імовірнісна машина Т'юринга

Імовірнісна машина Т'юринга являє собою детерміновану машину Т'юринга, що має додаткове апаратне джерело випадкових бітів, будь-яке число яких, наприклад, вона може «замовити» і «завантажити» на окрему стрічку і потім використовувати в обчисленнях звичайним для МТ чином [9].

1.3.4 Універсальна машина Т'юринга

Універсальною машиною Тьюрінга називають машину Т'юринга, яка може замінити собою будь-яку машину Т'юринга. Отримавши для входу програму і вхідні дані, вона обчислює відповідь, який вирахувала б за вхідними даними машина Т'юринга, чия програма була дана для входу [10].

1.4. Огляд систем-аналогів

В даний час існує досить багато реалізацій МТ. Розглянемо можливості деяких з них.

1.4.1 Машина Поста

Тренажер «Машина Поста» - це навчальна модель універсального виконавця (абстрактної обчислювальної машини), заснованого на роботах Е.Л. Посту по уточненню поняття алгоритм [11].

Згідно тези Посту, будь-який алгоритм може бути записаний у вигляді програми для машини Поста. Як приклад розглянемо програму К. Полякова (Рис. 1.8).

2		r' Kr.	K.	(H)	- KI HI KA KA KA	TE'V
11				1.17	E TE KITE K	Аркуш
	(H)	KR	TEX	20	КНТЕУ 121 06-22.МР	11
Зм.	Аркуш	№ докум	Підпис	Дата	A FLANKER AND A	14

I Ma	эшина	Поста: U:\SO	FT\POSTM\EX	AMPLES\add.pst	X
Файл	Лент	га Выполнени	е Алфавит I	Вид Скорость ?	
D	\$	8 🖫 🚅		& FENJTER KITE	
Усл	овие	задачи:	21	NO LA KENDER KEND	1
мето Сфор А+В г	ік, а в рмир метоі	ю втором В овать спрає к.	меток. Каре за от них чер	этка стоит на самой правой метке первого числа. рез пустую секцию результирующий массив, в котором	N N
	-14 -1	3 -12 -11 -10 -3	9 -8 -7 -6 -5	5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14	
	-14 -1	3 -12 -11 -10 -5 Команда	3 -8 -7 -6 -5 Переход	5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14	
	-14 -1	31211105 Команда 0	9 -8 -7 -6 -5 Переход	5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 Комментарий Временно убираем метку.	
	-14 -1	3-12-11-10 Команда 0 >	3 -8 -7 -6 -5 Переход	5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 Комментарий Временно убираем метку.	
11 타 년	-14 -1 1 2 3	3 -12 -11 -10 -5 Команда 0 > ?	9 -8 -7 -6 -5 Переход 4, 2	5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 Комментарий Временно убираем метку. проходим 1 слагаемое	
世界を	-14 -1 1 2 3 4	3-12-11-10 Команда 0 > ? >	9 -8 -7 -6 -5 Переход 4, 2	5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 Комментарий Временно убираем метку. проходим 1 слагаемое	
	-14 -1 1 2 3 4 5	3-12-11-10-4 Команда 0 > ? > ?	9 -8 -7 -6 -5 Переход 4, 2 6, 4	5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 Комментарий Временно убираем метку. проходим 1 слагаемое проходим 2 слагаемое	



Машина Поста складається з каретки (зчитуючої і записуючої головки) і нескінченної стрічки, яка розбита на осередки. Кожен осередок стрічки може бути або порожній («0»), або містити мітку («1»).

Програма складається з пронумерованих рядків. У кожному рядку записується одна з наступних команд:

> N перемістити каретку вправо на 1 клітинку і перейти до рядка з номером N;

<N перемістити каретку вліво на 1 клітинку і перейти до рядка з номером N;

0 N записати в поточну комірку «0» (стерти мітку) і перейти до рядка з номером N;

1 N записати в поточну комірку «1» (поставити мітку) і перейти до рядка з номером N;

	10				K K K K	JV V
11					EN MELKITE K	Аркуш
	(H)	KNUK	TEX	10	КНТЕУ 121 06-22.МР	15
Зм.	Аркуш	№ докум	Підпис	Дата	TE KHIEK KHIEK	15

? N, M якщо поточна комірка містить «0» (не позначена), то перейти до рядка з номером N, інакше перейти до рядка M;

. зупинити програму.

Номер рядка переходу в командах ",", 0 і 1 можна не вказувати, при цьому відбувається перехід до наступного рядка.

Для завершення роботи програми досить зробити перехід на рядок 0, наприклад, так:

? 25, 0 зупинити програму, якщо поточна комірка містить «1», інакше перейти до рядка 25.

1.4.2 «Algo2000»

«Algo2000» - інтерпретатор машини Посту і машини Т'юринга [12]. Перевірка правильності складання алгоритму для обчислення функції за допомогою комп'ютера, повністю виключає можливість помилки при роботі з програмою на стрічці. При цьому користувач має можливість регулювати швидкість виконання програми і бачити кожен крок машини при обробці числа, що дозволяє вчасно вносити зміни в програму при виявленні помилки. Програма має зручний інтерфейс, передбачена можливість збереження складених програм, підтримуються різні зовнішні алфавіти машини Т'юринга, а також є довідник користувача (Рис. 1.9).

Програма має помірні вимоги: комп'ютер IBM PC AT 486 і вище, наявність операційної системи Windows'95 / 98 / NT.

1	(Y)	EV	(ALL)	EKA,	HT	EXAMPLITE XAU	TEX
	11		11.1		1.17	ENTER KITE KI	Аркуш
2	10	(4)	KAUN	ALLIN	NV.	КНТЕУ 121 06-22.МР	16
0	Зм.	Аркуш	№ докум	Підпис	Дата	REP KHER KHER	10

Ucconuccon			₩ 🛛 н	er •	TT	TEKTE
Эсловие зад.	NPE	TE	E KN	TEY	KN	TES KAN
1 2	3 4 5	6789	10 11 12 13	14 15 16 1	7 18 19 20	0 21 22 23 24 25 26 27
K	KE	1 A	KE	1.5	J.S.K	Y LIKELY
Внешний ал	іфавит:		HE	EKAN	LE I	EXAMPLEY
Внешний ал	фавит: Q0	-+ Q1	Q2	03	Q4	Комментарий:
Внешний ал А\Q Пробел	іфавит: Q0	Q1	Q2	Q3	Q4	Комментарий:

Рис. 1.9. Екранна форма програми «Algo2000» Джерело: «Скріншот від автора»

1.5 Діаграма об'єктів предметної області

Об'єктно-орієнтована декомпозиція - розбиття системи на суті, будьякими об'єктами, що діють в тій ситуації, яку якраз і моделює система.

Об'єкт - елемент системи, який об'єднує в собі дані і операції, що мають властивості спадкування, інкапсуляції і поліморфізму [13].

Об'єктна декомпозиція може застосовуватися багаторазово, або бути багаторівневою. Це означає, що кожен об'єкт може розглядатися як система, яка складається з елементів, що взаємодіють один з одним. При багаторівневій декомпозиції, на кожному рівні ми отримуємо об'єкти з більш

11		11.1		. 17	EN MITEN KITE KI	Аркуш
	(H)	KAN	TEX	10	КНТЕУ 121 06-22.МР	17
Зм.	Аркуш	№ докум	Підпис	Дата	CAEL RATE CURLER	CH L

просою поведінкою, що дозволяє розробляти системи підвищеної складності по частинах.

На рисунку 1.10 наведена діаграма об'єктів даної предметної області.



Рис. 1.10 - Діаграма об'єктів предметної області

Джерело: «Розробка автора»

1.6 Постановка задачі

Перед авторами стоїть задача - розробити систему моделювання роботи машини Т'юринга.

Перед початком моделювання користувач може або створити власний (новий) алгоритм, або завантажити стандартний (в програмі повинна бути реалізація двох алгоритмів: складання і НОД), або з файлу.

11		11. 53		. 17	EVENTE KITE K	Аркуш
	H	- ANO	TEX	10	КНТЕУ 121 06-22.МР	18
Зм.	Аркуш	№ докум	Підпис	Дата	TE WATER WHIER	

Алгоритм повинен бути представлений в табличному вигляді (див. Рис. 1.3), число рядків в таблиці фіксоване (обмежене алфавітом), кількість стовпців (станів) користувач може змінювати (в межах від 1 до 100). Користувач повинен мати можливість додавати / видаляти будь-які стани. Щоб додати новий стан, користувач повинен вибрати яку кількість стовпців йому необхідно додати. Ці стовпці додадуться в кінець. Для видалення стану слід виділити відповідний стовпець. Команди управління МТ повинні розташовуватися в осередках таблиці. Семантичний і синтаксичний аналіз команд на даному етапі не проводиться. Користувач повинен мати можливість зберегти алгоритм в файл.

В системі повинен бути передбачений алфавіт «За замовчуванням» (Е * 10 =), але при необхідності користувач повинен мати можливість його змінити (максимальна кількість символів в алфавіті - 100). Перед запуском МТ користувачеві необхідно задати значення операндів або в цифровому вигляді (в десятковій системі числення), або в символьному (безпосередньо на стрічці), а також вибрати один з режимів роботи (покроковий, автоматичний, миттєвий). При автоматичному способі, користувач повинен задати швидкість виконання за допомогою повзунка в межах від 15 мс до 500 мс. В процесі виконання, повинна підсвічуватися поточна комірка стрічки і команда управління МТ.

У програмі повинна бути можливість отримання траси з можливістю запису її в текстовий файл.

Після запуску МТ буде проведений синтаксичний і семантичний аналіз введених команд, при помилці користувач повинен бути повідомлений про ці помилки. Якщо помилок немає, то МТ почне свою роботу відповідно до обраного режиму роботи.

~	N.	, Kr.	2 M' V	N.	K HI KI JI KK	TEIL
11		11. 53		11	E TE KITE K	Аркуш
1	(H)	KAUK	TEX	10	КНТЕУ 121 06-22.МР	10
Зм.	Аркуш	№ докум	Підпис	Дата	TE KHTER KHTEK	19

В системі також повинна бути забезпечена можливість отримання довідкової інформації, як про саму систему, так і про запропоновані нею можливості.

Таким чином, система повинна виконувати наступні функції:

- візуалізація процесу роботи МТ;
- синтаксичний аналіз алгоритму;
- семантичний аналіз алгоритму;
- видача довідкової інформації про систему;
- вибір способу завдання операндів;
- редагування символу на стрічці;
- вибір режиму роботи МТ;
- вибір швидкості візуалізації роботи МТ;
- створення алфавіту;
- редагування алфавіту;
- вибір стандартного алгоритму;
- ручне складання алгоритму;
- запис алгоритму в файл;
 - завантаження алгоритму з файлу;
- збереження траси виконуваного алгоритму;
 - збереження траси в файл.

L		NV V	K		1 Y I	C. C. M. M. M. M.	JV V
	11		E7		1.17		Аркуш
5		(H)	KNU	TEX	10	КНТЕУ 121 06-22.МР	20
	Зм.	Аркуш	№ докум	Підпис	Дата	TE KHTE KUHTEK	20

1.7. Висновки до розділу 1

Отже, машина Т'юринга є розширенням кінцевого автомата і, згідно з тезою Черча-Т'юринга, здатна імітувати всіх виконавців (за допомогою задання правил переходу), будь-яким чином реалізувати процес покрокового обчислення, в якому досить елементарний кожен крок обчислення.

Перед автором стоїть задача - розробити систему моделювання роботи машини Т'юринга.

Перед початком моделювання користувач може або створити власний (новий) алгоритм, або завантажити стандартний (в програмі повинна бути реалізація двох алгоритмів: складання і НОД), або з файлу.

5		N. KAT	- K' V	HI	- KI HI - KN JI - KN	SE' V
11				. 17	EN MITE KITE K	Аркуш
	H	KNUK	TEX	10	КНТЕУ 121 06-22.МР	21
Зм.	Аркуш	№ докум	Підпис	Дата	TEN KY TEN KHIEK	21

РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Структурна схема системи

Під системою розуміється єдність цілей, ресурсів і будови (структури), властивості яких визначаються відношенням і входять у розглянуту сукупність об'єктів (елементів).

Структурна схема - це сукупність елементарних ланок об'єкта і зв'язків між ними, один з видів графічної моделі, яка розробляється на початкових стадіях проектування. Вона визначає основні функціональні частини, призначення і взаємозв'язки між ними [14]. Структурна схема системи приведена на рисунку 2.1.



Рис. 2.1. Структурна схема системи

A IL			зрооки	ивтор	кнтеу 12	21 06-22.1	MP	NHT
Зм.	Аркуш	№ докум.	Підпис	Дата	HI-KANTE	- KR	TE	, Ar
Зав	. каф.	Криворучко О.В.	11.	07.09.20	Візуалізація роботи	Стадія	Арку	Аркушів
Кеј	оівник	Десятко А.М.	N	07.09.20	алгоритмів машини	P2	22	55
Гар	ант	Криворучко О.В.	NK	07.09.20	Т'юрінга	Факультет	інформ	аційних
Pos	вробив	Чернієнко А.І.	121-1	07.09.20	TANKE	mex	нологій	
7	T	I'NK'	FE.	Kriz	проектування системи	2м кур	ос, 6 гру	ma

Джерело: «Розробка автора»

До складу системи входять наступні підсистеми:

 файлова підсистема, що дозволяє зберігати і завантажувати алгоритми;

 підсистема складання алгоритму, яка відповідає за складання алгоритму;

3) *підсистема налаштування параметрів*, що дозволяє задавати алфавіт і кількість станів;

4) *підсистема моделювання роботи МТ*, яка проводить синтаксичний і семантичні аналізи, а також виконує команди;

5) *підсистема налаштування параметрів моделювання*, яка дозволяє налаштовувати режим роботи і спосіб завдання операндів;

довідкова підсистема, яка видає відомості про систему (керівництво користувача) і про її розробників;

7) *підсистема візуалізації*, яка відображає процес виконання алгоритму.

2.2 Специфікація системи

Специфікація вимог до ПО (SRS) - процес формалізованого опису функціональних і не функціональних вимог і вимог до характеристик якості, відповідно до стандарту якості ISO / IEC 9126-94, які будуть відпрацьовуватися на етапах ЖЦ ПЗ [15].

5	1CV	N. Kri	- K' V	H.	2 KI HI KA JI F KA	TE'V
11				. 17	EN MITE KITE KI	Аркуш
	(H)	KNUK	TEX	10	КНТЕУ 121 06-22.МР	23
Зм.	Аркуш	№ докум	Підпис	Дата	KET KNIE MUCH EK	20

2.2.1 Функціональна специфікація

Функціональна специфікація включає в себе перелік всіх функцій системи з прив'язкою їх до конкретної підсистемі і до інформаційного середовища (вхідні та вихідні дані), перелік виняткових ситуацій і реакцію системи на їх виникнення [15].

Функціональна специфікація системи наведена в таблиці 2.1.

Таблиця 2.1

Назва	Назва	KHIEK	Інфомаційне	середовище	TE	
підсистеми	функції	Вхідні ,	данні	Вихідні	ідані	
Y KHUTE	EXKIN	Призначення (назва)	Тип, обмеження	Призначення (назва)	Тип, обмеження	
E MILK	2	3	4	5	6	
	1.1. Видати відомості про розробників	Відомості про розробників системи (ПІБ, номер групи)	Текст (МЕМО)	Візуальне відображення	TERK	
ГСправка	1.2. Видати відомості	Файл справки	Текстовий	пформаци	KHTE	
KHI	розробників	KRHTE	(*.cnm)	Код помилки	Ціле	
EXNUL	(EXNU)	Кількість станів	Ціле,1100	EYKINT	ET KAN	
2 Склонония	2.1. Ввести	Алфавіт	Масив символів	Комонна	Рядок	
алгоритму	переходу	Команди управління МТ	R, L, S	Команда		
Джер	рело: «Розроб	ка автора»	EKNUTE EEXKNUTE TEXKNUTE	E KNUTE TEX KNUT TEX KNUT		
NEE	IT SEL	L'ELL	MITEJ	KITE	Аркуш	
Зм. Аркуш Л	© докум Підпи	с Дата	КНТЕУ 121	06-22.MP	24	

Перелік функцій, які виконуються системою

2.2.2 Специфікація якості

Кожна система повинна виконувати певні функції, володіти рядом властивостей, що дозволить успішно її використовувати протягом тривалого періоду, тобто володіти певною якістю.

Якість - це сукупність рис і характеристик, які впливають на здатність системи задовольнити задані потреби користувачів. В даний час критеріями якості системи прийнято вважати:

формулюються вимоги по швидкодії;

вимоги до розробки інтерфейсу і т.д;

контроль введення всіх параметрів в системі повинен
 здійснюватися з прив'язкою до діапазонів допустимих значень;

- повинна бути реалізована обробка виняткових ситуацій.

2.2.3 Перелік виняткових ситуацій

Виняткова ситуація - це ситуація, при якій система не може виконати покладених на неї функцій або яка може привести до денормалізації роботи системи [15].

У таблиці 2.2 наведено перелік виняткових ситуацій для розроблюваної системи і описані реакції системи на їх виникнення.

Y	2.	EV	N' KH'	2KY	sH1	KATEKPE	TE'V
	11				1.17	EN MITEN KITENK	Аркуш
5	10	(H)	XN.	TEX	10	КНТЕУ 121 06-22.МР	25
	Зм.	Аркуш	№ докум	Підпис	Дата	TEN KATER KATER	20

2.3.Розробка прототипу інтерфейсу користувача системи

Інтерфейс - сукупність елементів, що дозволяють користувачеві управляти роботою програми або обчислювальної системи і отримувати необхідні результати. Розробка користувальницького інтерфейсу включає наступні основні етапи:

постановка завдання - визначення типу інтерфейсу і загальних вимог до нього;

аналіз вимог і визначення специфікацій - визначення сценаріїв використання і користувальницької моделі інтерфейсу;

проектування - проектування діалогів і їх реалізація у вигляді процесів введення-виведення;

реалізація - програмування і тестування інтерфейсних процесів.

Таблиця 2.2

Назва підсистеми	Назва Виключної ситуації	Реакція системи		
1 Справка	1.1 Неможливо відкрити файл справки	Видача повідомлення «Файл довідки пошкоджений»		
	1.2 Неможливо відкрити файл справки	Видача повідомлення «Відсутній файл довідки»		
2 Файлова	2.1 Неможливо відкрити файл алгоритму	Видача повідомлення «Файл алгоритму пошкоджений»		
3 Моделювання	3.1 Знайдені семантичні помилки	Видача повідомлення «Семантична помилка: <показує на помилку>»		
EVKITE	3.2 Знайдені синтаксичні помилки	Видача повідомлення «Синтаксична помилка: <покажчик на помилку>»		

Перелік виняткових ситуацій

Джерело: «Розробка автора»

11		11.1			EN MITENKITE KI	Аркуш
	(1)	KNUK	TEX	NV.	КНТЕУ 121 06-22.МР	26
Зм.	Аркуш	№ докум	Підпис	Дата	A ST AN A ST AN A ST AN	20

При проектуванні, призначених для користувача інтерфейсів, необхідно враховувати психофізичні особливості людини які пов'язані зі сприйняттям, запам'ятовуванням і обробкою інформації [16].

Прототип інтерфейсу користувача представлений на рисунках 2.2-2.5.

На рисунку 2.2 представлений прототип головної екранної форми, на якій повинні розташовуватися: стрічка, таблиця переходів, параметри режиму роботи, режим візуалізації, операнди і їх режим завдання, збереження і відкриття траси.



Операнди можна задати як на формі, так і безпосередньо на стрічці, для цього слід вибрати відповідний перемикач в панелі «Операнди».

Якщо є необхідність вести файл траси, то потрібно вибрати відповідну кнопку «Зберігати» на панелі «Траса». Після виконання алгоритму трасу можна відкрити відповідною кнопкою.

Прототип екранної форми налаштувань представлений на рисунку 2.3, де можна вибрати такі параметри як: довжина алфавіту, довжина стрічки, кількість станів, а також повинна бути реалізована можливість створити свій алфавіт.

На рисунку 2.4 і рисунку 2.5 представлені форми «Про систему» і «Керівництво користувача» відповідно.

На рисунку 2.6 представлена навігаційна модель додатки.



Рис. 2.3 - Прототип екранної форми «Налаштування»

Джерело: «Скріншот автора»



Рис. 2.4 - Прототип екранної форми «Про систему»

Джерело: «Скріншот автора»

5	EV	ST KH	Kr.	(H)	KRUTEKRUTEKR	TEX
11		11. 53			EN MITER KITERK	Аркуш
	(H)	KNUT	TEX	AV.	КНТЕУ 121 06-22.МР	28
Зм.	Аркуш	№ докум	Підпис	Дата	THE KATER KHIEK	20


Рис. 2.6. Навігаційна модель додатки

Запуск/пауза

Джерело: «Скріншот автора»

2.4 Розробка і опис алгоритмів обробки даних

Поняття алгоритму обробки даних використовується для опису методу розв'язання задачі, який в подальшому можливо реалізувати в вибраному середовищі програмування [17].

Алгоритм - набір інструкцій, що описують порядок дій виконавця для досягнення певного результату [18].

Діаграма діяльності при роботі з програмою приведена на рисунках 2.7 і 2.8.

2	K	r. Kri	- K'	H'	2 KI HI KALITE KAL	TE'V
11		11.1		. 17	EN MITE WITE W	Аркуш
	(H)	KNUK	TEX	20	КНТЕУ 121 06-22.МР	20
Зм.	Аркуш	№ докум	Підпис	Дата	KEN KNIEN KHIEK	29





Рис. 2.8 – UML-діаграма Діяльності «Моделювання роботи МТ» Джерело: «Розробка автора»

При виборі пункту «Файл» з'являється можливість зберегти або завантажити алгоритм з файлу.

При виборі пункту «Робота з алгоритмом» спочатку необхідно задати алфавіт. Для цього слід вказати довжину алфавіту і допустимі символи. Крім завдання алфавіту можна змінити кількість станів (за замовчуванням 3) <u>Аркуш</u> <u>3м.</u> *Аркуш* № докум Підпис Дата шляхом додавання по одному, або вказати кількість, а також задати команди. Для задання команди потрібно вибрати комірку, в якій вона знаходиться і змінити її.

Пункт «Параметри моделювання» відповідає за можливість вибору режиму роботи програми, а також дозволяє зберігати трасу програми в файл.

Вся інформація про систему може бути отримана при виборі відповідних пунктів.

2.5 Вибір і обгрунтування комплексу програмних засобів

2.5.1 Вибір мови програмування і середовища розробки

Обраною мовою програмування є С #. Це мова програмування для розробки додатків, призначених для виконання в середовищі .NET Framework. Мова С # проста, безпечна і об'єктно-орієнтована. Завдяки безлічі нововведень С # забезпечує можливість швидкої розробки додатків.

У середовищі .NET можна розробляти практично будь-який додаток для запуску в Windows. Наприклад, із застосуванням С # і .NET Framework можна створювати динамічні веб-сторінки, додатки Windows Presentation Foundation, веб-служби XML, компоненти для розподілених додатків, компоненти для доступу до баз даних, класичні настільні додатки Windows i навіть клієнтські програми нового інтелектуального типу, володіти можливостями для роботи в оперативному і автономному режимах [19].

R	K	r' Kr'	· K ·	HI	KI HIE KA JE KA	TE'V
1	115			. 17	EN MITEN KITE KI	Аркуш
12	(H)	KR	TEX	30	КНТЕУ 121 06-22.МР	32
Зм	. Аркуш	№ докум	Підпис	Дата	TEN KATEN KHIEK	JZ

2.5.2 Вибір операційної системи

В якості операційної системи була обрана система MS Windows 10. Windows 10 - операційна система сімейства Windows NT, виробництва корпорації Microsoft. Вона підтримує MS Visual Studio 2017 і надає розробникам всі необхідні засоби для створення якісного додатка.

2.5.3. Вибір середовища програмування

Як середовище розробки була обрана MS Visual Studio 2017. Microsoft Visual Studio надає кілька способів розробки додатків для Windows, що працюють локально на комп'ютерах користувачів. Перевага використання Visual Studio полягає в тому, що це середовище надає кошти, завдяки яким процес розробки додатків стає набагато швидше, простіше і надійніше [20].

У Visual Studio 2017 є все необхідне для створення відмінних додатків для пристроїв, настільних додатків, веб-додатків і хмарних рішень. Володіти перевагами простої навігації по коду, швидкої збірки і розгортання в найкоротші терміни. Visual Studio підвищує продуктивність і дозволяє зручно працювати самостійно або в складі команди [21].

2.6. Висновки до розділу 2

У розділі 2 мною досліджено особливості системи, її специфікація та було обрано мову програмування і середовища розробки.

Обраною мовою програмування є С #. Це мова програмування для розробки додатків, призначених для виконання в середовищі .NET Framework. Мова С # проста, безпечна і об'єктно-орієнтована.

	1			1		
11				.17	EN A TENKITE K	Аркуш
	H	KAU	TEX	10	КНТЕУ 121 06-22.МР	33
Зм.	Аркуш	№ докум	Підпис	Дата	TEN KATEN KHIEK	00

Як середовище розробки була обрана MS Visual Studio 2017. Microsoft Visual Studio надає кілька способів розробки додатків для Windows, що працюють локально на комп'ютерах користувачів.

3	E V	S. KHI	KA	(HI	KR HTE KRUTE KRU	TEX
11				1.17	EN MITEN KITE KI	Аркуш
	(H)	KNOK	TEX	NV.	КНТЕУ 121 06-22.МР	34
Зм.	Аркуш	№ докум	Підпис	Дата	シューエッショーショー	34

РОЗДІЛ З РЕАЛІЗАЦІЯ СИСТЕМИ

3.1. Розробка і опис інтерфейсу користувача

Інтерфейс користувача - сукупність засобів і методів, за допомогою яких користувач взаємодіє з різними, найчастіше складними, машинами, пристроями та апаратурою [22].

Після запуску системи відкриється головна форма (Рис. 3.1).

Алго	орити	M	Дові	цник	кори	стува	на	До	відка	B	ихід												
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	1
Е	E	E	E	E	E.	E	E	E	E	E	E	E	Ê	E	E	E	E	E	E	E	E	E	T
<	SU	1	EX		in	1	E	6	X	11	1	X	N	T	1	Y	1	1	E		4	.1	3
./	q0	Ĵ,	q1	기	g2	<u> </u>	TF	Y	1	ÚL	1	1	1-1	11		-	K	1	J	X	1K	11	1
E		KX		E	(· ·	N																	
R		N	3	1	Z																		
1<	N.S		XY		E	6																	
0	18	Ņ	1	S	1																		
R		- \	24	N	11																		
	11	1	1	L.	11	F	1	K	1	TE		K		TF		, L	H	17	E		H	11	<
Pe	жим	робо	м					N		Опер	анди	X		3			N	pacca	Ż				
9	Покр	око	зий			1	E		X	• H	а фор	ima		A		Į] 36e	рігат	и тра	ccy		
0	Авто	мат	ичний		T	P	T	7	'L	OH	а стр	ічці		B	1	Ð		Відк	PKIN	Ŀ	36ep	егти	1
0	Мит	гевий	1		15	50 10	00 25	0 50	0			1	Очис	тити	стріч	ку	E						
	ухат	истр	ічку		5	A	10	Th	E						P		KE	Кро	ZX		Ст	арт	7
7	7		Ń	0	1	E	Р	ис.	3.1	. Fo	лог	вне	вік	но	про	гра	ми		7	K.	ίÚ	15	
	1	E			5		1	X		~~	N.		N			- P -	2						
	07	Įж	epe.	ло:	«Po	озра	οοκι	а ав	зто	pa»	1												
1	K		9		14		~	X		(Y)	1	7 2		N	Y	2	4	N	T		V		Ś
~							VV										1.1	KI					
	10		1	V	1	C		· Y		110			$\supset K$	H7	FV	11	21	06-	22	MF	2		
И.	Аркуі	u	N₂ d	докул	1.	П	ідпис		Дат	a			K	ΉĪ	ΈУ	/ 12	21 (06-	22.	Mł	2		

Зав. каф.	Криворучко О.В.	19.10.20	Візуалізація роботи	Стадія	Арку	Аркушів
Керівник	Десятко А.М.	19.10.20	алгоритмів машини	P3	34	55
Гарант	Криворучко О.В.	19.10.20	Т'юрінга	Факультет	1 інформ	аційних
Розробив	Чернієнко А.І.	19.10.20	Degainquig quomoun	mes	кнологій	
7.7	E'NK'SE	Kr.	Реалізація системи	2м ку	рс, 6 груг	na

3.1.1 Розробка і опис користувальницького меню

Меню - елемент інтерфейсу користувача, що дозволяє вибрати одну з декількох перерахованих опцій програми [23].

На рисунку 3.2 наведена структура меню. В меню верхнього рівня включені наступні пункти: «Алгоритм», «Керівництво користувача», «Довідка», «Вихід». При натисканні на пункт «Алгоритм» можна вибрати одну з трьох функцій: новий алгоритм (Рис. 3.3), відкрити алгоритм (Рис. 3.4), зберегти як алгоритм (Рис. 3.5) і одного розділу - стандартні. При наведенні на цей розділ з'явиться можливість вибрати алгоритм складання (Рис. 3.6) або множення (Рис. 3.7).

		1.10		2		K		K	-	15	1,	1	-		1			1
Riakoutu	5	6	17	8	9	10	(1D	12	13	14	15	16	17	18	19	20	21	22
Зберегти як	E	E	E	E	E	E	E	Ę	E	E	E	E	E	E	E	E	E	E
Стандартні	2	Д	одава	ння	Ż		K	<u>1</u>	1			4			10		$\overline{\mathbf{b}}$	
uv ur		M	ноже	ення		16												
ATE V		A	F	N	20	1												
E'TE'	14																	
A ATE		44																
THE L		A																
END	TE	1																
Режим роботи	TE	1	K	Ľ,	-	Опер	ранди	4	~	<u>z</u> Y		H	Ţ	pacca	<u>Y</u>	N	17	-0
						OH	а форг	ni 🔪		A	1	Ð	P	36e	рігат	и тра	ccy	
О Покроковий												()					\sim	
 Покроковий Автоматичний 	KH			Ì ¥		OH	la стріч	ų,		В	1	÷	L A	Відку	рити	4	Збер	егти
 Покроковий Автоматичний Миттевий 	KH (15)	50 1	00 2!	50 50	0	O H	la стріч	ii (Очис	В	1 cmiu			Відку	PUTU	T	36ep	er tu
 Покроковий Автоматичний Миттєвий 	(15) (15)	50 1	00 2	50 50	0	O H	la стріч		Очис	В	1	A KY	A	Відку	рити	Y.	35ep	ertu JU

При натисканні на пункт меню «Керівництво користувача» відкриється файл формату .chm для отримання докладної інформації про програму (Рис. 3.8). При виборі пункту «Довідка» можна подивитися відомості про розробників. Пункт «Вихід» закриває програму.





Джерело: «Розробка автора»

	комутатор / Разочий стол / МП	i z Mig		тюйск: А	9
порядочить 🔻 Созда	ть папку				III - 🔟 🔇
	Имя	Дата изменения	Тип	Размер	
🖈 Быстрый доступ	Alg1.tma	19.12.2017 3:58	Файл "ТМА"	1 K5	
💻 Этот комутатор 🖈	Alg2.tma	19.12.2017 3:58	Файл "ТМА"	1 K5	
📃 Рабочий стол 🖈	BEST ALGO EVER.tma	19.12.2017 3:58	Файл "ТМА"	1 K5	
🖊 Загрузки 🛛 🖈	Nod.tma	19.12.2017 3:59	Файл "ТМА"	1 K6	
🝐 Google Диск 🖉 🖈	Sum.tma	19.12.2017 3:59	Файл "ТМА"	1 КБ	
🛖 Seagate (F:) 🎐 Сеть					
🔩 Домашняя группа					
	KHITEK	KHTT	KIN	ATE	ANTI



Джерело: «Розробка автора»

X	EV	Ch KHI	EKA,	TH	KRUTE KRUTE KNO	TEI
1		11. 53		1.17	EN MITER KITERK	Аркуш
	(H)	KNUK	TEX	NU.	КНТЕУ 121 06-22.МР	37
Зм.	Аркуш	№ докум	Підпис	Дата	THE KATER KHIEK	57

Упорядочить 👻 Созда	ать папку				
	Имя	Дата изменения	Тип	Размер	
🖈 Быстрый доступ	Alg1.tma	19.12.2017 3:58	Файл "ТМА"	1 КБ	
💻 Этот комутатор 🖈	Alg2.tma	19.12.2017 3:58	Файл "ТМА"	1 K6	
📃 Рабочий стол 🖈	BEST ALGO EVER.tma	19.12.2017 3:58	Файл "ТМА"	1 K6	
👆 Загрузки 🛛 🖈	Nod.tma	19.12.2017 3:59	Файл "ТМА"	1 КБ	
💧 Google Диск 🛛 🖈	Sum.tma	19.12.2017 3:59	Файл "ТМА"	1 KG	
👝 Seagate (F:) 💣 Сеть					
• Домашняя группа				KHTE	
Имя файла: Аlgo10	0	11 (3)		1,20	
True 1 and 2					



Джерело: «Розробка автора»

1	q0	q1	q2	q3	q4	q5
E	q0 E R	q2 = R	q31L	4	q5 E R	1.1
•	q0*R	q1 . R	1.75	q3*L	q4 * L	EK.
1	q10 R	q11R	q21R	q31L	UL	q51S
0	(H)	EK	1H2	q0 0 R	q4 1 L	(EE)
Ŧ	q4 = L	q2 = R	11	q3 = L	K	JTE

Рис. 3.6 Стандартний алгоритм складання

Джерело: «Розробка автора»

	q0	q1	q2	q3	q4	q5	q 6	q7	q 8	q9	q10	q11
E	q0 E R	q2 = L	q3 E R	q3 E R	1	10	q9 E R	q8 1 R	q5 E L	16.	101	Y L
•	K	q1°B	q2°L	q4 * R	FR	q6 * L	L.C.	q7*R	TE.	q10 * R	171	q11 * S
1	q11R	q11R	q21L	q3 1 R	q5 0 R	q51L	q70 R	q71 R	11.	FIL	q5 0 R	KE
0	KL	141	EV	N.J	TE	q5 0 L	q6 0 L	q70 R	40.7	q91R	q10 0 R	q111L
5		C	E.	IKI	TE	q5 = L	1	q7 = R	XX	AF.	q11 = L	ALF



Джерело: «Розробка автора»

1	-CV	N. Krig	- K' .	'H'	- KI HI KIN JI F KR	TE'V
11		11.1			ENTERKITE KI	Аркуш
	(H)	KNUK	TEX	10	КНТЕУ 121 06-22.МР	28
Зм.	Аркуш	№ докум	Підпис	Дата	ションアンエントン	- 30

3.1.2 Робота з алгоритмом

Основними елементами форми є «Стрічка» (Рис. 3.8) і «Правила переходу» (Рис. 3.9) або «Алгоритм».

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	2
E	1	1	$) \odot$	1	1	1	E	E	E	E	Е	E	E	E	E	E	E	E	E	E	Е	E	E
<	2	1		11		V	1	17	F	-	V	14	1	0		1	200	1	J		P	<u> </u>	2



Джерело: «Розробка автора»

	q0	q1	q2
E	q0 E R	IL	
5		q2 0 L	J.F.
1)	q11R	Kui	TE
0	TE	1K	q3 1 S
-	171	P.V	MY.

Рис. 3.9. Правила переходу МТ

Джерело: «Розробка автора»

При натисканні правою кнопкою миші на стан з'явиться контекстне меню стану (Рис. 3.10) з трьома пунктами: «Додати стан», при виборі його підпунктів в кінець додається потрібна кількість станів, «Видалити стан», при натисканні на який поточний стан видалиться і «Форматувати стан », який перейменує їхні капітали в такий вигляд: q0, q1, q2, ..., qn.

r an	E V	1 KHI	2KM	(H)	KRUTEKRUTEKR	TEX
11				1.17	EN MITEN KINTENK	Аркуш
	(H)	KAU	TEX	10	КНТЕУ 121 06-22.МР	30
Зм.	Аркуш	№ докум	Підпис	Дата	A HILL AN A HILL A	



Рис. 3.10. Контекстне меню стану

Джерело: «Розробка автора»

При натисканні правою кнопкою миші на алфавіт, з'явиться контекстне меню алфавіту (Рис. 3.11) з трьома пунктами: «Додати рядок», який додає цілий рядок, «Видалити рядок», при виборі якого поточний рядок видаляється і «Відновити алфавіт», при натисканні на який алфавіт заміниться на стандартний.

Додати стан Додати строку Форматувати стани Відновити алфавіт

Рис. 3.11 - Контекстне меню алфавіту

Джерело: «Розробка автора»

В осередку на перетині алфавіту і станів можна записувати правила (Рис. 3.9)

5	E	A KH	EKI,	141	KR HTE KRUTE KR	TE'V
11				1.17	E MITE KITE KI	Аркуш
	(H)	KNOW	TEX	10	КНТЕУ 121 06-22.МР	40
Зм.	Аркуш	№ докум	Підпис	Дата	TENKNIEN KHIEK	40



Рис. 3.12. Панель налаштування параметрів системи Джерело: «Розробка автора»

В панелі «Режим роботи» є можливість вибору одного з трьох режимів роботи: «Покроковий», при якому МТ робить 1 крок після натискання кнопки «Крок», «Автоматичний», при якому МТ працює з певною затримкою (15мс, 50мс, 100мс, 250мс, 500мс) і «Миттєвий», при виборі якого візуалізація не передбачена, а швидкість роботи дуже висока (понад 9000 ітерацій в секунду).

На панелі «Операнди» можна вибрати яким способом буде вироблено завдання операндів: «На формі» або «На стрічці». Також присутня кнопка, яка очищає стрічку, тобто замінює всі її символи на порожні («Е» за замовчуванням).

В панелі «Траса» існує перемикач, що дозволяє зберігати трасу і дві кнопки: «Відкрити», яка відкриває трасу в текстовому поданні (Рис. 3.13) і «Зберегти», яка зберігає трасу в текстовий файл (Рис. 3.14).

1	s	EV	CI KHI	EKA,	KII	KRUTE KRUTE KR	TEXI
	11		11. 53			E MITE KITE K	Аркуш
2		(1)	NN.	ALX	NV.	КНТЕУ 121 06-22.МР	11
0	Зм.	Аркуш	№ докум	Підпис	Дата	TEN KHIEK KHIEK	HIL





Джерело: «Розробка автора»

орядочить 🔻 Созд	дать папку				III - ()
Быстрый доступ	Имя	Дата изменения	Тип	Размер	
Этот комутатор * Рабочий стол	M5.tmt	19.12.2017 4:35 19.12.2017 4:35	Файл "ТМТ" Файл "ТМТ"	40 K6 40 K6	
🖡 Загрузки 🖈					
oneDrive					
Этот комутатор					
Seagate (F:)					
Сеть Домашняя группа					
Имя файла:	TE! NU	TE AL		KUTE	
Тип файла: Turin	g machine track (*.tmt)	ILL KI	TE		2.1



Джерело: «Розробка автора»

Підпис

Дата

Аркуш

Зм.

№ докум

3.1.4 Моделювання роботи МТ

Після налаштування всіх параметрів необхідно натиснути кнопку «Старт», після чого відбудеться синтаксичний і семантичний аналізи введеного алгоритму. Приклади помилок приведені на рисунках 3.15-3.18. Аркуш <u>Аркуш</u> <u>42</u>

10	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	2
E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
10	7	1	JU		1	K	11	27		K		15	V	N	T	
02	1	X	1 al	111	1	Y	-	17	1-		A	. ~	E	2	K	1
qz																
q311	- /			20				1	0		- 6		10			~
q3 7 I	4			10											>	<
q3 7 1				KY												<
q3 7 1	4			14	нтак	ГИНН		K	ABQ	цейш	E	ВИЛГ	lenex	0.11/2		
q37				Си	інтак	сичн	іа поі відпо	милк відає	а в я алф	чейці авіту	пра	вил г	iepex	оду 2	;3	
q3 7	K V			CM	інтак Імвол	сичн л не в	іа поі відпо	милк відає	а в я алф	чейці авіту	пра	вил г	iepex	оду 2	3	
q3 7	K			2.2	інтак імво	сичн л не п	іа поі відпо	милк відає	а в яч алф	чейці авіту	i npa	вил г	iepex	оду 2	3	

Рис. 3.15 - Синтаксична помилка алгоритму

Джерело: «Розробка автора»

_															
на До	відка	В	ихід			11			-		5	·	1		2
7 8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
EE	E	E	Е	Е	E	E	Е	E	E	E	Е	E	E	E	
		1	11		1	1	5	. 1		V		Y	5	1	1
	1	9	. <	K	1	. /	5	10		- C	-		15	1	
		~	11				<u> </u>		1	\bigcirc		1			
													X	V	
	-												×	X	
TE		Син	такси	чна	поми	ілка в чейці	з яче	йці п	рави	л пер	реход	ıy 3;1	×	X	
jTF		Син Неві	такси ірне ч	чна	поми о в яч	ілка в нейці	з яче	йціп	рави	лпер	oexod	ıy 3;1	×	X. GU	
TE	144	Син Неві	такси ірне	числ	поми о в яч	ілка і іейці	яче	йці п	рави	л пер	реход	ay 3;1	×	N. C.V.Y.Y	
TE	1 4 1	Син Неві	такси	числ	поми о в яч	илка в нейці	яче	йціп	рави	л пер	реход	цу 3;1 ОК	×	Y ANY VE	
0 250 50	•	Син Неві	такси	числ	поми о в яч Очис	илка е нейці стити	стріч	йці п	рави		реход	цу 3;1 ОК	×	Y GUL VE	
	4a 240 7 8 E E	1а Довідка 7 8 9 Е Е Е Е	13 ДОБІДКА В 7 8 9 10 E E E E E	13 Довідка вихід 7 8 9 10 11 Е Е Е Е Е Е	4a Довідка вихід 7 8 9 10 11 12 E E E E E E E	13 206/2Ka BUX/2 7 8 9 10 11 12 13 E E E E E E E E	13 ДОВІДКЗ БИХІД 7 8 9 10 11 12 13 14 E E E E E E E E E	та довідка вихід 7 8 9 10 11 12 13 14 15 Е Е Е Е Е Е Е Е Е Е	та довідка вижід 7 8 9 10 11 12 13 14 15 16 Е Е Е Е Е Е Е Е Е Е Е	13 2061QK3 BHX1Q 7 8 9 10 11 12 13 14 15 16 17 E E E E E E E E E E E E E	13 ДОВІДКЯ БИХІД 7 8 9 10 11 12 13 14 15 16 17 18 E E E E E E E E E E E E E E E	та довідка вихід 7 8 9 10 11 12 13 14 15 16 17 18 19 Е Е Е Е Е Е Е Е Е Е Е Е Е Е Е	та довідка вихід 7 8 9 10 11 12 13 14 15 16 17 18 19 20 Е Е Е Е Е Е Е Е Е Е Е Е Е Е Е С	13 ДОБІДКЯ БИХІД 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 E E E E E E E E E E E E E E E E E E E	13 ZOBLAX BUXIA 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 E E E E E E E E E E E E E E E E E E E



Джерело: «Розробка автора»

11	TE.	EJ KIT	Ear	1.17	F-J PATE KI TE KI	Аркуш
20	(H)	KNOK	TEX	NO.	КНТЕУ 121 06-22.МР	12
Зм.	Аркуш	№ докум	Підпис	Дата	KEN KNYEN KHIEK	43

истув	зача	До	відка	i B	ихід										
6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	1º
E	E	Ε	E	E	E	E	E	E	E	E	E	E	E	E	I
L	ST	- - C-	17		J	F	Ì	5		T		K	×		
F)	J.	E	1,	N	U	E	, F	4	j	1	1	K	×		
L.	11-1	Синта	аксич	нап	омил	кав	ячей	ці пр	абил	пере	ходу	4;3	× N×		
L	144	Синта Симв	аксич	чна по е відп	омил овіда	ка в н ка в по	ячейі фавіт	ų np	абил	пере	ходу	4;3	XXXX		
	1-19 1 C	Синта	аксич	нна по	омил овіда	ка в и	ячей фавіт	ų p	авил	пере	ходу	4;3	X	A.Y.C.	L' L' L'
	The bar	Синта Симв	аксичол не	нна по е відп	омил овіда	Ka B I	ячейі	ų np	БИЛ	пере	ходу	43	×	итра	× 1

Рис. 3.17 - Синтаксична помилка алфавіту

Джерело: «Розробка автора»

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	1
D	ŀ	1	E	E	E	E	E	E	E	E	E	E	Е	E	E	E	E
-	<u>h</u>				K		17	1	1	Ń		. E	1	×	¥	K	6
000	1.4.1	1	$\wedge \vee$												1000		
lani			Ce	мант	ична	пом	илка	В ЯЧ	йці п	рави	лпе	pexo,	ay 1;1				
	1	14	Ce He	мант існує	ична тако		илка гану в	в яче в спи	ейці п ску ст	ірави анів	IN RE	pexo,	ay 1;1		X		
		Y D T	Cel	мант	ична тако		илка гану в	в яче	ейці п ску ст	ірави анів	Г	pexo,	ду 1;1 ОК	X	XX		
	pofo		Cel	мант	ична тако		илка гану в	в яче	ейці п ску ст Опер	анів		pexo,	ay 1;1 OK	Y LING KA	CX X	H. K. Y. I.	pa

Рис. 3.18 - Семантична помилка алгоритму

S		Д.	жерело: «	Розробк	а автор	a»	
					TL	ZI MITELIKI TE IK	Аркуш
2		(M)	KAN	ALK	NUIT	КНТЕУ 121 06-22.МР	44
2	Зм.	Аркуш	№ докум	Підпис	Дата	EL KALE CKHLEN	HIL

Після проходження всіх перевірок, в залежності від обраного режиму і налаштувань, МТ почне свою роботу. На рисунку 3.19 наведено приклад.

0	1 2	3 4	5 6	Z	8 9	10 11 1
E	0 1	\ <mark>`</mark> _1	VI.	E	È	E E E
<		JU L	J K	JUL-	TRU	UT
R	qO	q1	q2	q3	q4	q5
E	q0 E R	q2 = R	q31L	NU	q5 E R	NU
6	q0*R	q1*R	TE	q3*L	q4 ° L	SHI
1	q10 R	q11R	q2 1 R	q31L	PE	q51S
0	KU	TE	KY.	q0.0 R	q4 1 L	En
-	q4 = L	q2 = R	1AN	q3 = L	1AV	ET

Рис. 3.19 - Візуалізація роботи МТ

Джерело: «Розробка автора»

Коли МТ перейде в фінальне стан «S», з'явиться повідомлення про завершення моделювання (Рис. 3.20).



3.2 Реалізація класів

Основні параметри і функції класів описані в таблицях 3.1-3.5. Діаграма сутнісних класів представлена на рисунку 3.21.

Таблиця 3.1

Реалізація класу «ViewTrack»

.	private string toN – приведення до потрібного числа цифр pryvedennya do potribnoho chysla tsyfr
Функції	public void delTrack – видалення траси
E KY KAT	public void addTrack – додавання в трасу
Дже	грело: «Розробка автора» Додаток А

Таблиця 3.2

	Геалізація класу «ugv»
KH	public static DataGridView StatesStart – ініцаіалізація роботи переходів
	public static DataGridView RibbonStart – ініціаоізація роботи стрічки
	public static DataGridView StatesFormat – пріведення до стандартного формату після змін
Функції	public static DataGridView RibbonFormat – приведення до стандартного формату після змін
	public static DataGridView toWhite – обілення всіх зафарбованих клітинок в різні кольори
	public static DataGridView RibbonClear – очищення стрічки
	public static DataGridView RibbonReadOnly – стрічка тільки для читання

Джерело: «Розробка автора» Додаток А, Б.

Таблиця 3.3

Реалізація класу «File»

	жерело: «Розробка автора» Лодаток В.						
Y is	Prove source source and a soch successive the						
	public static void saveTrack – збереження траси						
	public static void save – збереження алгоритму						
Функції	public static DataGridView open – відкриття алгоритму						
	public static DataGridView openDefault – відкриття стандартного алгоритму						
	private static string toN – приведення до потрібного числа цифр						

11		11. 53			EN MITEN KITE KI	Аркуш
	(H)	LAN I	TEX	10	КНТЕУ 121 06-22.МР	16
Зм.	Аркуш	№ докум	Підпис	Дата	A STANKE A STANKE	40

Таблиця 3.4

	Реалізація класу «Analyzer»
KALT	private static void error – виведення помилок
	public static bool syntacticAnalysis – синтаксичний аналіз
Функції	public static bool semanticAnalysis – семантичний аналіз
	public static bool ribbonAnalysis – аналіз стрічки
	public static bool alphabetAnalysis – аналіз алфавіту
Пжере	ло: «Розпобка автора» Лодаток В

		Таблиця 3.5				
	Реалізація класу «Main»					
AN TEN	private string alphabet – поточний алфавіт	ANTE!				
	private string defaultAlphabet – стандартний алфавіт	KM. TE				
	private int operatingMode – режим роботи	in No. E				
	private int speed – швидкість	ELUT				
	private bool operandsModeForm – спосіб завдання операндів	JK'N				
	private int operand – операнд А	E.K.				
	private int operand – операнд Б	ITE KH				
	private bool saveTrack – чи зберігати трасу	N. EJ				
	Private List <string> fullTrack – траса</string>	HILEN				
	private int currentMouseOverRow – становище миші в dgv					
H. K.	private int currentMouseOverColumn – становище миші в dgv					
Параметри	private int addCount – скільки станів додати					
	private string[] states – список станів					
	private int currentRibonIndex – індекс поточного символу на стрічці					
	private string currentRibonSymbol – поточний символ на стрічці					
	private int currentStateIndex – індекс поточного стану					
	private string currentState – поточний стан					
	private int currentAlphabetIndex – поточний індекс алфавіту					
	private int colorIndexRibon – осередок для перефарбування траси					
	private int colorIndexStates1 – для перефарбування стрічки					
	private int colorIndexStates2 – для перефарбування стрічки					
	private int shift – зсув стрічки					
KR HI	private void blockOnStart() – блокування елементів форми					
D	private void unblockOnFinish() – розблокування елементів фо	рми				
Функції	private void start – дії на старті	IL KI				
F.K.K	private bool step – 1 ітерація МТ					
Джер	рело: «Розробка автора» Додаток Ж.	JUTE K				
J K K	JUE A LIES MITEL WITEL	Аркуш				
JHL V	КНТЕУ 121 06-22.МР	47				
Зм. Аркуш М	è докум Підпис Дата	MILLE				



Рис. 3.21 – UML-діаграма класів

Джерело: «Розробка автора»

2	E V	N KH .	· KI	st'	KAN KANTE KA	TE'V
11		11.1		1.17	ENTERKITE KI	Аркуш
	(H)	KNOK	TEX	NV.	КНТЕУ 121 06-22.МР	18
Зм.	Аркуш	№ докум	Підпис	Дата	LE CAPECANER	40

3.3 Діаграма модулів



Рис. 3.22 - Діаграма модулів

Джерело: [15]

3.4 Вибір і обгрунтування комплексу технічних засобів

Вибір комплексу технічних засобів в значній мірі визначає функціональні можливості і надійність створюваної програми. Вибір комплексу технічних засобів проводиться на основі переліку вирішуваних завдань і їх інформаційних характеристик, специфікою оброблюваної інформації, технічними можливостями обладнання і т.д.

X		A KHI	SKR.	sH1	KRUTEKRUTEKR	TEN
1				1.17	ENTERKITE KI	Аркуш
270	(H)	KNU	TEX	10	КНТЕУ 121 06-22.МР	10
Зм	. Аркуш	№ докум	Підпис	Дата		73

3.4.1 Розрахунок обсягу займаної пам'яті

Розрахунок обсягу зовнішньої пам'яті

Для того щоб зробити і обґрунтувати вибір комплексу технічних засобів, необхідно провести розрахунок обсягу займаної пам'яті (жорсткого диска і ОП).

Для розрахунку необхідного обсягу вільної зовнішньої пам'яті, необхідної для функціонування системи, скористаємося наступною формулою:

VЖД = VOC + VПР + VД,

де VOC - обсяг пам'яті, займаний операційною системою (операційна система Windows 10, VOC = $20 \Gamma 6$);

VПР - обсяг пам'яті, займаний безпосередньо файлами додатка (VПР = 43 Кб);

VД - об'єм пам'яті, займаний стандартними алгоритмами і файлом довідки (VД = ОБСЯГ ФАЙЛА ДОВІДКИ + 1кб)

Таким чином, сумарний обсяг зовнішньої пам'яті складе:

VЖД = $20 \Gamma 6 + 43 K6 + ... \approx 20 \Gamma 6$.

Розрахунок обсягу ОП

Для розрахунку необхідного обсягу ОП скористаємося наступною формулою:

 $VO\Pi = VOC + V\Pi P,$

де VOC - ОП, займане операційною системою (1,6 Гб);

VПР - ОП, яке займе сам додаток (не перевищить 400 Мб);

Сумарні обсяги ОП складе:

VOII = 1,6 Γ 6 + 400 M6 \approx 2 Γ 6.

11			1	1.17
10	1H	- ANO	TEX	10
Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-22.МР

Аркуш 50

3.4.2 Мінімальні вимоги, що пред'являються до системи

Для коректного функціонування системи необхідно:

- 1) монітор з роздільною здатністю 1024х768;
- 2) маніпулятор миша, клавіатура;
- 3) обсяг оперативної пам'яті не менше 2 Гб;
- 4) обсяг вільного дискового простору не менше 1 Мб.

3.5. Висновки до розділу 3

Досліджено та наведено основні елементи системи, її особливості та параметри функціонування.

2		KA	24.	sH'	- KI HI KA JIE KA	TE'N
1				1.17	EN MITEN KITENK	Аркуш
	(H)	- KRUCK	TEX	10	КНТЕУ 121 06-22.МР	51
Зм.	Аркуш	№ докум	Підпис	Дата	TEN KATER KATER	SIL

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В рамках лабораторного практикуму була розроблена система моделювання роботи машини Т'юринга, яка дозволяє користувачеві створювати алгоритми і перевіряти їх працездатність на різних алфавітах і в різних режимах.

У першому розділі була вивчена структура МТ, були знайдені аналоги її програмної реалізації, розібрані деякі алгоритми і сформульована постановка задачі.

На етапі проектування системи були розроблені: структурна схема і специфікація системи, продуманий і проілюстрований прототип інтерфейсу користувача, вибрано і обґрунтовано комплекс технічних і програмних засобів.

На етапі реалізації приведено опис інтерфейсу системи, описана діаграма класів, розрахований обсяг займаної пам'яті і сформульовані вимоги, що пред'явлені до системи.

Ця програма буде корисна студентам і людям, які цікавляться кінцевими автоматами.

L'I	EX	WILLE	NUT	FEY.	КНТЕУ 121 06-22.М.			JK
Зм.	Аркуш	№ докум.	Підпис	Дата	KALE MALE KULLE			
Зав. н	саф.	Криворучко О.В.	EN	30.10.20	Візуалізація роботи	Стадія	Аркуш	Аркушів
Керів	вник	Десятко А.М.	EK.	30.10.20	алгоритмів машини	ВП	52	55
Гарант		Криворучко О.В.	TEY	30.10.20	Т'юрінга	Факультет інформацій		аційних
Розробив		Чернієнко А.І.	H	30.10.20	те		т інформаціиних хнологій	
		THEY	TE	14	Висновки та пропозици	2м курс, 6 група		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- А.А. Шевченко. Концепції моделювання: технології і рішення. // Прикладна інформатика. №6. 2006. URL: https://cyberleninka.ru/article/n/kontseptsii-modelirovaniya-tehnologiii-resheniya/
- Алфавіт (формальна мова) [Електронний ресурс] // Вікіпедія: електрон. енциклопедія. 2001-2017. URL: https://ru.wikipedia.org/ wiki/Алфавіт_(формальна_мова)
- 3. Єнікєєва Є.М. Особливості перекладу комп'ютерних термінів на українську мову. Вісник СумДУ. 2001. № 5 (26). С. 54–59.
- Машина Поста [Електронний ресурс] // Методичні матеріали програмне забезпечення. 2008-2017.
 URL: http://kpolyakov.spb.ru/prog/post.
- Машина Тюринга [Електронний ресурс] // Вікіпедія: електрон. енциклопедия. 2001-2017. URL: https://ru.wikipedia.org/ wiki/Машина_Тюринга
- 6. Машина Тюрінга [Електронний ресурс] // Навчальні матеріали з інформатики. Основні поняття інформатики. Режим доступу : http://www.ua5.org/osnovi/158-mashina-tjuringa.html.
- 7. Основи програмування [Електронний ресурс] // Електронна бібліотека Libr.Org.Ua. Режим доступу : http://libr.org.ua/books/74.html.
- 8. Петрова О., Бурменський Р. Функціональна схема машини Тьюринга для множення числа на 11 // Матеріали VII Українсько-польської науково-

LT	EX	NUTE	TUN I	FEY,	КНТЕУ 121	1P	EJ K	
Зм.	Аркуш	№ докум.	Підпис	Дата	KATERKH			
Зав. каф.		Криворучко О.В.	E'N	24.01.20	Візуалізація роботи алгоритмів машини	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.	SK.	24.01.20		СВД	53	55
Гарант		Криворучко О.В.	TEY	24.01.20	Т'юрінга	Факультет інформацій		апійних
Розробив		Чернієнко А.І.	HI	24.01.20		технологій		20
X		THEY	TE	11	Список використаних джерел	2м курс, 6 група		

практичної конференції «Електроніка та інформаційні технології» (ЕлІТ - 2015), Львів-Чинадієво-2015. С.139-141.

- Петрова О. О., Бурменський Р.В. Моделювання машини Тьюринга обчислення додатку чисел / / Всеукраїнська студентська наукова конференція «Наукова Україна» (з міжнародною участю), м. Дніпропетровськ, 2015. С. 215-218.
- 10.Принципи розробки навчальних програм [Електронний ресурс] //

 StudFiles:
 файловий архів
 студентів.
 2014-2017.

 URL: https://studfiles.net/preview/2114024
- 11.Психофізичні особливості людини [Електронний ресурс] // ТеhProgелектроннийпосібник.2009-2010URL: http://tehprog.ru/index.phppage=lecture43.html2009-2010
- 12.Ресурсова ефективність алгоритмів [Електронний ресурс] // Intuit HOУ «ИНТУИТ». 2003-2017.
 URL: http://www.intuit.ru/studies/courses/648/504/lecture/11461%2525253F
 page%2525253D3
- 13.Спірін О. М. Інформаційно-комунікаційні та інформатичні компетентності як компоненти системи професійно-спеціалізованих компетентностей вчителя інформатики [Електронний ресурс] // Інформаційні технології і засоби навчання. 2009. № 5 (13). Режим доступу до журн. : http://www.ime.edu.ua.net/em.html.
- 14. Теорія алгоритмів [Електронний ресурс] // StudFiles: файловий архів студентів. 2014-2017. URL: <u>https://studfiles.net/preview/5903677/</u>.
- 15. Техніка обчислень і алгоритмізація : навчальний посібник /
 І. Ф. Следзінський, А. М. Ломакович, Ю. С. Рамський, Р. І. Зароський. –
 К. : Вища школа, 1991 199 с.
- Фаліна М. М. Машина Тьюрінга / М. М. Фаліна // Інформатика. № 26. 2005. – С. 12–15.

7	L'.Y	ENK	1ET	14	E MALEN HIEK	Аркуш
1	N	ITE'NI	JITE		КНТЕУ 121 06-22.МР	54
Зм.	Аркуш	№ докум	Підпис	Дата	"HILKPUTE KPUTE	04
		NV N	1	1 .		

- 17. Ю. Г. Карпов. Теорія автоматів. СПб.: Питер, 2003. С. 195.
- Ящик О. Б. Компетентнісний підхід у навчанні об'єктно-орієнтованого програмування як основа підготовки учнів старших класів / О. Б. Ящик // Наукові записки ТНПУ ім. В. Гнатюка – Серія «Педагогіка». – 2011. – № 1. – С. 103–108
- 19. Visual Studio 2013 [Електронний ресурс] // Microsoft: [сайт]. URL: https://msdn.microsoft.com/ru-ru/library/dd831853(v=vs.120).aspx

<u> </u>	A.	EKIK	1 EX	N. L	ALE REALE REALER	Аркуш
Es	UL.	TEN	JTE		KHTEY 121 06-22.MP	ripkym FF
Зм.	Аркуш	№ докум	Підпис	Дата	HTE KRUTE KRUTE?	55

ТЕХНІЧНЕ ЗАВДАННЯ

Однією з найголовніших запорук відповідності вимогам та високої якості розробленого програмного рішення є грамотно спроектоване до початку розробки технічне завдання. Отже, далі розкриємо невід'ємні положення технічного завдання до розробки программи «Машина Т'юрінга»

<u>1. Загальні відомості</u>

Назва проекту: «Машина Т'юрінга».

Планові строки розробки: серпень 2019 – вересень 2019.

Потенційні користувачі: користувачі стаціонарних ПК з встановленою OC Windows.

Призначення програмного рішення: надати можливість користувачу ПК працювати з алгоритмами МТ і бачити хід роботи з точністю до кроку.

Мета створення програмного рішення: розвинути свої навички програмування на С#.

2. Структура проекту

Структура проекту а саме самої системи вказана в пункті «Структура системи». Вона має такий вигляд:

FE	270	KT EK	KHI	E KN	KHTEV 121	06-22 N	ЛР	KAL
Зм.	Аркуш	№ докум.	Підпис	Дата		00 22.1		
Зав. каф	þ.	Криворучко О.В.	JK	28.02.20	Візуалізація роботи	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.	EN	28.02.20	алгоритмів машини	T3	56	55
Гарант		Криворучко О.В.	EM	28.02.20	Т'юрінга	Факультет інформаційни технологій 2м курс, 6 група		аційних
Розробив Чернієнко		Чернієнко А.І.	HTE	28.02.20	Технічне завдання			ла
			AF					



3. Функціональні вимоги до програми

Перелік функціональних можливостей визначений базовими діями, що виконує МТ:

- Створення траси алгоритмів;
- Тестування послідовностей на алгоритмізацію;
- Вирішення задач за допомогою побудованого алгоритму;
- Збереження процесу виконання.

4. Вимоги до програмного забезпечення

Розробка програмного рішення здійснюватиметься із використанням мови програмування С#, у середовищі розробки Visual Studio.

5. Вимоги до технічного забезпечення

Найважливішою вимогою до задіяних пристроїв ϵ наявність комп'ютера з операційною системою Windows не нижче Windows XP.

Серверна частина для ПК має виконуватися за наступних мінімальних конфігурацій:

- Microsoft Windows 10/8/7.
- RAM: 1 Гб і більше.
- Наявність клавіатури і мишки, або тачпаду.

	jUT	TEN	JEF	1		Аркуш
Y	14	L'EKP	KH1.	12	КНТЕУ 121 06-22.МР	58
Зм.	Аркуш	№ докум	Підпис	Дата	KITE WITE WITE	50

додатки

Додаток А

Код класу DVG

using System; using System.Collections.Generic; using System.Drawing; using System.Linq; using System.Text; using System.Threading.Tasks; using System.Windows.Forms; using System.Reflection;

namespace MT

class DGV

//При старте

public static DataGridView StatesStart(DataGridView dgv, string alphabet, int a)
{

dgv.GetType().GetProperty("DoubleBuffered", BindingFlags.NonPublic | BindingFlags.I nstance).SetValue(dgv, true, null);

dgv.Rows.Clear(); //Разрешаем добавлять строки dgv.AllowUserToAddRows = true; //Количество строк = 4(3+1 для определения алфавита) dgv.ColumnCount = a; dgv.Rows.Clear(); //Задаю количество строк = длине алфавита+1, т.к. нулевая ячейка dgv.Rows.Add(alphabet.Length + 1); //Заполняю столбец алфавитом for (int i = 0; i < alphabet.Length; i++) dgv.Rows[i + 1].Cells[0].Value = alphabet[i];

//Рисую состояния

for (int i = 1; i < dgv.ColumnCount; i++) dgv.Rows[0].Cells[i].Value = "q" + (i - 1);

```
//Инициализация каждой ячейки
for (int i = 1; i < dgv.Rows.Count; i++)
for (int j = 1; j < dgv.Columns.Count; j++)
dgv.Rows[i].Cells[j].Value = "";
```

//Запрещаем добавлять строки dgv.AllowUserToAddRows = false;

return dgv;

public static DataGridView RibbonStart(DataGridView dgv, string alphabet, int a, int b)

59

dgv.GetType().GetProperty("DoubleBuffered", BindingFlags.NonPublic | BindingFlags.I nstance).SetValue(dgv, true, null);

```
dgv.AllowUserToAddRows = true;
      dgv.ColumnCount += 1;
      dgv.Rows.Add(b);
      for (int i = 0; i < a - 1; i++)
         dgv.ColumnCount += 1;
         dgv.Rows[1].Cells[i].Value = alphabet[0];
         dgv.Rows[1].Cells[i].ReadOnly = true;
         dgv.Columns[i].FillWeight = 1;
      // dgvRibbon.Rows.Add(b);
      dgv.AllowUserToAddRows = false;
      return dgv;
    //0 строка и 0 столбец - ReadOnly, обрабока 0;0, определяем FillWeight и ширину ячее
к ну и всякую фигню.
    public static DataGridView StatesFormat(DataGridView dgv)
      for (int i = 0; i < dgv.ColumnCount; i++)
         dgv.Columns[i].FillWeight = 1;
         dgv.Columns[i].Width = 50;
      dgv.Columns[0].Width = 20;
      dgv.Rows[0].Cells[0].Style.BackColor = System.Drawing.Color.DarkGray;
      dgv.Rows[0].Cells[0].Value = " ";
      return dgv;
    public static DataGridView RibbonFormat(DataGridView dgv)
      for (int i = 0; i < dgv.ColumnCount; i++)
```

```
dgv.Rows[0].Cells[i].Value = i;
dgv.Rows[0].Cells[i].ReadOnly = true;
dgv.Rows[0].Cells[i].Style.BackColor = Color.Gray;
dgv.Rows[0].Cells[i].Style.SelectionBackColor = Color.Gray;
dgv.Rows[0].Cells[i].Style.SelectionForeColor = Color.Black;
dgv.Rows[1].Cells[i].ReadOnly = true;
dgv.Columns[i].FillWeight = 1;
if (dgv.ColumnCount <= 100)
  dgv.Columns[i].Width = 20;
else
```

```
ł
  return dgv;
ł
public static DataGridView toWhite(DataGridView dgv, int index)
  for (int i = index; i < dgv.Rows.Count; i++)
     for (int j = 0; j < dgv.Columns.Count; j++)
       dgv.Rows[i].Cells[j].Style.BackColor = Color.White;
  return dgv;
public static DataGridView RibbonClear(DataGridView dgv, string alphabet)
  for (int i = 0; i < dgv.ColumnCount; i++)
     dgv.Rows[1].Cells[i].Value = alphabet[0];
  return dgv;
}
public static DataGridView RibbonReadOnly(DataGridView dgv, bool value)
  for (int i = 0; i < dgv.ColumnCount; i++)
     dgv.Rows[1].Cells[i].ReadOnly = value;
  return dgv;
}
```

```
}
```

Код класу File

using System; using System.Collections.Generic; using System.IO; using System.Linq; using System.Text; using System.Threading.Tasks; using System.Windows.Forms;

```
namespace MT
```

{

class File

}

```
//Для сохранения трассы подбиваем до нужного количества символов private static string toN(int i, int N)
```

```
string s = Convert.ToString(i);
while (s.Length != N) s = "0" + s;
return s;
```

public static DataGridView openDefault(DataGridView dgv1, string path, string defaultAlphabet)

```
try
  StreamReader sr = new StreamReader(@path);
  try
    //длина и ширина
    int a = 0;
    int b = 0;
    //Считываем длину и ширину
    a = Convert.ToInt32(sr.ReadLine());
    b = Convert.ToInt32(sr.ReadLine());
    //Создаём новые размеры
    dgv1.Rows.Clear();
    dgv1.Rows.Add(a);
    dgv1.ColumnCount = b;
    string newAlphabet = "";
    for (int i = 0; i < a; i++)
    ł
       string[] mas = sr.ReadLine().Split(';');
       for (int j = 0; j < b; j++)
         //переписываем в dgv
         dgv1.Rows[i].Cells[j].Value = mas[j];
         //Если не нулевая ячейка, то считываем алфавит
```

```
if (i = 0 \&\& i != 0)
             newAlphabet += mas[j];
         }
      dgv1 = DGV.StatesFormat(dgv1);
      return dgv1;
    catch
      MessageBox.Show("Проблемы считывания стандартного алгоритма");
      dgv1 = DGV.StatesStart(dgv1, defaultAlphabet, 4);
      return dgv1;
    finally
      sr.Close();
  }
  catch
    MessageBox.Show("Стандартный алгоритм не найден");
    dgv1 = DGV.StatesStart(dgv1, defaultAlphabet, 4);
    return dgv1;
public static DataGridView open(DataGridView dgv1, string defaultAlphabet)
  //Файловый диалог
  OpenFileDialog ofd = new OpenFileDialog();
  ofd.Filter = "Algorithm of Turing machines|*.tma";
  ofd.Title = "Открыть алгоритм";
  //Открываем и ждём нажатия кнопки
  if (ofd.ShowDialog() == DialogResult.OK)
  {
    //Открываем поток с нашим файлом
    StreamReader sr = new StreamReader(ofd.FileName);
    //ошибка считывания либо синтаксиса
    bool ex = false;
    //длина и ширина
    int a = 0;
    int b = 0;
    try
    ł
      //Считываем длину и ширину(пытаемся)
      a = Convert.ToInt32(sr.ReadLine());
      b = Convert.ToInt32(sr.ReadLine());
      //Создаём новые размеры
      dgv1.Rows.Clear();
      dgv1.Rows.Add(a);
      dgv1.ColumnCount = b;
```

```
//заводим переменную для нового алфавита, который будет считан
  string newAlphabet = "";
  for (int i = 0; i < a; i++)
    //Считываем в массив строк
    string[] mas = sr.ReadLine().Split(';');
    //Если количество ; и столбцов не совпало хотя бы раз, то ех и уходим
    if (mas.Length != b + 1)
     ł
       ex = true;
       MessageBox.Show("Неверный формат файла");
      break:
    for (int j = 0; j < b; j++)
       //переписываем в dgv
      dgv1.Rows[i].Cells[j].Value = mas[j];
      //Если не нулевая ячейка, то считываем алфавит
      if (j == 0 \&\& i != 0)
         newAlphabet += mas[j];
      if (ex) break;
    if (ex) break;
  if(ex)
    dgv1 = DGV.StatesStart(dgv1, defaultAlphabet, 4);
    dgv1 = DGV.StatesFormat(dgv1);
  else
    dgv1 = DGV.StatesFormat(dgv1);
    return dgv1;
//Если вылетели по ошибке, то загружаем новый алгоритм
catch
  MessageBox.Show("Неверный формат файла");
  ex = true;
  dgv1 = DGV.StatesStart(dgv1, defaultAlphabet, 4);
  dgv1 = DGV.StatesFormat(dgv1);
  return dgv1;
//В любом случае нужно закрыть поток
finally
{ sr.Close(); }
```

64
```
return dgv1;
    }
    public static void save(DataGridView dgv1)
       SaveFileDialog sfd = new SaveFileDialog();
       sfd.Filter = "Turing machine algorithm|*.tma";
       sfd.Title = "Сохранить алгоритм как";
       //Ждём кнопку ОК
       if (sfd.ShowDialog() == DialogResult.OK)
         StreamWriter sw = new StreamWriter(sfd.FileName);
         sw.WriteLine(dgv1.RowCount);
         sw.WriteLine(dgv1.ColumnCount);
         for (int i = 0; i < dgv1.RowCount; i++)
           for (int j = 0; j < dgv1.ColumnCount; j++)
              sw.Write(dgv1.Rows[i].Cells[j].Value + ";");
           if (i != dgv1.RowCount - 1)
              sw.WriteLine("");
         sw.Close();
    public static void saveTrack(List<string> fullTrack)
       SaveFileDialog sfd = new SaveFileDialog();
       sfd.Filter = "Turing machine track|*.tmt";
       sfd.Title = "Сохранить трассу как";
       //Ждём кнопку ОК
       if (sfd.ShowDialog() == DialogResult.OK)
         StreamWriter sw = new StreamWriter(sfd.FileName);
         for (int i = 0; i < fullTrack.Count; i++)
           sw.WriteLine(toN(i + 1, Convert.ToString(fullTrack.Count).Length) + ". " +
fullTrack[i]);
         sw.Close();
```

Код класу Analizer

using System; using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks; using System.Windows.Forms;

namespace MT

```
class Analyzer
```

```
{
```

private static void error(DataGridView dgv, int a, int b, char error, char where)

//с какой dgv работаем

//позиции a,b

//e=0 вернуть обратно цвет ячейки, e=1 - синтаксическая ошибка, e=2 -

семантическая, е=3 - синтаксическая на ленте

//where = ' ' - пустой параметр, where='0,1,2' - позиция ошибки(либо её характер)

```
dgv.Rows[a].Cells[b].Style.BackColor = System.Drawing.Color.OrangeRed;
//CHAЧAЛA СВИТЧИМ ERROR, ПОТОМ WHERE
switch (error)
```

case ('1'):

switch (where)

case ('0'):

1

MessageBox.Show("Синтаксична помилка в ячейці правил переходу " + a + ";" + b + "\nПроверьте название состояния");

```
return;
}
case ('1'):
```

MessageBox.Show("Синтаксична помилка в ячейці правил переходу " + a + ";" + b + "\nСимвол не відповідає алфавіту");

return;

} case ('2'):

{

MessageBox.Show("Синтаксична помилка в ячейці правил переходу " + a + ";" + b + "\nНевірна команда переміщення");

```
return;
```

```
case ('3'):
```

66

MessageBox.Show("Синтаксична помилка в ячейці правил переходу " + a + ";" + b + "\nНевірне число в ячейці");

return;

} case ('4'):

MessageBox.Show("Синтаксична помилка в ячейці правил переходу " + а + ";" + b + "\nСтан не може бути пустим");

return;

} case (' '):

{

MessageBox.Show("Синтаксична помилка в ячейці правил переходу " + a + ";" + b + "\nНепередбачувана помилка");

return;

} break;

K

case ('2'):

}

switch (where)

case ('0'):

MessageBox.Show("Семантична помилка в ячейці правил переходу " + a + ";" + b + "\nПеревірте правильність стану");

break;

case ('1'):

}

MessageBox.Show("Семантична помилка в ячейці правил переходу " + a + ";" + b + "\nHe існує такого стану в списку станів");

break;

case ('2'):

1

MessageBox.Show("Семантична помилка в ячейці правил переходу " + a + ";" + b + "\nНема обробки початкового стану");

```
break;
```

} break;

}

orean,

case ('3'):

ł

Ł

switch (where)

case ('0'):

MessageBox.Show("Синтаксична помилка в стрічці " + b + "\nЯчейка стрічки має матив в собі 1 символ або не не мати взагалі");

```
return:
                  \left\{ \right\}
                 case ('1'):
                      MessageBox.Show("Синтаксична помилка в стрічці " + b + "\nЯчейка
має містити символи алфавіту");
                     return:
              break;
            }
         case ('4'):
              switch (where)
                 case ('0'):
                      MessageBox.Show("Помилка алфавіту в ячейці правил переходу " + а
+ ";" + b + "\nВ в ячейці має бути 1 символ алфавіту");
                     return:
                 case ('1'):
                      MessageBox.Show("Помилка алфавіту в ячейці правил переходу " + а
+ ";" + b + "\nВ алфавіті не можуть бути одинакові символи");
                      return;
                   ł
                 case ('2'):
                      MessageBox.Show("Помилка алфавіту в ячейці правил переходу " + а
+ ";" + b + "\nВ алфавіті не може бути пробілу");
                      return:
              break;
       ł
     }
    public static bool syntacticAnalysis(DataGridView dgv, string alphabet, string[] states)
       //Для состояний
       states = new string[dgv.ColumnCount];
       //Определяем
       for (int k = 1; k < states.Length; k++)
         try{ states[k] = dgv.Rows[0].Cells[k].Value.ToString(); }
         catch {dgv.Rows[0].Cells[k].Value = ""; }
       for (int i = 1; i < dgv.RowCount; i++)
         //Проверка на пустое состояние
         for (int k = 1; k < states.Length; k++)
```

if (dgv.Rows[0].Cells[k].Value.ToString() == "")

```
68
```

```
error(dgv, 0, k, '1', '4');
     return false;
for (int j = 1; j < dgv.ColumnCount; j++)
  //Если строка не пустая
  try
     if (dgv.Rows[i].Cells[j].Value.ToString().Length != 0)
       try
          string[] words = dgv.Rows[i].Cells[j].Value.ToString().Split(' ');
          //Если три слова
          if (words.Length != 3)
             error(dgv, i, j, '1', '3');
            return false;
          ł
          //Если только одна буква во втором слове и она содержится в алфавите
          if (words[1].Length != 1 || !alphabet.Contains(words[1]))
          ł
             error(dgv, i, j, '1', '1');
             return false:
          //Если только одна буква в последнем слове и если эта буква R,L,S
          if (words[2].Length != 1 || !"RLS".Contains(words[2]))
             error(dgv, i, j, '1', '1');
             return false;
          }
       }
       catch
          error(dgv, i, j, '1', ' ');
          return false;
       }
  }
```

catch

//При написании чего-ибо в ячейку, а потом удаление онного, в ячейке не "", а null. Исправляем.

```
dgv.Rows[i].Cells[j].Value = "";
```

```
return true;
```

public static bool semanticAnalysis(DataGridView dgvStates, DataGridView dgvRibbon, string alphabet, string[] states)

```
//Для состояний
       states = new string[dgvStates.ColumnCount];
       //Определяем
       for (int k = 1; k < states.Length; k++)
          states[k] = dgvStates.Rows[0].Cells[k].Value.ToString();
       for (int i = 1; i < dgvStates.RowCount; i++)
          for (int j = 1; j < dgvStates.ColumnCount; j++)
            //Если строка не пустая
            try
               if (dgvStates.Rows[i].Cells[j].Value.ToString().Length != 0)
                 try
                    if (Array.IndexOf(states,
dgvStates.Rows[i].Cells[j].Value.ToString().Substring(0,
dgvStates.Rows[i].Cells[j].Value.ToString().IndexOf(' '))) == -1)
                      error(dgvStates, i, j, '2', '1');
                      return false;
                 }
                 catch
                    error(dgvStates, i, j, '2', '0');
                    return false;
                 ł
            catch
               //При написании чего-ибо в ячейку, а потом удаление онного, в ячейке не "",
a null. Исправляем.
               dgvStates.Rows[i].Cells[j].Value = "";
       if (dgvStates.Rows[alphabet.IndexOf(dgvRibbon.Rows[1].Cells[0].Value.ToString()) +
1].Cells[1].Value.ToString() == "")
          error(dgvStates, alphabet.IndexOf(dgvRibbon.Rows[1].Cells[0].Value.ToString()) + 1,
1, '2', '2');
          return false;
       return true;
     public static bool ribbonAnalysis(DataGridView dgv, string alphabet)
       for (int i = 0; i < dgv.ColumnCount; i++)
          try
            if (dgv.Rows[1].Cells[i].Value.ToString().Length > 1)
```

```
error(dgv, 1, i, '3', '0');
               return false;
             if (!alphabet.Contains(dgv.Rows[1].Cells[i].Value.ToString()))
                error(dgv, 1, i, '3', '1');
               return false;
           Ł
          catch
             //При написании чего-ибо в ячейку, а потом удаление онного, в ячейке не "", а
null. Исправляем.
             dgv.Rows[1].Cells[i].Value = alphabet[0];
        return true;
     }
     public static bool alphabetAnalysis(DataGridView dgv, string alphabet)
        string S = "";
       for (int i = 1; i < dgv.RowCount; i++)
          S += dgv.Rows[i].Cells[0].Value;
          if (dgv.Rows[i].Cells[0].Value.ToString().Length != 1)
             error(dgv, i, 0, '4', '0');
             return false;
        }
        for (int i = 0; i < alphabet.Length; i++)
          for (int j = 0; j < alphabet.Length - i; j++)
             if (i != j)
                if (alphabet[i] == alphabet[j])
                  error(dgv, j, i, '4', '1');
                  return false;
       for (int i = 0; i < alphabet.Length; i++)
          if (alphabet[i] == ' ')
             error(dgv, i + 1, 0, '4', '2');
             return false;
       return true:
  ł
}
```

```
Додаток Г
```

Код класу TouchPadActivity

public class TouchPadActivity extends Activity {

ImageView _touchPad; TextView tvInfo; TextView _tvindex; TextView tvCoord1; TextView tvCoord2; private static final int ClickTime = 300; private static final int ClickDragTime = 400; int downx0; int downy0; int _downx1; int _downy1; int _lastx0; int _lasty0; int _lastx1; int _lasty1; int _x0; int _y0; int_x1; int _y1; boolean _motion0; boolean motion1; boolean _isClickAndDragging; boolean _isPinching; boolean _isWheeling; Date dateLastPrimaryClick; Date _dateDown0;

Date _dateDown0; Date _dateDown1; int _pixelTolerance; int _pixelToleranceSqr;

@Override

protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_board);

getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
 Point size = new Point();
 getWindowManager().getDefaultDisplay().getSize(size);
 int width = size.x;
 int height = size.y;
 _pixelTolerance = (int) ((float) (Math.min(width, height)) *
Settings.getPIXEL_TOLERANCE_PERCENTAGE());
 _pixelToleranceSqr = (int) Math.pow(_pixelTolerance, 2);
 _dateLastPrimaryClick = new Date(0);

Код класу Program

using System; using System.Collections.Generic; using System.Linq; using System.Threading.Tasks; using System.Windows.Forms;

namespace MT

ł

ł

static class Program

```
/// <summary>
/// Главная точка входа для приложения.
/// </summary>
[STAThread]
static void Main()
{
Application.EnableVisualStyles();
```

Application.EnablevistalStyles(), Application.SetCompatibleTextRenderingDefault(false); Application.Run(new FormMain());

ł

} }

Код класу Connection

```
public class Connection {
```

private static DatagramSocket _socket = null;

protected static void CreateConnection(DatagramSocket socket) {

```
_socket = socket;
```

}

```
protected synchronized static void Send(String str) {
    if (_socket != null && _socket.isConnected() && !_socket.isClosed()) {
```

try {

```
byte[] bytes = str.getBytes("UTF-8");
DatagramPacket packet = new DatagramPacket(bytes,
```

bytes.length);

}

ł

```
_socket.send(packet);
```

} catch (Exception e) {

e.printStackTrace(); Log.i("air", e.getMessage());

```
public static boolean IsConnected() {
    if (_socket == null)
        return false;
    return _socket.isConnected();
```

```
protected static void Disconnect() {
    if (_socket == null)
        return;
    if (_socket.isConnected()) {
        Send(Commands.Bye);
        // _socket.disconnect();
    }
}
```

_socket.close();

Код класу Main

namespace MT

partial class FormMain

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

private System.ComponentModel.IContainer components = null;

```
/// <summary>
```

/// Освободить все используемые ресурсы.

/// </summary>

/// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</param>

protected override void Dispose(bool disposing)

```
if (disposing && (components != null))
```

components.Dispose();

```
base.Dispose(disposing);
```

```
}
```

#region Код, автоматически созданный конструктором форм Windows

```
/// <summary>
```

```
/// Требуемый метод для поддержки конструктора — не изменяйте /// содержимое этого метода с помощью редактора кода. /// </summary>
```

private void InitializeComponent()

{

this.components = new System.ComponentModel.Container(); this.groupBoxWork = new System.Windows.Forms.GroupBox(); this.labelSpeed = new System.Windows.Forms.Label(); this.trackBarSpeed = new System.Windows.Forms.TrackBar(); this.radioButtonNow = new System.Windows.Forms.RadioButton(); this.radioButtonAuto = new System.Windows.Forms.RadioButton(); this.radioButtonStep = new System.Windows.Forms.RadioButton(); this.groupBoxOperands = new System.Windows.Forms.GroupBox(); this.buttonClearRibbon = new System.Windows.Forms.Button(); this.labelB = new System.Windows.Forms.Label(); this.labelA = new System.Windows.Forms.Label(); this.numericUpDownB = new System.Windows.Forms.NumericUpDown(); this.numericUpDownA = new System.Windows.Forms.NumericUpDown(); this.radioButtonRibbon = new System.Windows.Forms.RadioButton(); this.radioButtonForm = new System.Windows.Forms.RadioButton(); this.groupBoxTrack = new System.Windows.Forms.GroupBox();

this.buttonSaveTrack = new System.Windows.Forms.Button(); this.buttonOpenTrack = new System.Windows.Forms.Button(); this.checkBoxSaveTrack = new System.Windows.Forms.CheckBox(); this.dgvStates = new System.Windows.Forms.DataGridView(); this.dgvRibbon = new System.Windows.Forms.DataGridView(); this.buttonStart = new System.Windows.Forms.Button(); this.menuStrip1 = new System.Windows.Forms.MenuStrip(); this.anropитмToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem(); this.HOBЫйToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem(); this.orкpытьToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem(); this.orkpытьToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

System.Windows.Forms.ToolStripMenuItem();

this.стандартныeToolStripMenuItem = new

System.Windows.Forms.ToolStripMenuItem();

this.сложениеToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem(); this.нOДToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem(); this.pyководствоПользователяToolStripMenuItem = new

System.Windows.Forms.ToolStripMenuItem();

this.справкаToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem(); this.выходToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem(); this.buttonStep = new System.Windows.Forms.Button();

this.timerAuto = new System.Windows.Forms.Timer(this.components);

this.checkBoxMove = new System.Windows.Forms.CheckBox();

this.groupBoxWork.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.trackBarSpeed)).BeginInit(); this.groupBoxOperands.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.numericUpDownB)).BeginInit(); ((System.ComponentModel.ISupportInitialize)(this.numericUpDownA)).BeginInit(); this.groupBoxTrack.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.dgvStates)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.dgvRibbon)).BeginInit();

this.menuStrip1.SuspendLayout();

this.SuspendLayout();

// \

// groupBoxWork

//

this.groupBoxWork.Controls.Add(this.labelSpeed); this.groupBoxWork.Controls.Add(this.trackBarSpeed); this.groupBoxWork.Controls.Add(this.radioButtonNow); this.groupBoxWork.Controls.Add(this.radioButtonAuto); this.groupBoxWork.Controls.Add(this.radioButtonStep); this.groupBoxWork.Location = new System.Drawing.Point(19, 318); this.groupBoxWork.Margin = new System.Windows.Forms.Padding(4); this.groupBoxWork.Name = "groupBoxWork"; this.groupBoxWork.Padding = new System.Windows.Forms.Padding(4); this.groupBoxWork.Size = new System.Drawing.Size(327, 123); this.groupBoxWork.TabIndex = 0; this.groupBoxWork.TabStop = false; this.groupBoxWork.Text = "Режим роботи"; //

// labelSpeed

```
this.labelSpeed.AutoSize = true;
      this.labelSpeed.Font = new System.Drawing.Font("Microsoft Sans Serif", 7.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
      this.labelSpeed.Location = new System.Drawing.Point(168, 84);
      this.labelSpeed.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
      this.labelSpeed.Name = "labelSpeed";
      this.labelSpeed.Size = new System.Drawing.Size(129, 16);
      this.labelSpeed.TabIndex = 4;
      this.labelSpeed.Text = "15 50 100 250 500";
      // trackBarSpeed
      11
      this.trackBarSpeed.LargeChange = 1;
      this.trackBarSpeed.Location = new System.Drawing.Point(163, 49);
      this.trackBarSpeed.Margin = new System.Windows.Forms.Padding(4);
      this.trackBarSpeed.Maximum = 5;
      this.trackBarSpeed.Minimum = 1;
      this.trackBarSpeed.Name = "trackBarSpeed";
      this.trackBarSpeed.Size = new System.Drawing.Size(156, 56);
      this.trackBarSpeed.TabIndex = 3;
      this.trackBarSpeed.Value = 3;
      this.trackBarSpeed.Scroll += new System.EventHandler(this.trackBarSpeed_Scroll);
      11
      // radioButtonNow
      this.radioButtonNow.AutoSize = true;
      this.radioButtonNow.Location = new System.Drawing.Point(9, 81);
      this.radioButtonNow.Margin = new System.Windows.Forms.Padding(4);
      this.radioButtonNow.Name = "radioButtonNow";
      this.radioButtonNow.Size = new System.Drawing.Size(92, 21);
      this.radioButtonNow.TabIndex = 2;
      this.radioButtonNow.Text = "Миттєвий";
      this.radioButtonNow.UseVisualStyleBackColor = true;
      this.radioButtonNow.CheckedChanged += new
System.EventHandler(this.radioButtonNow_CheckedChanged);
      // radioButtonAuto
      11
      this.radioButtonAuto.AutoSize = true;
      this.radioButtonAuto.Checked = true;
      this.radioButtonAuto.Location = new System.Drawing.Point(9, 53);
      this.radioButtonAuto.Margin = new System.Windows.Forms.Padding(4);
      this.radioButtonAuto.Name = "radioButtonAuto";
      this.radioButtonAuto.Size = new System.Drawing.Size(124, 21);
      this.radioButtonAuto.TabIndex = 1;
      this.radioButtonAuto.TabStop = true;
      this.radioButtonAuto.Text = "Автоматичний";
      this.radioButtonAuto.UseVisualStyleBackColor = true;
      this.radioButtonAuto.CheckedChanged += new
```

System.EventHandler(this.radioButtonAuto_CheckedChanged);

// radioButtonStep

this.radioButtonStep.AutoSize = true; this.radioButtonStep.Location = new System.Drawing.Point(9, 25); this.radioButtonStep.Margin = new System.Windows.Forms.Padding(4); this.radioButtonStep.Name = "radioButtonStep"; this.radioButtonStep.Size = new System.Drawing.Size(108, 21); this.radioButtonStep.TabIndex = 0; this.radioButtonStep.Text = "Покроковий"; this.radioButtonStep.UseVisualStyleBackColor = true; this.radioButtonStep.CheckedChanged += new

System.EventHandler(this.radioButtonStep_CheckedChanged);

```
11
```

// groupBoxOperands

this.groupBoxOperands.Controls.Add(this.buttonClearRibbon); this.groupBoxOperands.Controls.Add(this.labelB); this.groupBoxOperands.Controls.Add(this.labelA); this.groupBoxOperands.Controls.Add(this.numericUpDownB); this.groupBoxOperands.Controls.Add(this.numericUpDownA); this.groupBoxOperands.Controls.Add(this.radioButtonRibbon); this.groupBoxOperands.Controls.Add(this.radioButtonForm); this.groupBoxOperands.Location = new System.Drawing.Point(353, 318); this.groupBoxOperands.Margin = new System.Windows.Forms.Padding(4); this.groupBoxOperands.Name = "groupBoxOperands"; this.groupBoxOperands.Padding = new System.Windows.Forms.Padding(4); this.groupBoxOperands.Size = new System.Drawing.Size(255, 123); this.groupBoxOperands.TabIndex = 1;this.groupBoxOperands.TabStop = false; this.groupBoxOperands.Text = "Операнди"; // buttonClearRibbon this.buttonClearRibbon.Location = new System.Drawing.Point(103, 87); this.buttonClearRibbon.Margin = new System.Windows.Forms.Padding(4); this.buttonClearRibbon.Name = "buttonClearRibbon"; this.buttonClearRibbon.Size = new System.Drawing.Size(144, 28); this.buttonClearRibbon.TabIndex = 9; this.buttonClearRibbon.Text = "Очистити стрічку"; this.buttonClearRibbon.UseVisualStyleBackColor = true; this.buttonClearRibbon.Click += new System.EventHandler(this.buttonClearRibbon_Click); 11 // labelB this.labelB.AutoSize = true; this.labelB.Location = new System.Drawing.Point(168, 53); this.labelB.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0); this.labelB.Name = "labelB"; this.labelB.Size = new System.Drawing.Size(17, 17); this.labelB.TabIndex = 8; this.labelB.Text = "B";

```
// labelA
      this.labelA.AutoSize = true;
      this.labelA.Location = new System.Drawing.Point(168, 23);
      this.labelA.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
      this.labelA.Name = "labelA";
      this.labelA.Size = new System.Drawing.Size(17, 17);
      this.labelA.TabIndex = 7;
      this.labelA.Text = "A";
      // numericUpDownB
      this.numericUpDownB.Location = new System.Drawing.Point(191, 49);
      this.numericUpDownB.Margin = new System.Windows.Forms.Padding(4);
      this.numericUpDownB.Minimum = new decimal(new int[] {
      1.
      0,
      0,
      0}):
      this.numericUpDownB.Name = "numericUpDownB";
      this.numericUpDownB.Size = new System.Drawing.Size(56, 22);
      this.numericUpDownB.TabIndex = 6;
      this.numericUpDownB.Value = new decimal(new int[] {
      1,
      0.
      0,
      0}):
      this.numericUpDownB.ValueChanged += new
System.EventHandler(this.numericUpDownB_ValueChanged);
      //
      // numericUpDownA
      this.numericUpDownA.Location = new System.Drawing.Point(191, 20);
      this.numericUpDownA.Margin = new System.Windows.Forms.Padding(4);
      this.numericUpDownA.Minimum = new decimal(new int[] {
      1,
      0,
      0.
      0};
      this.numericUpDownA.Name = "numericUpDownA";
      this.numericUpDownA.Size = new System.Drawing.Size(56, 22);
      this.numericUpDownA.TabIndex = 5;
      this.numericUpDownA.Value = new decimal(new int[] {
      1,
      0,
      0.
      0}):
      this.numericUpDownA.ValueChanged += new
System.EventHandler(this.numericUpDownA_ValueChanged);
      // radioButtonRibbon
```

```
79
```

this.radioButtonRibbon.AutoSize = true; this.radioButtonRibbon.Location = new System.Drawing.Point(8, 53); this.radioButtonRibbon.Margin = new System.Windows.Forms.Padding(4); this.radioButtonRibbon.Name = "radioButtonRibbon"; this.radioButtonRibbon.Size = new System.Drawing.Size(95, 21); this.radioButtonRibbon.TabIndex = 4; this.radioButtonRibbon.Text = "Ha crpi4ni"; this.radioButtonRibbon.UseVisualStyleBackColor = true; this.radioButtonRibbon.CheckedChanged += new

System.EventHandler(this.radioButtonRibbon_CheckedChanged);

// radioButtonForm

this.radioButtonForm.AutoSize = true; this.radioButtonForm.Checked = true; this.radioButtonForm.Location = new System.Drawing.Point(8, 25); this.radioButtonForm.Margin = new System.Windows.Forms.Padding(4); this.radioButtonForm.Name = "radioButtonForm"; this.radioButtonForm.Size = new System.Drawing.Size(90, 21); this.radioButtonForm.TabIndex = 3; this.radioButtonForm.TabStop = true; this.radioButtonForm.Text = "Ha фopMi"; this.radioButtonForm.UseVisualStyleBackColor = true; this.radioButtonForm.CheckedChanged += new

System.EventHandler(this.radioButtonForm_CheckedChanged);

// groupBoxTrack

1

this.groupBoxTrack.Controls.Add(this.buttonSaveTrack); this.groupBoxTrack.Controls.Add(this.buttonOpenTrack); this.groupBoxTrack.Controls.Add(this.checkBoxSaveTrack); this.groupBoxTrack.Location = new System.Drawing.Point(616, 318); this.groupBoxTrack.Margin = new System.Windows.Forms.Padding(4); this.groupBoxTrack.Name = "groupBoxTrack"; this.groupBoxTrack.Padding = new System.Windows.Forms.Padding(4); this.groupBoxTrack.Size = new System.Drawing.Size(219, 123); this.groupBoxTrack.TabIndex = 1; this.groupBoxTrack.TabStop = false; this.groupBoxTrack.Text = "Tpacca";

//

// buttonSaveTrack
//

this.buttonSaveTrack.Enabled = false; this.buttonSaveTrack.ForeColor = System.Drawing.SystemColors.ControlText; this.buttonSaveTrack.Location = new System.Drawing.Point(111, 53); this.buttonSaveTrack.Margin = new System.Windows.Forms.Padding(4); this.buttonSaveTrack.Name = "buttonSaveTrack"; this.buttonSaveTrack.Size = new System.Drawing.Size(100, 28); this.buttonSaveTrack.TabIndex = 2; this.buttonSaveTrack.Text = "Зберегти"; this.buttonSaveTrack.UseVisualStyleBackColor = true; this.buttonSaveTrack.Click += new System.EventHandler(this.buttonSaveTrack_Click);

// buttonOpenTrack

this.buttonOpenTrack.Enabled = false; this.buttonOpenTrack.ForeColor = System.Drawing.SystemColors.ControlText; this.buttonOpenTrack.Location = new System.Drawing.Point(8, 53); this.buttonOpenTrack.Margin = new System.Windows.Forms.Padding(4); this.buttonOpenTrack.Name = "buttonOpenTrack"; this.buttonOpenTrack.Size = new System.Drawing.Size(100, 28); this.buttonOpenTrack.TabIndex = 1; this.buttonOpenTrack.Text = "Відкрити"; this.buttonOpenTrack.UseVisualStyleBackColor = true; this.buttonOpenTrack.Click += new System.EventHandler(this.buttonOpenTrack_Click); // checkBoxSaveTrack // this.checkBoxSaveTrack.AutoSize = true; this.checkBoxSaveTrack.Checked = true; this.checkBoxSaveTrack.CheckState = System.Windows.Forms.CheckState.Checked; this.checkBoxSaveTrack.Location = new System.Drawing.Point(8, 21); this.checkBoxSaveTrack.Margin = new System.Windows.Forms.Padding(4); this.checkBoxSaveTrack.Name = "checkBoxSaveTrack"; this.checkBoxSaveTrack.Size = new System.Drawing.Size(142, 21); this.checkBoxSaveTrack.TabIndex = 0; this.checkBoxSaveTrack.Text = "Зберігати трассу"; this.checkBoxSaveTrack.UseVisualStyleBackColor = true; this.checkBoxSaveTrack.CheckedChanged += new System.EventHandler(this.checkBoxSaveTrack_CheckedChanged);

// dgvStates

11

```
11
      this.dgvStates.AllowUserToAddRows = false;
      this.dgvStates.AllowUserToDeleteRows = false;
      this.dgvStates.AllowUserToResizeColumns = false;
      this.dgvStates.AllowUserToResizeRows = false;
      this.dgvStates.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
      this.dgvStates.ColumnHeadersVisible = false;
      this.dgvStates.Location = new System.Drawing.Point(19, 123);
      this.dgvStates.Margin = new System.Windows.Forms.Padding(4);
      this.dgvStates.MultiSelect = false;
      this.dgvStates.Name = "dgvStates";
      this.dgvStates.RowHeadersVisible = false;
      this.dgvStates.RowHeadersWidth = 51;
      this.dgvStates.Size = new System.Drawing.Size(816, 188);
      this.dgvStates.TabIndex = 2;
      this.dgvStates.MouseClick += new
System.Windows.Forms.MouseEventHandler(this.dgvStates_MouseClick);
      11
      // dgvRibbon
      this.dgvRibbon.AllowUserToAddRows = false;
```

this.dgvRibbon.AllowUserToDeleteRows = false; this.dgvRibbon.AllowUserToResizeColumns = false; this.dgvRibbon.AllowUserToResizeRows = false; this.dgvRibbon.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize; this.dgvRibbon.ColumnHeadersVisible = false; this.dgvRibbon.Location = new System.Drawing.Point(19, 36); this.dgvRibbon.Margin = new System.Windows.Forms.Padding(4); this.dgvRibbon.MultiSelect = false; this.dgvRibbon.Name = "dgvRibbon"; this.dgvRibbon.RowHeadersVisible = false; this.dgvRibbon.RowHeadersVisible = false; this.dgvRibbon.RowHeadersWidth = 51; this.dgvRibbon.Size = new System.Drawing.Size(816, 80); this.dgvRibbon.TabIndex = 3; this.dgvRibbon.MouseClick += new

System.Windows.Forms.MouseEventHandler(this.dgvRibbon_MouseClick);

// buttonStart

11

this.buttonStart.ForeColor = System.Drawing.SystemColors.ControlText; this.buttonStart.Location = new System.Drawing.Point(725, 448); this.buttonStart.Margin = new System.Windows.Forms.Padding(4); this.buttonStart.Name = "buttonStart"; this.buttonStart.Size = new System.Drawing.Size(109, 28); this.buttonStart.TabIndex = 2; this.buttonStart.Text = "CTapt"; this.buttonStart.UseVisualStyleBackColor = true; this.buttonStart.Click += new System.EventHandler(this.buttonStart_Click); //

// menuStrip1

 \parallel

this.menuStrip1.ImageScalingSize = new System.Drawing.Size(20, 20); this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] { this.aлгоритмToolStripMenuItem, this.pyкoBoдстBoПoльзoBateляToolStripMenuItem, this.cпpaBkaToolStripMenuItem, this.BыxoдToolStripMenuItem}); this.menuStrip1.Location = new System.Drawing.Point(0, 0); this.menuStrip1.Name = "menuStrip1"; this.menuStrip1.Size = new System.Drawing.Size(855, 30); this.menuStrip1.TabIndex = 10; this.menuStrip1.Text = "menuStrip1";

// алгоритмToolStripMenuItem

this.aлгоритмToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.новыйToolStripMenuItem, this.открытьToolStripMenuItem,

this.coxpaнитьКакToolStripMenuItem,

this.стандартныeToolStripMenuItem});

this.aлгоритмToolStripMenuItem.Name = "алгоритмToolStripMenuItem";

this.алгоритмToolStripMenuItem.Size = new System.Drawing.Size(91, 26); this.aлгоритмToolStripMenuItem.Text = "Алгоритм";

// новыйToolStripMenuItem

this.новыйToolStripMenuItem.Name = "новыйToolStripMenuItem"; this.новыйToolStripMenuItem.Size = new System.Drawing.Size(174, 26); this.новыйToolStripMenuItem.Text = "Новий"; this.новыйToolStripMenuItem.Click += new

System.EventHandler(this.новыйToolStripMenuItem_Click);

// открытьToolStripMenuItem

this.oткрытьToolStripMenuItem.Name = "открытьToolStripMenuItem"; this.oткрытьToolStripMenuItem.Size = new System.Drawing.Size(174, 26); this.oткрытьToolStripMenuItem.Text = "Відкрити"; this.oткрытьToolStripMenuItem.Click += new

System.EventHandler(this.загрузитьАлгоритмToolStripMenuItem_Click);

// сохранитьКакToolStripMenuItem

1

11

11

this.coxpaнитьКакToolStripMenuItem.Name = "coxpaнитьКакToolStripMenuItem"; this.coxpaнитьКакToolStripMenuItem.Size = new System.Drawing.Size(174, 26); this.coxpaнитьКакToolStripMenuItem.Text = "Зберегти як";

this.coxpaнитьКакToolStripMenuItem.Click += new

System.EventHandler(this.coxранитьАлгоритмКакToolStripMenuItem_Click);

// стандартныеToolStripMenuItem

this.стандартныeToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.сложениеToolStripMenuItem,

this.нOДToolStripMenuItem});

this.стандартныeToolStripMenuItem.Name = "стандартныeToolStripMenuItem"; this.стандартныeToolStripMenuItem.Size = new System.Drawing.Size(174, 26); this.стандартныeToolStripMenuItem.Text = "Стандартні";

// сложениеToolStripMenuItem

//

//

```
this.сложениеToolStripMenuItem.Name = "сложениеToolStripMenuItem";
this.сложениеToolStripMenuItem.Size = new System.Drawing.Size(169, 26);
this.сложениеToolStripMenuItem.Text = "Додавання";
this.сложениеToolStripMenuItem.Click += new
```

System.EventHandler(this.сложениеToolStripMenuItem_Click);

// нОДToolStripMenuItem

this.нОДТооlStripMenuItem.ForeColor = System.Drawing.Color.Black; this.нОДТоolStripMenuItem.Name = "нОДТоolStripMenuItem"; this.нОДToolStripMenuItem.Size = new System.Drawing.Size(169, 26); this.нОДToolStripMenuItem.Text = "Множення"; this.нOДToolStripMenuItem.Click += new System.EventHandler(this.нOДToolStripMenuItem_Click);

// руководствоПользователяToolStripMenuItem

this.pyководствоПользователяToolStripMenuItem.ForeColor = System.Drawing.Color.Black;

this.pyководствоПользователяToolStripMenuItem.Name = "pyководствоПользователяToolStripMenuItem";

this.pyководствоПользователяToolStripMenuItem.Size = new System.Drawing.Size(177, 26);

this.pyководствоПользователяToolStripMenuItem.Text = "Довідник користувача"; this.pyководствоПользователяToolStripMenuItem.Click += new

System.EventHandler(this.pyководствоПользователяToolStripMenuItem_Click);

// справкаToolStripMenuItem

this.справкаToolStripMenuItem.Name = "справкаToolStripMenuItem"; this.справкаToolStripMenuItem.Size = new System.Drawing.Size(77, 26); this.справкаToolStripMenuItem.Text = "Довідка"; this.справкаToolStripMenuItem.Click += new

System.EventHandler(this.справкаToolStripMenuItem Click);

// выходToolStripMenuItem

//

11

11

11

this.выходToolStripMenuItem.Name = "выходToolStripMenuItem"; this.выходToolStripMenuItem.Size = new System.Drawing.Size(60, 26); this.выходToolStripMenuItem.Text = "Вихід"; this.выходToolStripMenuItem.Click += new

System.EventHandler(this.выходToolStripMenuItem_Click);

// buttonStep

```
11
this.buttonStep.Enabled = false;
this.buttonStep.Location = new System.Drawing.Point(616, 448);
this.buttonStep.Margin = new System.Windows.Forms.Padding(4);
this.buttonStep.Name = "buttonStep";
this.buttonStep.Size = new System.Drawing.Size(100, 28);
this.buttonStep.TabIndex = 11;
this.buttonStep.Text = "Kpok";
this.buttonStep.UseVisualStyleBackColor = true;
this.buttonStep.Click += new System.EventHandler(this.buttonStep_Click);
//
// timerAuto
11
this.timerAuto.Tick += new System.EventHandler(this.timerAuto_Tick);
//
// checkBoxMove
this.checkBoxMove.AutoSize = true;
this.checkBoxMove.Location = new System.Drawing.Point(19, 444);
this.checkBoxMove.Margin = new System.Windows.Forms.Padding(4);
```

this.checkBoxMove.Name = "checkBoxMove"; this.checkBoxMove.Size = new System.Drawing.Size(126, 21); this.checkBoxMove.TabIndex = 12; this.checkBoxMove.Text = "Pyxaти стрічку"; this.checkBoxMove.UseVisualStyleBackColor = true; //

// FormMain

11

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F); this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font; this.ClientSize = new System.Drawing.Size(855, 480); this.Controls.Add(this.checkBoxMove); this.Controls.Add(this.buttonStep); this.Controls.Add(this.dgvStates); this.Controls.Add(this.buttonStart); this.Controls.Add(this.dgvRibbon); this.Controls.Add(this.groupBoxTrack); this.Controls.Add(this.groupBoxOperands); this.Controls.Add(this.groupBoxWork); this.Controls.Add(this.menuStrip1); this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle; this.MainMenuStrip = this.menuStrip1; this.Margin = new System.Windows.Forms.Padding(4); this.MaximizeBox = false; this.Name = "FormMain"; this.Text = "Машина Тюрінга"; this.groupBoxWork.ResumeLayout(false); this.groupBoxWork.PerformLayout(); ((System.ComponentModel.ISupportInitialize)(this.trackBarSpeed)).EndInit(); this.groupBoxOperands.ResumeLayout(false); this.groupBoxOperands.PerformLayout(); ((System.ComponentModel.ISupportInitialize)(this.numericUpDownB)).EndInit(); ((System.ComponentModel.ISupportInitialize)(this.numericUpDownA)).EndInit(); this.groupBoxTrack.ResumeLayout(false); this.groupBoxTrack.PerformLayout(); ((System.ComponentModel.ISupportInitialize)(this.dgvStates)).EndInit(); ((System.ComponentModel.ISupportInitialize)(this.dgvRibbon)).EndInit(); this.menuStrip1.ResumeLayout(false); this.menuStrip1.PerformLayout(); this.ResumeLayout(false); this.PerformLayout();

.

#endregion

private System.Windows.Forms.GroupBox groupBoxWork; private System.Windows.Forms.RadioButton radioButtonNow; private System.Windows.Forms.RadioButton radioButtonAuto; private System.Windows.Forms.RadioButton radioButtonStep; private System.Windows.Forms.GroupBox groupBoxOperands; private System.Windows.Forms.RadioButton radioButtonRibbon;

private System.Windows.Forms.RadioButton radioButtonForm; private System.Windows.Forms.GroupBox groupBoxTrack; private System.Windows.Forms.CheckBox checkBoxSaveTrack; private System.Windows.Forms.NumericUpDown numericUpDownB; private System.Windows.Forms.NumericUpDown numericUpDownA; private System.Windows.Forms.TrackBar trackBarSpeed; private System.Windows.Forms.Button buttonOpenTrack; private System.Windows.Forms.Label labelB; private System.Windows.Forms.Label labelA; private System.Windows.Forms.Button buttonStart; private System.Windows.Forms.Label labelSpeed; private System. Windows. Forms. MenuStrip menuStrip1; private System.Windows.Forms.ToolStripMenuItem алгоритмToolStripMenuItem; private System.Windows.Forms.ToolStripMenuItem новыйToolStripMenuItem; private System.Windows.Forms.ToolStripMenuItem открытьToolStripMenuItem; private System.Windows.Forms.ToolStripMenuItem сохранитьКакToolStripMenuItem; private System.Windows.Forms.ToolStripMenuItem руководствоПользователяToolStripMenuItem; private System.Windows.Forms.ToolStripMenuItem справкаToolStripMenuItem; private System. Windows. Forms. ToolStripMenuItem выходToolStripMenuItem; private System.Windows.Forms.ToolStripMenuItem стандартныeToolStripMenuItem; private System.Windows.Forms.ToolStripMenuItem сложениеToolStripMenuItem; private System.Windows.Forms.ToolStripMenuItem HOДToolStripMenuItem; private System. Windows. Forms. Button buttonStep; private System.Windows.Forms.DataGridView dgvRibbon; public System.Windows.Forms.DataGridView dgvStates; private System.Windows.Forms.Timer timerAuto;

private System.Windows.Forms.Button buttonSaveTrack;

private System.Windows.Forms.CheckBox checkBoxMove;

private System.Windows.Forms.Button buttonClearRibbon;

5