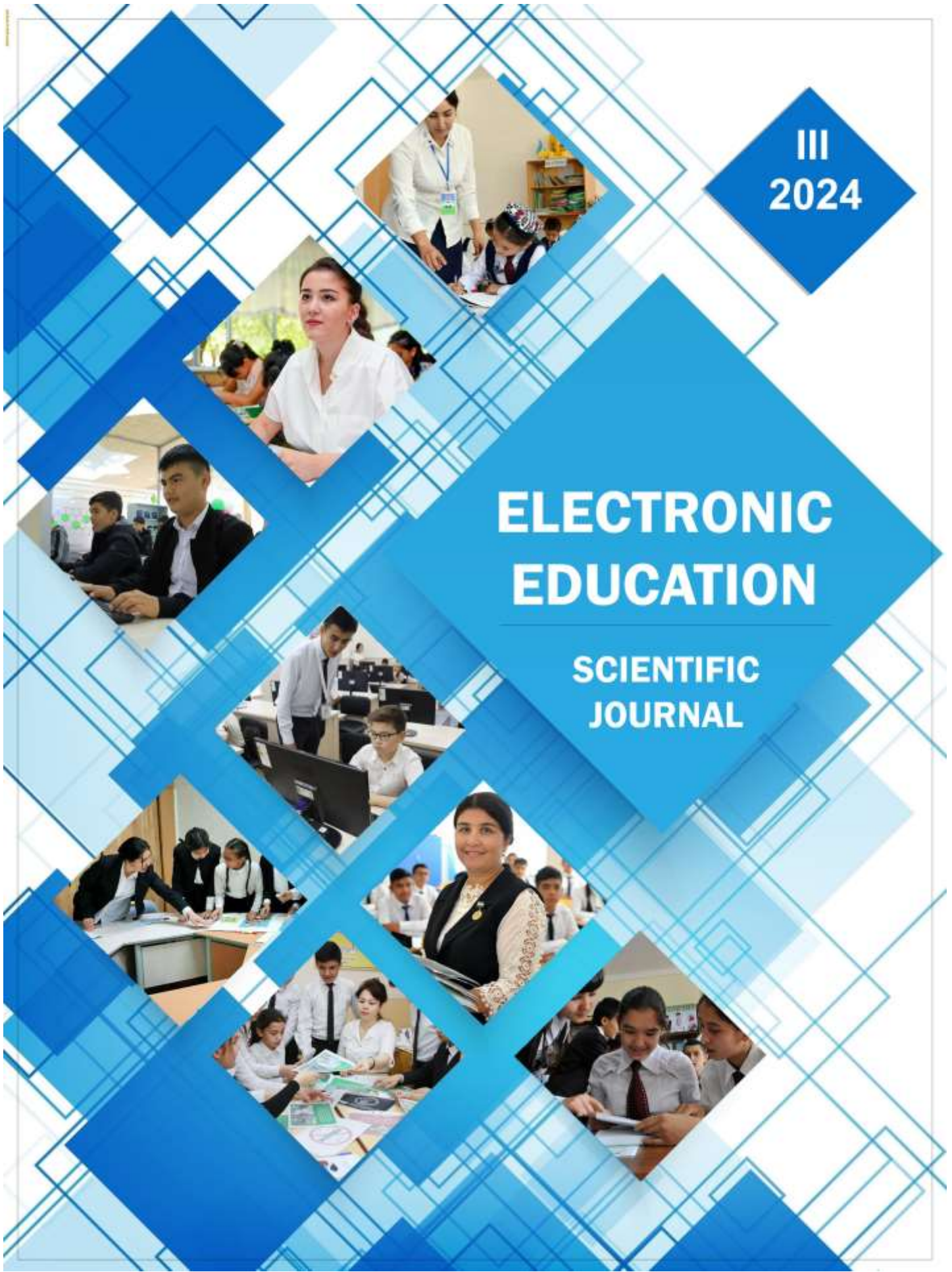


III
2024

ELECTRONIC EDUCATION

SCIENTIFIC
JOURNAL



TAHRIRIYAT

Bosh muharrir

Laqayev Saidaxmad Norjigitovich
fizika-matematika fanlari doktori, akademik

Bosh muharrir o‘rinbosari

Ro‘ziyev Rauf Axmadovich
fizika-matematika fanlari nomzodi, dotsent

Mas’ul muharrir

Mirsanov Uralboy Mukhammadiyevich
pedagogika fanlari doktori DSc, dotsent

Editor-in-Chief

Saidaxmad Norjigitovich Lakayev
doctor of physical and mathematical sciences,
academician

Deputy Editor-in-Chief

Ruziyev Raup Akhmadovich
Candidate of Physical and Mathematical Sciences,
Associate Professor

Responsible editor

Mirsanov Uralboy Mukhammadiyevich
doctor of Pedagogical Sciences DSc, Associate
Professor

TAHRIRIYAT A’ZOLARI

Sobirov Baxodir Boypulatovich – NavDPI rektori,
texnika fanlari doktori, professor (O‘zbekiston)

Djurayev Risbay Xaydarovich – akademik
(O‘zbekiston)

Shokin Yuriy Ivanovich – akademik (Rossiya)

Negmatov Sayibjon Sodiqovich – akademik
(O‘zbekiston)

Aripov Mersaid Mirsiddikovich – fizika-matematika
fanlari doktori, professor (O‘zbekiston)

Turabdjano Sadriddin Maxamatdinovich – texnika
fanlari doktori, akademik (O‘zbekiston)

Raximov Isomiddin Sattarovich – fizika-matematika
fanlari doktori, professor (Malayziya)

Shariy Sergey Petrovich – fizika-matematika fanlari
doktori, professor (Rossiya).

Ajimuxammedov Iskandar Maratovich – texnika
fanlari doktori, professor (Rossiya).

Ibraimov Xolboy – pedagogika fanlari doktori,
akademik (O‘zbekiston)

Yunusova Dilfuza Isroilovna – pedagogika fanlari
doktori, professor (O‘zbekiston)

Aloyev Raxmatillo Djurayevich – fizika-matematika
fanlari doktori, professor (O‘zbekiston)

Abdullayeva Shaxzoda Abdullayevna – pedagogika
fanlari doktori, professor (O‘zbekiston)

Mo‘minov Bahodir Boltayevich – texnika fanlari
doktori, professor (O‘zbekiston)

Korshunov Igor Lvovich – texnika fanlari nomzodi,
dotsent. (Rossiya)

Kolbanyov Mixail Olegovich – texnika fanlari
doktori, professor. (Rossiya)

Verzun Natalya Arkadyevna – texnika fanlari
nomzodi, dotsent. (Rossiya)

Maxmudova Dilfuza Mileyevna – pedagogika
fanlari doktori, professor (O‘zbekiston)

Xudjayev Muxiddin Kushshayevich – texnika
fanlari doktori, dotsent (O‘zbekiston).

Xolmurodov Abdulhamid Erkinovich – fizika-
matematika fanlari doktori, professor (O‘zbekiston)

Stelmashonok Yelena Viktorovna – iqtisod fanlari
doktori, professor. (Rossiya)

Tatarnikova Tatyana Mixaylovna – texnika fanlari
doktori, professor. (Rossiya)

Alekseyev Vladimir Vasilyevich – texnika fanlari
doktori, professor. (Rossiya)

Satikov Igor Abuzarovich – fizika-matematika fanlari
nomzodi, dotsent. (Belarus)

Boyarshinova Oksana Aleksandrovna – fizika-
matematika fanlari nomzodi, dotsent. (Belarus)

Makarenya Sergey Nikolayevich – texnika fanlari
nomzodi, dotsent. (Belarus)

Sednina Marina Aleksandrovna – texnika fanlari
nomzodi, dotsent. (Belarus)

Ibragimov Abdusattar Turgunovich – texnika fanlari
doktori, dotsent (O‘zbekiston)

Kabiljanova Firusa Azimovna – fizika-matematika
fanlari nomzodi, dotsent. (O‘zbekiston)

Lutfillayev Maxmud Xasanovich – pedagogika fanlari
doktori, dotsent (O‘zbekiston).

Ergasheva Gulruksor Surxonidinovna – pedagogika
fanlari doktori (DSc), dotsent (O‘zbekiston).

Norov Abdusait Muradovich – texnika fanlari
bo‘yicha falsafa doktori, dotsent (O‘zbekiston).

Yuldoshev Ismoil Abriyevich – pedagogika fanlari
bo‘yicha falsafa doktori, dotsent (O‘zbekiston).

Karaxonova Oysara Yuldoshevna – pedagogika fanlari
bo‘yicha falsafa doktori, (O‘zbekiston).

Kurbaniyazova Zamira Kalbaevna – pedagogika
fanlari doktori, dotsent (O‘zbekiston).

Nasirova Shaira Narmuradovna – texnika fanlari
doktori, professor (O‘zbekiston).

Nasridinov Ilxam Burxanidinovich – texnika fanlari
nomzodi, dotsent (O‘zbekiston).

Xujjiyev Sodiq Oltiyevich – biologiya fanlari nomzodi,
dotsent (O‘zbekiston).

Suvonov Olim Omonovich – texnika fanlari nomzodi,
dotsent (O‘zbekiston).

O‘tapon Toyir Usmonovich – pedagogika fanlari nomzodi, dotsent (O‘zbekiston).

Ibragimov Alimjon Artikbayevich – fizika-matematika fanlari nomzodi, dotsent (O‘zbekiston).

Yodgorov G‘ayrat Ro‘ziyevich – fizika-matematika fanlari nomzodi, dotsent (O‘zbekiston).

Xudoyorov Shuxrat Jumaqulovich – fizika-matematika fanlari nomzodi, dotsent (O‘zbekiston)

Baxodirova Umida Baxodirovna – pedagogika fanlari bo‘yicha falsafa doktori (O‘zbekiston).

Toxirov Feruz Jamoliddinovich – pedagogika fanlari bo‘yicha falsafa doktori (O‘zbekiston).

Xamroyeva Dilafro‘z Namozovna – fizika-matematika fanlari bo‘yicha falsafa doktori (O‘zbekiston).

Jo‘rakulov Tolib Toxirovich – texnik muharrir

© Mazkur jurnal O‘zbekiston Respublikasi Vazirlar Mahkamasi huzuridagi Oliy Attestatsiya komissiyasi rayosatining 2022-yil 28-fevraldagi 312/6 qaroriga asosan Pedagogika fanlari bo‘yicha falsafa doktori (PhD) va fan doktori (DSc) ilmiy darajasiga talabgorlarning dissertatsiya ishlari yuzasidan dissertatsiyalari asosiy ilmiy natijalarini chop etish uchun tavsiya etilgan ilmiy nashrlar ro‘yxatiga kiritilgan

Adress: Navoiy sh., Janubiy ko‘chasi, 1-A uy. (1-A, South Street, Navoi city) URL: <http://www.el-nspi.uz>

MUNDARIJA

Aniq fanlarda axborot texnologiyalari

Nekboyev X. X. <i>MEDIATA’LIMNING PEDAGOGIK - PSIXOLOGIK XUSUSIYATLARI</i>	6
Otaxanov N. A. <i>C# DASTURLASH TILIDA GRAFIKLAR BILAN ISHLASH</i>	15
Xo’jamberdiyeva J. N., Xoliyorova E. <i>FIZIKADAN RAQAMLI O’QUV MATERIALLARI ASOSIDA O’QUVCHILARNI TADQIQOTCHILIK FAOLIYATLARINI RIVOJLANTIRISH METODLARI</i>	35
Djumabayev K. N. <i>DASTURLASH TILLARINI O’QITISH SAMARADORLIGINI OSHIRISHDA RAQAMLI TEXNOLOGIYALARNING DIDAKTIK IMKONIYATLARI</i>	42
Mirsanov U. M. <i>DASTURLASH TILLARINI O’QITISH SAMARADORLIGINI OSHIRISHDA WEB-KVEST TA’LIM TEXNOLOGIYASINING IMKONIYATI</i>	51

Tabiiy fanlarda axborot texnologiyalari

Ahtamov Abror Anvar o’g’li <i>BIOLOGIYA DARSLARIDA «BLENDED LEARNING» VA «FLIPPED CLASS» TEXNOLOGIYALARIDAN FOYDALANISHNING METODIK ASOSLARI</i>	60
--	----

СОДЕРЖАНИЕ

Информационные технологии в точных науках

Некбоев X. X. <i>ПЕДАГОГО-ПСИХОЛОГИЧЕСКАЯ ХАРАКТЕРИСТИКА МЕДИАОБРАЗОВАНИЯ</i>	6
Отаханов Н. А. <i>РАБОТА С ГРАФИКАМИ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ C#</i>	15
Ходжамбердиева Дж.Н., Холиерова Э. <i>МЕТОДЫ РАЗВИТИЯ НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ СТУДЕНТОВ НА ОСНОВЕ ЦИФРОВЫХ УЧЕБНЫХ МАТЕРИАЛОВ ПО ФИЗИКЕ</i>	35
Джумбабаев К. Н. <i>ДИДАКТИЧЕСКИЕ ВОЗМОЖНОСТИ ЦИФРОВЫХ ТЕХНОЛОГИЙ В ПОВЫШЕНИИ ЭФФЕКТИВНОСТИ ПРОГРАММИРОВАНИЯ ОБУЧЕНИЯ ЯЗЫКАМ</i>	42
Мирсанов У. М. <i>ВОЗМОЖНОСТИ ОБРАЗОВАТЕЛЬНОЙ ТЕХНОЛОГИИ WEB-QUEST В ПОВЫШЕНИИ</i>	

ЭФФЕКТИВНОСТИ ОБУЧЕНИЯ ЯЗЫКУ ПРОГРАММИРОВАНИЯ 51

Информационные технологии в естественных науках

Ахтамов А.А.
МЕТОДИЧЕСКИЕ ПРИНЦИПЫ ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИЙ СМЕШАННОГО
ОБУЧЕНИЯ И ПЕРЕВЕРНУТЫХ КЛАССОВ НА УРОКАХ БИОЛОГИИ 60

CONTENT

Information technologies in exact sciences

Nekboyev Khurshid
PEDAGOGICAL - PSYCHOLOGICAL CHARACTERISTICS OF MEDIA EDUCATION 6

Otaxanov Nurillo
WORKING WITH GRAPHS IN THE C# PROGRAMMING LANGUAGE 15

Khodjamberdieva Jamilya, Holierova Ezoza
METHODS OF DEVELOPING STUDENTS' RESEARCH ACTIVITIES BASED ON DIGITAL
TEACHING MATERIALS IN PHYSICS 35

Dzhumbabaev Kuanishbai
DIDACTIC POSSIBILITIES OF DIGITAL TECHNOLOGIES IN IMPROVING THE
EFFICIENCY OF LANGUAGE TEACHING PROGRAMMING 15

Mirsanov Uralbay
POSSIBILITIES OF EDUCATIONAL TECHNOLOGY WEB-QUEST IN INCREASING THE
EFFICIENCY OF LEARNING PROGRAMMING LANGUAGE 51

Information technologies in natural sciences

Akhtamov Abror
DIDACTIC POSSIBILITIES OF DIGITAL TEXTBOOK IN THE EDUCATIONAL PROCESS 60

Aniq fanlarda axborot texnologiyalari

C# DASTURLASH TILIDA GRAFIKLAR BILAN ISHLASH

Otaxanov Nurillo Abdumalikovich
Namangan davlat universiteti, professor

Annotatsiya: Ushbu maqolada C# dasturlash tilining grafiklar bilan ishlash imkoniyatlari ochib berilgan. bayon etilgan. Ma’lumotlar tizimli ketma-ketlikda, o‘quvchilarga tushunarli va sodda tilda bayon etilgan. Unda ekranni grafik rejimga o‘tkazish, sodda geometrik shakllarni yasash, figuralarga fon berish hamda nuqtalar orqali funktsiya grafigini qurish bo‘yicha nazariy ma’lumotlar va ulardan amaliyotda foydalanish namunalari keng o‘rin berilgan.

Tayanch so‘zlar: tasvir, grafika, usul, kod, xolst, chiziq, xususiyat, rang.

РАБОТА С ГРАФИКАМИ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ C#

Otaxanov Нурилло Абдумаликович
Наманганский государственный университет, профессор

Аннотация: В данной статье раскрываются графические возможности языка программирования C#. Информация изложена в систематической последовательности понятным и простым для читателей языком. В нем широко представлены теоретические данные по переводу экрана в графический режим, созданию простых геометрических фигур, заданию фона фигурам и построению графика функции с помощью точек, а также конкретные примеры их использования на практике.

Ключевые слова: изображение, график, метод, код, холст, линия, свойство, цвет..

WORKING WITH GRAPHS IN THE C# PROGRAMMING LANGUAGE

Otaxanov Nurillo
Namangan State University, professor

Abstract: This article reveals the graphical capabilities of the C# programming language. The information is presented in a systematic sequence in a language that is understandable and easy for readers. It provides extensive theoretical data on converting the screen to graphical mode, creating simple geometric shapes, setting the background to shapes and plotting functions using dots, as well as specific examples of their use in practice.

Key words: image, graph, method, code, canvas, line, property, color

Kirish. Ochiq manbalarga ko‘ra, XX asrning ikkinchi yarmidan boshlab 9000 ga yaqin dasturlash tillari ishlab chiqilgan bo‘lib, ularning ellikka yaqini zamonaviy dasturlash amaliyotida qo‘llaniladi. Amalda qo‘llashning qiyinligi, imkoniyatlarining cheklanganligi, shuningdek, zamonaviy talablarga mos kelmasligi

sababli ko‘plab dasturlash tillari dasturchilar orasida o‘z muxlislarini topa olmadi. Dasturlash tillarining mashhurlik reytingining ko‘tarilmasligining yana bir sababi ularni o‘rganishning murakkabligi bilan bog‘liq. Tabiiyki, o‘rganishdagi muammolar o‘quv jarayoni samarasiga ham salbiy ta’sir etadi.

Raqobatchi kompaniyalar tomonidan zamonaviy talablar asosida dasturchilar o‘rtasida mashhur bo‘lgan dasturlash tillarini takomillashtirish va imkoniyatlarini kengaytirish yuzasidan usluksiz ish olib borilmoqda. Natijada, bunday tillarning yangi versiyalari ishlab chiqilmoqda. Ularning yosh yoki tajribali dasturchilar orasida keng tarqalishida tegishli o‘quv adabiyotlarining mavjudligi muhim ahamiyat kasb etadi. Ma'lumki, dasturlash tillari bo‘yicha o‘quv adabiyotlari ikki toifa olimlar tomonidan yaratiladi: yetakchi dasturchilar va metodist o‘qituvchilar.

Yetakchi dasturchilar tomonidan yozilgan kitoblarda dasturlash tilining barcha imkoniyatlarini ochib berishga qaratilgan katta hajmli ma’lumotlar bayon qilinadi. Bunga misol qilib [3-6] adabiyotlarni shu toifaga kiritish mumkin. Uslubchi olimlar esa o‘z kitoblarida har bir mavzuni o‘rganish bo‘yicha zarur va yetarli bo‘lgan minimal hajmdagi ma’lumotlarni uslubiy jihatdan ishlab chiqqan holda taqdim etadilar [1]. Zamonaviy dasturlash tillari (ayniqsa, C#) shu qadar kengaydiki, ular haqida batafsil ma’lumotni bitta kitob doirasida qamrab olish mushkul vazifaha aylandi. Shu munosabat bilan N. Virt shunday deb yozadi: “Dasturlash tillari juda murakkab bo‘lib qoldi, chunki ularning yangi mo‘jizakor va ajoyib imkoniyatlari haqiqatdan ham zarur bo‘lgani uchun emas, balki shunday imkoniyatlarni taqdim etish mumkin bo‘lgani uchun yaratishga harakat qilindi. Buning evaziga bir raqobatchi boshqasidan ustunlikka erishishga umid qilishadi. Haddan tashqari avzalliklarni taqdim etish ortiqcha noqulaylik, foydalanishdagi murakkablik va ishonchlilikning pasayishiga olib keladi [2].

1969 yilda Ci bilan o'z tarixini boshlagan zamonaviy va qudratli C# dasturlash tili birinchi marta 2000 yilda Microsoft tomonidan yangi dasturlash tili sifatida taqdim etilgan. So‘nggi yillarda bu til imkoniyatlari spektrining beqiyos kengayishi tufayli dasturlash tillari reytingining kuchli beshligiga kira oldi. Bizning fikrimizcha,

dastur sohalarining boy imkoniyatlari va kengligi va ma’lumotlarni taqdim etish shakllarining xilma-xilligi C# dasturlash tilini o’rganish jarayoniga salbiy ta’sir ko’rsatmoqda. Ushbu muammoni hal qilish C#tilining imkoniyatlarini tavsiflashga qaratilgan o’quv materiallarini uslubiy jihatdan qayta ko’rib chiqishni talab qiladi. Quyida ushbu muammo C# dasturlash tilining grafik imkoniyatlarini ochib berish misolida hal qilinadi.

Adabiyotlar tahlili. Niklaus Virt 2002 yilda Aarxushda bo’lib o’tgan ITiCSE informatika fanini o’qitishga bag’ishlangan xalqaro konferentsiyada dasturlash tillaridagi murakkabliklar to’g’risida so’zlagan nutqida ko’plab dasturlash tillarining “mo’jizaviy” imkoniyatlarini reklamadan boshqa narsa emas deb ta’kidlaydi va “nokerak va ortiqcha murakkabliklar ko’plab muammolarni keltirib chiqaradi. Odamlar murakkabliklarni maftunkor deb o’ylab xato qiladilar" deb yozadi [2].

C# dasturlash tili imkoniyatlarini to’la tavsiflash olimlardan katta mehnat talab qiladi. Shunda ham bu imkoniyatlarni to’la qamrab olish mushkul fmasala. Masalan, balandparvoz nomlariga qaramay, G. Shildtning “C# 4: To’liq yo’riqnoma” [6] va X. Deytelning "C#: Eng to’liq ma’lumot” kitoblarida [3] C# ning tasvirlar bilan ishlash imkoniyatlariga doir o’quv ma’lumotlari umuman uchramaydi. N.E. Flenovning “C # ning Injil” [5] kitobida esa C# ning grafik imkoniyatlarini ochish uchun 11 sahifa ajratilgan va faqat elementar chiziqqlar va raster grafik elementlari haqidagi asosiy tushunchalar bilan cheklangan.

“C# va.NET. platformasi: dasturchi kutubxonasi” kitobida E. Troelsen C# tilining grafik vositalari bilan ishlash haqida qisman to’liq ma’lumotlarni bayon qilgan U *GDI* va *System.Drawing* nomlar fazosini, *Point*, *Rectangle*, *Size*, *Region* sinflari, ranglar va shriftlar hamda tayyor tasvirlar bilan ishlash jarayonini ochib berichga harakat qildi. Biroq, bu kitobda ma’lumotlarni bayon qilishga hos bo’lgan “oddiydan murakkabga” va “ma’lumdan noma’lumga” didaktika printsiplariga rioya etilmagan. Bu holat taqdim etilayotgan o’quv materiallarni o’zlashtirish jarayonini murakkablashtiradi.

Shunga o‘xshash adabiyotlarning tahlili shuni ko‘rsatdiki, “oddiydan murakkabga” va “ma’lumdan noma’lumga” didaktik printsiplar ko‘plab mualliflarning e’tiboridan chetda qolmoqda. Taklif qilinayotgan o‘quv materiallarning kattagina hajmi “ma’naviy eskirgan” va shu sababli bunday adabiyotlardan talabalarni zamonaviy dasturlash tillarni o‘qitish jarayonida foydalanish yuqori samaradorlikni ta’minlay olmaydi. Shu sababli, o‘zlashtirish uchun kerak bo‘lgan yetarli va minimal hajmdagi o‘quv materiallarini taqdim etish masalasini dasturlash tillarining alohida mavzu yoki bo‘limlari (masalan, C# ning grafik imkoniyatlari) bo‘yicha hususiy takliflar asosida tashkil qilish maqsadga muvofiq bo‘ladi.

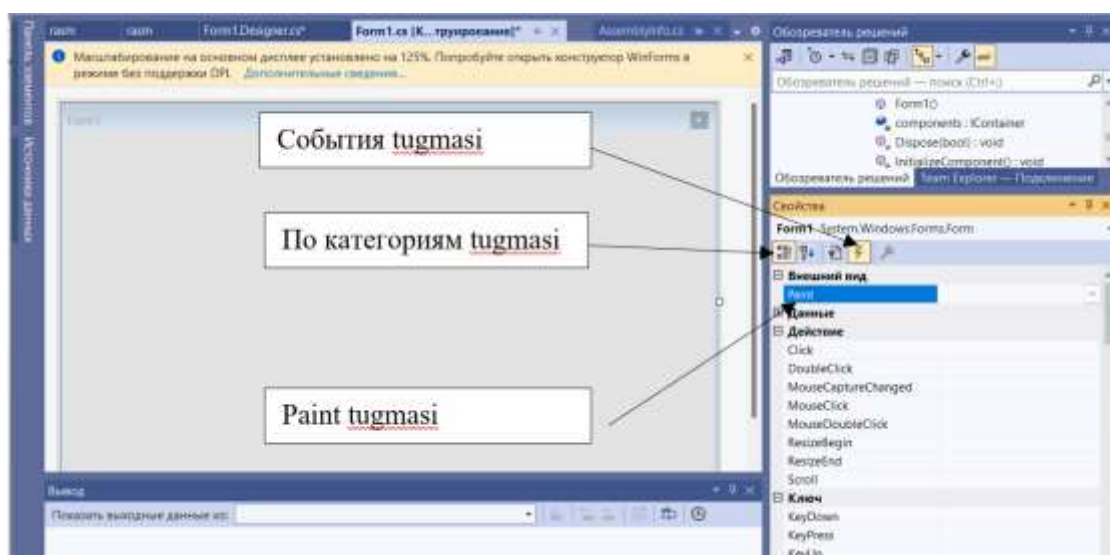
Tadqiqot metodologiyasi. C# tili dasturchilarga tasvirlar bilan ishlash uchun juda katta imkoniyatlarni taqdim etadi. Ushbu maqolada GDI + yordamida tasvirlar bilan ishlash jarayonini dasturlash asoslari bo‘yicha boshlang‘ich ma’lumotlar bayon qilinadi. Asosiy e’tibor *Graphics*, *Pen*, *Brush* va *Color* obyektlari, tasvirlar chizish, matnlar bilan ishlash, rasmlarni ko‘rsatish kabi amaliyotda keng qo‘llanadigan muammolarni hal qilish usullari haqida ma’lumotlar bayon qilinadi.

C# tilida tasvirlar bilan ishlash uchun *System.Drawing* kutubxonasi ishlab chiqilgan. Tasvirli loyiha (kod) yaratilganda bu kutubxona to‘g‘ridan-to‘g‘ri kodga birlashtiriladi. Grafiklar bilan ishlashning asosiy sinfi *System.Drawing* nomlar fazosida joylashgan *Graphics* sinfi hisoblanadi. Huddi shu nomlar maydonida tasvirlar bilan ishlashda yordam beradigan boshqa sinflarni ham topish mumkin.

Graphics sinfi turli chizma yasash sirtlari (xolst) hamda chizish usullaridan foydalanishga imkon beradi. GDI+ yordamida nafaqat displeyda, balki boshqa grafik qurilmalarda (masalan, printerda) ham tasvirlar bilan ishlash mumkin.

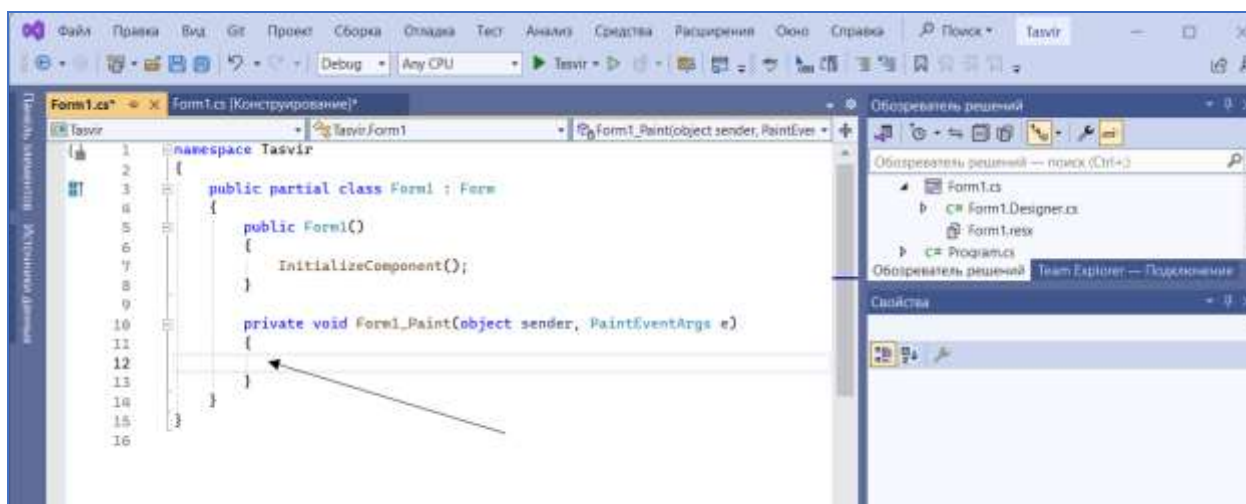
Graphics sinfi displey qurilmasida obyektlarni chizish usullarini qo‘llash imkonini taqdim etadi. Bu sinf obyektini hosil qilishning eng oddiy va ommabop usuli yangi ilova yaratish va forma oynasida *Paint* hodisasini tashkil qilishni nazarda tutadi. Bu muammo quyidagi tartibda amalga oshiriladi.

I. Paint hodisasi yordamida yangi ilova loyahasini tashkil qilish. Visual Studio dasturini ishga tushirib, **Выберите действие** → **Создание проекта** tugmalari chertiladi. Ochilgan **Создание проекта** muloqot oynasining **Приложение Windows Form** bandi belgilanadi va **Далее** tugmasi chertiladi. Shundan keyin **Настроить новый проект** – oynasining **Имя проекта** qismiga loyihaning nomi, **Расположение** oynasiga – manzili, **Имя решения** oynasiga esa yechin kodining nomi kiritiladi. Soʻngra **Далее** hamda keyingi **Дополнительные сведения** oynasidan **Создать** tugmalari chertiladi.



1-rasm. Tasvirlar bilan ishlash uchun boshlangʻich xususiyatlarni belgilash.

Ochilgan Visual Studio muhitining C# ishchi dialog oynasining oʻng tomonidagi **Свойства** boʻlimidan **СОБЫТИЯ** va **По категориям**, soʻngra **Paint** bandlari tanlanadi (1-rasm). Ekranda C# tilida kod yozish uchun tayyor ishlanma ochiladi (3-rasm). Bu ishlanmaning *private void ...* satridan keyingi “{” belgisidan keyin ilova kodini yozish mumkin boʻladi. Shu bilan, yangi loyihani yaratishning 1-qadami yakunlanadi.



2-rasm. C# tilidagi kodlarni yozish uchun tayyor ishlanma.

II. Paint hodisasini tashkil qilish. C# dasturlash tilida tasvirlar bilan ishlashning bir qator usullari mavjud bo’lib, ular orasida eng soddasi *Paint* hodidasidan foydalanishni nazarda tutadi. Bu amalni *Paint* usulining quyidagi tayyor ishlanmasi (shabloni) yordamida tashkil qilish mumkin:

```

private void Form1_Paint(object sender, PaintEventArgs e)
{
}
    
```

Bunda sinfning *PaintEventArgs* o‘zgaruvchisi orqali ikkita muhim *ClipRectangle* (qayta chizilishi talab qilinadigan maydon) hamda *Graphics* (rasm chiziladigan sirtning ifodalovchi *Graphics* sinfining nusxasi) xususiyatlari ochiladi. Tasvirlar aynan shu obyektida yaratiladi. *Paint* hodisasi yordamida tasvirlar bilan ishlash uchun formaga *Graphics* obyektining nusxasi (masalan, *e*) joylanadi va tegishli usullardan foydalaniladi.

Graphics sinfi tasvirlar bilan ishlash jarayonida qo‘llash uchun ko‘plab usul va xususiyatlarga ega. 1- va 2-jadvallarda dasturchilik amaliyotida ko‘p qo‘llanadigan ayrim usul va xususiyatlar haqidagi ma’lumotlar bayon qilinadi.

Graphics sinfi tasvirlar bilan ishlash xususiyatlari.

1-jadval.

Xususiyatlar	Tavsifi
Clip	xolst sirtini belgilovchi to‘rtburchakli soha;
ClipBounds	xolst uchun to‘rtburchakli soha (<i>RectangleF sinfi</i>);
CompositingMode	kompozit ¹ rasmlarni chizish usuli;
CompositingQuality	kompozit tasvirlarni ko‘rsatish sifatini belgilash;
Dpix	xolstning gorizontaal o‘lchami;
DpiY	xolstning vertikal o‘lchami;
PageScale	masshtablash;
PageUnit	Xolst uchun o‘lchov birliklari.

Graphics sinfining asosiy usullari.

2-jadval.

<i>Clear(Color)</i>	xolstni tozalash va uni <i>Color</i> rangi bilan to‘ldirish;
<i>CopyFromScreen (Point, Point, Size)</i>	Xolstning ko‘rsatilgan to‘rtburchakli sohasiga mos rangni aniqlash;
<i>DrawArc(Pen, Rectangle, Single, Single)</i>	<i>Rectangle</i> tuzilmasiga mos to‘rtburchak ichiga yoy chizish;
<i>DrawBezier(Pen, Point, Point, Point, Point)</i>	Beze egri chizig‘ini chizish;
<i>DrawBeziers(Pen, Point[])</i>	Bir nechta Beze egri chiziqlarini chizish;
<i>DrawCurve(Pen, Point[])</i>	egri chiziq chizish;
<i>DrawClosedCurve(Pen, Point[])</i>	yopiq egri chiziq chizish;
<i>DrawEllipse(Pen, Rectangle)</i>	ellips chizish;
<i>DrawIcon(Icon, Rectangle)</i>	piktogramma chizish;
<i>DrawImage(Image image, int x, int y)</i>	(x, y) nuqtadan boshlab <i>Image</i> rasmini chizish;
<i>DrawLine(Pen, Point, Point)</i>	Ikki nuqta orqali o‘tuvchi kesma chizish;
<i>DrawLines(Pen, Point[])</i>	siniq chiziqlar chizish;

¹ Kompozit rasmlar – ikki va undan ortiq rasmlar kompozitsiyasidan tashkil topgan rasmlar.

<i>DrawPie(Pen, Rectangle, Single, Single)</i>	<i>Rectangle</i> tuzilma hamda ikkita radial chiziq bilan aniqlanadigan sektor chizish;
<i>DrawPolygon(Pen, Point[])</i>	nuqtalar massivi bo‘yicha ko‘pburchak chizish;
<i>DrawPath(Pen, gp)</i>	<i>GraphicsPath gp</i> obyektini chizish;
<i>DrawRectangle(Pen, Rectangle)</i>	to‘rtburchak chizish;
<i>DrawString(String, Font, Brush, PointF)</i>	Ekranga <i>Brush</i> va <i>Font</i> bilan ko‘rsatilgan joydan matnli satrni chiqarish;
<i>ExcludeClip(Rectangle)</i>	<i>Graphics</i> obyektining <i>Rectangle</i> bilan aniqlanadigan sohasini olib tashlash;
<i>FillEllipse(Brush, Rectangle)</i>	Ellipsni rang bilan to‘ldirish;
<i>FillPolygon(Brush, Point[])</i>	ko‘pburchakni rang bilan to‘ldirish;
<i>FillRectangle(Brush, Rectangle)</i>	to‘rtburchakni rang bilan to‘ldirish;
<i>FillRegion(Brush, Region)</i>	yopiq sohani rang bilan to‘ldirish;
<i>FromImage()</i>	tasvir asosida <i>Graphics</i> obyektini yaratishning statik usuli;
<i>MeasureString()</i>	Muayyan shriftdan foydalanganda sirt-dagi matnli satr o‘lchamlarini hisoblash.

Bunday usullar ro‘yxatini davom ettirish mumkin. Hozircha keltirilgan usullar ro‘yxati yetarli deb hisoblanadi. Bu usullarning har biri *Pen* (qalam) yoki *Brush* (mo‘yqalam, cho‘tka) obyektidan foydalanadi. Bunda to‘g‘ridan-to‘g‘ri obyektarning o‘zi emas, balki ularning nusxalari nazarda tutiladi.

Tasvirlar bilan ishlash jarayoni rasm chiziladigan soha – xolstni yaratishdan boshlanadi. Odatda bu soha to‘g‘ri to‘rtburchak shaklidagi formadan iborat bo‘ladi. Xolstdagi har bir nuqta piksellarda beriladigan gorizontaal – *X* va vertikal – *Y* koordinatalari bilan belgilanadi. Bunda *X* ning koordinatasi joriy pozitsiya chapga o‘ngga, *Y* – esa yuqoridan pastga qarab surilganda ortib boradi.

Xolstning *g* sohasi hoshiyasini chizish uchun *DrawRectangle* usulidan foydalaniladi. Bu usulning umumiy ko‘rinishi quyidagicha:

```
g.DrawRectangle(SystemPens.ActiveBorder, new Rectangle(X0, Y0, X1, Y1));
```

Bu yerda *SystemPens.ActiveBorder* – soha hoshiyasini tizim uchun belgilangan qalamda chizishni anglatadi.

Har qanday tasvir nuqtalar to‘plamidan iborat ekanligi hammaga ma’lum. Demak, tasvir yasash uchun muayyan nuqtalarga *Color* yordamida xolst fonidan boshqa rang berish talab qilinadi. Ranglarni tanlashda ularning ingliz tilidagi shakllaridan foydalanish mumkin. Masalan:

```
Color Rang = Color.FromName("SlateBlue");
```

ko‘rsatmasi *Color* ning *Rang* nusxasi salat rangi, ya’ni, “ochiq yashil rang” qiymatiga teng bo‘lishini ta’minlaydi.

Tabiiyki, rasm chizish qalam tanlashdan boshlanadi. Turli chiziqlarni chizish uchun *Pen* sinfi nusxalaridan foydalaniladi. Nusxani yaratish (rangli qalam tanlash) variantlari juda ham ko‘p bo‘lib, ular orasida quyidagi shakllar keng qo‘llaniladi:

- 1) *Pen (Color)*; – *Pen* ning *Color* rangli yangi nusxasini yaratadi;
- 2) *Pens.Color*; – *Pen* ning *Color* rangli yangi nusxasini yaratadi;
- 3) *Pen (Color, Single)* – *Pen* ning yangi nusxasi *Color* (rang) va *Width* (chiziqning qalinligi) xususiyatlari bilan yaratiladi. Masalan:

```
Pen Qalam = new Pen(Color.Red); // chiziq qalinligi 1 pikselli qizil qalam
```

Shundan so‘ng, *Pen* o‘rniga uning qizil chiziq chizadigan *Qalam* nusxasidan foydalanib tasvirlar yasash mumkin bo‘ladi.

Qalam chizig‘i qalinligi *Pen.Width* xususiyati yordamida belgilanadi:

```
Pen.Width=8.0F;
```

Pen nusxalarini yaratishda oraliq o‘zgaruvchilardan ham foydalanish mumkin:

```
Green Color green = Color.Green;
```

```
Pen greenPen = new Pen(green, 4.5f); // jigar rangli qalam
```

Yangi obyektini *Pen* nomini oshkoq ko‘rsatmasdan (masalan, *Pen* dinamik obyektini yaratish) aniqlash mumkin. Masalan, *g* – xolstda *r* – qizil to‘g‘ri to‘rtburchakni 3 piksel qalinlikda chizish uchun buyruq quyidagicha yoziladi:

```
g.DrawRectangle(new Pen(Color.Red, 3), r);
```

Brush sinfi to‘rtburchak, ellips, doira, ko‘pburchak va yo‘llar kabi grafik shakllar ichini fon bilan to‘ldirish uchun foydalaniladigan obyektlarni aniqlaydi. *Brush* obyektini nusxasini yaratish uchun *Brush* ning *SolidBrush*, *TextureBrush* va *LinearGradientBrush* kabi hosila sinflaridan foydalaniladi. Masalan:

```
SolidBrush br = new SolidBrush(Color.Aqua);
```

```
g.FillRectangle (br, r); // g-xolstda r-to‘rtburchakni bo‘yash
```

Bu ko‘rsatmani *Brush* obyektini oshkor e‘lon qilmasdan ham yozish mumkin:

```
g.FillRectangle(new SolidBrush(Color.Red), r);
```

C# tilida yopiq sohalarga rang berish, ya’ni, bo‘yash uchun *Brush* mo‘yqalam (chytka) obyektini yoki uning nusxalari xususiyatlarini belgilash amalidan foydalaniladi. Masalan:

```
Brushes.Blue; // havo rangli mo‘yqalam
```

```
Brush myBrush = Brushes.Blue; // havo rangli mo‘yqalam nusxasi
```

Tasvirlarni hosil qilish uchun rang, qalam (yoki mo‘yqalam) tanlanganidan so‘ng, xolstda shakllarni yasashni boshlash mumkin. Buning uchun 1- va 2-jadvallarda keltirilgan usullardan ehtiyojga ko‘ra xususiyatlarni ko‘rsatgan holda foydalanish mumkin. Quyida bu usullardan ayrimlarini qo‘llash haqidagi ma’lumotlar bayon qilinadi. Bunda turli variantlarda amalga oshirilishi mumkin bo‘lgan xususiyatlarni tayinlash amalining eng ko‘p qo‘llanadigan shaklidan foydalaniladi.

To‘g‘ri chiziqlar chizish uchun

DrawLine(Pen, X0, Y0, X1, Y1);

usulidan foydalaniladi. Bu yerda *Pen* – qalam nusxasi, $(X0, Y0)$ – chiziqning boshlanish nuqtasi, $(X1, Y1)$ – esa tugash nuqtasi. Masalan:

DrawLine(Pens.Black, 100, 100, 500, 500);

Ichki sohasi bo‘yalgan to‘g‘ri to‘rtburchak tasvirini hosil qilish uchun

FillRectangle(Brushes.Color, X0, Y0, X1, Y1.); //sohani bo‘yash

bo‘yalmagan bo‘sh to‘g‘ri to‘rtburchak uchun esa

DrawRectangle(Pens.Color, X0, Y0, X1, Y1);// hoshiyasini chizish

ko‘rinishidagi buyruqlardan foydalaniladi. Bu buyruqlarda $(X0, Y0)$ – to‘g‘ri to‘rtburchakning chap yuqori uchi, $(X1, Y1)$ – esa uning balandligi va kengligi. Masalan:

FillRectangle(Brushes.Color, 300, 300, 350, 350.);

DrawRectangle(Pens.Red, 400,400, 450, 450);

Ichki sohasi bo‘yalgan ellips tasvirini hosil qilish uchun

FillEllipse (Brushes.Color, X0, Y0, X1, Y1.); //sohani bo‘yash

Bo‘yalmagan bo‘sh ellipsni chizish uchun

DrawEllipse(Pens.Color, X0, Y0, X1, Y1);// hoshiya chizish

buyrug‘idan foydalaniladi. Masalan:

g.DrawEllipse(sky, 250, 120, 50, 75);

e.Graphics.FillEllipse(Brushes.Red, 350, 120, 50, 75);

Quyidagi kodga e’tibor bering:

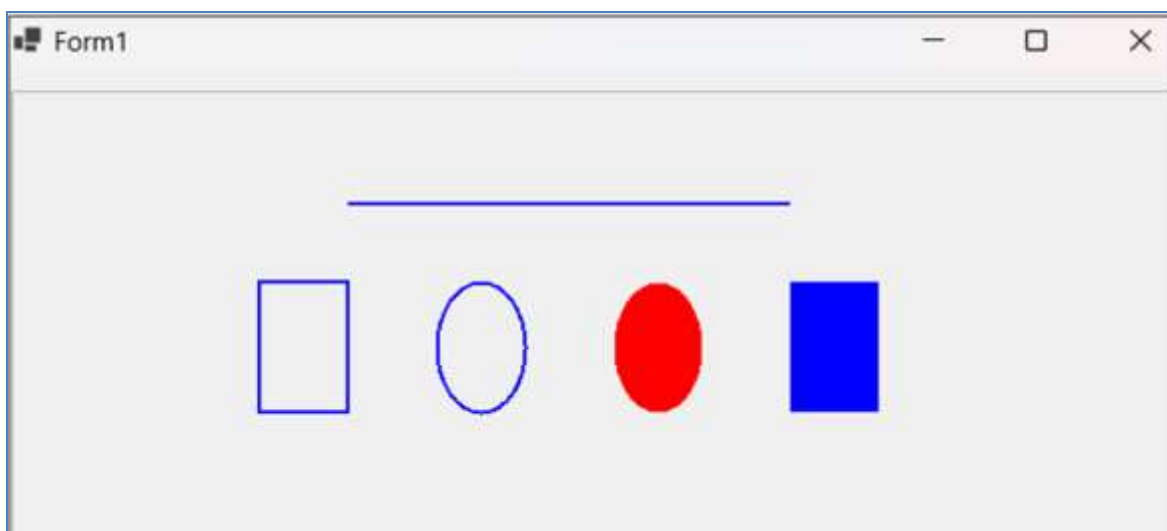
Color Blue = Color.FromName("Blue");


```
Pen sky = new Pen(Blue, 2.0f);  
int x = ClientSize.Width;  
int y = ClientSize.Height;  
Graphics g = Graphics.FromHwnd(Handle);  
g.DrawRectangle(SystemPens.ActiveBorder, 10, 10, x - 20, y - 20);  
g.DrawLine(sky, 200, 75, 450, 75);  
g.DrawRectangle(sky, 150, 120, 50, 75);  
e.Graphics.FillRectangle(Brushes.Blue, 450, 120, 50, 75);  
g.DrawEllipse(sky, 250, 120, 50, 75);  
e.Graphics.FillEllipse(Brushes.Red, 350, 120, 50, 75);
```

Keltirilgan ushbu kod ekranda 3-rasmdagi shakllarni hosil qiladi.

Ellipsning bo‘sh hamda bo‘yalgan sectorlari quyidagi ko‘rsatmalar asosida yasaladi:

```
FillPie(Brushes.Color, X0, Y0, X1, Y1, X3, Y3);  
DrawPie(Pens.Color, X0, Y0, X1, Y1, X3, Y3);
```



3-rasm. Keltirilgan kod natijasi.

Bu usullardan foydalanish uchun 7 ta parametrlarni ko‘rsatish lozim:

- 1) bo‘yash uchun rang;
- 2) chap yuqori burchak koordinatalari;

- 3) kengligi va balandligi;
- 4) Sektor burchagining boshlanishi;
- 5) Sektor burchagining oxiri.

Masalan:

```
FillPie(Brushes.Beige, 100, 100, 200, 200, 0, 90);
```

```
DrawPie(Pens.Red, 100, 100, 200, 200, 30, 45);
```

Tahlil va natijalar. C# tilida tasvirlar bilan ishlash uchun nuqtalar massivini parametr sifatida qabul qilishga mo‘jallangan ko‘plab usullar mavjud. Bu usullar bilan ishlash uchun dastlab nuqtalar massivini shakllantirish talab qilinadi. Masalan:

```
Point [] t={new Point(20, 10), new Point(10, 40), new Point(45,100), new  
Point(50, 80)};
```

Shundan so‘ng, bu massivdan foydalangan holda usullarga murojaat qilish mumkin.

Ko‘pburchaklarni yasash uchun *DrawPoligon* usulidan foydalaniladi. Bu usulning umumiy ko‘rinishi quyidagicha:

```
DrawPolygon(Pens.Color, nuqtalar_massivi);
```

Shu ko‘pburchakni bo‘yash uchun *FillPolygon* usuli quyidagi shaklda yoziladi:

```
FillPolygon(Brushes.Color, nuqtalar_massivi);
```

Masalan, yuqorida aniqlangan nuqtalar uchun bu buyruqni quyidagicha yozish mumkin:

```
g.DrawPolygon(Pens.Green, t);
```

```
g.FillPolygon(Brushes.Red, t);
```

Quyidagi kodga e’tibor bering:

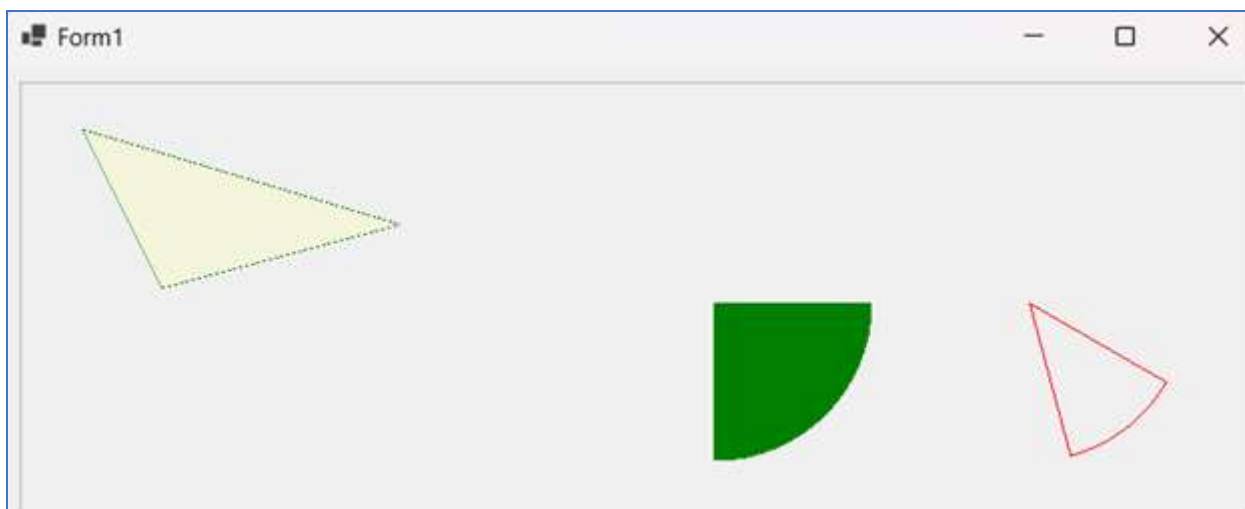
```
Color Blue = Color.FromName("Blue");
```

```

Pen sky = new Pen(Blue);
sky.Width = 2.0F;
Graphics g = Graphics.FromHwnd(Handle);
g.DrawRectangle(SystemPens.ActiveBorder,
    new Rectangle(10, 10, ClientSize.Width - 20,
        ClientSize.Height - 20));
e.Graphics.FillPie(Brushes.Green, 350, 50, 200, 200, 0, 90);
g.DrawPie(Pens.Red, 550, 50, 200, 200, 30, 45);
Point[] t = { new Point(50, 40), new Point(100, 140), new Point(250, 100),
new Point(50, 40) };
g.DrawPolygon(Pens.Green, t);
e.Graphics.FillPolygon(Brushes.Beige, t);
    
```

Ushbu kod natijasi 4-rasmda ifodalangan.

Funksiya grafigini nuqtalar orqali yasash. C# tilida funksiya grafigini nuqtalar orqali yasash uchun ko‘plab variantlar mavjud bo‘lib, ulardan biri maktab matematika kursidan ma’lum bo‘lgan an’anaviy usulni takomillashtirgan foydalanishni nazarda tutadi. Bu usulga ko‘ra, berilgan $y = f(x)$ funksiya grafigini yasash uchun 3 bosqichdan iborat amallar ketma-ketligi bajariladi:



4-rasm. Kodning natijasi.

- 1) x ga qiymatlar beriladi;
- 2) $y = f(x)$ funksiyaning qiymati hisoblanadi;
- 3) Olingan (x, y) nuqta Dekart koordinatalar sistemasida belgilanadi.

Ishning yakunida hosil qilingan barcha nuqtalar silliq chiziq bilan birlashtiriladi.

Bu usulni C# ga quyidagicha moslashtirish mumkin. Ma’lumki, *DrawLine()* usuli ikkita nuqtani birlashtirib, kesma tasvirini hosil qiladi. Shunga ko‘ra, birinchi uchi koordinatasi (x, y) , ikkinchi uchi koordinatasi $(x1, y1)$ bilan belgilangan chiziqni *DrawLine(rang, x, y, x1, y1)* yordamida birlashtirish orqali chizish mumkin.

Masala. $x = a \cos t(1 + \cos t)$ hamda $y = a \sin t(1 + \cos t)$, ($t \in [0, 2\pi]$) tenglamalar bilan berilgan kardioda tasvirini yasang.

Yechish g‘oyasi. Grafik kesmalar orqali yasaladi. Bunda t ning hamma qiymatlari Dekart va qutb koordinatalari sistemasi o‘rtasidagi bo‘g‘lanishdan foydalanish tavsiya etiladi. Buning uchun x va y o‘zgaruvchilarning qiymatlarini berilgan chiziq tengamalaridan foydalanib birma-bir hisoblash lozim. Dastlabki kesmaning birinchi uchi (x, y) bo‘lsin. Buni $t=0$ uchun hisoblanadi. Qolgan kesmalar uchun navbatdagi kesmaning ikkinchi uchi $(x1, y1)$ berilgan funksiyaning $t=t+0,01$ dagi qiymatidan foydalangan holda topiladi. So‘ngra, (x, y) va $(x1, y1)$ nuqtalar orqali o‘tuvchi kesma chiziladi. Joriy kesmaning ikkinchi uchi navbatdagi kesmaning birinchi uchi bilan ustma-ust tushgani uchun $x=x1$, $y=y1$ almashtirish bajariladi. Bunda aniqlangan har bir (x, y) koordinatali nuqta ekranga koordinatalar boshi ekranning markaziy qismida (uning koordinatasi (320, 240)) ekanligini hisobga olgan holda joylashtiriladi. Bu jarayon to t ning qiymati 2π dan katta bo‘lmaguncha davom etadi.

Ushbu jarayonni ifodalovchi kod quyidagicha yoziladi.

```
Color Black = Color.FromName("Black");
```

```
Pen sky = new Pen(Black);
```

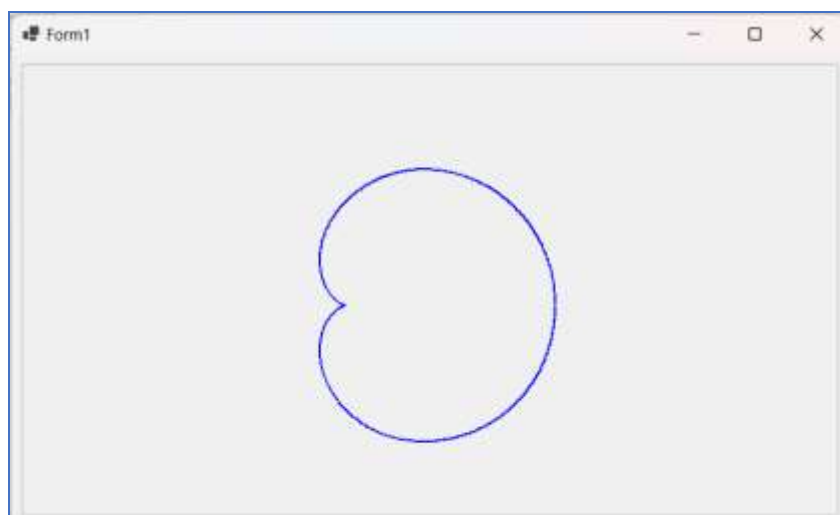
```
sky.Width = 2.0F;
```



```
Graphics g = Graphics.FromHwnd(Handle);
g.DrawRectangle(SystemPens.ActiveBorder,
    new Rectangle(10, 10, ClientSize.Width - 20,
        ClientSize.Height - 20));
int x, y, x1, y1, a = 100;
double fi = 0;
x = Convert.ToInt32(a * Math.Cos(fi) * (1 + Math.Cos(fi)) + 320);
y = Convert.ToInt32(240 - a * Math.Sin(fi) * (1 + Math.Cos(fi)));
while (fi <= 6.28)
{
    x1 = Convert.ToInt32(a * Math.Cos(fi) * (1 + Math.Cos(fi)) + 320);
    y1 = Convert.ToInt32(240 - a * Math.Sin(fi) * (1 + Math.Cos(fi)));
    fi = fi + 0.01;
    g.DrawLine(sky, x, y, x1, y1);
    x = x1; y = y1;
}
```

Keltirilgan kod ekranda 5-rasmdagi tasvirni hosil qiladi.

Ekranga matnlarni chiqarish. Tasvirlar bilan ishlash jarayonida ekranga qandaydir matnlarni ham chiqarishga to‘g‘ri keladi. Bu amalni bajarish uchun *DrawString* usulidan foydalaniladi. *DrawString()* usuli belgilangan *Pen* va *Font*



5-rasm. Kardioida.

obyektlarining xususiyatlari hamda formatlash atributlaridan foydalangan holda ekranda berilgan matn satrini chizadi. Bu usulning umumiy ko‘rinishi quyidagiha:

DrawString(S, Font, Brush, RectangleF, StringFormat);

Bu yerda: *S* – satrli matn; *Font* – matn uchun shrift; *Brush* – matnning rangi va shaklini belgilaydigan cho‘tka; *RectangleF* – matn chiqariladigan soha; *StringFormat* – matn uchun satrlararo interval, tekislash usuli kabi atributlarini belgilovchi bichim.

DrawString usulini qo‘llashda ayrim parametrlarni ko‘rsatmaslik ham mumkin. Bu holda ko‘rsatilmagan parametr o‘rniga tizim tomonidan o‘rnatilgan argumentlardan foydalaniladi.

Font – parametri satr uchun shriftni tayinlaydi. Bunda *Font* sinfi usullaridan foydalaniladi. Shriftning hajmi va uslubini tanlash imkoniyatini beradi. Bir-biridan argumentlar soni va yangi shriftni o‘rnatish usuli bilan farq qiladigan bir nechta murojaat variantlari mavjud. Masalan:

font shrift1 = new Font ("Arial", 24 , FontStyle.Bold) ;

yoki

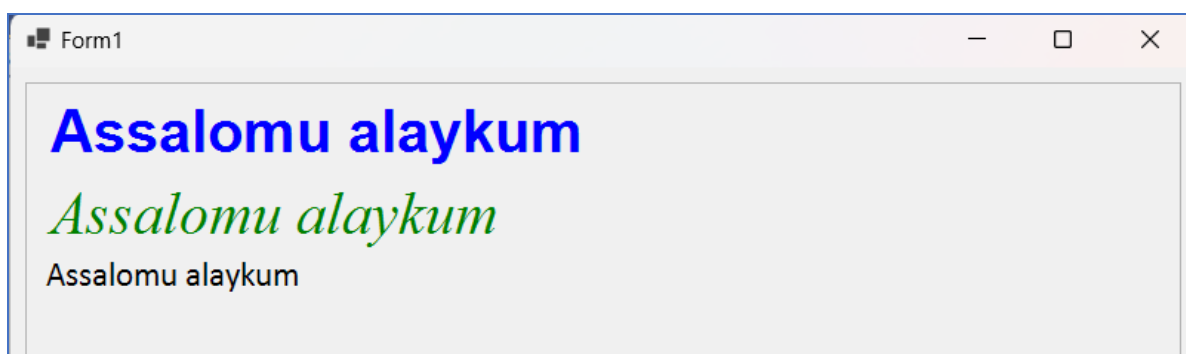
shrift shrift1 = yangi shrift ("Arial Narrow", 14) ;

RectangleF – parametri matn satri chiqariladigan to‘rtburchakli sohani ko‘rsatadi. Uni ikki xil usulda yozish mumkin:

- 4 ta koordinata orqali (chap yuqori ($X1, Y1$) va quyi o‘ng ($X2, Y2$) uchlar);
- Obyektning chap yuqori burchagi koordinatalari (P) va o‘lchamlari (h - to‘rtburchakning kengligi va W – balandligi).

Quyidagi buyruqlarining bajarilishiga e’tibor bering:

```
float width =600.0F, height = 100.0F;
RectangleF rect = new RectangleF(20, 20, width, height);
Color Blue = Color.FromName("Blue");
Pen sky = new Pen(Blue);
sky.Width = 2.0F;
Graphics g = Graphics.FromHwnd(Handle);
g.DrawRectangle(SystemPens.ActiveBorder,
new Rectangle(10, 10, ClientSize.Width-20, ClientSize.Height-20));
string s = "Assalomu alaykum";
Font shrift1 = new Font("Arial", 24, FontStyle.Bold);
g.DrawString(s, shrift1, Brushes.Blue, rect);
Font shrift2 = new Font("Times New Roman", 24, FontStyle.Italic);
g.DrawString(s, shrift2, Brushes.Green, 20, 75);
Font shrift3 = new Font("Calibri", 14);
g.DrawString(s, shrift3, Brushes.Black, 20, 125);
```



C# dasturlash tilining grafiklar bilan ishlashga mo‘ljallangan vositalari yuqorida bayon qilingan ma’lumotlar bilan chaegaralanmaydi. Keltirilgan

ma’lumotlar faqat sodda tasvirlar bilan ishlash uchun xizmat qiladi holos. Uning murakkab imkoniyatlari keyingi maqolalarda bayon etiladi.

Xulosa va takliflar. Talabalarga C # dasturlash tili bo‘yicha kerak bol‘gan yetarli va minimal hajmdagi o‘quv materiallarini ajratib olishda quyidagi ta’limiy maqsadlarni e’tiborga olish talab qilinadi. Bu o‘rinda biz taklif etayotgan C# ning tasvirlar bilan ishlashga doir materiallari quyidagi ta’limiy muammolarni hal qilishga yordam beradi:

- kompyuterda tasvirlar bilan ishlash bo‘yicha bilimlar doirasini kengaytiradi;
- tasvirni qayta ishlash kodlarini ishlab chiqish usullarini o‘rgatadi;
- kompyuterda oddiy tasvirlar bilan ishlashning shakl va usullari bo‘yicha bilimlarini shakllantiradi va takomillashtiradi;
- tasvirni yaratish algoritmlarini ishlab chiqish qobiliyatini rivojlantiradi.

C# dasturlash tilining tasvir bilan ishlash bo‘yicha egallagan bilim va malakalari quyidagi ta’limning quyidagi rivojlanturuvchi o‘quv maqsadlariga erishishga yordam beradi:

- mantiqiy va tizimli fikrlashni takomillashtirish;
- estetik va algoritmik dunyoqarashni rivojlantirish;
- natijalarni bashorat qilish buyruqlarning bajarilishi.

Adabiyotlar

1. Aripov M.M., Otaxanov N.A. Dasturlash asoslari. –Toshkent: Tafakkur gulshani, 2016. – 240 b.
2. Вирт Н. Преподавание информатики: потерянная дорога. <https://informatika-21.ru/texts/2002-06-Aarhus/ru.htm>
3. Дейтель Х. С# : Наиболее полное руководство. – СПб.: БХВ-Петербург, 2006. – 1024 с.
4. Троелсен Э. С# и платформа .NET. Библиотека программиста. –СПб.: Питер, 2004. –796 с.
5. Фленов М.Е. Библия С#. – СПб.: БХВ-Петербург, 2019. –512 с.

6. Шилдт Г. С# 4.0: полное руководство. –М.: ООО "И.Д. Вильяме", 2011. –1056 с
7. С++. Основы програмування. Теорія та практика: підручник [/О.Г. Трофименко, Ю.В. Прокоп, та ін.] – Одесса: Фенікс, 2010. – 544 с.
8. Седжвик Р. Фундаментальные алгоритмы на С++. / Седжвик Р. –М. ДиаСофт., 2001. – 688 с.
9. Топп У. Структуры данных в С++. / Топп У., Форд У. – М.: Бином., 2000. – 816 с.
10. Вирт Н. Алгоритмы и структуры данных. / Вирт Н. – М.: Мир, 1989. – 896 с.
11. Архангельський А. Я. Программирование в С++ Builder 6 и 2006./ Архангельський А. Я., Тагин М.А. – М.: Бином-Пресс, 2007. – 1184 с.
12. Примеры выполнения лабораторных работ по алгоритмам компьютерной графики: Метод. указания. /Сост.: Хайдаров Г.Г., Алексеев С.Ю. – СПб., СПбГТИ(ТУ), 2005. – 30 с.
13. Шикин А.В. Компьютерная графика. Динамика, реалистические изображения. / Шикин А.В., Боресков А.В. – М.: ДИАЛОГ-МИФИ, 1996. – 288 с.
14. Порев В.Н. Компьютерная графика./ Порев В.Н. - СПб.: БХВ-Петербург, 2002. – 432 с.