Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №4

по курсу «Компьютерная графика»

# «Реализация алгоритмов растровой графики для заполнения сплошных областей»

Выполнил студент группы ИВТ-21\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Коржавина А.С./

Киров 2021

**Цель работы**: закрепить лекционный материал по изучению базовых алгоритмов компьютерной графики алгоритмов закраски.

**Краткие теоретические сведения:**

***Построчный алгоритм заливки***

Применяется идея построчного сканирования и используется стек для хранения одного затравочного пиксела для любого непрерывного интервала на сканирующей строке. Непрерывный интервал – это группа примыкающих друг к другу пикселов, ограниченная уже закрашенными или граничными пикселами. Если непрерывный интервал пикселов принадлежит внутренней части области, пикселы на и под этим интервалом либо граничные, либо тоже находятся внутри области и могут служить затравочными для своих строк. Поэтому можно предложить следующую схему заполнения:

***Описание алгоритма:***

Дано : Pop(x,y) -процедура извлечения из стека координат (x,y) очередного пиксела

Push(x,y) - процедура помещения в стек координат (x,y) очередного пиксела

Нарисовать\_точку(x,y,col) - процедура для подкрашивания цветом col пиксела с координатами(x,y)

с(x,y) - цвет пиксела с координатами (x,y)

cb - цвет для подкрашивания пикселов границы области

ci - цвет для подкрашивания пикселов внутри области

(х0,y0) - координаты затравочного пиксела

Получить : Перекрасить сплошную область с данным затравочным пикселом и цветом границы в цвет ci.

1.Инициализировать стек:Push(x0,y0).

2.Пока стек не пуст выполнить: начало

Извлечь пиксел из стека и закрасить его в цвет области:

Pop(x,y);

Нарисовать\_точку(x,y,ci);

3.Присвоить: xw=x; x=x+1.

4.Пока c(x,y) # cb заполняем интервал справа от затравки:

Нарисовать\_точку(x,y,ci); x=x+1;по окончанию идти к 5.

5.Присвоить: xr=x-1; x=xw; x=x-1.

6.Пока c(x,y) # cb заполнть интервал слева от затравки: Нарисовать\_точку(x,y,ci);x=x-1; по окончанию идти к 7.

7.Сохранить крайний слева пиксел: xl=x+1;

8.Проверить строки ниже и выше данной, если там есть еще незаполненные пикселы искать затравку, начиная с левого края:

Для j от -1 до 2 с шагом 3 выполнить: начало

x=xl; y=y+j;

Пока x <= xr искать затравку на строке ниже(выше): начало

fl="ложь";

Пока ( c(x,y)#cb and c(x,y)#ci and x<xr ) заполнить точки внутри:

начало

Увеличить х на единицу.

Если (not fl) = "истина" то присвоить fl="истина". конец

Если fl="истина" то крайний справа пиксел - в стек: начало

Если ( x=xr and c(x,y)#cb and c(x,y)#ci ) то Push(x,y) иначе Push(x-1,y).

Присвоить fl="ложь". конец

Продолжить проверку,если интервал был прерван: Присвоить xb=x.

Пока (c(x,y)=cb or c(x,y)=ci and x<xr) увеличить x на 1. Если x=xb то присвоить: x=x+1.

конец

конец

конец пункта 2.

6.Закончить.

Алгоритм правильно заполняет любую область, справляется с внутренними дырами и зубцами на границе. Используется при закраске, когда известна хотя бы одна точка внутри необходимой для закрашивания области.

**Разработка программы**

Пользователь может менять фигуру нажатием правой кнопкой мыши и закрашивать область, выделенную щелчком левой кнопки мыши.

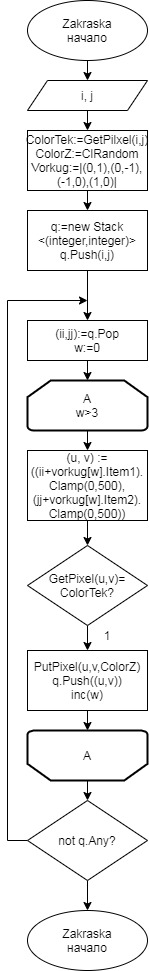
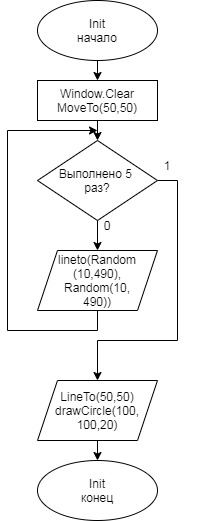
Схемы алгоритмов приведены в приложении А, листинг программ приведет в приложении Б, экранные формы – в приложении В

**Вывод:**

В ходе выполнения лабораторной работы были получены знания об алгоритмах закраски. В результате была написана программа, реализующая изученный метод.

Приложение А

Схемы алгоритмов

Приложение Б

Листинг программы

**procedure** Zakraska(i,j:integer);

**begin**

**var** ColorTek := GetPixel(i,j);//техущий цвет

**var** ColorZ := ClRandom;//выбор случайного цвета

**var** vorkug := |(0,1), (0,-1),(-1,0),(1,0)|; //(влево вправо вниз вверх )

**var** q := **new** Stack<(integer,integer)>; //создаем стек q

q.Push((i, j));//инициализировать стек

**repeat**

**var** (ii, jj) := q.Pop;//достать пиксель из стека

**For var** w := 0 **to** 3 **do** //пробежка с разных сторон затравки

**begin**

**var** (u, v) :=((ii+vorkug[w].Item1).Clamp(0,500), (jj+vorkug[w].Item2).Clamp(0,500) );//заполняем интервал 1) слева от затравки,2) справа, 3)внизу

**if** GetPixel(u,v) = ColorTek **then begin**

PutPixel(u,v,ColorZ);//нарисовать пиксель

q.Push((u,v)) //подожить его в стек

**end**;

**end**;

**until not** q.Any;//пока затравленный пиксель внутри

**end**;

**procedure** Init;

**begin**

Window.Clear;

Moveto(50,50);//начинаем из этой координаты

**loop** 5 **do** lineto(Random(10,490),Random(10,490));//случайные координаты для рисования линий

lineto(50,50);//замыкаем

DrawCircle(100,100,20);//кружочек - не подвижен

**end**;

Приложение В

Экранные формы

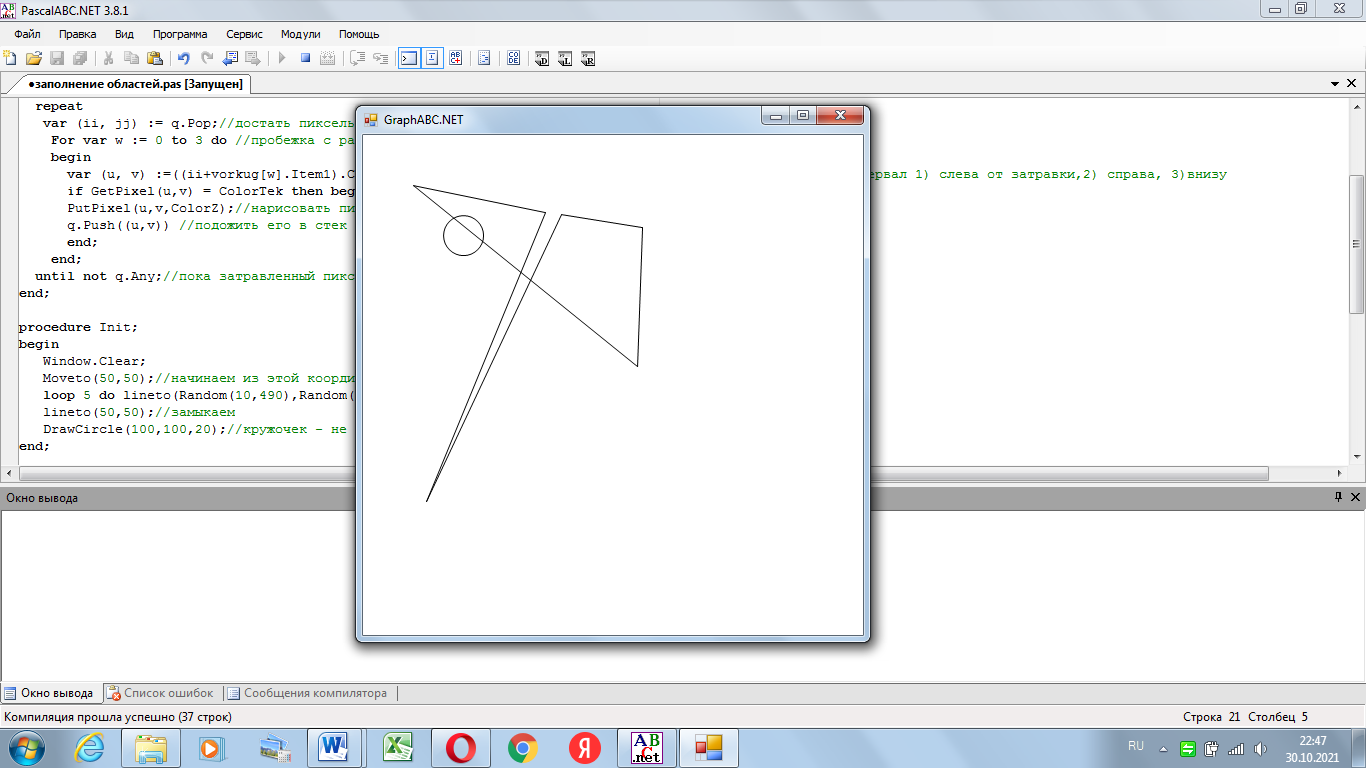


Рисунок 1 –первоначально сгенерированная фигура

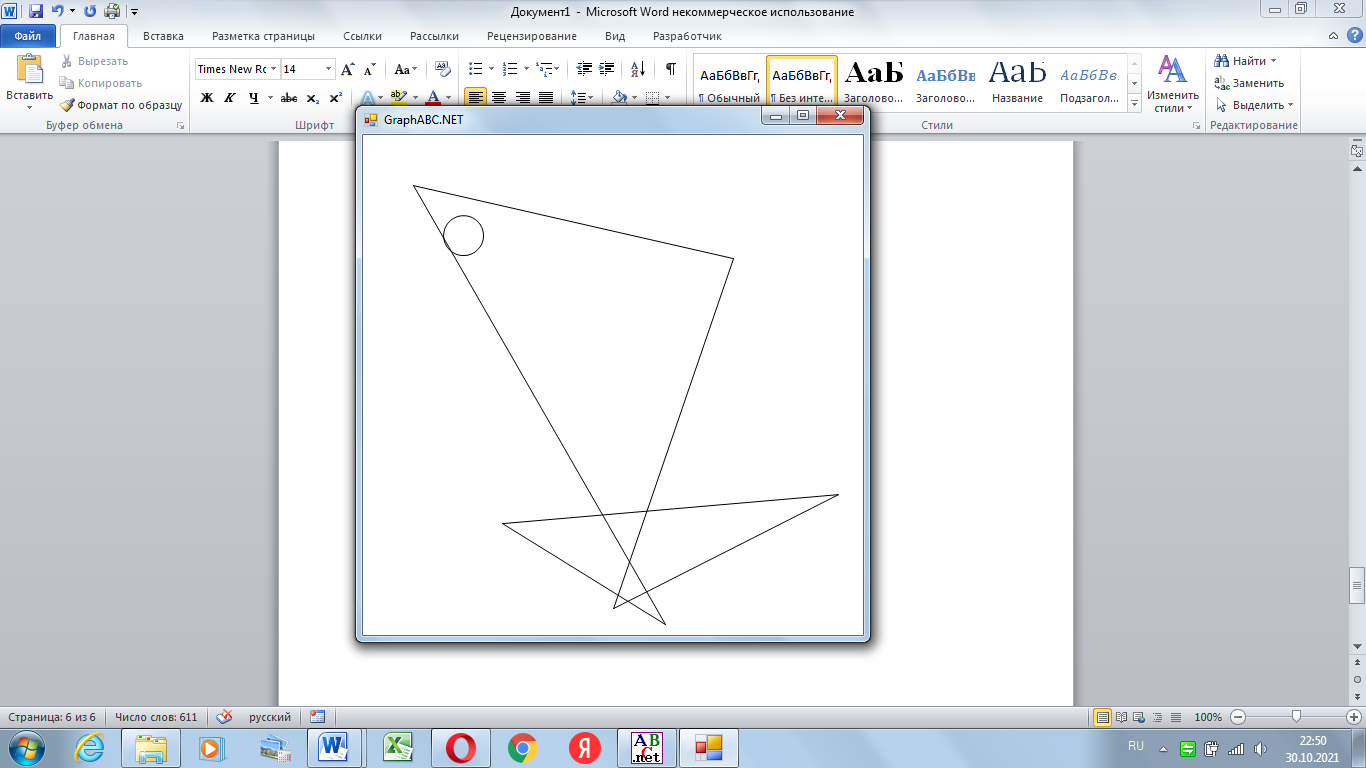


Рисунок 2 – изменение фигуры при щелчке мыши

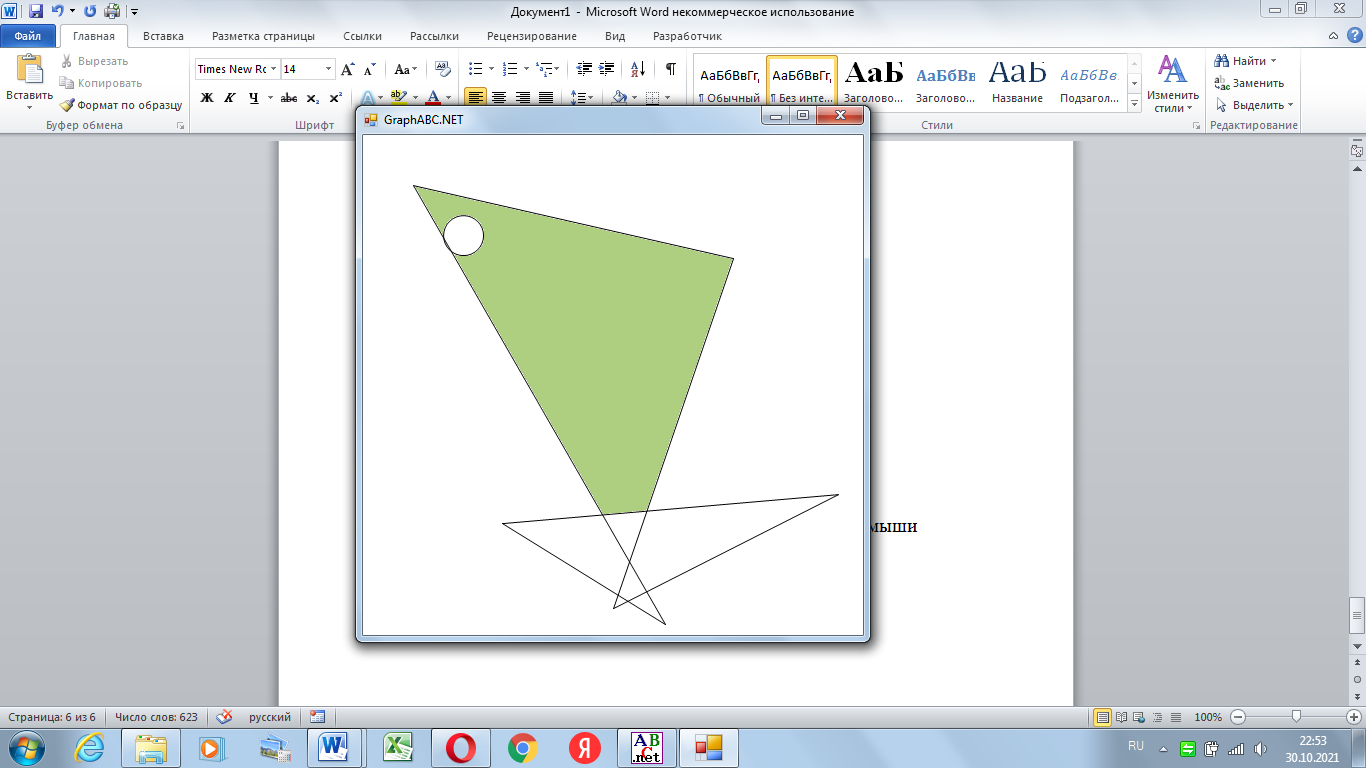


Рисунок 3 – закраска выделенной области