

Управление образования Лидского районного исполнительного комитета
Государственное учреждение образования
«Лидский районный центр технического творчества»

**Методическая разработка занятия по теме
«Работа с графикой в С++»**

Разработчик:
Петровский Илья Леонидович,
педагог дополнительного
образования
без категории
Лидского РЦТТ
Методическое сопровождение:
Янковская Анна Владимировна,
методист
высшей квалификационной
категории
Лидского РЦТТ

Методическая разработка занятия по теме «Работа с графикой в C++»

Тема занятия: Работа с графикой в C++

Цель занятия: Получить общие сведения о графическом режиме. Научиться создавать элементарные графические изображения программными средствами C++.

Задачи:

Образовательная: Знать основные сведения о графическом режиме в процедурном программировании. Уметь использовать функции для рисования примитивных фигур в C++.

Развивающая: Развитие навыков самостоятельного логического мышления, развивать у учащихся умение выделять главное, существенное в изученном материале, развитие навыков сравнения, обобщения, анализа, умение работать с текстом, развивать культуру поведения, воспитывать у учащихся наблюдательность, внимание и усидчивость.

Воспитательная: Привитие интереса к избранной профессии, воспитание казахстанского патриотизма, учиться быть толерантным, оказывать взаимопомощь, воспитывать уверенность в своих силах, трудолюбие, активность, внимание, ответственность за порученное дело.

Тип урока: Комбинированный

Оборудование урока: ПЭВМ, доска, мультимедиа проектор, экран, раздаточный материал

Ход занятия:

I Орг момент :

- a) Отметить отсутствующих.
- b) Сообщение темы и цели урока.

II Объяснение новой темы: «Работа с графикой в C++»

- 1. Графический режим
- 2. Функции черчения и заполнения
- 3. Функции установок и настроек

III Закрепление новой темы: Выполнение задания на компьютере

IV Подведение итогов: Обобщение результатов урока – оценка знаний студентов.

Обобщение новой темы.

План

1. Опрос и актуализация знаний.
2. Стадия вызова. Как создаются игры или другие современные программы (Какие данные преобладают? ГРАФИКА). Как создавать графические элементы.
3. Стадия осмысления. Объяснение новой темы с использованием мультимедиа проектора и экрана. Рисование домика
4. Выполнение практического задания индивидуально.
5. Рефлексия. Ответить на четыре вопроса.

Ход занятия

1. Приветствие. Проверка присутствующих и отсутствующих
2. На предыдущих занятиях мы работали с различными функциями, типами данных и операторами для написания программ. Давайте вспомним их.

Задание: Даны зарезервированные слова языка программирования. Определить для чего они используются.

1. `main()`
2. `getch()`
3. `int`
4. `return`
5. `if`
6. `iostream`
7. `while`
8. `double`
9. `float`
10. `rand()`
11. `char`
12. `for`
13. `string`
14. `short`
15. `cout`
16. `cin`
17. `void`
18. `max()`
19. `sqrt()`
20. `pow()`
21. `endl`

Педагог задает вопросы: Назовите ...

Функция вычисления степени числа (20)

Целый тип данных (3)

Вещественный тип данных с двойной точностью (8)

Главная функция программы (1)

Символьный тип данных (11)

Оператор условия (5)

Команда завершения строки (21)

Функция генерации случайного числа (10)

Оператор цикла с предусловием (7)

Оператор цикла с параметром (12)

Функция вывода данных на экран (15)

Функция ввода данных (16)

Библиотека функций ввода-вывода (6)

и т.д.

Определить значение переменной `s` после выполнения фрагмента программы.

```
int i, s=0;
for (i=0; i<10; i++)
s=s+1;
```

Ответ: s=10

```
int i, s=0;
for (i=0; i<10; i++)
s=s+i;
```

Ответ: s=45

```
int i, s=0;
for (i=0; i<10; i++)
if (i%2== 0) s=s+i;
```

Ответ: s=20

В основном, с какими данными мы с вами работали? (числовые и текстовые).

Современные программы (например, игры) какие данные содержат? (графика)

А как создается графика? (Предположения учащихся)

А можно ли создавать графические элементы в языке программирования c++?

Значит тема нашего урока: Работа с графикой в C++.

Цель нашего урока. Научиться использовать функции графического режима для построения элементарных изображений в языке C++.

3. Объяснение нового материала с использованием демонстрации на экране интерактивной доски

При работе с графикой и текстом используются различные видеорежимы.

Компьютер обязательно имеет некоторый видеоадаптер. Это может быть монохромный дисплейный адаптер (MDA) для базового (только текстового) дисплея, либо это может быть графический адаптер, например цветной графический адаптер (CGA), улучшенный графический адаптер (EGA), монохромный графический адаптер Hercules или видеографическая матрица (VGA/SVGA). Каждый из этих адаптеров может работать в нескольких режимах. Режим определяет величину экрана - 80 или 40 символов в строке (только в текстовом режиме), разрешающую способность экрана (только в графическом режиме) и тип дисплея (цветной или черно-белый).

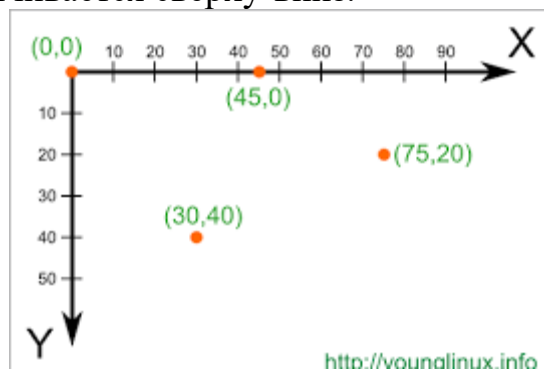
Рабочий режим экрана определяется, когда ваша программа вызывает одну из функций определения режима (textmode, initgraph или setgraphmode).

- В текстовом режиме экран компьютера разделен на ячейки (80 или 40 столбцов в ширину и 25, 43 или 50 строк по высоте). Каждая

ячейка состоит из атрибута и символа. Символ представляет собой имеющий графическое отображение символ кода ASCII, а атрибут задает, каким образом данный символ будет выведен на экран (его цвет, яркость, и т.д.). C++ предоставляет полный набор подпрограмм для манипулирования текстовым экраном, для вывода текста непосредственно на экран и управления атрибутами ячеек.

- В графическом режиме экран компьютера делится на элементы изображения (пикселы); каждый элемент изображения представляет собой отображение на экране одной точки. Число элементов изображения на экране (т.е. его разрешающая способность) зависит от типа подключенного к вашей системе видеоадаптера и режима, в который установлен этот адаптер. Для получения на экране графических изображений C++ предоставляет библиотеку графических функций: вы можете создавать на экране линии и формы, заполненные шаблонами замкнутые области, а также управлять цветом каждого элемента изображения.

В графическом режиме позиция верхнего левого угла определяется координатами (0,0), где x-координата растет слева-направо, а y-координата увеличивается сверху-вниз.



Для работы с графикой в C++ необходимо подключить библиотеку `graphics.h`

```
#include <graphics.h>
```

Также необходимо выбрать графический режим работы:

```
int gd=DETECT, gm;  
initgraph(&gd, &gm, " ");
```

Функция `initgraph` - Инициализирует графическую систему и переводит аппаратное обеспечение в графический режим.

Как правило размер графического экрана (640, 480).

Графический пакет компилятора C++ обеспечивает графические драйверы для следующих графических адаптеров (и полностью совместимых с ними):

- Цветной/графический адаптер (CGA);
- Многоцветная графическая матрица (MCGA);
- Улучшенный графический адаптер (EGA);
- Видеографическая матрица (VGA);
- Графический адаптер Hercules;

- Графический адаптер серии AT&T 400;
- Графический адаптер 3270 PC;
- Графический адаптер IBM 8514.

Для завершения работы в графическом режиме и возврат в первоначальный графический режим необходимо вызвать функцию `closegraph()`.

Мы рассмотрели некоторые функции управления графической системой, а теперь рассмотрим несколько функций для черчения и заполнения.

`putpixel( int x, int y, int color );` - рисование точки определенного цвета.

`line( int x1, int y1, int x2, int y2 );` - рисование линии от точки (x1,y1) до (x2,y2).

`rectangle( int left, int top, int right, int bottom );` - рисование прямоугольника.

`circle( int x, int y, int radius );` - рисование окружности с заданным радиусом.

`arc( int x, int y, int stangle, int endangle, int radius );` - дуга окружности.

`ellipse( int x, int y, int stangle, int endangle, int xradius, int yradius );` - эллиптическая дуга.

`sector( int x, int y, int stangle, int endangle, int xradius, int yradius );` - сектор.

`floodfill( int x, int y, int border );` - закрашивает замкнутую область до границы цвета border.

Для работы с цветами используют функции:

`setcolor( int color );` - выбор цвета рисования.

`setbkcolor( int color );` - выбор цвета фона.

Далее в таблице определены числовые значения цветов

Цвет	Константа	Цвет	Константа
Черный	Black = 0	Темно-серый	DarkGray = 8
Синий	Blue = 1	Ярко-синий	LightBlue = 9
Зеленый	Green = 2	Ярко-зеленый	LightGreen = 10
Голубой	Cyan = 3	Ярко-голубой	LightCyan = 11
Красный	Red = 4	Ярко-красный	LightRed = 12
Сиреневый	Magenta = 5	Светло-сиреневый	LightMagenta = 13
Коричневый	Brown = 6	Желтый	Yellow = 14
Светло-серый	LightGray = 7	Белый	White = 15

Для очистки экрана используется функция `cleardevice()`.

4. Теперь давайте попробуем все вместе нарисовать домик. На доске у нас будет воображаемый экран и исходный код. А вы на местах работаете в среде программирования.

```
#include <graphics.h>
#include <conio.h>

int main()
{
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "");
```

(Здесь пишем команды рисования!)

```
closegraph();
    getch();
```

```
return 0;
```

```
}
```

5. Самостоятельная работа. Нарисуйте в среде программирования `c++` произвольный рисунок (кораблик, самолет, машину и т.д.)

7.Рефлексия

Заполните таблицу записав продолжение фразы

Сегодня я узнал...	Я научился ...	У меня получилось...	Было трудно ...

8.Домашнее задание. Повторите все изученные функции и попробуйте нарисовать несколько деревьев.