Урок 12.

Задание для самостоятельной работы.

1. Составьте объектно-ориентированную программу, имитирующую работу автоматического турникета при въезде на парковку. Программа должна моделировать ситуацию, когда часть машин оплачивает парковку, а часть машин проезжает бесплатно. В кассе ведется учет числа проехавших за день машин и суммарной выручки от оплаты парковки.
   Для реализации программы создайте класс, например, с именем *gate*.
   В состав полей данных класса включите три переменные:
   - целочисленную переменную, например, с именем *payCars*, типа *unsigned int* для учета числа машин, оплативших парковку;
   - целочисленную переменную, например, с именем *nopayCars*, типа *unsigned int* для учета числа машин, не оплативших парковку;
   - целочисленную переменную, например, с именем *totalCash*, типа *unsigned int* для хранения суммарной выручки от оплаты парковки.

   В состав методов класса включите следующие функции:
   - конструктор класса, инициализирующий все поля нулевыми значениями;
   - функцию, например, с именем *payingCars()*, инкрементирующую количество оплативших парковку машин и увеличивающую суммарную выручку на величину тарифа за парковку;
   - функцию, например, с именем *nopayingCars()*, инкрементирующую количество не оплативших парковку машин и оставляющую суммарную выручку без изменения;
   - функцию, выводящую на экран результаты работы турникета за день.

   В функции *main()* продемонстрируйте работу класса. Для этого в ней:
   - создайте объект класса *gate*;
   - предложите пользователю:
     - нажать одну клавишу для имитации заплатившего водителя;
     - нажать другую клавишу для имитации недобросовестного водителя;
     - нажать клавишу *ESC* (код клавиши – 27).

   Нажатие клавиши *ESC* должно приводить к выводу всей информации о работе парковки и завершению работы программы.
   Для считывания введенных пользователем данных используйте функцию *_getche()*.
   Для передачи тарифа за парковку в метод *payingCars()* используйте глобальную переменную.
   Возможный вариант работы программы для тарифа за парковку в 50 рублей представлен ниже.

```
Нажмите клавиши: 0 - для каждой машины без оплаты
                 1 - для каждой оплачивающей машины
                 Esc - для выхода

11100101011111011101

Количество проехавших машин:    20
Из них: оплатили парковку:      14
        не оплатили парковку:   6

Общая сумма выручки: 700 рублей
```

2. Составьте объектно-ориентированную программу, в которой объектом будет прямоугольник, отображающийся на экране. Для этого создайте класс *rect*, содержащий три поля данных и два метода. Объектом класса *rect* будет прямоугольник произвольного размера, отображающийся произвольным цветом в произвольном месте экрана.
   В качестве полей данных класса *rect* задайте:
   - координаты левого нижнего угла прямоугольника (структурная переменная типа *COORD*, например, с именем *COORD left_bottom*);
   - координаты правого верхнего угла прямоугольника структурная переменная типа *COORD*, например, *COORD right_top*);
   - цвет прямоугольника.

   Для задания цвета прямоугольника:
   - предварительно создайте структуру (например, с именем *color*), полями которой будут три целочисленные переменные, определяющие интенсивность трех основных цветов (красного, зеленого и синего) для макроса *RGB*;
   - включите в состав третьего поля класса *rect* структурную переменную типа *color*, например, *color color_type*).

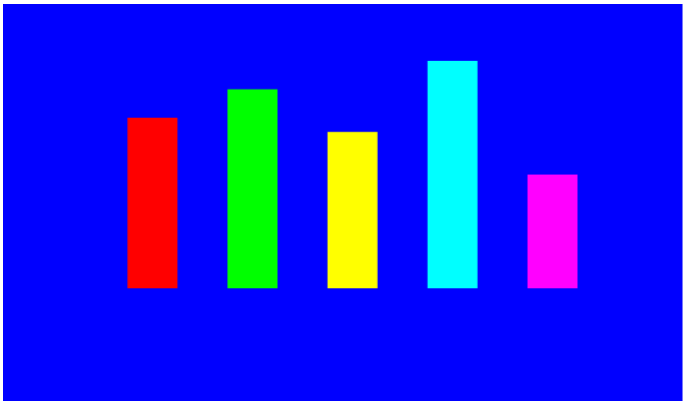
В качестве методов класса *rect* создайте:

- конструктор без аргументов, создающий прямоугольник, цвет которого совпадает с цветом фона экрана, а размер равен размеру экрана;
- функцию *set_rect()* для задания параметров прямоугольника;
- функцию *draw_rect()* для отображения прямоугольника на экране.

Протестируйте работу класса в функции *main()*. Для этого:

- создайте несколько объектов класса *rect*;
- с помощью метода *set_rect()* задайте значения всем полям созданных объектов;
- с помощью метода *draw_rect()* выведите все объекты на экран.

Результат работы программы может выглядеть следующим образом.



3. Составьте объектно-ориентированную программу, которая будет последовательно изменять размеры и цвет объекта в виде прямоугольника от минимального, расположенного в центре экрана, до максимального, соответствующего размерам экрана, и обратно. Шаг изменения размеров объекта и его цвет на каждой итерации – произвольные.

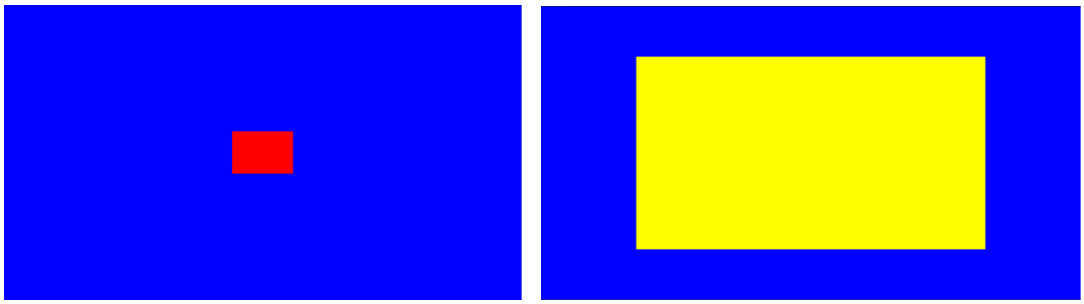
   Для этого модернизируйте класс *rect* следующим образом:

   - добавьте в состав методов класса *rect* функцию *get_rect()*, которая будет возвращать координаты объекта; для этого реализуйте передачу в эту функцию двух аргументов типа *COORD* по указателю;
   - замените конструктор класса без аргументов на конструктор с тремя аргументами;
   - установите в конструкторе *параметры по умолчанию*, которые будут располагать вновь создаваемый объект *по центру* экрана и окрашивать его в какой-то *определенный цвет* (например, в красный).

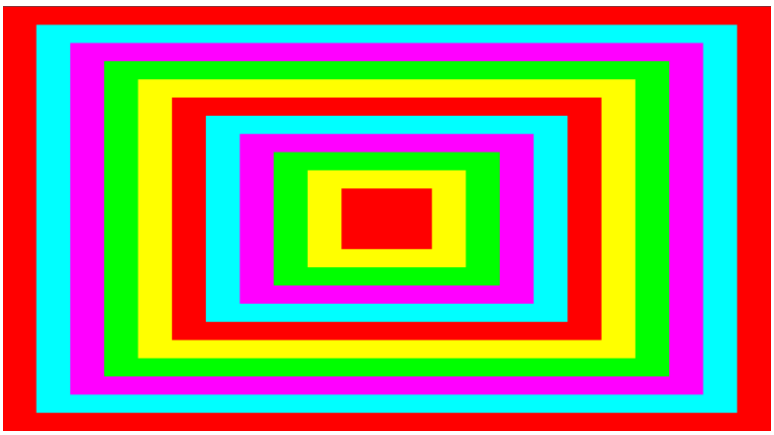
   Алгоритмы изменения координат прямоугольника реализуйте *в отдельных функциях* (например, с именами *increase()* и *decrease()*), *не входящими в состав методов класса rect*. *Передачу аргументов* (т.е. структурных переменных типа *COORD*) в эти функции осуществите *по указателям*.

   Для визуализации процесса изменения параметров объекта реализуйте вывод объекта с измененными параметрами с задержкой по времени (например, в 1 секунду). Воспользуйтесь для этого библиотечной функцией *Sleep()*.

   Возможный результат работы программы при прямом проходе (начальный и один промежуточный) представлен ниже.



Возможный результат работы программы при обратном проходе представлен ниже.



4. Реализуйте объектно-ориентированный вариант программы из второй части урока 10, которая выводит на экран изображение мишени. Для этого создайте класс, например, с именем *circle*, объектом которого будет круг произвольного радиуса и цвета, расположенный в произвольном месте экрана.
   В качестве полей данных класса *circle* задайте:
   - координаты центра круга (структурная переменная типа *COORD*);
   - значение радиуса круга (переменная типа *int*);
   - цвет круга (структурная переменная типа *color* (см. программу из задания №2)).

   В качестве методов класса *circle* создайте три функции:
   - конструктор с тремя аргументами; установите в конструкторе *параметры по умолчанию*, которые будут задавать вновь создаваемому объекту *определенный радиус*, располагать его *по центру* экрана и окрашивать в какой-то *определенный цвет* (например, в красный цвет);
   - функцию *set_circle()* для задания параметров круга;
   - функцию *draw_circle()* для отображения круга на экране.

   Протестируйте работу класса в функции *main()*. Для этого:
   - создайте объект класса *circle*;
   - реализуйте алгоритм изменения радиуса объекта и его цвета;
   - с помощью метода *set_circle()* реализуйте задание значений всем полям объекта;
   - с помощью метода *draw_circle()* реализуйте вывод объекта на экран.

   Результат работы программы должен совпадать с результатом работы программы из урока 10.