

UNI: Объединение

Код.

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

#define PI 3.1415

typedef enum {
    RED,
    GREEN,
    BLUE,
} Color;

typedef enum {
    CIRCLE,
    RECTANGLE,
    POLYGON,
} ShapeType;

typedef struct {
    ShapeType type;
    Color color;
    union {
        struct {
            float radius;
        } circle;

        struct {
            float width;
            float height;
        } rectangle;

        struct {
            int sides;
            float side_length;
        } polygon;
    } dimensions;
} Shape;

void analyzeShapes(const char *filename) {
    FILE *file = fopen(filename, "rb");

    fseek(file, 0, SEEK_END);
    long fileSize = ftell(file);
    fseek(file, 0, SEEK_SET);

    size_t shapeSize = sizeof(Shape);
    size_t shapeCount = fileSize / shapeSize;

    Shape *shapes = (Shape *)malloc(shapeCount * shapeSize);
    size_t readedCount = fread(shapes, shapeSize, shapeCount, file);

    fclose(file);

    printf("Theoretical shape count: %zu\n", shapeCount);
```

```

printf("Readed shape count: %zu\n", readedCount);

int circleCount = 0, rectangleCount = 0, polygonCount = 0;
int redCount = 0, greenCount = 0, blueCount = 0;
double totalSquare;

for(int i = 0; i < shapeCount; i++){
    switch (shapes[i].type)
    {
        case CIRCLE:
            circleCount++;
            totalSquare += PI * pow(shapes[i].dimensions.circle.radius, 2);
            break;
        case RECTANGLE:
            rectangleCount++;
            totalSquare += (
                shapes[i].dimensions.rectangle.height *
                shapes[i].dimensions.rectangle.width
            );
            break;
        case POLYGON:
            polygonCount++;
            totalSquare += (
                (shapes[i].dimensions.polygon.sides / 4.) *
                pow(shapes[i].dimensions.polygon.side_length, 2) *
                pow(tan(PI / shapes[i].dimensions.polygon.sides), -1)
            );
            break;
        default:
            break;
    }

    switch (shapes[i].color)
    {
        case RED:
            redCount++;
            break;
        case GREEN:
            greenCount++;
            break;
        case BLUE:
            blueCount++;
            break;
        default:
            break;
    }
}

printf("Circle count: %d\n", circleCount);
printf("Rectangle count: %d\n", rectangleCount);
printf("Polygon count: %d\n", polygonCount);

printf("Red shape count: %d\n", redCount);
printf("Green shape count: %d\n", greenCount);
printf("Blue shape count: %d\n", blueCount);

printf("Total square: %.2lf\n", totalSquare);
}

int main() {
    analyzeShapes("uni_shapes.bin");
}

```

```
    return 0;  
}
```

Пример работы программы.

```
Theoretical shape count: 42  
Readed shape count: 42  
Circle count: 13  
Rectangle count: 14  
Polygon count: 15  
Red shape count: 14  
Green shape count: 12  
Blue shape count: 16  
Total square: 1137.21
```