



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

Imię i Nazwisko:  Jan Perko Radosław Wojtukiewicz		Wydział Inżynierii Mechanicznej i Mechatroniki	Grupa: LA01 MECHATRONIKA Semestr VI
PRZEDMIOT: INŻYNIERIA PROGRAMOWANIA (PROJEKT)		OCENA: .....	
PROWADZĄCY: dr inż. Kamil Stateczny		PODPIS: .....	

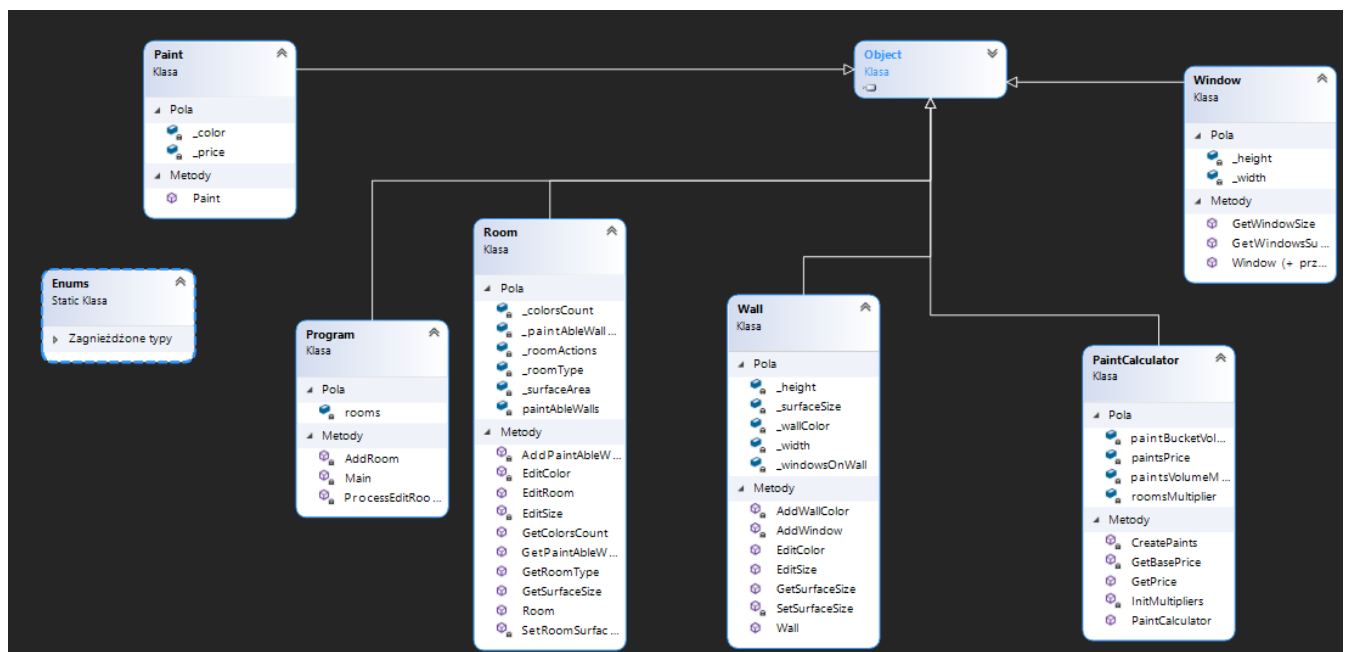
## 1. Cel projektu

Celem projektu było stworzyć program który ułatwi obliczanie ilości farby która będzie potrzebna na malowanie ścian w różnych pomieszczeniach. Dzięki temu można oszczędzić czas na obliczeniach.

## 2. Założenia projektu

Głównym założeniem było aby można było dobrać rozmiary ścian do każdego pomieszczenia które chcemy malować z uwzględnieniem okien. Pomaga to w szybszym rozpisaniu oraz obliczeniu ilości farby którą będziemy potrzebować oraz szacowany koszt farby.

## 3. Diagram klas



#### 4. Omówienie kodu

Poniżej widać początek programu gdzie wywoływana funkcja która zapisze nam ilość pokoi podanych przez użytkownika aby można było ją uwzględnić w dalszej części programu.

```
static void Main(string[] args)
{
    Program mainProgram = new Program();

    //początek programu gdzie wywołujemy pierwszą funkcję

    PaintCalculator calculator = new PaintCalculator();
    mainProgram.rooms = new List<Room.Room>();

    Console.WriteLine("Ile pokoi chcesz pomalować?");

    int roomsCount = 0;
    while (!int.TryParse(Console.ReadLine(), out roomsCount) || roomsCount <= 0)
    {
        Console.WriteLine("Liczba pokoi musi być większa niż 0!");
    }

    float totalPrice = 0f;

    for (int i = 0; i < roomsCount; i++)
    {
        Console.Clear();
        var room = mainProgram.AddRoom();
        mainProgram.rooms.Add(room);
    }
}
```

Zdeklarowane wartości są poprzez Enums aby kod był bardziej przejrzysty.

```
1 odwołanie
private void InitMultipliers()
{
    roomsMultiplier = new Dictionary<Enums.RoomType, float>();

    roomsMultiplier.Add(Enums.RoomType.Bathroom, 1.5f);
    roomsMultiplier.Add(Enums.RoomType.Bedroom, 1f);
    roomsMultiplier.Add(Enums.RoomType.Corridor, 1f);
    roomsMultiplier.Add(Enums.RoomType.Garage, 1.2f);
    roomsMultiplier.Add(Enums.RoomType.Salon, 1f);

    paintsVolumeMultiplier = new Dictionary<int, float>();

    paintsVolumeMultiplier.Add(5, 5);
    paintsVolumeMultiplier.Add(10, 9);
    paintsVolumeMultiplier.Add(15, 13);
}

1 odwołanie
private void CreatePaints()
{
    paintsPrice = new Dictionary<Enums.PaintColor, float>();

    paintsPrice.Add(Enums.PaintColor.Black, 10);
    paintsPrice.Add(Enums.PaintColor.Blue, 20);
    paintsPrice.Add(Enums.PaintColor.Green, 15);
    paintsPrice.Add(Enums.PaintColor.Purple, 20);
    paintsPrice.Add(Enums.PaintColor.Red, 30);
    paintsPrice.Add(Enums.PaintColor.Violet, 10);
    paintsPrice.Add(Enums.PaintColor.White, 5);
    paintsPrice.Add(Enums.PaintColor.Yellow, 10);
}
```

Fragment kodu odpowiadający za dobieranie pojemników z farbą które są podzielone na 3 rozmiary.

```
private float GetBasePrice(float surfaceSize)
{
    float price = 0;

    //ile 15l pojemnikow mozemy miec
    var bigBuckets = (int)(surfaceSize / 15);

    surfaceSize -= bigBuckets * 15;

    //ile 10l pojemnikow mozemy miec
    var mediumBuckets = (int) (surfaceSize / 10);

    surfaceSize -= mediumBuckets * 10;

    //ile 5l pojemnikow mozemy miec
    var smallBuckets = 0;

    if (surfaceSize > 0)
    {
        if (surfaceSize > 5)
        {
            //przyklad
            //jezeli mamy 7m to lepiej kupic 1x 10l niz 2x 5l
            mediumBuckets++;
        }
        else
        {
            //przyklad:
            //jezeli mamy 2m to musimy kupic pojemnik 5l
            smallBuckets++;
        }
    }

    Console.WriteLine("Wynik: ");
    Console.WriteLine($"Duze pojemniki => {bigBuckets}");
    Console.WriteLine($"Srednie pojemniki => {mediumBuckets}");
    Console.WriteLine($"Male pojemniki => {smallBuckets}");
}
```

Fragment odpowiadający za rozpoczęcie edycji ściany/koloru farby

```
do
{
    Console.Clear();
    Console.WriteLine("Chcesz zmienic jakas sciane? t/n");
    edit = Console.ReadLine();

} while (edit != "t" && edit != "n");

if (edit == "t") mainProgram.ProcessEditRooms(mainProgram.rooms);

for (int i = 0; i < mainProgram.rooms.Count; i++)
{
    Console.WriteLine($"Pokoj {i} ({mainProgram.rooms[i].GetRoomType()}) powierzchnia => {mainProgram.rooms[i].GetSurfaceSize()}");
}

foreach (var room in mainProgram.rooms)
{
    totalPrice += calculator.GetPrice(room, room.GetColorsCount());
}

Console.WriteLine($"Cena => {totalPrice}");

Console.ReadKey();
```

Część kodu odpowiedzialna za wybór ściany oraz akcje chcemy wykonać

```
Console.Clear();
Console.WriteLine("Która ścianę chcesz zmienić?");
for (int i = 0; i < paintAbleWalls.Count; i++)
{
    Console.WriteLine($"{i}");
}

int wallIndex = -1;
while (!int.TryParse(Console.ReadLine(), out wallIndex) || wallIndex < 0 || wallIndex >= paintAbleWalls.Count)
{
    Console.WriteLine("Nie mamy ściany z takim numerem ! Spróbuj ponownie");
}

Console.Clear();
Console.WriteLine("Co chcesz zmienić ?");

foreach (var action in _roomActions)
{
    Console.WriteLine($"{action.Key}. {action.Value}");
}

int actionIndex;
while (!int.TryParse(Console.ReadLine(), out actionIndex) || !_roomActions.ContainsKey(actionIndex))
{
    Console.WriteLine("Taka akcja nie istnieje ! Spróbuj ponownie");
}

switch (_roomActions[actionIndex])
{
    case Enums.RoomAction.EditColor:
    {
        EditColor(paintAbleWalls[wallIndex]);
        break;
    }

    case Enums.RoomAction.EditSize:
    {
        EditSize(paintAbleWalls[wallIndex]);
        SetRoomSurfaceSize();
        break;
    }
}
```

Tutaj mamy podawanie nowych wartości ściany po edycji

```
public void EditSize()
{
    Console.WriteLine("Jaka jest nowa szerokość ściany ?");
    float w = -1;
    while (!float.TryParse(Console.ReadLine(), out w) || w <= 0)
    {
        Console.WriteLine("Szerokość musi być większa niż 0 !");
    }

    Console.WriteLine("Jaka jest nowa wysokość ściany ?");
    float h = -1;
    while (!float.TryParse(Console.ReadLine(), out h) || h <= 0)
    {
        Console.WriteLine("Wysokość musi być większa niż 0 !");
    }

    _width = w;
    _height = h;

    // zliczanie ilości okien
    var windowsOnWall = _windowsOnWall.Count;

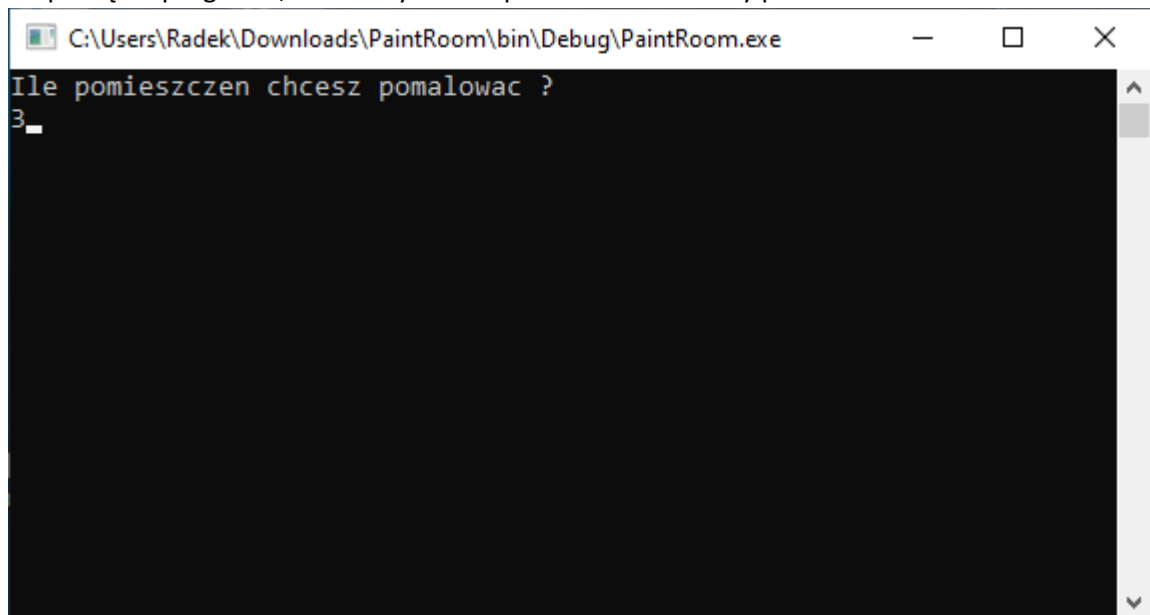
    _windowsOnWall = new List<Window>();

    for (int i = 0; i < windowsOnWall; i++)
    {
        AddWindow();
    }

    SetSurfaceSize();
}
```

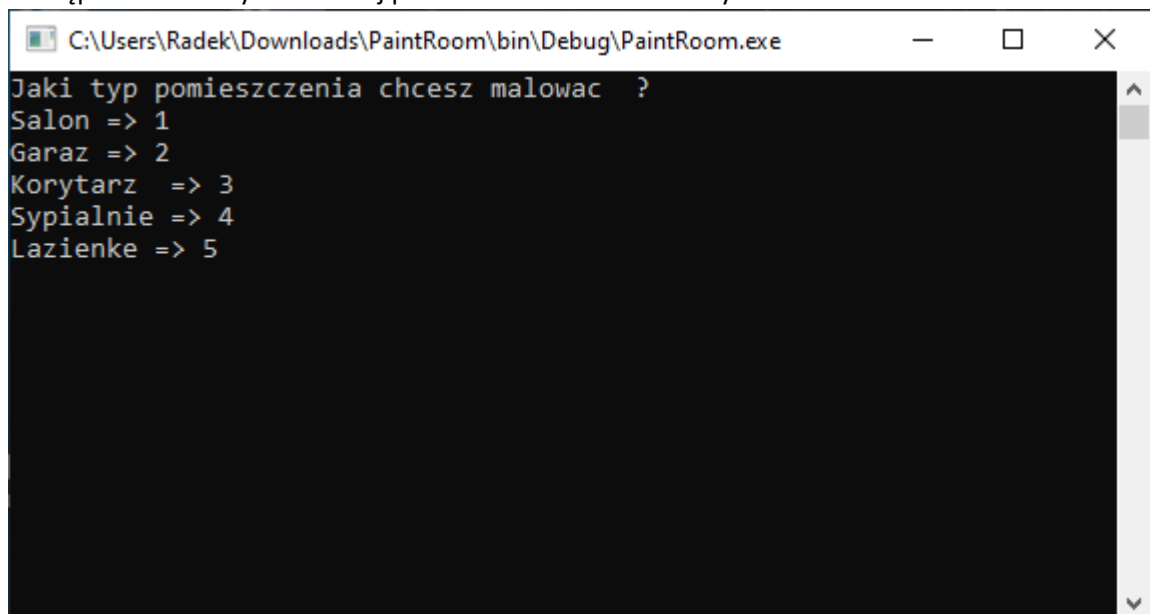
## 5. Instrukcja użytkownika

Na początku programu, trzeba wybrać ile pomieszczeń chcemy pomalować.



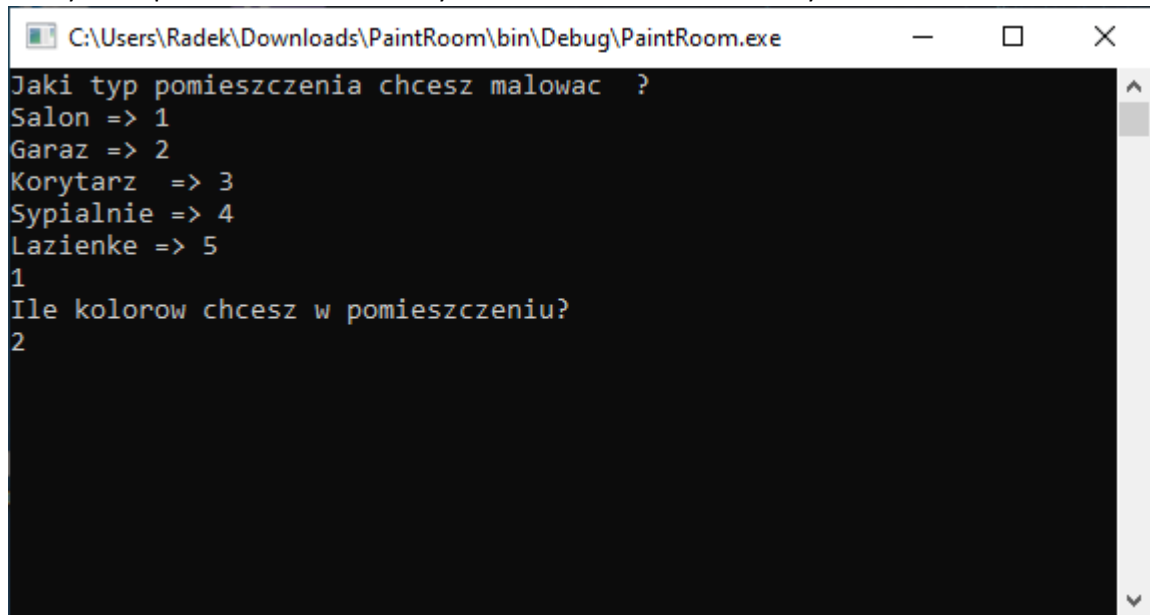
```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Ile pomieszczen chcesz pomalowac ?
3_
```

Następnie trzeba wybrać rodzaj pomieszczenia które chcemy malować.



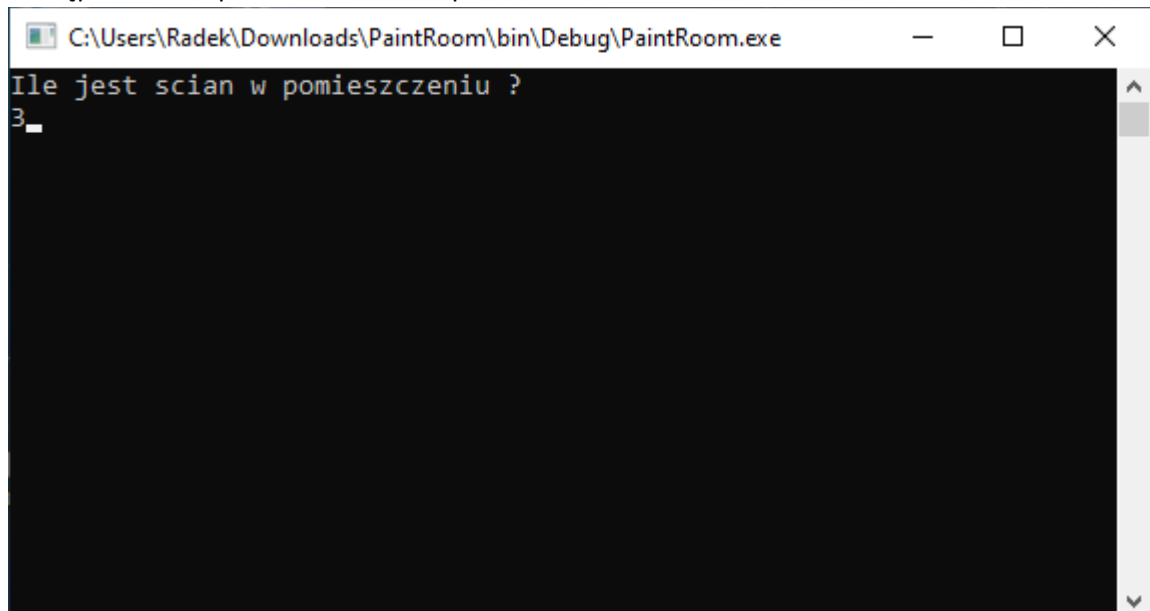
```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Jaki typ pomieszczenia chcesz malowac ?
Salon => 1
Garaz => 2
Korytarz  => 3
Sypialnie => 4
Lazienke => 5
```

Po wyborze pomieszczenia trzeba wybrać ile kolorów w nim chcemy



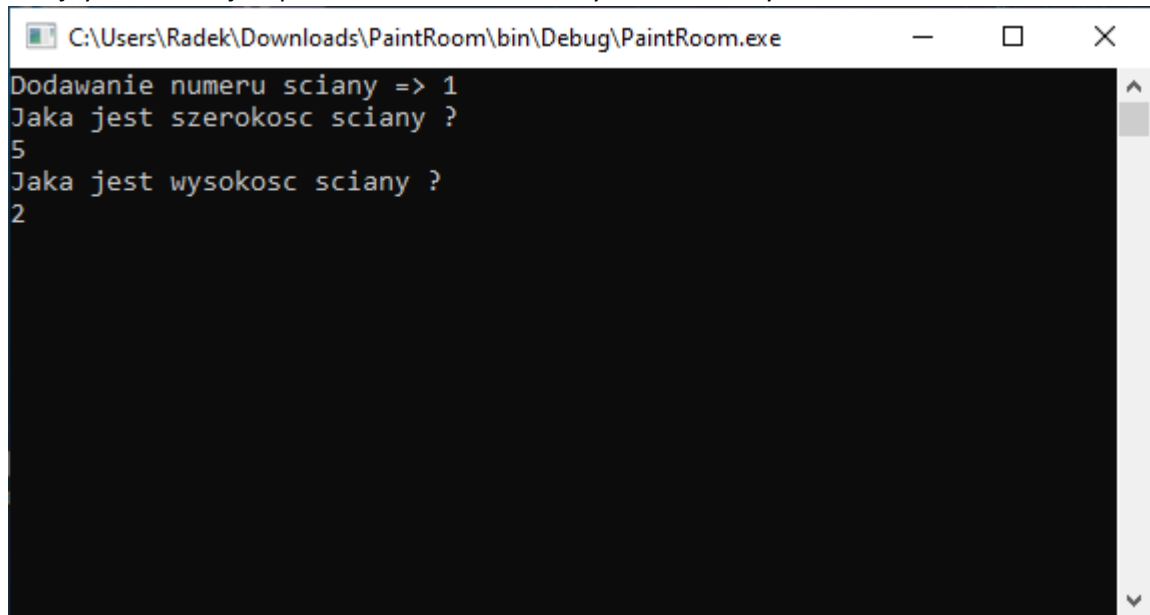
```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Jaki typ pomieszczenia chcesz malowac ?
Salon => 1
Garaz => 2
Korytarz  => 3
Sypialnie => 4
Lazienke => 5
1
Ile kolorow chcesz w pomieszczeniu?
2
```

Następnie trzeba podać ilość ścian w pomieszczeniu



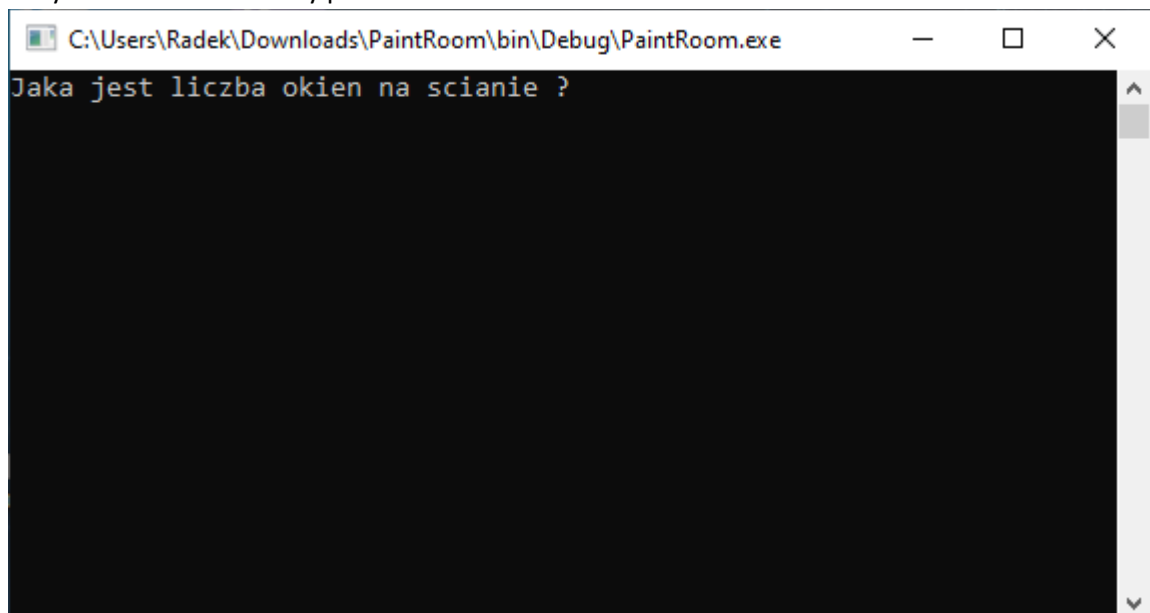
```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Ile jest scian w pomieszczeniu ?
3
```

Kolejnym krokiem jest podanie szerokości oraz wysokości ściany



```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Dodawanie numeru sciany => 1
Jaka jest szerokosc sciany ?
5
Jaka jest wysokosc sciany ?
2
```

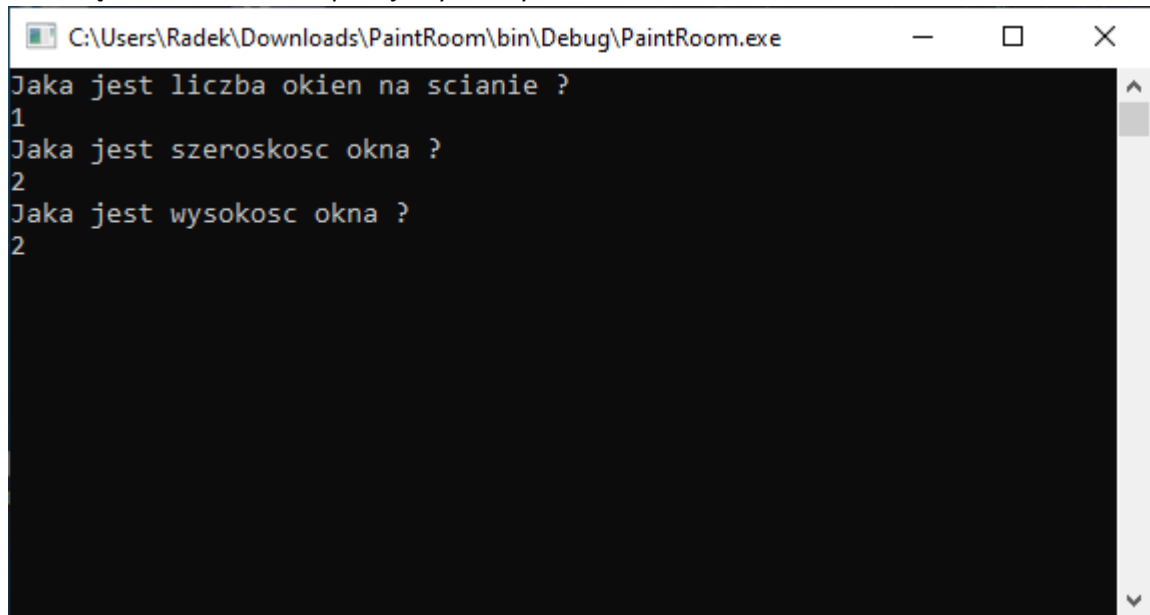
W tym momencie musimy podać ilość okien na ścianie



```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Jaka jest liczba okien na scianie ?
```

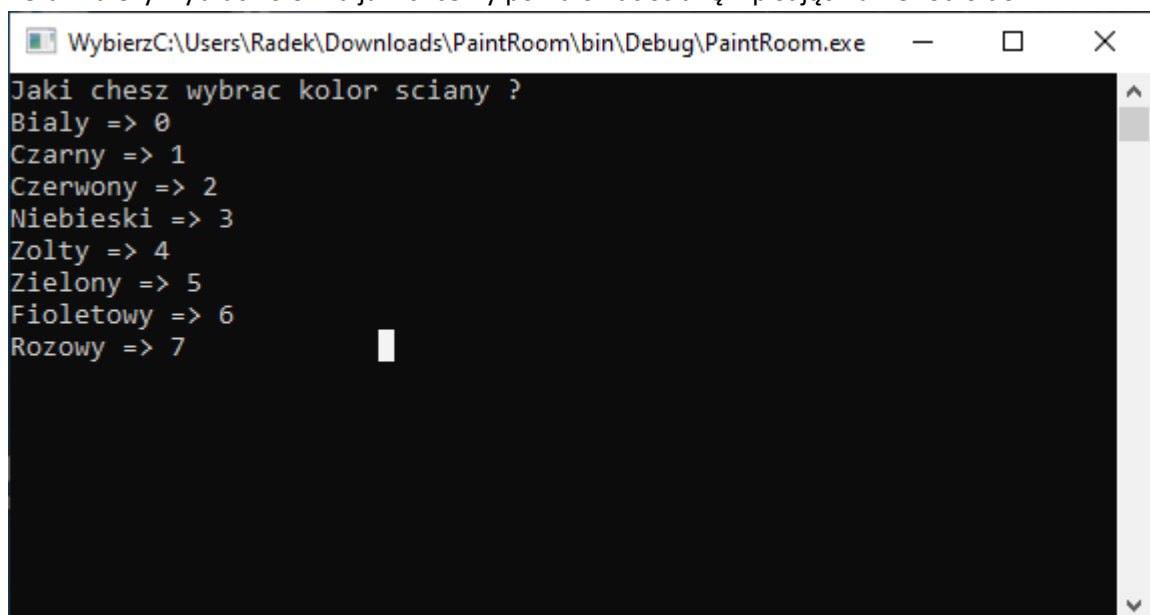


Jeżeli są okna na ścianie to podajemy ich wysokość



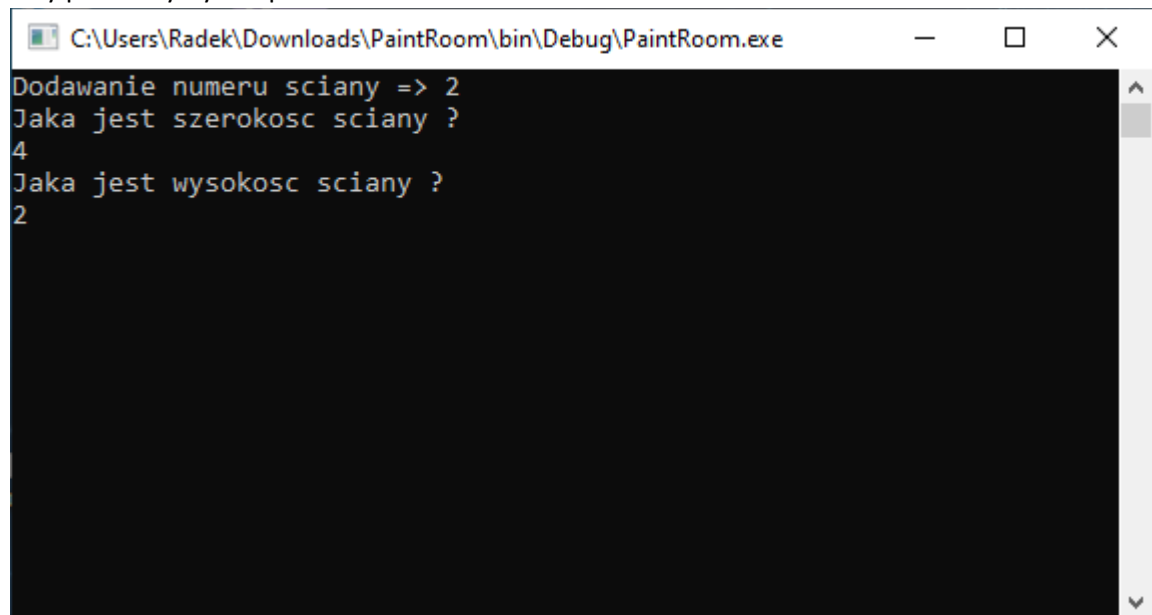
```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Jaka jest liczba okien na scianie ?
1
Jaka jest szerokosc okna ?
2
Jaka jest wysokosc okna ?
2
```

Teraz należy wybrać kolor na jaki chcemy pomalować ścianę wpisując numer od 0 do 7



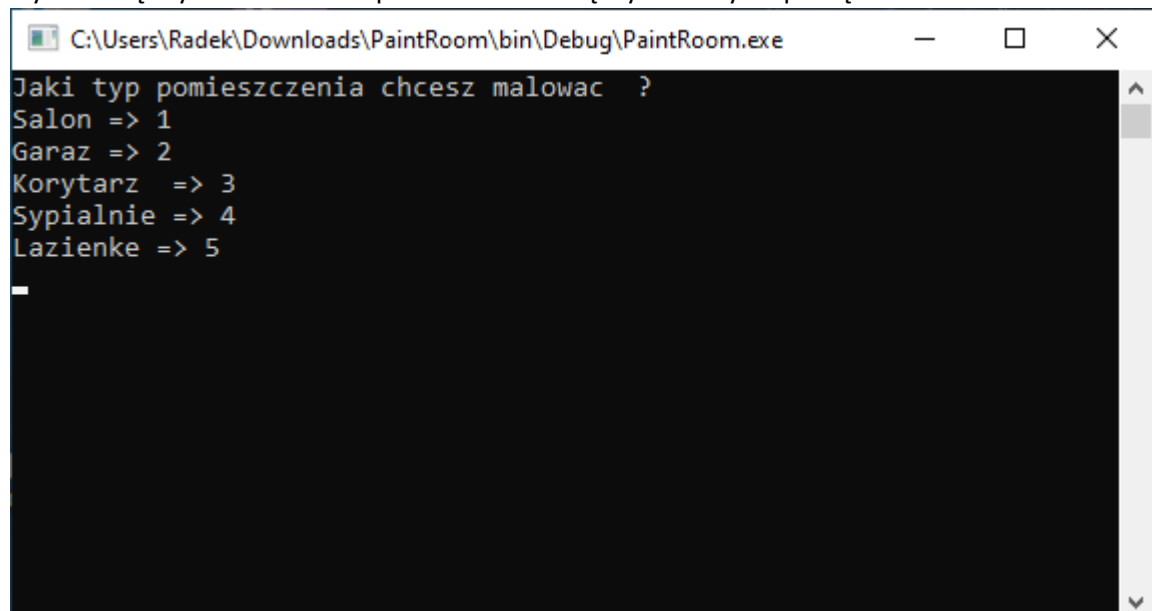
```
WybierzC:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Jaki chesz wybrac kolor sciany ?
Bialy => 0
Czarny => 1
Czerwony => 2
Niebieski => 3
Zolty => 4
Zielony => 5
Fioletowy => 6
Rozowy => 7
```

Teraz dochodzimy do momentu gdzie ponownie wprowadzamy wymiary ścian itp. Cała ta procedura powtarza się dla wszystkich ścian które podaliśmy np. ilość ścian wynosi 4 zatem 4 razy powtórzmy ten proces.



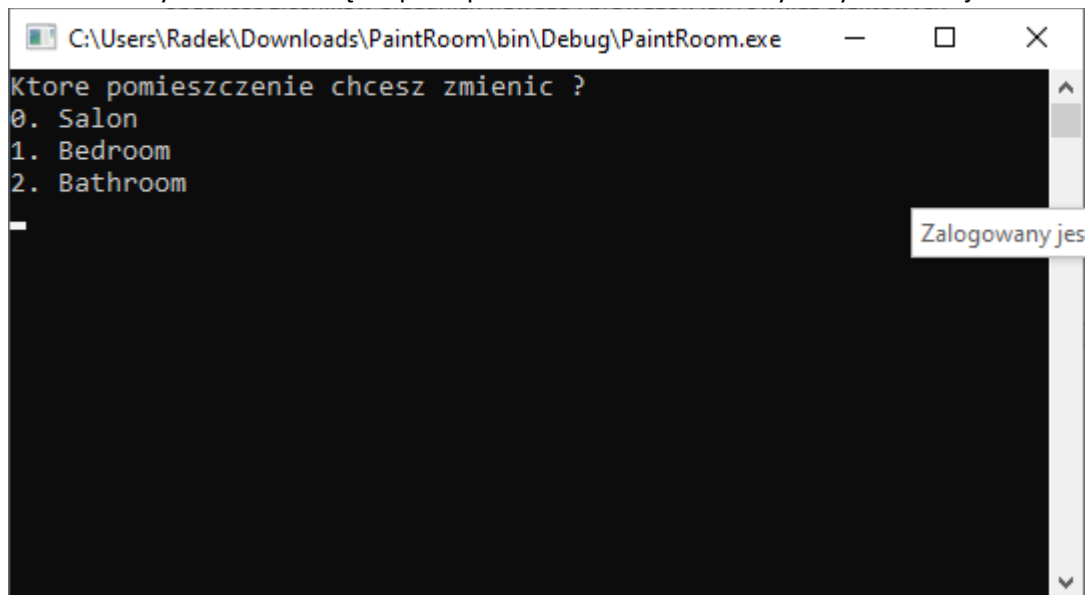
```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Dodawanie numeru sciany => 2
Jaka jest szerokosc sciany ?
4
Jaka jest wysokosc sciany ?
2
```

Po podaniu wartości dla wszystkich ścian, ponownie wybieramy typ malowanego pomieszczenia i powtarza się procedura wymiarowania ścian wraz z wyborem koloru. Trzeba wykonać tę czynność dla ilości pomieszczeń którą wybraliśmy na początku.



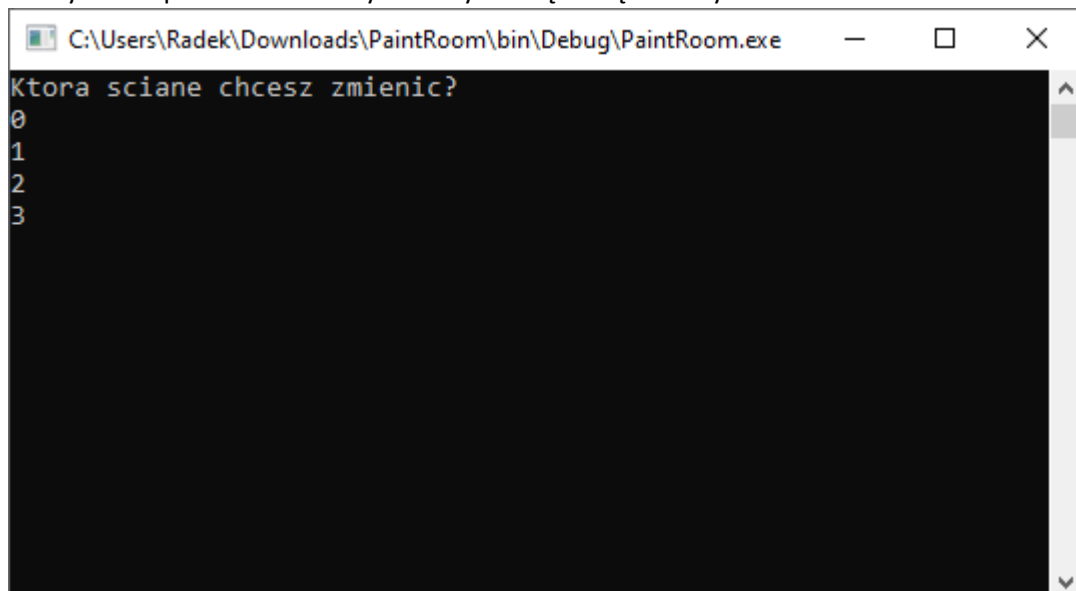
```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Jaki typ pomieszczenia chcesz malowac ?
Salon => 1
Garaz => 2
Korytarz => 3
Sypialnie => 4
Lazienke => 5
_
```

Po podaniu wszystkich wymiarów mamy możliwość dokonania zmian w pomieszczeniach. Aby dokonać zmian trzeba wpisać „t” jeżeli nie to „n”. Jeżeli wybraliśmy, że chcemy coś zmienić to wyświetla nam się rozpiska pomieszczeń które stworzyliśmy wcześniej.



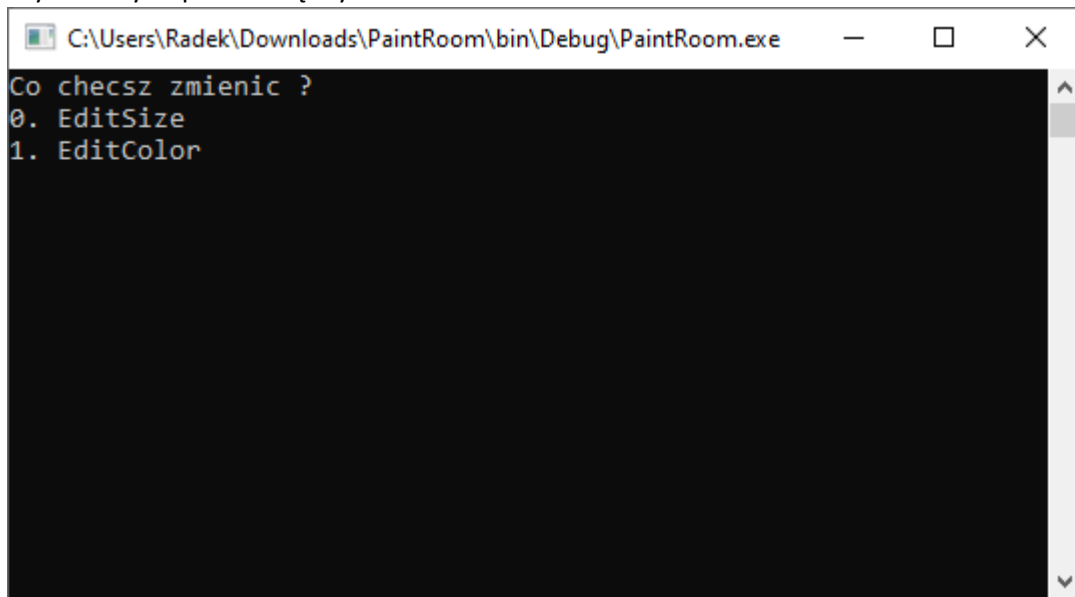
```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Ktore pomieszczenie chcesz zmienic ?
0. Salon
1. Bedroom
2. Bathroom
_
```

Po wybraniu pomieszczenia wybieramy ścianę którą chcemy zmienić.



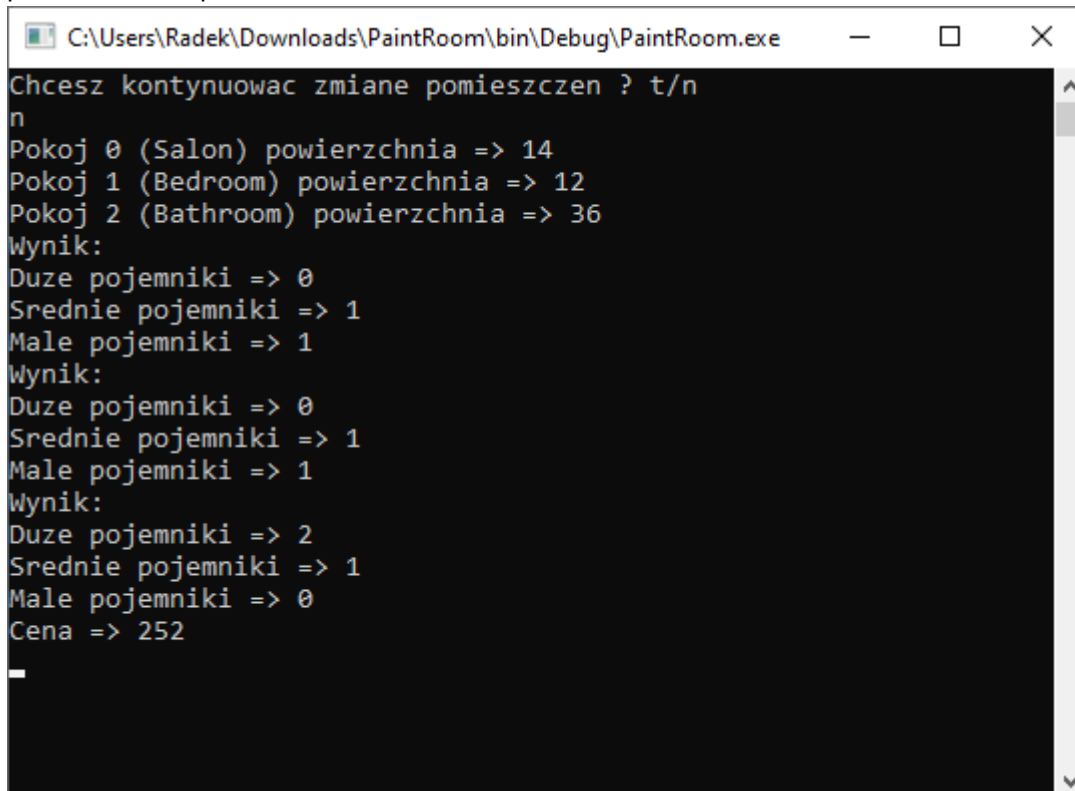
```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Ktore sciane chcesz zmienic?
0
1
2
3
```

Następnie musimy wybrać co chcemy zmienić w wybranej ścianie rozmiar czy kolor.  
Wybieramy odpowiednią czynność.



```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Co chcesz zmienić ?
0. EditSize
1. EditColor
```

Po edycji ponownie jesteśmy pytani o to czy chcemy coś zmienić jeżeli odpowiemy, że nie przenosi nas do podsumowania.



```
C:\Users\Radek\Downloads\PaintRoom\bin\Debug\PaintRoom.exe
Chcesz kontynuować zmianę pomieszczenia ? t/n
n
Pokoj 0 (Salon) powierzchnia => 14
Pokoj 1 (Bedroom) powierzchnia => 12
Pokoj 2 (Bathroom) powierzchnia => 36
Wynik:
Duze pojemniki => 0
Srednie pojemniki => 1
Male pojemniki => 1
Wynik:
Duze pojemniki => 0
Srednie pojemniki => 1
Male pojemniki => 1
Wynik:
Duze pojemniki => 2
Srednie pojemniki => 1
Male pojemniki => 0
Cena => 252
-
```

## 6. Podsumowanie

Wykonany projekt programu liczącego ilości farby do malowania pomieszczeń spełnia zakładane założenia projektowe. Program rozróżnia różne powierzchnie ścian, została także zastosowana funkcja uwzględniająca okna. Funkcje wyliczające ilości i wielkości puszek farby także zostały uwzględnione i spełniają wcześniej zaimplementowane założenia projektowe.