Министерство образования и науки Российской Федерации

Федеральное государственное образовательное учреждение высшего образования

«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра программной инженерии

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Отчёт

По лабораторной работе №2

“Структуры языка C”

по дисциплине “Программирование”

Студент группы ПИ-92 А.П. Красников

Преподаватель Троицкий В.С.

Барнаул 2020

## Задача

Сделать структуру похожую на [Vector2](https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.vector2).

Реализовать функции для: вычисления суммы векторов, нахождения длины вектора, угла между векторами, скалярного произведения векторов.

## Код

<https://github.com/AVIX8/LR2>

### Vector2.h:

#ifndef VECTOR\_h\_

#define VECTOR\_h\_

#include <stdio.h>

#include <malloc.h>

#include <math.h>

#include "Window.h"

typedef struct

{

    int x;

    int y;

    int \*color;

} Vector2;

void Vector2Init(Vector2 \*newVector, int x, int y, int \*color)

{

    newVector->x = x;

    newVector->y = y;

    newVector->color = color;

}

void Vector2Read(Vector2 \*vector)

{

    while (

        printf("Enter x and y: "),

        fflush(stdin),

        2 != scanf("%d%d", &vector->x, &vector->y) &&

        printf("[!] You have entered incorrect data, please try again.\n"));

    fflush(stdin);

}

double Length(Vector2 \*vector)

{

    return sqrt(pow(vector->x, 2) + pow(vector->y, 2));

}

void Display(Vector2 \*vector)

{

    printf("<Vector2 (x:%d, y:%d), Length: %lf>\n", vector->x, vector->y, Length(vector));

}

double Scalar(Vector2 \*a, Vector2 \*b)

{

    return a->x \* b->x + a->y \* b->y;

}

double Angle(Vector2 \*a, Vector2 \*b)

{

    double ang = acos(Scalar(a, b) / Length(a) / Length(b)) / M\_PI \* 180;

    return fmin(ang, 360 - ang);

}

Vector2 Add(Vector2 \*a, Vector2 \*b)

{

    Vector2 newVector;

    newVector.x = a->x + b->x;

    newVector.y = a->y + b->y;

    return newVector;

}

#endif

### Window.h:

#ifndef WINDOW\_h\_

#define WINDOW\_h\_

#include <stdio.h>

#include "SDL2/SDL.h"

#include "Vector2.h"

SDL\_Window \*WindowInit()

{

    if (SDL\_Init(SDL\_INIT\_EVERYTHING) != 0)

    {

        printf("SDL\_Error: %s\n", SDL\_GetError());

    }

    SDL\_Window \*window = SDL\_CreateWindow("DA", SDL\_WINDOWPOS\_CENTERED, 25, 501, 501, 0);

    SDL\_CreateRenderer(window, -1, SDL\_RENDERER\_ACCELERATED);

    return window;

}

void WindowClear(SDL\_Window \*window)

{

    SDL\_Renderer \*renderer = SDL\_GetRenderer(window);

    SDL\_SetRenderDrawColor(renderer, 0, 0, 0, 255);

    SDL\_RenderClear(renderer);

    SDL\_SetRenderDrawColor(renderer, 96, 96, 96, 255);

    for (int x = 0; x <= 500; x += 25)

    {

        SDL\_RenderDrawLine(renderer, x, 0, x, 500);

    }

    for (int y = 0; y <= 500; y += 25)

    {

        SDL\_RenderDrawLine(renderer, 0, y, 500, y);

    }

    SDL\_SetRenderDrawColor(renderer, 192, 192, 192, 255);

    SDL\_RenderDrawLine(renderer, 250, 0, 250, 500);

    SDL\_RenderDrawLine(renderer, 0, 250, 500, 250);

    SDL\_RenderPresent(renderer);

}

void DrawVector2(SDL\_Window \*window, Vector2 \*vector)

{

    SDL\_Renderer \*renderer = SDL\_GetRenderer(window);

    SDL\_SetRenderDrawColor(renderer, vector->color[0], vector->color[1], vector->color[2], 255);

    SDL\_RenderDrawLine(renderer, 250, 250, 250 + vector->x \* 25, 250 + vector->y \* -25);

    SDL\_RenderPresent(renderer);

}

void Quit(SDL\_Window \*window)

{

    SDL\_DestroyWindow(window);

    SDL\_Quit();

}

#endif

### main.c:

#include "SDL2/SDL.h"

#include "Window.h"

#include "Vector2.h"

static int GREEN[3] = {0, 255, 0};

static int RED[3] = {255, 0, 0};

static int WHITE[3] = {255, 255, 255};

static int Script(void \*window)

{

    while (1)

    {

        WindowClear(window);

        // статическая переменная

        Vector2 v1;

        Vector2Read(&v1);

        printf("v1 = ");

        Display(&v1);

        v1.color = GREEN;

        DrawVector2(window, &v1);

        // динамическая переменная

        Vector2 \*v2 = (Vector2 \*)malloc(sizeof(Vector2));

        Vector2Init(v2, 0, 0, WHITE);

        Vector2Read(v2);

        printf("v2 = ");

        Display(v2);

        v2->color = GREEN;

        DrawVector2(window, v2);

        printf("Angle between v1 and v2: %f\n", Angle(&v1, v2));

        Vector2 sum = Add(&v1, v2);

        printf("v1 + v2 = ");

        Display(&sum);

        sum.color = RED;

        DrawVector2(window, &sum);

        printf("\n");

        system("pause");

        system("cls");

    }

    return 0;

}

int main(int argc, char \*argv[])

{

    SDL\_Window \*window = WindowInit();

    SDL\_Thread \*thread = SDL\_CreateThread(Script, "Script", (void \*)window);

    SDL\_Event event;

    while (SDL\_WaitEvent(&event) >= 0)

    {

        switch (event.type)

        {

        case SDL\_QUIT:

        {

            Quit(window);

            exit(0);

        }

        break;

        }

    }

    return 0;

}

### 

### 

### 

### 