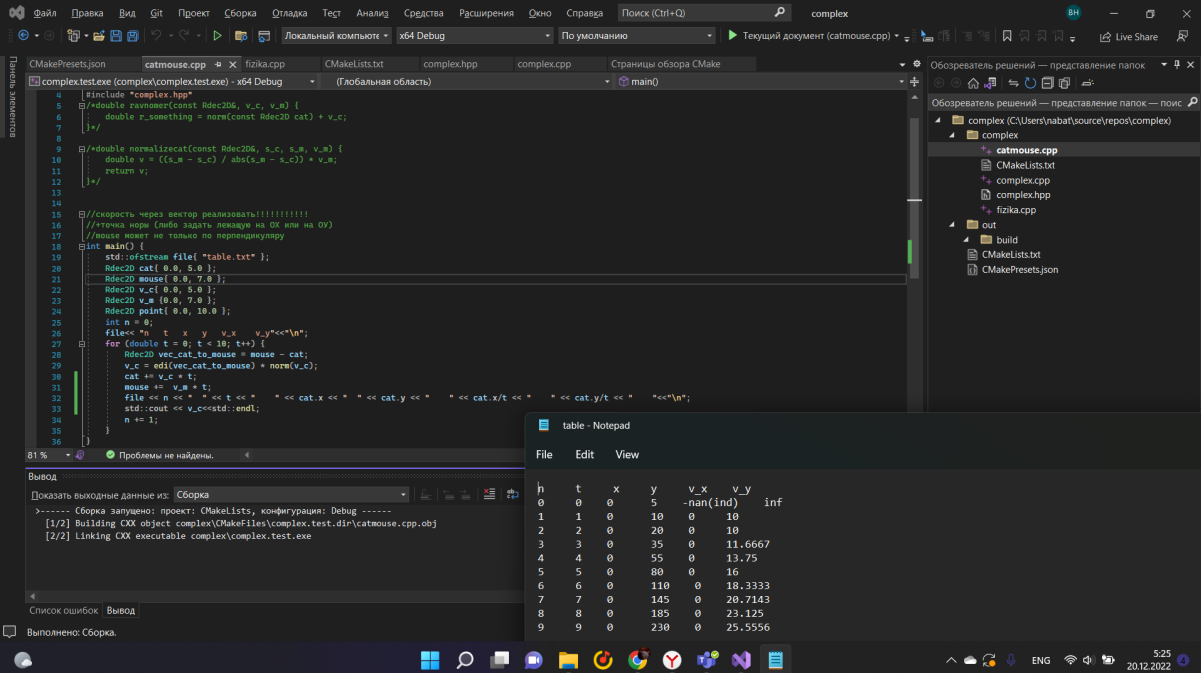


Кошки-мышки

По результатам исследования должна выполняться следующая задача: Используя реализацию библиотеки для работы с векторами написать консольное приложение, для оценки параметров начальной скорости кошки, если место встречи - норка.

В ходе решения задачи скорость кошки менялась в зависимости от положения мышки. Также, как и положение самой кошки. Движение направлено в сторону движения мышки. Мной была написана программа, выполняющая задачу и результаты были выведены в файл в табличном виде:



```
1 #include <complex.h>
2 double random(const RVec2D, v, s, v_m) {
3     double r_something = norm(const RVec2D cat) + v;
4 }
5
6 double normalizecat(const RVec2D, s, s_m, v_m) {
7     double v = ((s_m - s_c) / abs(s_m - s_c)) * v_m;
8     return v;
9 }
10
11 //скорость через вектор реализовать!!!!!!
12 //точка норы (либо задать левую на 0x или на 0y)
13 //mouse может не только по перпендикуляр
14
15 int main() {
16     std::ofstream file("table.txt");
17     RVec2D cat(0.0, 5.0);
18     RVec2D mouse(0.0, 7.0);
19     RVec2D v_c(0.0, 5.0);
20     RVec2D v_m(0.0, 7.0);
21     RVec2D point(0.0, 10.0);
22     int n = 0;
23     while (1) {
24         for (double t = 0; t < 10; t++) {
25             RVec2D vec_cat_to_mouse = mouse - cat;
26             v_c = ediv(vec_cat_to_mouse) * norm(v_c);
27             cat += v_c * t;
28             mouse += v_m * t;
29             file << n << " " << t << " " << cat.x << " " << cat.y << " " << cat.x/t << " " << cat.y/t << " " << "\n";
30             std::cout << v_c << std::endl;
31             n++;
32         }
33     }
34 }
```

n	t	x	y	v_x	v_y
0	0	0	5	-nan(ind)	inf
1	1	0	10	0	10
2	2	0	20	0	10
3	3	0	35	0	11.6667
4	4	0	55	0	13.75
5	5	0	80	0	16
6	6	0	110	0	18.3333
7	7	0	145	0	20.7143
8	8	0	185	0	23.125
9	9	0	230	0	25.5556

Колебания с затуханием

По результатам исследования должна выполняться следующая задача: моделирования движения материальной точки на плоскости, определяемое соотношением

$$\ddot{\vec{r}} = -k(\vec{r} - \vec{r}_0) - \mu \frac{\dot{\vec{r}}}{|\dot{\vec{r}}|}$$

В ходе решения задачи нужно было смоделировать маятник с затухающими колебаниями, результат вывести. Мной была написана программа, выполняющая задачу и результаты были выведены в файл в табличном виде:

```
1 #include <iostream>
2 #include <cmath>
3 #include <fstream>
4 #include "complex.hpp"
5 int main() {
6     std::ofstream file("fiziktable.txt");
7     RVec2D body{ 2.3, 5.6 };
8     RVec2D v_body{ 3.0, 9.7 };
9     double k = 2.0;
10    double m = 4.6;
11    double mas = 5.0;
12    file << "t x y v_x v_y a_x a_y" << "\n";
13    int n = 0;
14    for (int t = 0; t < 10; t++) {
15        RVec2D rx = (body - (body + v_body * t)) * (-k);
16        RVec2D rt = (edl(body)) * (-m);
17        RVec2D r = rx + rt;
18        file << n << " " << t << " " << r.x << " " << r.y << " " << v_body.x << " " << v_body.y << " " << r.x / mas << " " << r.y / mas << "\n";
19        n++;
20    }
21 }
```

n	t	x	y	v_x	v_y	a_x	a_y
0	0	-1.74763	-4.25509	3	9.7	-0.349525	-0.851018
1	1	7.25237	24.8449	3	9.7	1.45047	4.96898
2	2	16.2524	53.9449	3	9.7	3.25047	10.789
3	3	25.2524	83.0449	3	9.7	5.05047	16.609
4	4	34.2524	112.145	3	9.7	6.85047	22.429
5	5	43.2524	141.245	3	9.7	8.65047	28.249
6	6	52.2524	170.345	3	9.7	10.4505	34.069
7	7	61.2524	199.445	3	9.7	12.2505	39.889
8	8	70.2524	228.545	3	9.7	14.0505	45.709
9	9	79.2524	257.645	3	9.7	15.8505	51.529