

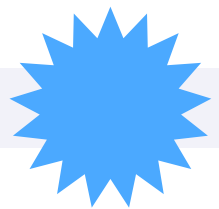
Программирование на C++



Минцифры
России

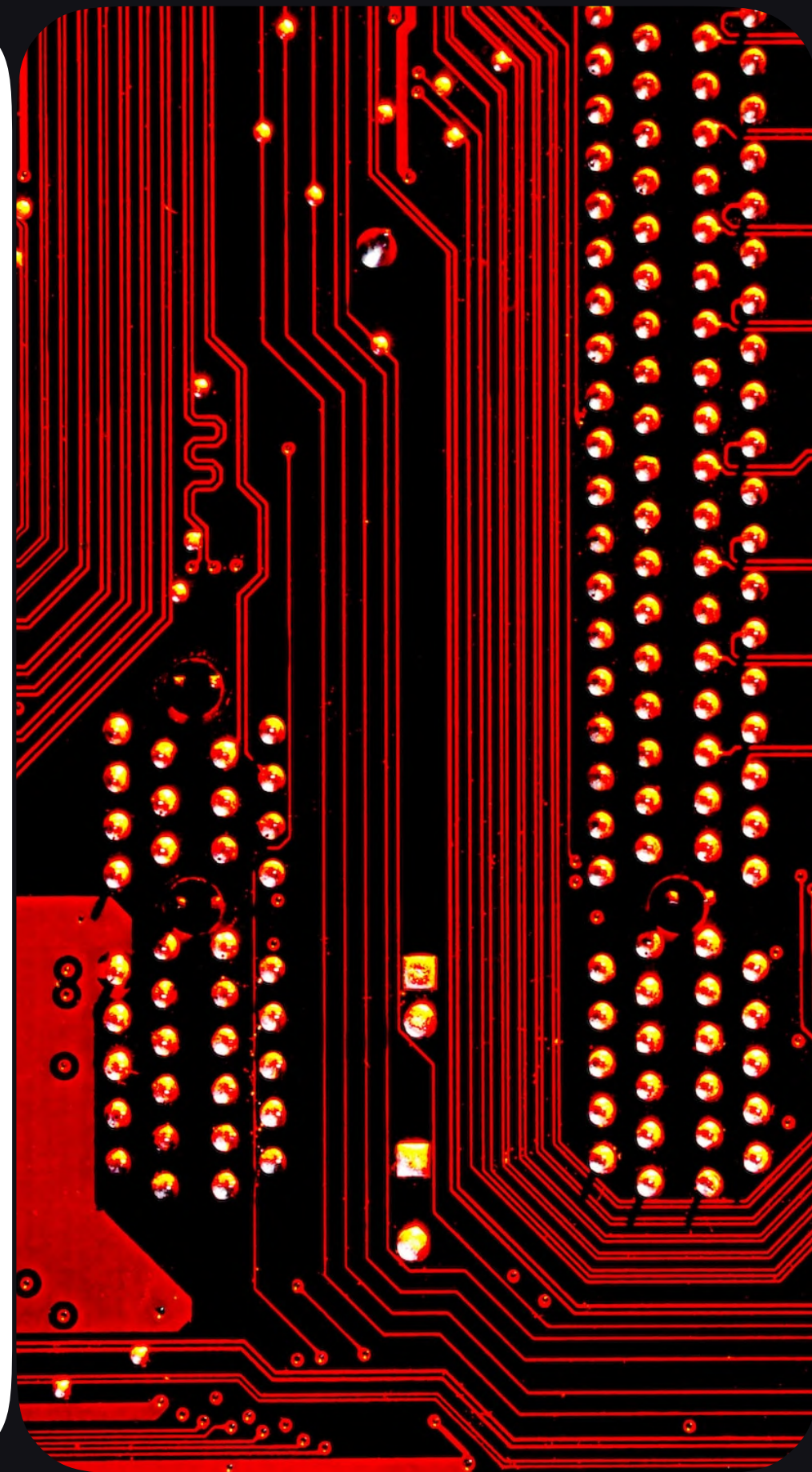
UCHi **DOMA**

20.35
УНИВЕРСИТЕТ

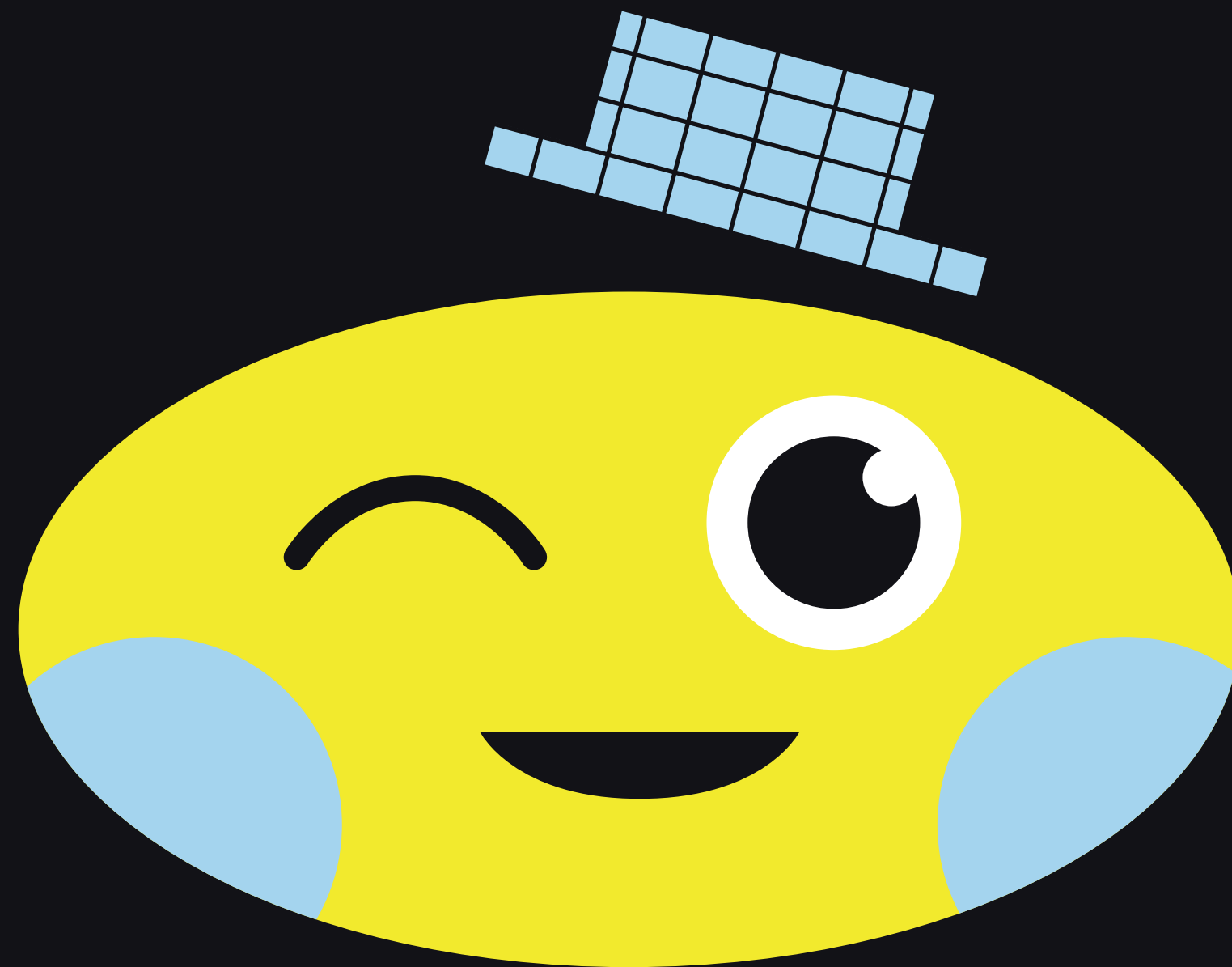


Модуль 2. Урок 2

Локальные и глобальные переменные



Привет!



проверка готовности



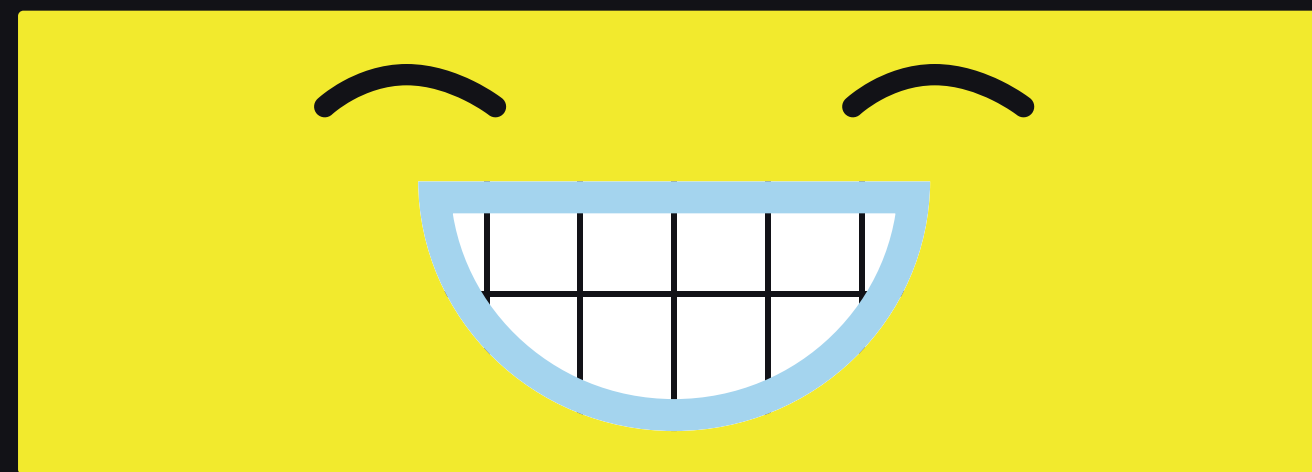
Видим и слышим друг друга без помех



Не опаздываем и не отвлекаемся



Сидим прямо



Улыбаемся, если всё ок

Как домашка?



Какие были трудности?



Какие остались вопросы?



Сколько заданий выполнено?



Разомнёмся



Что будет выведено на экран в результате работы программы если ввести 3?

```
1  #include <stdio.h>
2  int func(int a)
3  {
4      int sum=0;
5      for (int i=1;i<=a;i++)
6          sum+=i;
7      return sum;
8  }
9  int main()
10 {
11     int a=0;
12     scanf("%d",&a);
13     printf("%d\n",func(a));
14     return 0;
15 }
```

Разомнёмся



Что будет выведено на экран в результате работы программы если ввести 3?

```
1  #include <stdio.h>
2  int func(int a)
3  {
4      int sum=0;
5      for (int i=1;i<=a;i++)
6          sum+=i;
7      return sum;
8  }
9  int main()
10 {
11     int a=0;
12     scanf("%d",&a);
13     printf("%d\n",func(a));
14     return 0;
15 }
```

Результат работы программы:

3
6

Разомнёмся



Что будет выведено на экран в результате работы программы если ввести 3?

```
1  #include <stdio.h>
2  int func(int n, int a)
3  {
4      int sum=1;
5      for (int i=0;i<n;i++)
6          sum*=a;
7      return sum;
8  }
9  int main()
10 {
11     int n;
12     scanf("%d",&n);
13     printf("%d\n",func(n,3)-func(n,2));
14     return 0;
15 }
```


Разомнёмся



Что будет выведено на экран в результате работы программы если ввести 3?

```
1  #include <stdio.h>
2  int func(int n, int a)
3  {
4      int sum=1;
5      for (int i=0;i<n;i++)
6          sum*=a;
7      return sum;
8  }
9  int main()
10 {
11     int n;
12     scanf("%d",&n);
13     printf("%d\n",func(n,3)-func(n,2));
14     return 0;
15 }
```

Результат работы программы:

3
19

Задача



Что будет выведено на экран в результате работы программы?

```
1  #include <stdio.h>
2  void func()
3  {
4      int a=1;
5  }
6  int main()
7  {
8      int a=0;
9      func();
10     printf("%d\n",a);
11     return 0;
12 }
```



Задача



Что будет выведено на экран в результате работы программы?

```
1  #include <stdio.h>
2  void func()
3  {
4      int a=1;
5  }
6  int main()
7  {
8      int a=0;
9      func();
10     printf("%d\n",a);
11     return 0;
12 }
```

Результат работы программы:

0



Задача



Что будет выведено на экран в результате работы программы?

```
1  #include <stdio.h>
2  void func()
3  {
4      int a=1;
5      printf("%d\n",a);
6  }
7  int main()
8  {
9      int a=0;
10     func();
11     return 0;
12 }
```



Задача



Что будет выведено на экран в результате работы программы?

```
1  #include <stdio.h>
2  void func()
3  {
4      int a=1;
5      printf("%d\n",a);
6  }
7  int main()
8  {
9      int a=0;
10     func();
11     return 0;
12 }
```

Результат работы программы:

1



Задача



Почему разный результат?

```
1  #include <stdio.h>
2  void func()
3  {
4      int a=1;
5      printf("%d\n",a);
6  }
7  int main()
8  {
9      int a=0;
10     func();
11     return 0;
12 }
```

1

```
1  #include <stdio.h>
2  void func()
3  {
4      int a=1;
5  }
6  int main()
7  {
8      int a=0;
9      func();
10     printf("%d\n",a);
11     return 0;
12 }
```

0



Цели урока



Изучить различия локальных и глобальных переменных



Отработать на практике составление алгоритмов с хранением данных в локальных и глобальных переменных на Си



Область видимости

Область видимости объекта (переменной или функции) определяет набор функций или модулей, внутри которых допустимо использование имени этого объекта.



Область видимости объекта начинается в точке объявления объекта.

Локальные и глобальные переменные

1

Глобальные

Объявляются вне функции, доступны (видимы) во всем файле, в котором они объявлены. В течение всего времени выполнения программы с глобальным объектом ассоциирована некоторая ячейка памяти.

2

Локальные

Объявляются внутри блока или функции, доступны только внутри того блока, в котором они объявлены. Локальным переменным выделяется новая ячейка памяти каждый раз при осуществлении описания внутри блока. Когда выполнение блока завершается, память, выделенная под локальную переменную освобождается, и переменная теряет своё значение.

Пример локальной переменной



```
1  #include <stdio.h>
2  void func(void)
3  {
4      int k = 1;    // локальный объект
5      printf(" \n k = %d ", k);
6      k = k + 1;
7  }
8  int main()
9  {
10     for (int i = 0; i <= 5; i++) // область видимости i - цикл
11         func();
12     return 0;
13 }
```



Область видимости локальной переменной **k** — функция **func()**.



Область видимости локальной переменной **i** — функция **for()**.

Пример локальной переменной



Каждый раз при входе в функцию с идентификатором **k** ассоциируется некоторая ячейка памяти, в которую помещается значение равное 1.

Результат выполнения программы:

```
k = 1  
k = 1  
k = 1  
k = 1  
k = 1  
k = 1
```

Пример глобальной переменной



```
1  #include <stdio.h>
2  int k = 1;    // глобальный объект
3  void func(void)
4  {
5      printf(" \n k = %d ", k);
6      k = k + 1;
7  }
8  int main()
9  {
10     for (int i = 0; i <= 5; i++) // область видимости i - цикл
11         func();
12     return 0;
13 }
```



Глобальная переменная `k` объявлена не в функции и имеет область видимости — весь файл программы.

Пример глобальной переменной



Результат работы программы:

```
k = 1
```

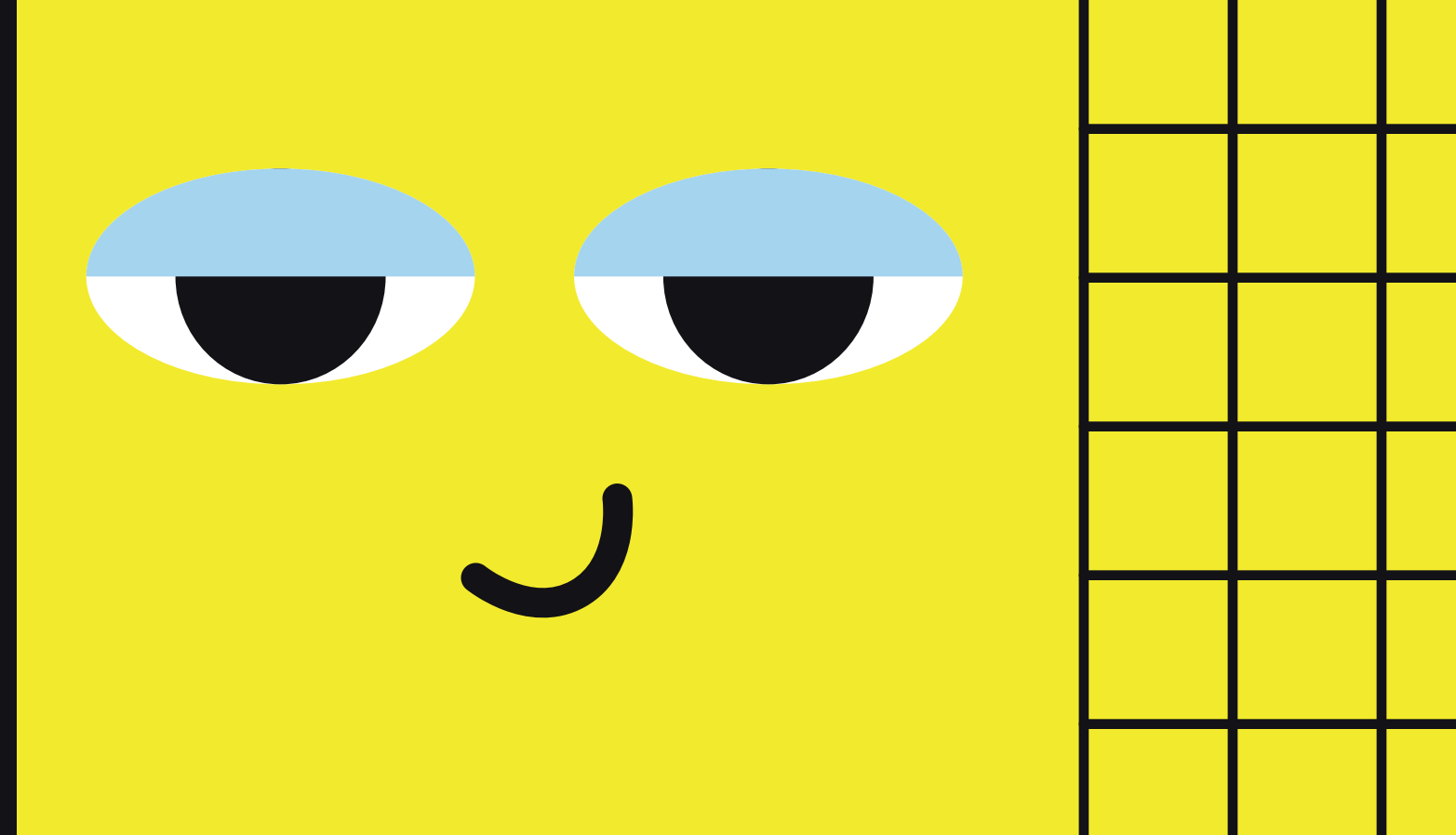
```
k = 2
```

```
k = 3
```

```
k = 4
```

```
k = 5
```

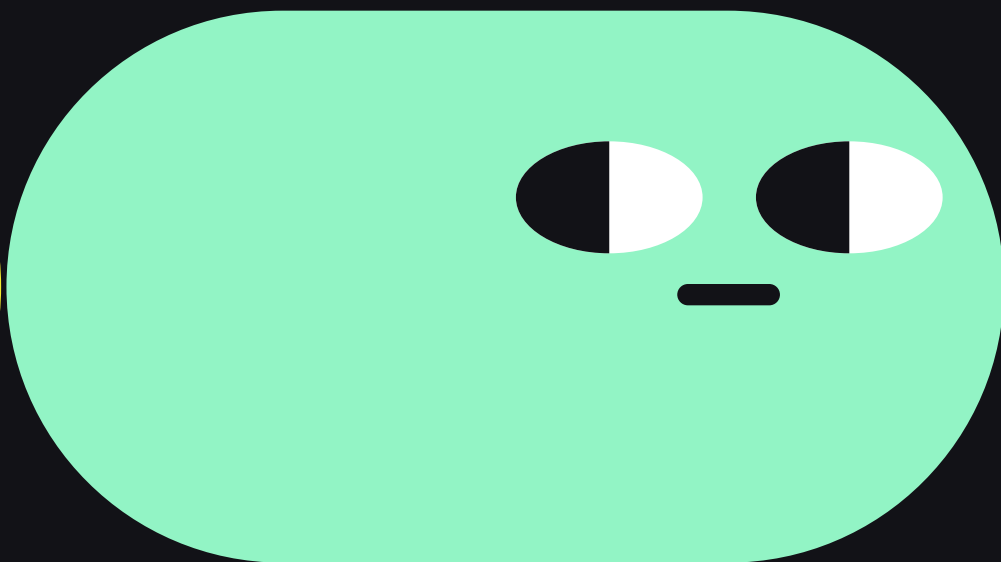
```
k = 6
```



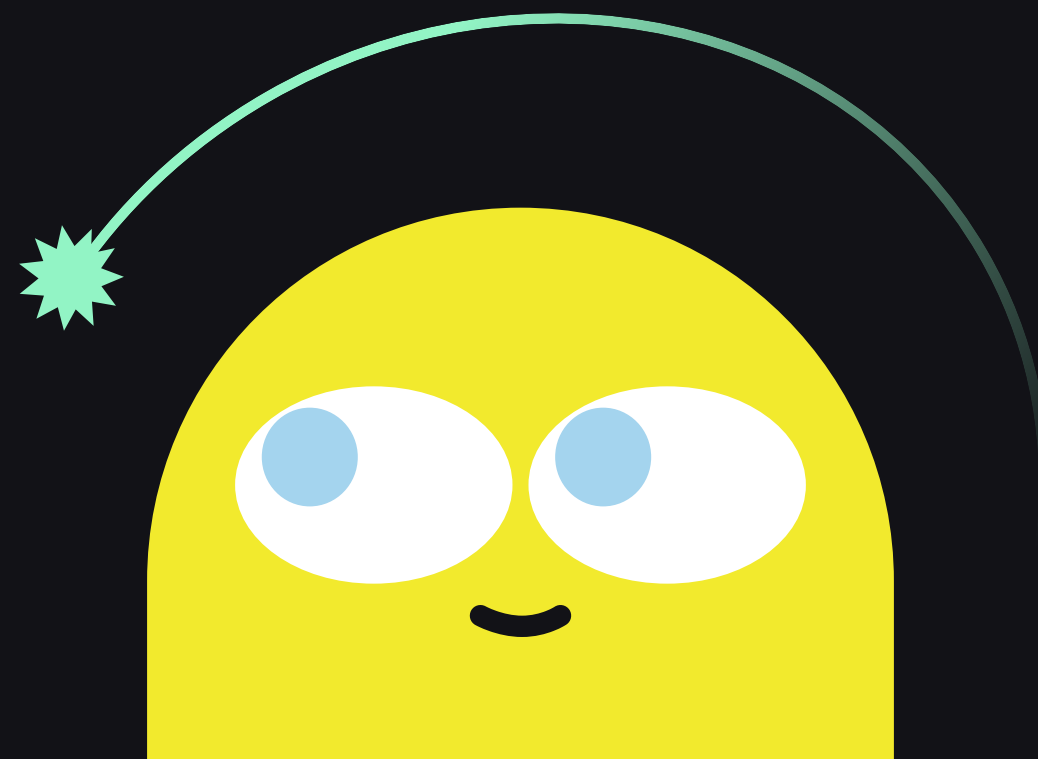
Практика

перерыв

физкультминутка



Смотрим вверх–вниз, вправо–влево



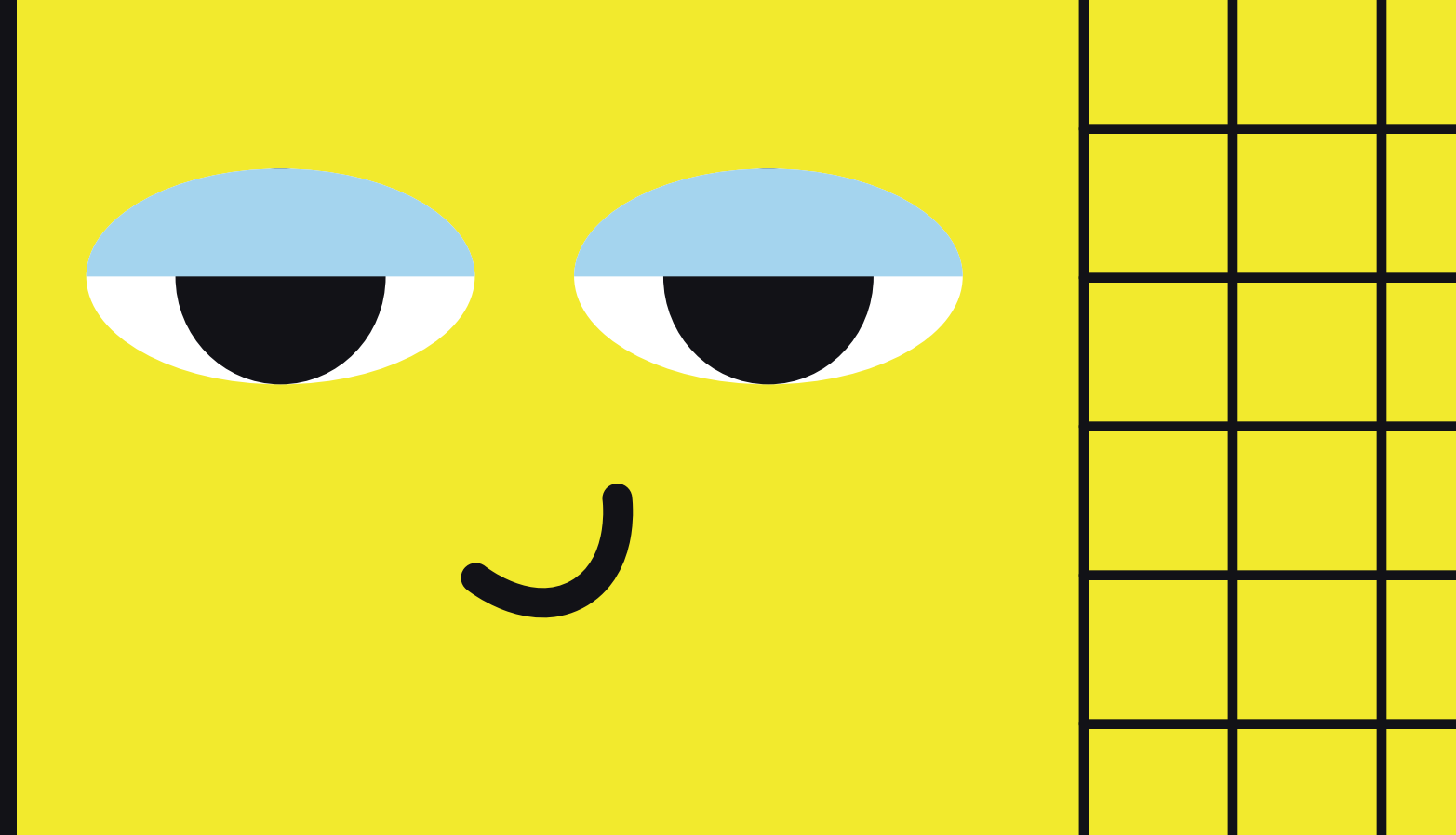
Вращаем по кругу туда–обратно



Крепко зажимаемся



Быстро моргаем



Практика

1

Закрепление

Что будет выведено на экран в результате работы программы если ввести 2, а затем 3?

```
1  #include <stdio.h>
2  int x, y, z;
3  void sum(void)
4  {
5      z = x + y;
6  }
7  int main()
8  {
9      printf("x= ");
10     scanf("%d", &x);
11     printf("y= ");
12     scanf("%d", &y);
13     sum();
14     printf("z= %d", z);
15     return 0;
16 }
```

1

Закрепление

Что будет выведено на экран в результате работы программы если ввести 2, а затем 3?

Результат работы программы:

x= 2
y= 3
z= 5

```
1  #include <stdio.h>
2  int x, y, z;
3  void sum(void)
4  {
5      z = x + y;
6  }
7  int main()
8  {
9      printf("x= ");
10     scanf("%d", &x);
11     printf("y= ");
12     scanf("%d", &y);
13     sum();
14     printf("z= %d", z);
15     return 0;
16 }
```

2

Закрепление

Что будет выведено на экран в результате работы программы если ввести 2, а затем 3?

```
1  #include <stdio.h>
2  int x=0, y=0, z=0;
3  void sum(void)
4  {
5      int x, y, z;
6      z = x + y;
7  }
8  int main()
9  {
10     printf("x= ");
11     scanf("%d", &x);
12     printf("y= ");
13     scanf("%d", &y);
14     sum();
15     printf("z= %d", z);
16     return 0;
17 }
```

2

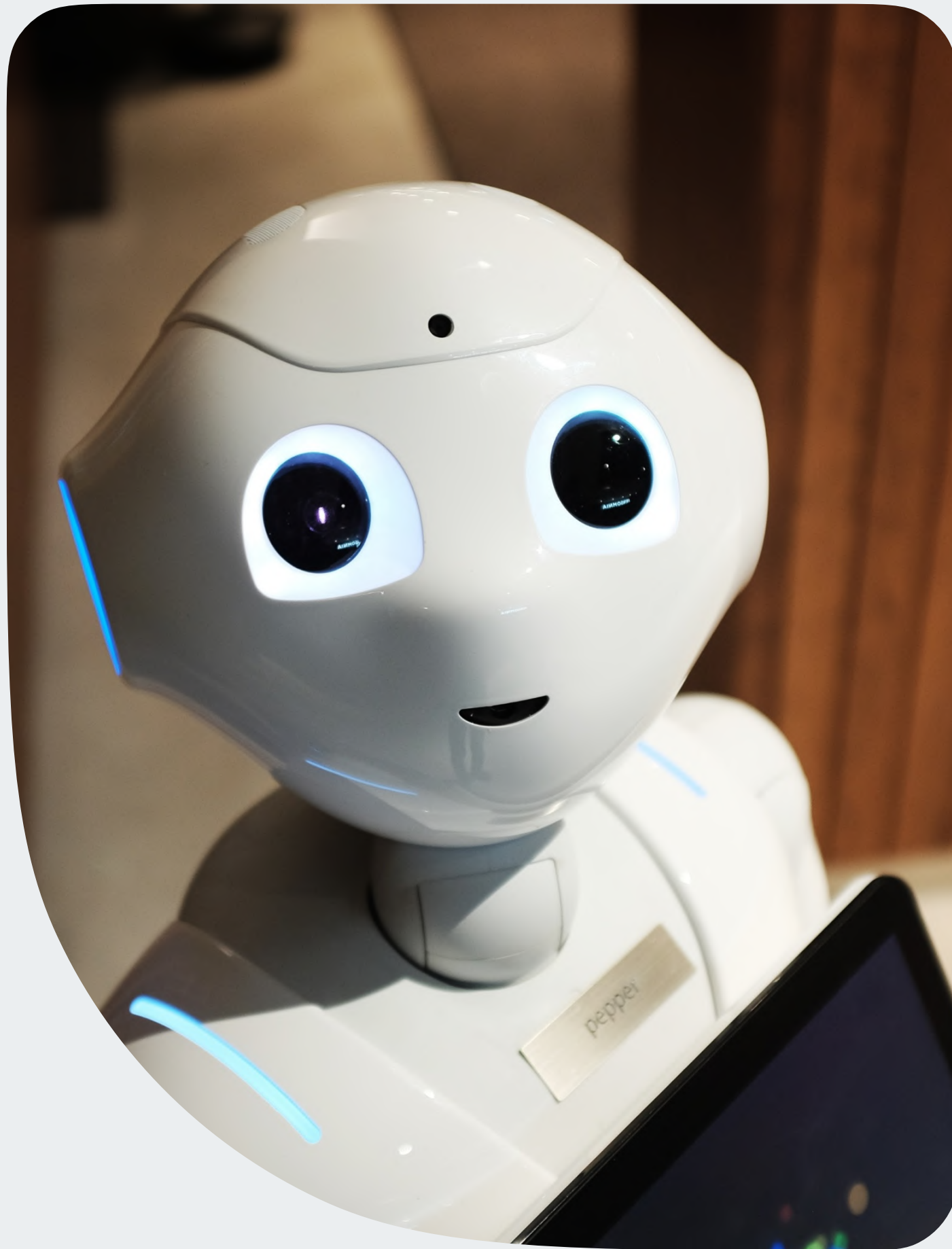
Закрепление

Что будет выведено на экран в результате работы программы если ввести 2, а затем 3?

Результат работы программы:

x= 2
y= 3
z= 0

```
1  #include <stdio.h>
2  int x=0, y=0, z=0;
3  void sum(void)
4  {
5      int x, y, z;
6      z = x + y;
7  }
8  int main()
9  {
10     printf("x= ");
11     scanf("%d", &x);
12     printf("y= ");
13     scanf("%d", &y);
14     sum();
15     printf("z= %d", z);
16     return 0;
17 }
```



Подведём итоги



Изучили различия локальных и глобальных переменных



Отработали на практике составление алгоритмов с хранением данных в локальных и глобальных переменных на Си

ОПИШИ СВОИ ЭМОЦИИ ПОСЛЕ УРОКА



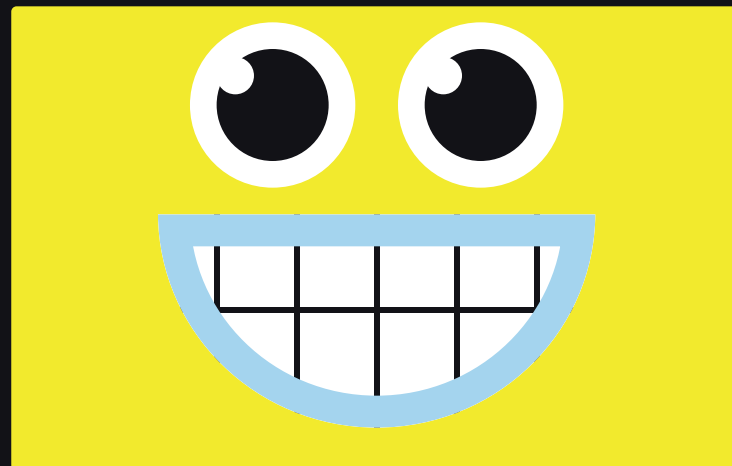
взял(а)
на вооружение ...

1



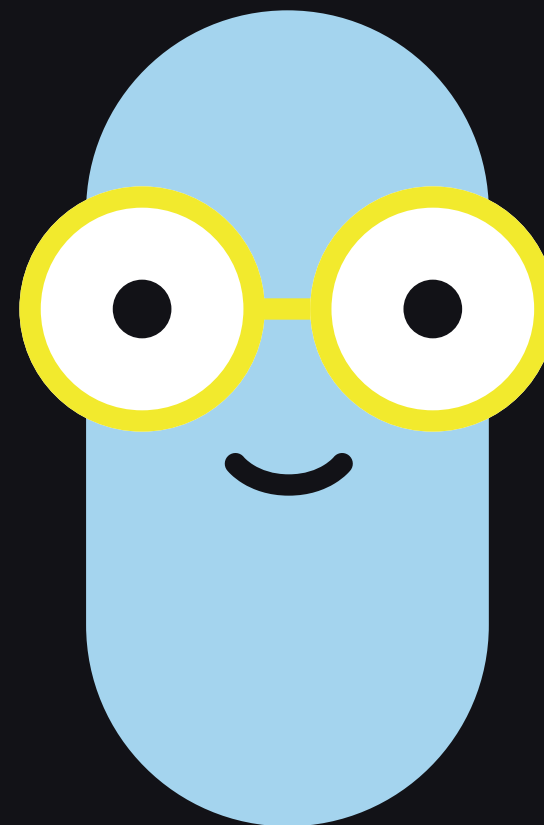
меня выбило
из колеи

2



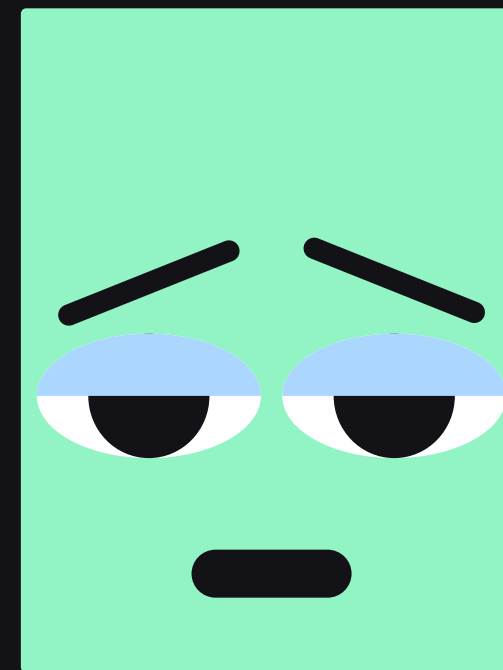
всё прошло без сучка,
без задоринки

3



мы били
в одну точку

4



нашлось моё
больное место

5

Домашнее задание



До встречи!