

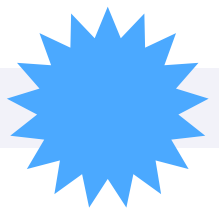
Программирование на C++



Минцифры
России

UCHi **DOMA**

20.35
УНИВЕРСИТЕТ

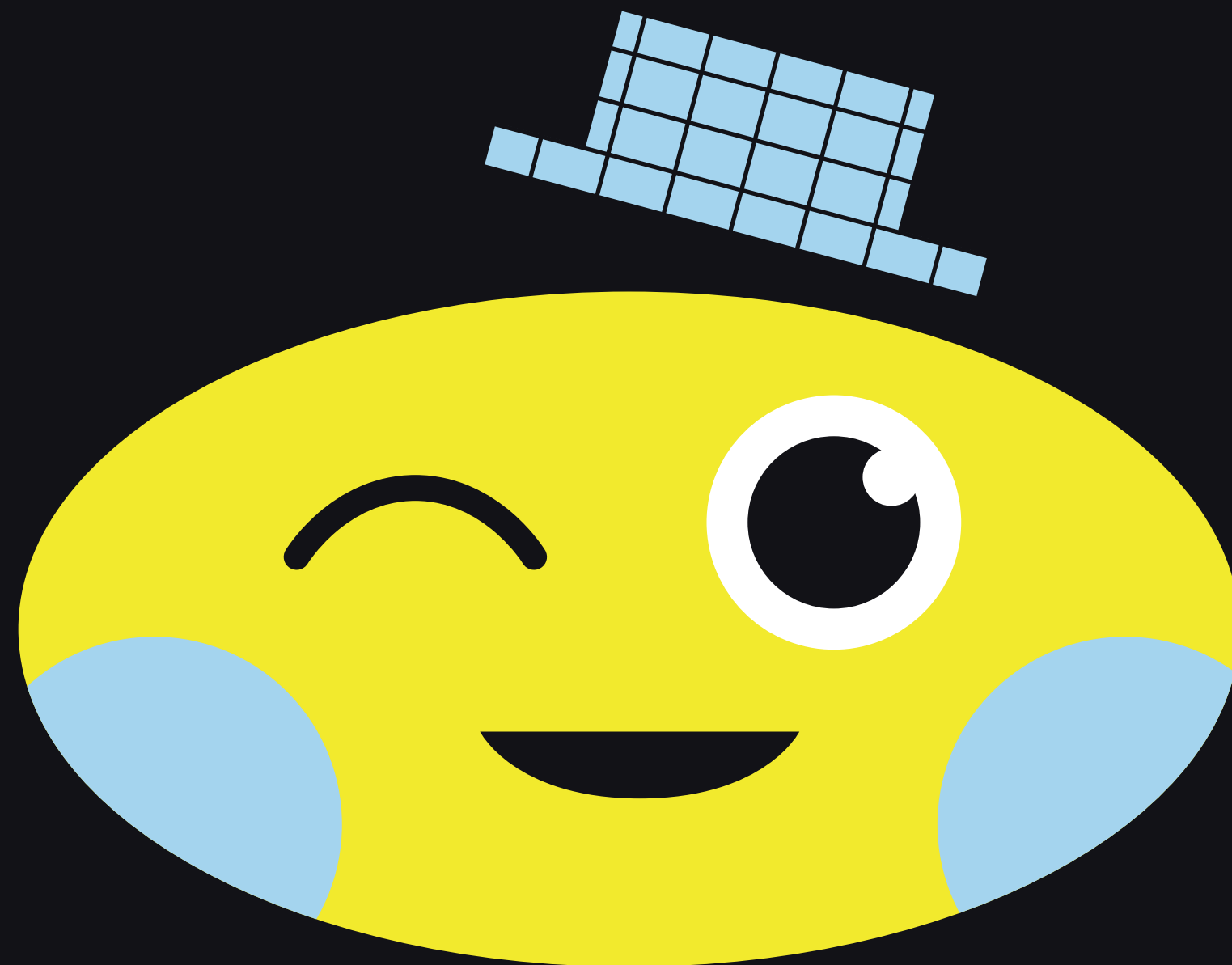


Модуль 1 Урок 2

Операторы ввода, вывода



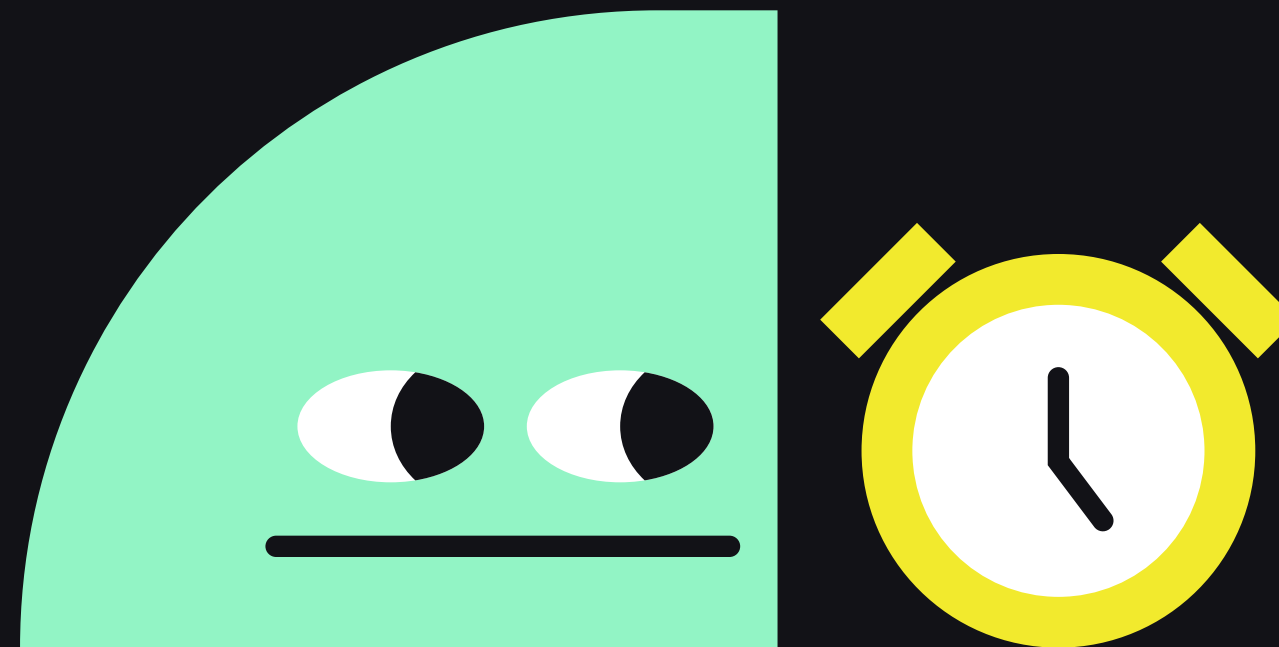
Привет!



проверка готовности



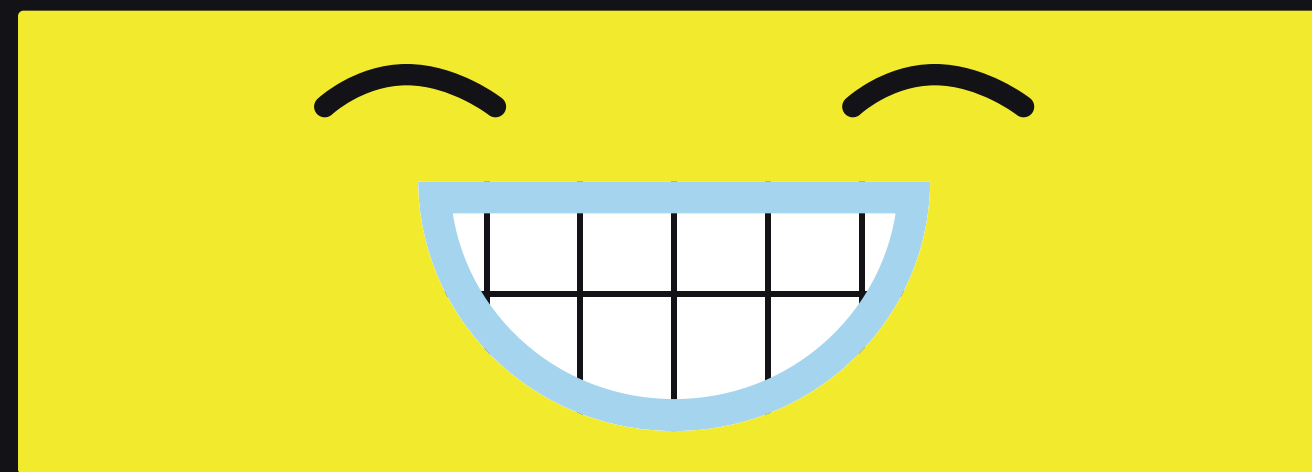
Видим и слышим друг друга без помех



Не опаздываем и не отвлекаемся



Сидим прямо



Улыбаемся, если всё ок

Как домашка?



Какие были трудности?



Какие остались вопросы?



Сколько заданий выполнено?



повторение



Чему будет равно значение переменной «с»?

```
1  int main ( )  
2  {  
3      int a=2,b=3,c;  
4      c=a+b*2;  
5      c++;  
6      return 0;  
7  }
```



Что будет результатом работы программы?

```
1  int main ( )  
2  {  
3      int a=2,b=3;  
4      a=a+3;  
5      b=a+1;  
6      c=b-a;  
7  }
```

1 c=1

2 c=2

3 Ошибка, программа
не выполнится



Что будет результатом работы программы?

```
1 int main ( )  
2 {  
3     int a=2,b=3;  
4     int c;  
5     a=a+3;  
6     b=a+1;  
7     c=b-a;  
8 }
```

1 c=1

2 c=2

3 Ошибка, программа
не выполнится

повторение



Инициализация переменных

```
int a=10;  
float b=1.5;
```

Как пользователю ввести данные
и вывести данные на экран?

Цели урока



Изучить операторы ввода, вывода



отработать на практике написание программ с выводом данных на Си

Операторы ввода, вывода



В базовом наборе языка C нет операторов ввода, вывода.

Ввод и вывод информации осуществляется через функции стандартной библиотеки `stdio.h`. Эта библиотека содержит функции:

★ `printf()` — для вывода информации

★ `scanf()` — для ввода информации

Для использования функций библиотеки, ее необходимо подключить:

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

Структура программы

```
1  #include <stdio.h>
2  int main ( )
3  {
4      return 0;
5  }
```

Оператор вывода

Функция `printf()` предназначена для форматированного вывода. Она переводит данные в символьное представление и выводит полученные изображения символов на экран. При этом у программиста имеется возможность форматировать данные, то есть влиять на их представление на экране.

Общая форма записи функции `printf()`:

```
printf("СтрокаФорматов", объект1, объект2, ..., объектn);
```

Оператор вывода

```
printf(“СтрокаФорматов”, объект1, объект2, ..., объектn);
```

СтрокаФорматов состоит из следующих элементов:

- ✓ управляющих символов
- ✓ текста, представленного для непосредственного вывода
- ✓ форматов, предназначенных для вывода значений переменных различных типов



Объекты могут отсутствовать.

Оператор вывода

Управляющие символы не выводятся на экран, а управляют расположением выводимых символов. Отличительной чертой управляющего символа является наличие обратного слэша \ перед ним.

Основные управляющие символы:



`\n` — перевод строки



`\t` — горизонтальная табуляция



`\v` — вертикальная табуляция



`\b` — возврат на символ



`\r` — возврат на начало строки



`\a` — звуковой сигнал

Оператор вывода

Форматы нужны для того, чтобы указывать вид, в котором информация будет выведена на экран. Отличительной чертой формата является наличие символа процент `%` перед ним:

- ✦ `%d` — целое число типа `int` со знаком в десятичной системе счисления
- ✦ `%u` — целое число типа `unsigned int`
- ✦ `%x` — целое число типа `int` со знаком в шестнадцатеричной системе счисления
- ✦ `%o` — целое число типа `int` со знаком в восьмеричной системе счисления
- ✦ `%hd` — целое число типа `short` со знаком в десятичной системе счисления
- ✦ `%hu` — целое число типа `unsigned short`
- ✦ `%hx` — целое число типа `short` со знаком в шестнадцатеричной системе счисления

Оператор вывода

Форматы нужны для того, чтобы указывать вид, в котором информация будет выведена на экран. Отличительной чертой формата является наличие символа процент % перед ним:

- ★ `%ld` — целое число типа `long int` со знаком в десятичной системе счисления
- ★ `%lu` — целое число типа `unsigned long int`
- ★ `%lx` — целое число типа `long int` со знаком в шестнадцатеричной системе счисления
- ★ `%f` — вещественный формат (числа с плавающей точкой типа `float`)
- ★ `%lf` — вещественный формат двойной точности (числа с плавающей точкой типа `double`)
- ★ `%e` — вещественный формат в экспоненциальной форме (числа с плавающей точкой типа `float` в экспоненциальной форме)
- ★ `%c` — символьный формат
- ★ `%s` — строковый формат

Оператор вывода

Строка форматов содержит форматы для вывода значений. Каждый формат вывода начинается с символа %. После строки форматов через запятую указываются имена переменных, которые необходимо вывести.

Количество символов % в строке формата должно совпадать с количеством переменных для вывода.

Тип каждого формата должен совпадать с типом переменной, которая будет выводиться на это место. Замещение форматов вывода значениями переменных происходит в порядке их следования.

Оператор вывода

Пример программы

```
1  #include <stdio.h>
2  int main ( )
3  {
4      int a = 5;
5      float x = 2.78;
6      printf("a=%d\n", a);
7      printf("x=%f\n", x);
8      return 0;
9  }
```

Результат работы программы

```
a=5
x=2.780000
```

Оператор вывода

Тот же самый код может быть представлен с использованием одного вызова `printf`:

Пример программы

```
1  #include <stdio.h>
2  int main ( )
3  {
4      int a = 5;
5      float x = 2.78;
6      printf("a=%d\nx=%f\n", a);
7      return 0;
8  }
```

Результат работы программы

```
a=5
x=2.780000
```

Оператор вывода

Табличный вывод

При указании формата можно явным образом указать общее количество знакомест и количество знакомест, занимаемых дробной частью:

Пример программы

```
1  #include <stdio.h>
2  int main ( )
3  {
4      float x = 1.2345;
5      printf("x=%10.5f\n", x);
6      return 0;
7  }
```

Результат работы программы

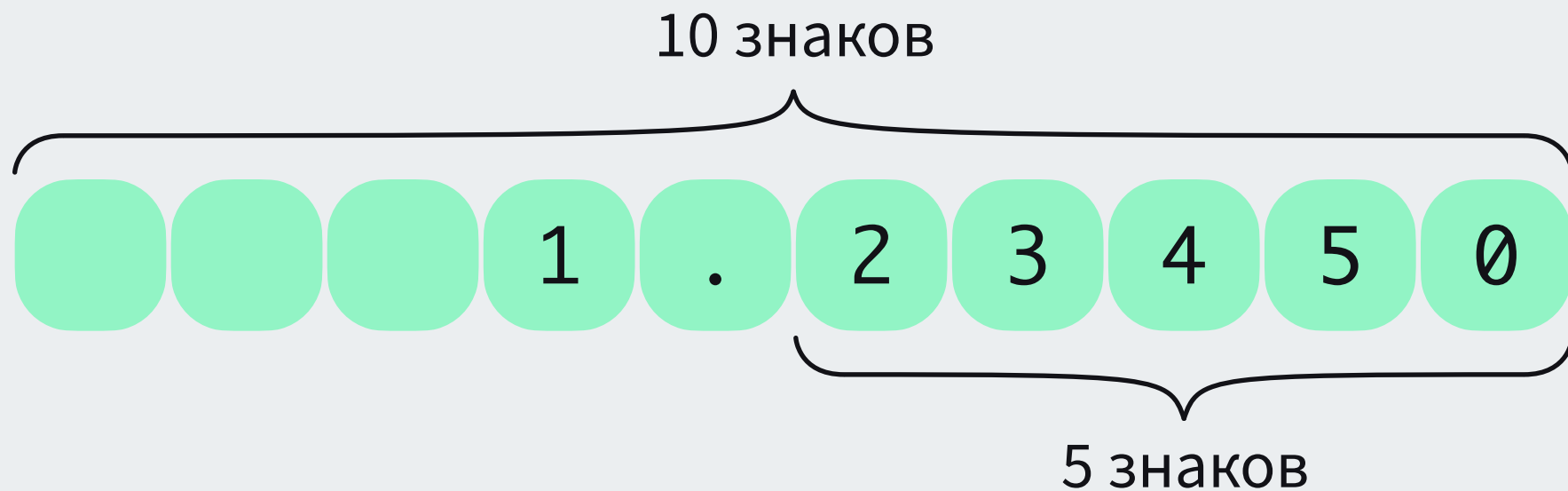
```
x=    1.23450
```

Оператор вывода

```
printf("x=%10.5f\n", x);
```

x= 1.23450

В приведенном примере 10 — общее количество знакомест, отводимое под значение переменной; 5 — количество позиций после разделителя целой и дробной части (после десятичной точки). В указанном примере количество знакомест в выводимом числе меньше 10, поэтому свободные знакоместа слева от числа заполняются пробелами. Такой способ форматирования часто используется для построения таблиц.



Оператор вывода

Пример вывода текста

```
1  #include <stdio.h>
2  int main ( )
3  {
4      printf("Привет \n");
5      printf("Вася\n");
6      return 0;
7  }
```

Результат работы программы

```
Привет
Вася
```

Оператор вывода

Пример вывода текста

```
1  #include <stdio.h>
2  int main ( )
3  {
4      printf("Привет \nВася \n");
5      return 0;
6  }
```

Результат работы программы

```
Привет
Вася
```


Оператор ввода

Функция форматированного ввода данных с клавиатуры `scanf()` выполняет чтение данных, вводимых с клавиатуры, преобразует их во внутренний формат и передает вызывающей функции. При этом программист задает правила интерпретации входных данных с помощью спецификаций форматной строки. Общая форма записи функции `scanf()`:

```
scanf("СтрокаФорматов", адрес1, адрес2, ...);
```

Строка форматов аналогична функции `printf()`. Для формирования адреса переменной используется символ амперсанд `&`:

адрес = `&объект`



Строка форматов и список аргументов для функции обязательны.

Оператор ввода

Пример программы

```
1  #include <stdio.h>
2  int main ( )
3  {
4      float y;
5      printf("Введите y: "); // выводим сообщение
6      scanf("%f", &y);      // вводим значения переменной y
7      printf("Значение переменной y=%f", y); // выводим значение переменной y
8      return 0;
9  }
```

Результат работы программы

Введите y: 5
Значение переменной y=5.000000

Оператор ввода

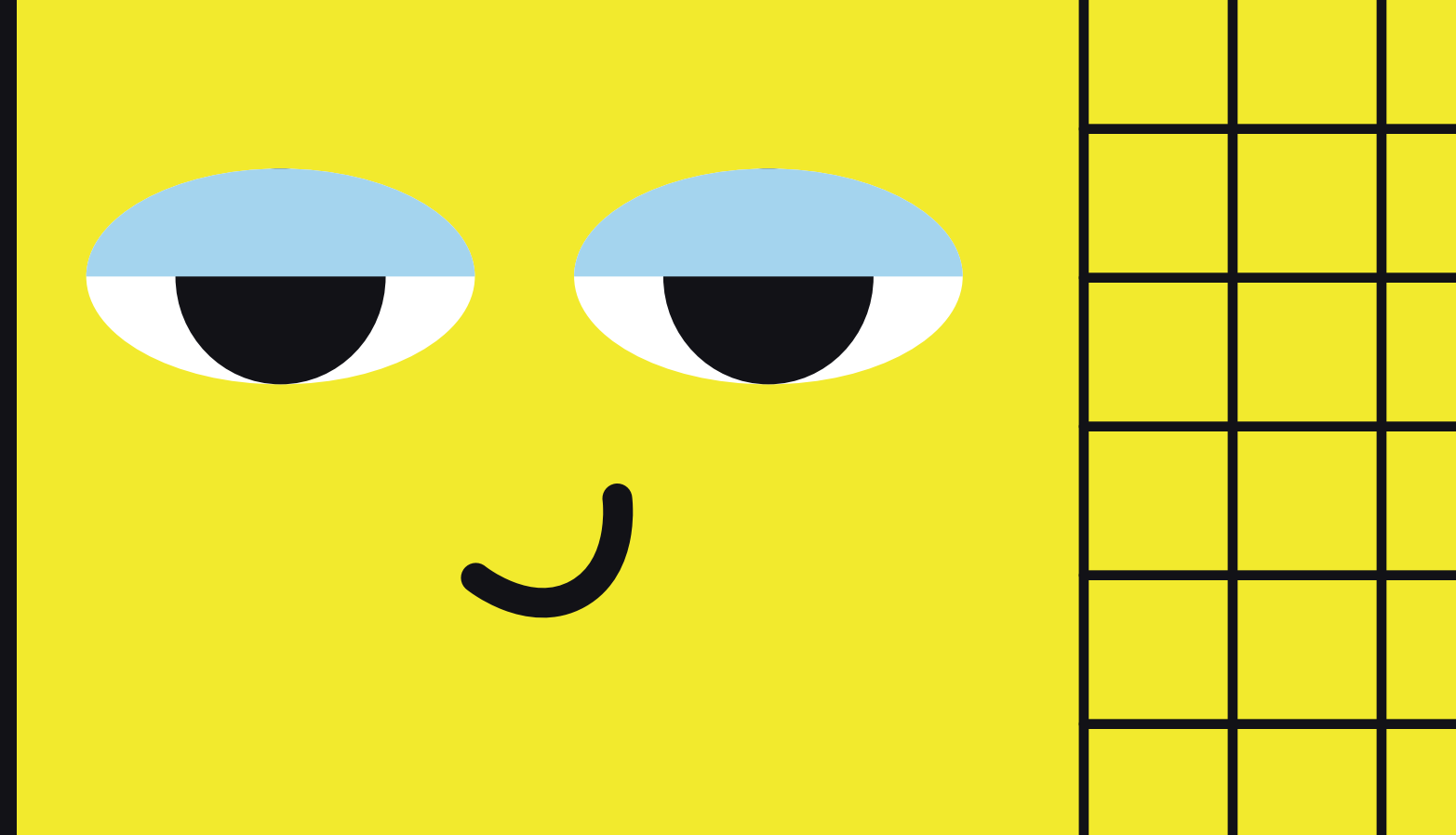
Пример

Пользователь вводит с клавиатуры 2 целых числа, программа должна вывести их сумму

```
1  #include <stdio.h>
2  int main ( )
3  {
4      int a,b;
5      printf("Введите первое число: "); // выводим сообщение
6      scanf("%d", &a); // вводим значения переменной a
7      printf("Введите второе число: ", y); // выводим сообщение
8      scanf("%d", &b); // вводим значения переменной b
9      printf("Сумма чисел=%d", a+b); // выводим значение суммы
10     return 0;
11 }
```

Пример работы
программы

```
Введите первое число: 1
Введите второе число: 2
Сумма чисел=3
```



Практика

Среда разработки

Интегрированная среда разработки, ИСР (англ. Integrated development environment — **IDE**) — комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).

Среда разработки включает в себя:



текстовый редактор



транслятор (компилятор и/или интерпретатор)



средства автоматизации сборки



отладчик

Среда разработки

Популярные среды разработки на C++:



Microsoft Visual Studio



Eclipse CDT



Sublime Text



NetBeans



Qt Creator



CLion



CodeLite



Code::Blocks

Среды разработки

Интегрированная среда разработки Visual Studio (IDE)

Помимо изменения кода, Visual Studio IDE объединяет графические конструкторы, компиляторы, средства завершения кода, системы управления версиями, расширения и многие другие функции в одном месте.



Среда разработки

- 1 Установите и запустите на своем компьютере Visual Studio с рабочей нагрузкой Разработка классических приложений на C++.
 - 2 Запустите Visual Studio, вы увидите диалоговое окно запуска. Выберите «Создать проект», чтобы приступить к работе.
- ★ Или в строке меню Visual Studio последовательно выберите Файл>Создать>Проект. Откроется окно «Создание проекта».





Visual Studio использует проекты, чтобы упорядочить код для приложения, и решения, чтобы упорядочить проекты. Проект содержит все параметры, конфигурации и правила, используемые для сборки приложения. Кроме того, он управляет связью между всеми файлами проекта и любыми внешними файлами. Чтобы создать приложение, сначала создайте проект и решение.

Visual Studio 2022

Open recent

Search recent (Alt+S)

Get started

-  **Clone a repository**
Get code from an online repository like GitHub or Azure DevOps
-  **Open a project or solution**
Open a local Visual Studio project or .sln file
-  **Open a local folder**
Navigate and edit code within any folder
-  **Create a new project**
Choose a project template with code scaffolding to get started

[Continue without code →](#)

Среда разработки

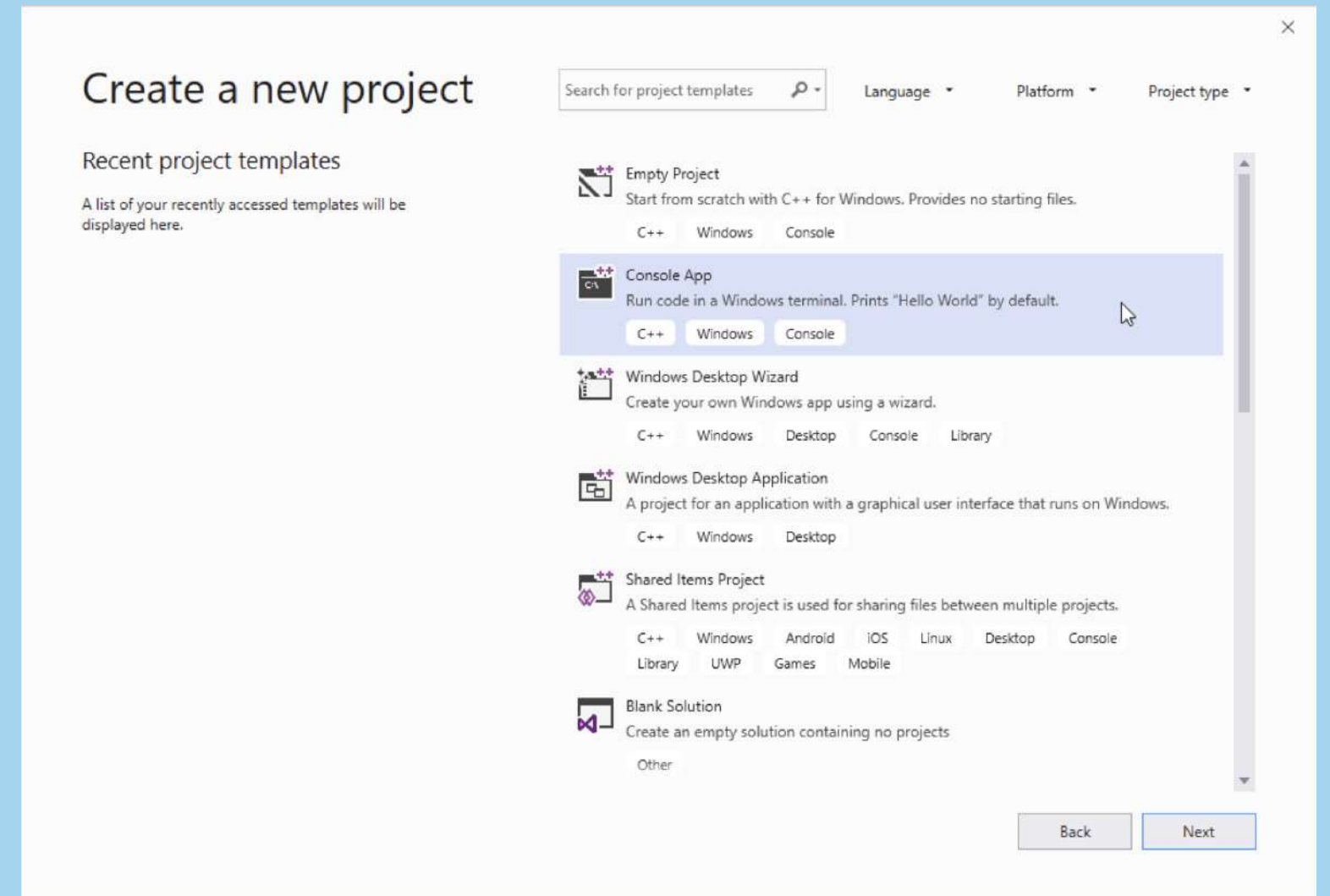
3

В списке шаблонов проектов выберите «Консольное приложение» и нажмите «Далее».

важно



Убедитесь, что вы выбрали версию C++ для шаблона Консольное приложение. Этот шаблон содержит теги C++ , Windows и Консоль, а в углу значка есть «++».



Среда разработки

4

В диалоговом окне «Настроить новый проект» выберите поле ввода «Имя проекта», назовите новый проект и нажмите «Создать».

Будет создано пустое консольное приложение Windows на C++. Консольные приложения используют окно консоли Windows для отображения выходных данных и приема данных, вводимых пользователем. В Visual Studio откроется окно редактора с созданным кодом.

Configure your new project

Console App C++ Windows Console

Project name

CalculatorTutorial

Location

C:\Users\username\source\repos

Solution name ⓘ

CalculatorTutorial

☐ Place solution and project in the same directory

Back Create

Запуск программы

5

Чтобы запустить этот код, в строке меню выберите «Отладка» и «Начать отладку» или нажмите на клавиатуре кнопку «F5».

Отладка Тест Анализ Средства Расширения Окно Справка

Окна ▶

Графика ▶

▶ Начать отладку F5

▶ Запуск без отладки Ctrl+F5

Online компиляторы

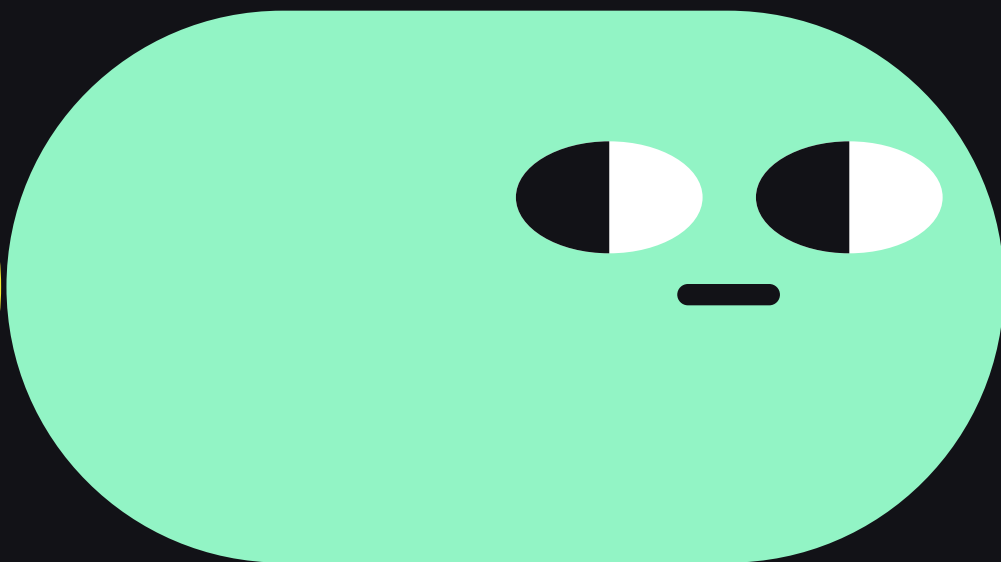
online-cpp.com

[onlinegdb.com/
online_c_compiler](https://onlinegdb.com/online_c_compiler)

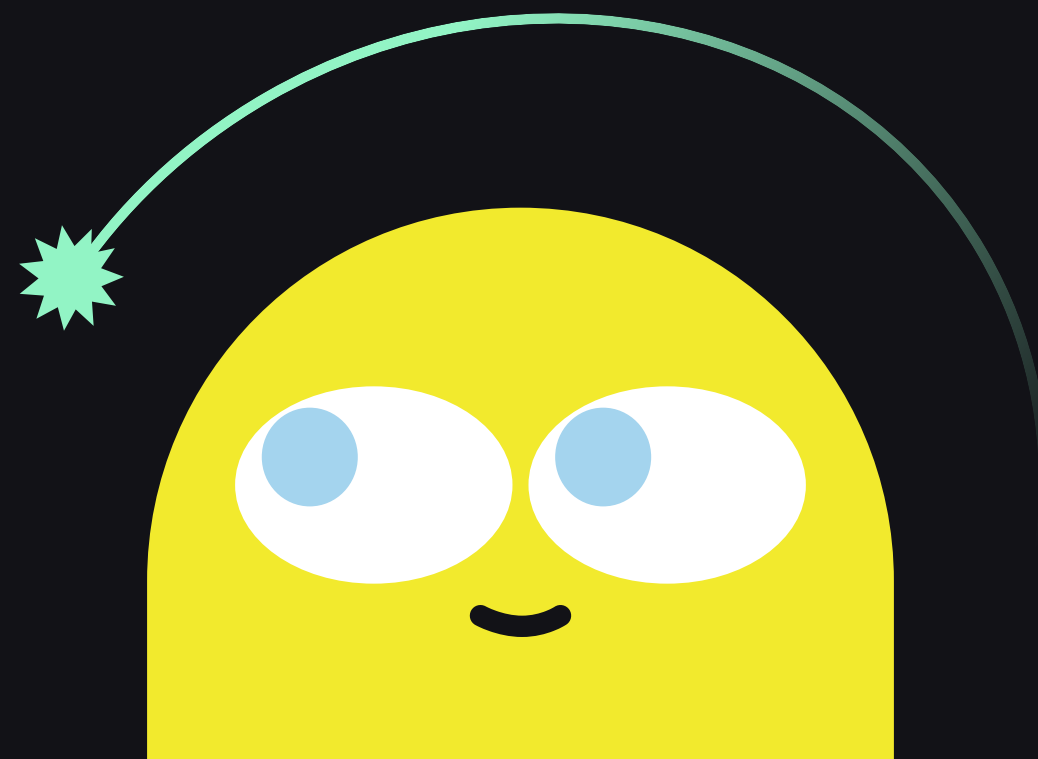
[replit.com/
languages/cpp](https://replit.com/languages/cpp)

перерыв

физкультминутка



Смотрим вверх–вниз, вправо–влево



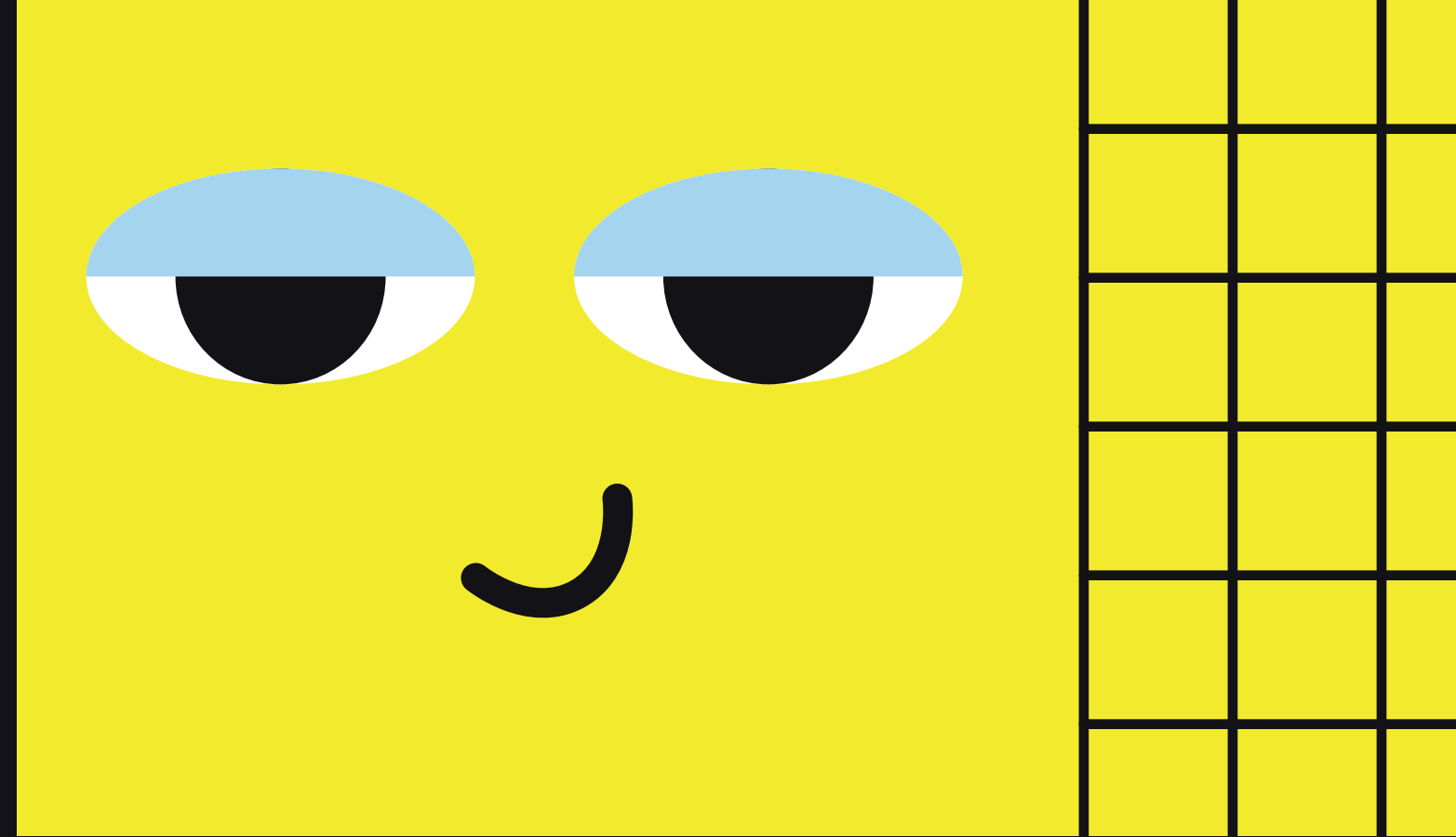
Вращаем по кругу туда–обратно



Крепко зажимаемся



Быстро моргаем



Практика

Задания в учебной платформе



Подведем итоги



Изучили операторы ввода, вывода



Отработали на практике написание программ с выводом данных на Си

1

Закрепление

Оператор вывода?

1

Закрепление

Оператор вывода:

```
printf(“СтрокаФорматов”, объект1, объект2, ..., объектn);
```

2

Закрепление

Какой тип данных будет выведен в консоль?

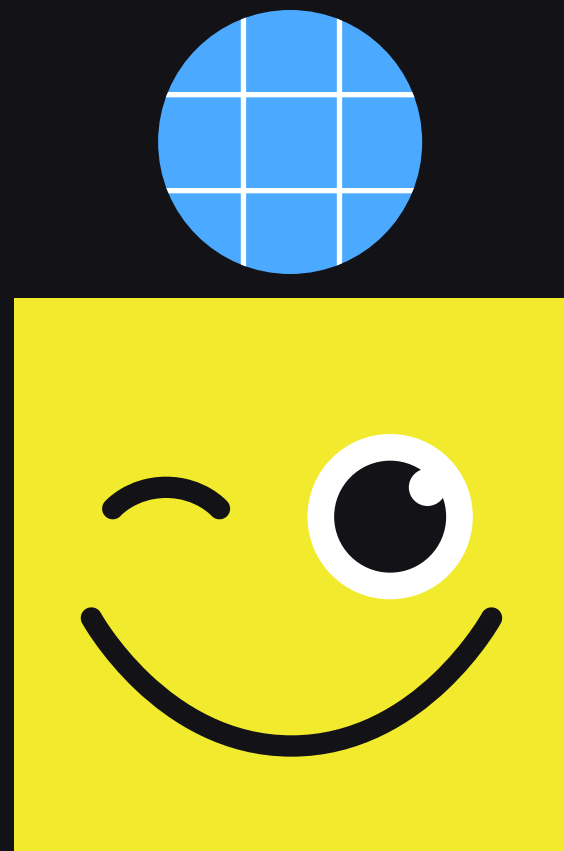
```
printf("%d", a+b);
```

Какой тип данных будет выведен в консоль?

```
printf("%d", a+b);
```

`%d` — целое число типа `int` со знаком в десятичной системе счисления

итоги урока



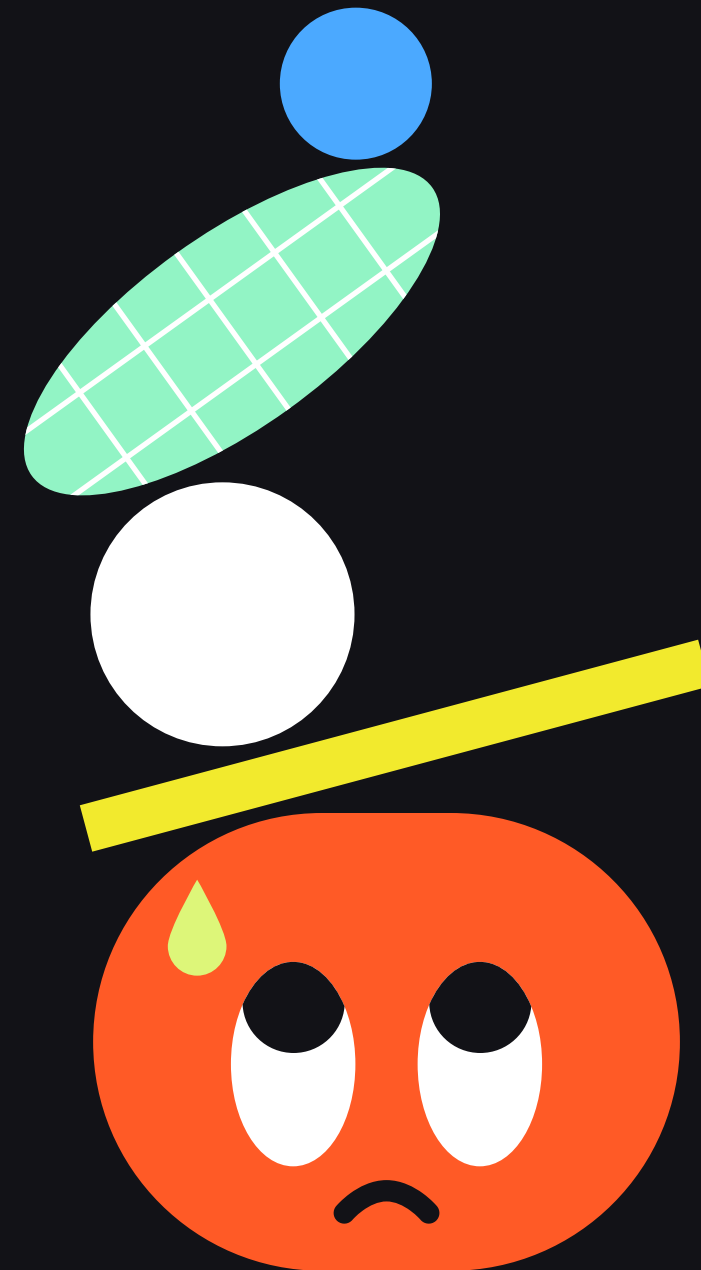
1

легко!



2

сложно, но можно



3

трудно

Домашнее задание



До встречи!