

Это у меня
хорошо
получается

Попробуйте, получится и у вас!



Кто я?

Педагог дополнительного образования

Наставник

Генератор идей

Современный человек

Теория < Практика

Задача – эффективно преподнести теорию и уделить больше времени практическим заданиям

Дизайн в педагогике

Текст, его размещение
Размер и принципы
типографики

- Положение элементов
- Общий стиль, цвета и акценты

Типографика

Искусство оформления текста

— Здесь каждое слово на вес золота —

Проблема

Необходимость повышения качества обслуживания и безопасности пассажиров в поездах. Современные поезда сталкиваются с рядом проблем, связанных с обслуживанием пассажиров и обеспечением их безопасности. Традиционные методы, такие как использование проводников, имеют свои ограничения: человеческий фактор, ограниченность ресурсов, отсутствие постоянного мониторинга.

Проблема

Современные поезда сталкиваются с рядом проблем, связанных с обслуживанием пассажиров и обеспечением их безопасности, поэтому имеется...

Необходимость повышения качества обслуживания и безопасности пассажиров в поездах

Но традиционные методы имеют свои ограничения:

- человеческий фактор
- ограниченность ресурсов
- отсутствие постоянного мониторинга

Игровой дизайн

«Искусство управления вниманием и мотивацией»



Шпаргалка для новичков в C++

Заголовочные файлы

Синтаксис

```
#include <заголовочный файл>
```

Стандартные заголовки/библиотеки

#include <stdio.h>	функции ввода/вывода
#include <string.h>	функции для работы со строками
#include <time.h>	функции для работы со временем
#include <stdlib.h>	память, рандом, ...
#include <math.h>	математические функции
#include <iostream.h>	функции ввода/вывода
#include <fstream.h>	функции ввода/вывода в файл
#include <"myfile.h">	включить файл из текущей директории

Пространства имён

Синтаксис

```
using namespace название пространства имён;
```

Стандартные пространства имён

```
using namespace std;
```

Комментарии

```
// Однострочный комментарий  
/* Многострочный комментарий */
```

Базовые типы данных

Числовые

```
int var1, var2 = 10;  
float var1, var2 = 10.5;
```

Символьные

```
char var1, var2 = 'a';  
string var1, var2 = "Hello World!";
```

Логический тип

```
bool var1, var2 = true;
```

Базовые операторы ввода/вывода

```
cin >> переменная    cin >> var1 >> var2;  
cout << данные        cout << "Hello World!" << var;
```

Базовые математические операторы

+	Добавить
-	Отнять
*	Умножить
/	Разделить
%	Остаток от деления

Порядок выполнения действий такой же как в математике.

var++;	Постфиксный и префиксный инкременты
++var;	
var--;	Постфиксный и префиксный декременты
--var;	

Префиксный инкремент/Декремент сначала увеличивает/уменьшает переменную, а затем возвращает её значение, а постфиксный, наоборот, сначала возвращает значение переменной, а затем увеличивает/уменьшает её.

Условия

a == b	Если a равно b, то результат true (истина), а иначе - false (ложь)
a != b	Если a НЕ равно b, то результат true (истина), а иначе - false (ложь)
a < b	Если a меньше b, то результат true (истина), а иначе - false (ложь)
a > b	Если a больше b, то результат true (истина), а иначе - false (ложь)
a <= b	Если a меньше или равно b, то результат true (истина), а иначе - false (ложь)
a >= b	Если a больше или равно b, то результат true (истина), а иначе - false (ложь)
a && b	Если a и b равны true (истина), то результат true (истина), а иначе - false (ложь)
a b	Если a или b равно true (истина), то результат true (истина), а иначе - false (ложь)

Логические выражения в C++ оцениваются справа налево, правила скобок действуют так же как в математике.

Массивы

Одномерный массив

```
название массива[количество элементов];  
int arr[10];  
string arr[100];
```

Двухмерный массив

```
название массива[количество элементов][количество элементов];  
int arr[10][10];  
string arr[100][5];
```

Порядок элементов в массиве начинается с 0.
Например вот вывод 3-го элемента: cout << var[2];

Разветвления

if

```
if (условие) {  
    Код, который выполнится если условие истинно  
}  
else if (другое условие) {  
    Код, который выполнится если первое условие ложно,  
    а второе - истинно  
}  
else {  
    Код, который выполнится если оба условия ложны  
}
```

while

```
while (условие) {  
    Код, который будет выполняться пока условие истинно  
}
```

for

```
for (инициализация; условие; действие) {  
    Код, который будет выполняться пока условие истинно  
}
```

инициализация - это команда которая выполняется перед запуском цикла for.
действие - это команда которая выполняется после каждой итерации цикла for

do while

```
do {  
    Код, который выполнится один раз, а затем будет выполняться если условие истинно  
} while (условие);
```

Команда break; останавливает выполнение цикла
Команда continue; переходит к следующей итерации цикла

switch case

```
switch (переменная) {  
    case значение 1:  
        Код, который выполнится если переменная равна значению 1  
        break;  
    case значение 2:  
        Код, который выполнится если переменная равна значению 2  
        break;  
    default:  
        Код, который выполнится если ни один case выше не был выполнен  
        break;  
}
```

НАЧАЛЬНЫЕ ЦИКЛЫ

```
void setup () { // код, выполнится 1 раз при старте программы }
void loop () { // код будет выполняться циклично после setup }
void yield () { // код будет выполняться во время работы delay }
void Ваше_название () { // код будет вызываться по запросу }
```

Переменные

```
Тип переменной Название ; // объявить
Тип переменной Название1, Название 2 ; // объявить несколько переменных
Тип переменной Название = значение ; // объявить и задать начальные значения
```

Типы переменных

	Значение
boolean или bool	true/false или 1/0
char	любой символ
float	3.4/10.84898 и т.д.
String	Могу писать любое предложение

Работа с пинами

```
pinMode (номер пина, режим работы);
OUTPUT // настраивает пин на выходной сигнал
INPUT // настраивает пин на входной сигнал
INPUT_PULLUP // настраивает пин на вход, и подключает подтягивающий резистор
```

Взаимодействие с портами

```
digitalWrite (номер пина, HIGH/LOW или 1/0); //максимум или минимум
digitalRead (номер пина); //считывает значения с пина (либо 0, либо 1)
analogWrite (номер пина, значение); //значения могут варьироваться от 0 до 255
analogRead (номер пина); // считывает значение пина (от 0 до 255)
```

Директивы

```
#define ИМЯ значение //каждый раз заменяет ИМЯ на определенное значение
#include <название> //подключает библиотеку из списка библиотек
#include "название" //подключает библиотеку из папки со скетчем
#if // аналог if в логической конструкции
#elif // аналог else if в логической конструкции
#else // аналог else в логической конструкции
#endif // директива, завершающая условную конструкцию
#ifdef // если "определено"
#endif // если "не определено"
```

Циклы и условные операторы

`==, !=, >, <, >=, <=` // равно, не равно, больше или равно, меньше или равно

`!, &&, ||` // НЕ, И, ИЛИ

```
switch (переменная) {
case 1:
// выполнить, если переменная == 1
break; //позволяет выйти из switch
case 2:
// выполнить, если переменная == 2
break;
default:
// выполнить, если переменная ни 1 ни 2
break;
```



```
if (условие) {
действие;
}
else if (дополнительное условие) {
действие;
}
else {
действие;
}
```

```
int c = (a > b) ? 10 : -20; // если a > b, то c = 10. Если нет, то c = -20
// "Короткий" вариант if-else
```

```
for ( int i = 0; i < 10; i++) { //начальное значение переменной; условие; изменение переменной
тело цикла; }
```

```
while (условие) {
//тело цикла, которое выполняется, пока соблюдено условие
}
do {
//сначала выполниться, а потом проверит условие
} while (условие);
```

Вычисления

min (a, b);	// Возвращает меньшее из чисел a и b
max (a, b);	// Возвращает большее из чисел
abs (x);	// Модуль числа
round (x);	// Математическое округление
sq (x);	// Квадрат числа
cos (x)	// Косинус (радианы)
sin (x)	// Синус (радианы)
tan (x)	// Тангенс (радианы)
sqr (x)	// Корень квадратный
sqr (x)	// Корень квадратный для float чисел
acos (x)	// Арккосинус (радианы)
asin (x)	// Арксинус (радианы)
atan (x)	// Арктангенс (радианы)
log (x)	// Натуральный логарифм x (ln(x))
log10 (x)	// Десятичный логарифм x (log ₁₀ x)

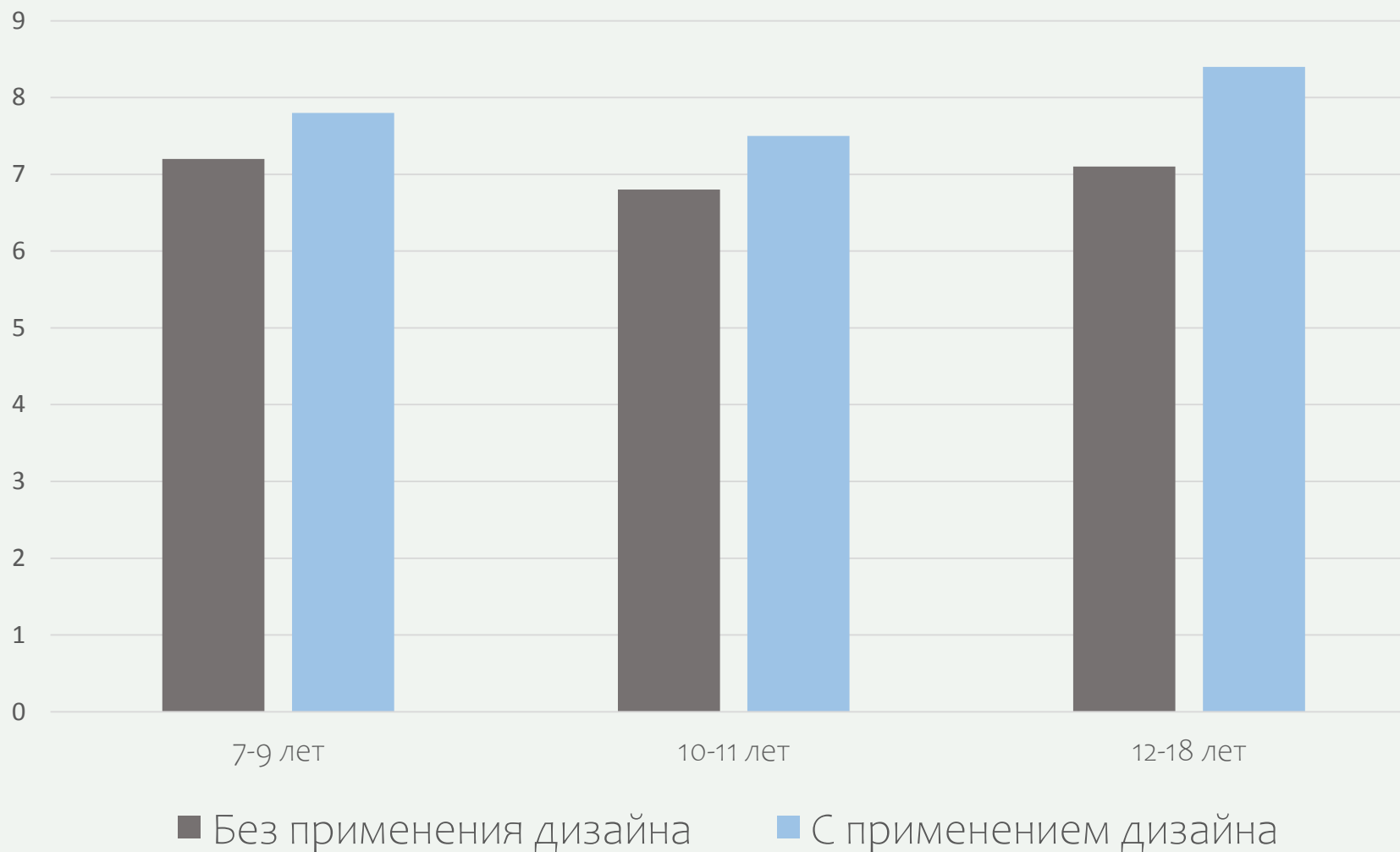
Монитор порта

```
Serial.begin (скорость); // открыть порт на скорости
Serial.end (); // закрыть порт
Serial.print (переменная); //отправляет в порт значение переменной
Serial.println (переменная); // отправляет значение в порт и переводит на новую строку
Serial.print (s + "text" + 1234); //сложение в строке
Serial.readString (); // принять строку
Serial.parseInt (); // принять целочисленное
```

Массивы

```
int myints (6); // указываем количество ячеек
int myPins [] = {2, 4, 8, 3, 6}; // указываем содержимое ячеек
float Sens [3] = {0.2, 0.4, -0.5}; // указываем и то и то, количество ячеек должно совпадать
char message (6) = "hello"; // храним символы
```

Средний бал по результатам итогового теста в разных группах



Иные возможности

Законы
Web дизайна

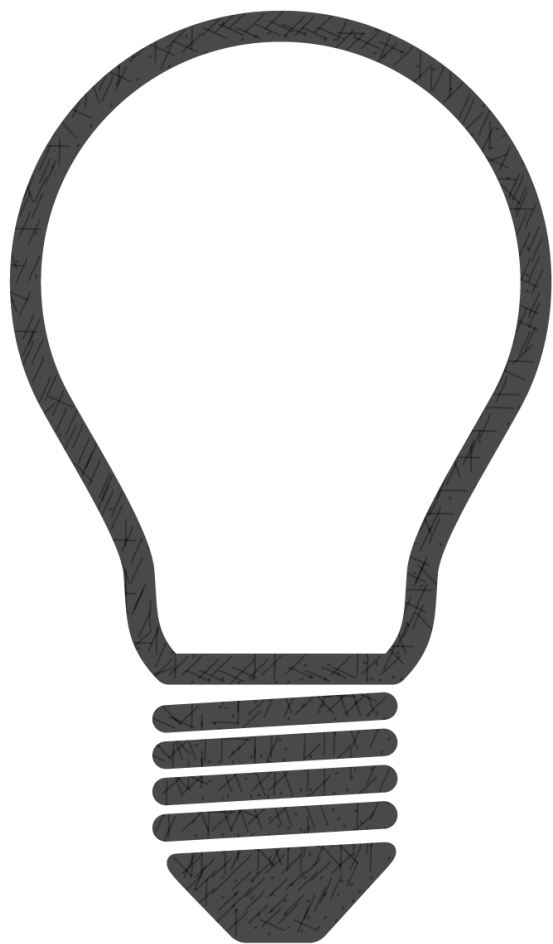
- Миллера
- Фиттса
- Хика

Принципы
фотографии

- Золотое сечение -
- Правило третей -

Узконаправленные
принципы

- Правила Нормана
- Бритва Окама
- Закон Парето



Это у меня
хорошо
получается

Попробуйте, получится и у вас!