

**Областной конкурс методических разработок педагогов
«Открытые уроки»**

**Муниципальное бюджетное образовательное учреждение
дополнительного образования
«Гуманитарный центр интеллектуального развития»
городского округа Тольятти**

Конкурсные материалы

Номинация: Методические рекомендации по организации занятий по образовательной программе

Название конкурсных материалов: Методическое пособие «Сборник контрольных заданий по дополнительной программе «Программирование – это интересно. С++» (первый год обучения)»

Разработчик:

Савина Дарья Александровна,
педагог дополнительного образования,
89277867220, e-mail: savina@cir.tgl.ru

Самара

2023

Пояснительная записка

Предлагаемое методическое пособие представляет собой сборник контрольных заданий для обучающихся первого года обучения по общеобразовательной дополнительной программе «Программирование это интересно. С++», реализуемой в МБОУ ДО ГЦИР городского округа Тольятти.

Методическое пособие создано в помощь педагогу дополнительного образования по осуществлению им контрольно-диагностической функции.

Контрольные задания рассчитаны на возраст обучающихся 11-12 лет. Предполагается, что их выполнение будет способствовать развитию в обучающихся таких личностных качеств, как

- самостоятельность в выполнении полученных заданий;
- профессиональная организованность;
- усидчивость и мотивация к достижению результата;
- уверенность в себе при решении задач;
- интерес к получению профессии программиста.

В методическое пособие включены контрольные задания по ключевым темам дополнительной программы «Программирование – это интересно. С++» (первый год обучения). Кроме текстов самих заданий, представляющих собой задания различного вида (тестовые задания с ключами, вопросы открытого и закрытого типа, решение задач, поиск ошибок и др.), сборник включает методические инструкции для педагога по организации выполнения контрольных заданий.

Задания были апробированы в течение учебного года.

Методическое пособие может быть использовано педагогами дополнительного образования, реализующими дополнительные программы технической направленности по обучению языку программирования С++. Оно поможет педагогу эффективно осуществлять контрольную функцию.



Муниципальное бюджетное образовательное учреждение дополнительного образования
«Гуманитарный центр интеллектуального развития»
городского округа Тольятти



**МЕТОДИЧЕСКОЕ ПОСОБИЕ
«СБОРНИК КОНТРОЛЬНЫХ ЗАДАНИЙ ПО
ДОПОЛНИТЕЛЬНОЙ ПРОГРАММЕ
«ПРОГРАММИРОВАНИЕ – ЭТО ИНТЕРЕСНО. C++»
(ПЕРВЫЙ ГОД ОБУЧЕНИЯ)»**

Разработчик:

Савина Дарья Александровна,
педагог дополнительного образования
МБОУ ДО ГЦИР

Тольятти, 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
СОДЕРЖАНИЕ КОНТРОЛЬНЫХ ЗАДАНИЙ И ИНСТРУКЦИИ ПО ОРГАНИЗАЦИИ ИХ ВЫПОЛНЕНИЯ.....	4
1. КОНТРОЛЬНЫЕ ЗАДАЧИ ПО ТЕМЕ «ВВОД-ВЫВОД ДАННЫХ. ПЕРЕМЕННЫЕ ТИПА INT».....	4
2. ТЕСТОВЫЕ ЗАДАНИЯ ПО ТЕМЕ «ВВОД-ВЫВОД ДАННЫХ. ПЕРЕМЕННЫЕ ТИПА INT».....	5
3. ТЕСТОВЫЕ ЗАДАНИЯ ПО ТЕМЕ «УСЛОВНЫЕ КОНСТРУКЦИИ C++»	8
4. КОНТРОЛЬНЫЕ ЗАДАНИЯ ПО ТЕМЕ «ВЫЧИСЛЕНИЕ. ЧИСЛА И СТРОКИ. УСЛОВНЫЕ КОНСТРУКЦИИ».....	11
5. ПРОМЕЖУТОЧНАЯ ИТОГОВАЯ РАБОТА ПО ТЕМАМ «ТИПЫ ПЕРЕМЕННЫХ. ОПЕРАТОРЫ. УСЛОВИЯ. ЦИКЛЫ».....	16
6. КОНТРОЛЬНЫЕ ЗАДАЧИ ПО ТЕМЕ «РАБОТА С СИМВОЛАМИ».....	20
7. ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ТЕМЕ «ИГРОВЫЕ СКРИПТЫ». ПРОЕКТИРОВАНИЕ МИНИ-ИГРЫ «ЛАБИРИНТ» В UNITY.	24
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	27

ВВЕДЕНИЕ

В сборник включены контрольные задания по ключевым темам программы «Программирование – это интересно. С++» первого года обучения.

Контрольные задания представляют собой задания различного вида: тестовые задания, вопросы открытого и закрытого типа, решение задач, поиск ошибок и т.п. Разные задания имеют разную схему решения.

Кроме текстов самих заданий, сборник включает методические рекомендации для педагога по организации выполнения контрольных заданий.

Методическое пособие включает в себя электронное приложение, которое содержит файлы с текстами контрольных заданий для распечатки.

СОДЕРЖАНИЕ КОНТРОЛЬНЫХ ЗАДАНИЙ И ИНСТРУКЦИИ ПО ОРГАНИЗАЦИИ ИХ ВЫПОЛНЕНИЯ

1. КОНТРОЛЬНЫЕ ЗАДАЧИ ПО ТЕМЕ «ВВОД-ВЫВОД ДАНЫХ. ПЕРЕМЕННЫЕ ТИПА INT»

Инструкция по организации выполнения контрольных заданий

Задание содержит 12 задач, которые необходимо распечатать и разрезать на полоски (билеты). Каждый учащийся вытягивает билет с одной задачей и выполняет ее на компьютере. Педагог контролирует правильность выполнения.

Содержание задания

1.	Задание - Пользователь вводит два числа - После чего на экран выводится надпись: «Вы ввели число ____ и число ____»
2.	Задание - Пользователь вводит два числа - После чего на экран выводится надпись: «Сумма чисел равна ____»
3.	Задание - Пользователь вводит два числа - После чего на экран выводится надпись: «Разность чисел равна ____»
4.	Задание - Пользователь вводит одно число - После чего на экран выводится надпись: «Если удвоить ваше число, то получится ____»
5.	Задание - Пользователь вводит два числа - После чего на экран выводится надпись: «Произведение чисел равно ____»

6.	Задание - Пользователь вводит число - После чего на экран выводится надпись: « Если к вашему числу прибавить 10, то получится _____ »
7.	Задание - Пользователь вводит число - После чего на экран выводится надпись: « Если ваше число умножить на 100, то получится _____ »
8.	Задание - Пользователь вводит два числа - После чего на экран выводится надпись: «Первое число больше второго на ____»
9.	Задание - Пользователь вводит два числа - После чего на экран выводится надпись: «Первое число больше второго в __ раз»
10.	Задание - Пользователь вводит три числа - После чего на экран выводится надпись: «Сумма двух первых чисел, умноженная на третье число равна _____»
11.	Задание - Пользователь вводит два числа - После чего на экран выводится надпись: «Частное чисел равно _____»
12.	Задание - Пользователь вводит три числа - После чего на экран выводится надпись: «Сумма двух первых чисел, разделенная на третье число равна _____»

2. ТЕСТОВЫЕ ЗАДАНИЯ ПО ТЕМЕ «ВВОД-ВЫВОД ДАННЫХ. ПЕРЕМЕННЫЕ ТИПА INT»

Инструкция по организации выполнения контрольных заданий

Задание содержит 10 тестовых вопросов с одним или несколькими правильными вариантами ответа и ключи. Каждый учащийся получает свой вариант теста и выполняет его в любом порядке, отмечая правильные ответы в бланке, в течение 10 минут. Педагог проверяет правильность выполнения теста согласно ключам.

Содержание задания

- 1. Переменные типа `int` – это.....**
 - А) Любые числа в диапазоне от -2,147,483,647 до 2,147,483,647
 - Б) Положительные целые числа
 - В) Целые числа в диапазоне от -2,147,483,647 до 2,147,483,647
 - Г) Дробные числа
- 2. Как выглядит оператор вывода данных?**
 - А) ++
 - Б) <<
 - В) >>
 - Г) “ ”
- 3. Как выглядит оператор ввода данных?**
 - А) ++
 - Б) <<
 - В) >>
 - Г) “ ”
- 4. При помощи какой из перечисленных строк можно вывести слово «Hello!» на экран?**
 - А) `std::cout>> “Hello!”;`
 - Б) `std::cin>> “Hello!”;`
 - В) `std::cin<< “Hello!”;`
 - Г) `std::cout<< “Hello!”;`
- 5. В каких вариантах ответа объявление переменной произведено корректно? (Несколько вариантов ответа).**
 - А) `int a = 5;`
 - Б) `int number_1 = 5;`
 - В) `int number`
 - Г) `int number_2;`
 - Д) `number = 12;`
 - Е) `int 5;`
 - Ж) `int number = number_1 + number_2;`
 - З) `int number = 2 * number_1;`
 - И) `intчисло = 4;`
 - К) `int number_1 = number_2;`
 - К) `int number 1 = 6;`
 - Л) `int a = 12`
- 6. При помощи какой из перечисленных строк обеспечивается ввод данных пользователем?**
 - А) `std::cout>> number;`

Б) `std::cin>> number`

В) `std::cin<< number;`

Г) `std::cin>> number;`

7. Как можно оставить комментарий или закомментировать строку в редакторе кода, чтобы она не обрабатывалась компилятором?

А) поставить перед строкой `//`

Б) поставить после строки `//`

В) поставить перед строкой `\n`

Г) поставить перед строкой `/n`

8. В каких вариантах ответа на экран выведется только число 5? (Несколько вариантов ответа).

А) `int number = 5;`

`std::cout<< number;`

Б) `std::cout<< 5`

В) `std::cout<< "5";`

Г) `int number = 5;`

`std::cout<< "number";`

Д) `int number = 5;`

`std::cin>> number;`

Е) `int number_1 = 2;`

`int number_2 = 3;`

`std:: cout<< number_1 + number_2;`

9. В каком из перечисленных вариантов ответа две строчки (части текста) выведутся на экран на разных строках? В таком виде:

Меня зовут Вася

Мне 10 лет

А) `std:: cout<< "Меня зовут Вася";`

`std:: cout<< "Мне 10 лет";`

Б) `std:: cout<< "Меня зовут Вася";`

`std:: cout<< "Мне 10 лет\n ";`

В) `std:: cout<< "Меня зовут Вася\n Мне 10 лет";`

Г) `std:: cout<< "Меня зовут Вася"\n;`

`std:: cout<< "Мне 10 лет ";`

10. В каком из перечисленных вариантов на экран выведется надпись «Каникулы начнутся через 10 дней»? (Несколько вариантов ответа).

А) `intnumber = 10;`

`std:: cout<<"Каникулы начнутся через number дней";`

Б) `intnumber = 10;`

`std:: cout<< "Каникулы начнутся через "<<number " дней";`

В) `intnumber = 10;`
`std:: cout<< ”;“Каникулы начнутся через “ <<number<< “ дней”;`

Г) `intnumber = 10;`
`std:: cout<< “Каникулы начнутся через “ number “ дней”;`

Д) `intnumber = 10;`
`std:: cout<< “Каникулы начнутся через>>number<< “ дней”;`

Е) `intnumber_1 = 5;`
`intnumber_2 = 5;`
`std:: cout<< “Каникулы начнутся через<<number_1 + number_2 <<“`
`дней”;`

Ж) `intnumber = 10;`
`std:: cout<< “Каникулы начнутся через \nnumber \ndней”;`

КЛЮЧИ

1. В	2. Б	3. В	4. Г	5. А,Б,Г,Ж,З,К
6. Г	7. А	8. А,В,Е	9. В	10. В,Е

3. ТЕСТОВЫЕ ЗАДАНИЯ ПО ТЕМЕ «УСЛОВНЫЕ КОНСТРУКЦИИ C++»

Инструкция по организации выполнения контрольных заданий

Задание содержит 10 тестовых вопросов с одним правильным вариантом ответа и ключи. Каждый учащийся получает свой вариант теста и выполняет его в любом порядке, отмечая правильные ответы в бланке, в течение 10 минут. Педагог проверяет правильность выполнения теста согласно ключам.

Содержание задания

1. Алгоритм, в котором в зависимости от выполнения некоторого условия совершается либо одна, либо другая последовательность действий.
 - А) линейный
 - Б) циклический
 - В) разветвляющийся
2. С какого ключевого слова начинаются условные конструкции?
 - А) else
 - Б) if
 - В) int
 - Г) string
3. Как переводится ключевое слово «if»?
 - А) пока

- Б) иначе
- В) также
- Г) если

4. Как переводится ключевое слово «else»?

- А) пока
- Б) иначе
- В) также
- Г) если

5. Как выглядит логический оператор «и»?

- А) <=
- Б) !=
- В) ||
- Г) &&

6. Как выглядит логический оператор «или»?

- А) <=
- Б) !=
- В) ||
- Г) &&

7. Какая из перечисленных записей условных конструкций не содержит ошибок?

А) `if (x>0);`
 {
 std::cout<< "Это число положительное";
 }

Б) `if (x>0)`
 {
 std::cout>> "Это число положительное";
 }

В) `else (x>0)`
 {
 std::cout<< "Это число положительное";
 }

Г) `if (x>0)`
 {
 std::cout<< "Это число положительное";
 }

8. Какая из перечисленных записей условных конструкций корректно проверяет, четное ли число x?

А) `if (x>2);`
 {
 std::cout<< "Это число четное";
 }

Б) `if (x%2)`
 {

```

std::cout<< “Это число четное”;
}
В) if (x%2==0)
{
std::cout<< “Это число четное”;
}
Г) if (x&&2==0)
{
std::cout<< “Это число четное”;
}

```

9. Как выглядит оператор «не равен/не равно»?

- А) !=
- Б) <=
- В) ||
- Г) &&

10. В какой из перечисленных условных конструкций при вводе чисел x равное 1 или x равное 2 будет выводиться надпись «Эти оценки нужно исправить», а при вводе любых других чисел программа будет завершаться?

```

А) if (x>0 && x<3)
{
std::cout<<“ Эти оценки нужно исправить ”;
}
Б) if (x>0 || x<3)
{
std::cout<<“ Эти оценки нужно исправить ”;
}
В) if (x>1&&x<2)
{
std::cout<<“ Эти оценки нужно исправить ”;
}
Г) if (x>1||x<2)
{
std::cout<<“ Эти оценки нужно исправить ”;
}

```

КЛЮЧИ

1. В	2. Б	3. Г	4. Б	5. Г
6. В	7. Г	8. В	9. А	10. А

4. КОНТРОЛЬНЫЕ ЗАДАНИЯ ПО ТЕМЕ «ВЫЧИСЛЕНИЕ. ЧИСЛА И СТРОКИ. УСЛОВНЫЕ КОНСТРУКЦИИ»

Инструкция по организации выполнения контрольных заданий

Файл содержит 4 варианта контрольных заданий по 5 задач в каждом. Каждый учащийся получает свой вариант задания. Задания построены по принципу «От простого – к сложному» (задача номер 1 – наиболее простая, задача номер 5 – наиболее сложная и объемная по записи). Учащиеся выполняют задания в любом порядке, но показывают правильный рабочий вариант педагогу после каждой выполненной задачи. Если программа работает некорректно, педагог сообщает об этом учащемуся, и задачу необходимо доработать. Таким образом, пока педагог не «примет» рабочий вариант задачи, к выполнению следующей задачи учащийся не переходит. Задача максимум – выполнить все 5 задач. Зачет – выполнены 4 задачи. Время выполнения – 2 учебных часа.

Содержание задания

ВАРИАНТ 1

№	ЗАДАЧА	ПРИМЕР
1	Расстояние от Тольятти до Москвы 1000 км. Напишите программу, которая рассчитывает, сколько времени потребуется водителю на машине, чтобы обратиться от Тольятти до Москвы и обратно.	<i>Эта программа рассчитывает, сколько времени вам потребуется, чтобы доехать от Тольятти до Москвы и обратно. Введите скорость машины: 50 -----Считаем----- Дорога туда и обратно займёт 40 часов</i>
2	Напишите программу, которая рассчитывает, сколько клиентов может обслужить один кассир за свою рабочую смену. Время обслуживания одного клиента складывается из времени, за которое клиент делает заказ, и времени, которое кассир собирает и отдает заказ. Время рабочей смены вводится в часах, но время заказа и сборки – в минутах. Все расчеты также производятся в минутах	<i>Эта программа рассчитывает, сколько клиентов успеет обслужить кассир за смену. Введите длительность смены в часах: 8 Сколько минут клиент делает заказ? 2 Сколько минут кассир собирает заказ? 4 -----Считаем----- За смену кассир успеет обслужить 80 клиентов.</i>
3	Пользователь вводит два числа. Программа вычисляет, <u>во сколько</u>	<i>Введите два числа: 25 5 Первое число больше второго в 5 раз</i>

	раз то число, которое больше, превосходит другое.	***** Введите два числа: 6 36 Второе число больше первого в 6 раз
4	Напишите программу, которая спрашивает имена и рост у трёх мальчиков в классе, и говорит, кто из них выше всех.	Введите имя и рост первого мальчика: Vasya 165 Введите имя и рост второго мальчика: Ivan 170 Введите имя и рост третьего мальчика: Vlad 158 Самый высокий мальчик это Ivan
5	Напишите программу, которая напоминает о домашнем задании для детей 1-4 класса по двум предметам: English и Math(английский/математика). По Английскому нужно выполнить упражнение 1 для 1 класса, упражнение 2 для 2ого класса, упражнение 3 для 3его, упражнение 4 для 4го. По математике тоже самое, но с примерами (1, 2, 3, 4). В случае ввода других классов и предметов выводилось «ошибка». Сделайте так, чтобы math, MATи Math (и то же самое с английским) считались как одно и то же слово	Введите свой класс: 1 Введите предмет: math Ваше домашнее задание: пример 1 ***** Введите свой класс: 2 Введите предмет: English Ваше домашнее задание: упражнение2

ВАРИАНТ 2

№	ЗАДАЧА	ПРИМЕР
1	Один урок в школе длится 40 минут. Вася решает 1 задачу по математике за 10 минут. Напишите программу, которая рассчитывает, сколько уроков нужно Васе, чтобы решить определённое количество задач (их вводит пользователь)	Эта программа рассчитывает, сколько уроков нужно Васе, чтобы решить задачи по математике. Введите количество задач: 20 -----Считаем----- Для решения 20 задач Васе потребуется 5 уроков.
2	Расстояние от Москвы до Турции 2000 км. Напишите программу, которая рассчитывает, какая	Эта программа рассчитывает, какая скорость должна быть у машины, чтобы доехать от Москвы до

	<p>скорость должна быть у машины, чтобы добраться от Москвы до Турции за определённое время (вводит пользователь).</p>	<p><i>Турции.</i> <i>Введите расчетное время в часах:</i> 20 -----Считаем----- <i>Скорость машины должны быть</i> 100 км/ч</p>
3	<p>Пользователь вводит два числа. Программа вычисляет, разность между большим и меньшим числом</p>	<p><i>Введите два числа: 25 5</i> <i>Разность первого и второго числа равна 20</i> ***** <i>Введите два числа: 6 36</i> <i>Разность второго и первого числа равна 30</i></p>
4	<p>Программа сравнивает, учебник по какому предмету служба доставки книжного магазина доставит быстрее всего. Пользователь вводит 3 товара и сроки их доставки. Программа выдает ответ.</p>	<p><i>Введите наименование первого учебника и срок его доставки: Math 3</i> <i>Введите наименование второго учебника и срок его доставки: English 5</i> <i>Введите наименование третьего учебника и срок его доставки: Physics 7</i> <i>Быстрее всего доставят учебник Math</i></p>
5	<p>Напишите программу, которая проверяет, знает ли пользователь таблицу умножения на 7, 8 или 9. Пользователь вводит, на какое число он хочет проверить свои знания (7, 8,9). Программа выдает ему пример на умножение выбранного числа на любое другое число (на ваше усмотрение). Пользователь пишет ответ. Программа выдает результат: верно/неверно.</p>	<p><i>Проверяем знания по таблице умножения! Введите число для проверки: 7, 8 или 9.</i> <i>Введите свой класс: 7</i> <i>Ваш пример: 7 * 6</i> <i>Введите ответ: 42</i> <i>Верно!</i> ***** <i>Проверяем знания по таблице умножения! Введите число для проверки: 7, 8 или 9.</i> <i>Введите свой класс: 8</i> <i>Ваш пример: 8 * 4</i> <i>Введите ответ: 42</i> <i>Неверно! Верный ответ: 32!</i></p>

ВАРИАНТ 3

№	ЗАДАЧА	ПРИМЕР
1	<p>Расстояние от Тольятти до Москвы 1000 км. Напишите программу, которая рассчитывает, сколько времени потребуется водителю на машине, чтобы обратиться от Тольятти до Москвы и обратно.</p>	<p><i>Эта программа рассчитывает, сколько времени вам потребуется, чтобы доехать от Тольятти до Москвы и обратно.</i> <i>Введите скорость машины: 50</i> <i>-----Считаем-----</i> <i>Дорога туда и обратно займёт 40 часов</i></p>
2	<p>Один урок в школе длится 40 минут. Вася решает 1 задачу по математике за 10 минут. Напишите программу, которая рассчитывает, сколько уроков нужно Васе, чтобы решить определённое количество задач (их вводит пользователь)</p>	<p><i>Эта программа рассчитывает, сколько уроков нужно Васе, чтобы решить задачи по математике.</i> <i>Введите количество задач: 20</i> <i>-----Считаем-----</i> <i>Для решения 20 задач Васе потребуется 5 уроков.</i></p>
3	<p>Пользователь вводит два числа. Программа вычисляет, <u>во сколько раз</u> то число, которое больше, превосходит другое.</p>	<p><i>Введите два числа: 25 5</i> <i>Первое число больше второго в 5 раз</i> <i>*****</i> <i>Введите два числа: 6 36</i> <i>Второе число больше первого в 6 раз</i></p>
4	<p>Напишите программу, которая спрашивает имена и рост у трёх мальчиков в классе, и говорит, кто из них выше всех.</p>	<p><i>Введите имя и рост первого мальчика: Vasya 165</i> <i>Введите имя и рост второго мальчика: Ivan 170</i> <i>Введите имя и рост третьего мальчика: Vlad 158</i> <i>Самый высокий мальчик это Ivan</i></p>
5	<p>Напишите программу, которая проверяет, знает ли пользователь таблицу умножения на 7, 8 или 9. Пользователь вводит, на какое число он хочет проверить свои знания (7, 8,9). Программа выдает ему пример на умножение выбранного числа на любое другое число (на ваше усмотрение). Пользователь пишет ответ. Программа выдает результат:</p>	<p><i>Проверяем знания по таблице умножения! Введите число для проверки: 7, 8 или 9.</i> <i>Введите свой класс: 7</i> <i>Ваш пример: 7 * 6</i> <i>Введите ответ: 42</i> <i>Верно!</i> <i>*****</i> <i>Проверяем знания по таблице умножения! Введите число для проверки: 7, 8 или 9.</i></p>

	верно/неверно.	<p><i>Введите свой класс: 8</i></p> <p><i>Ваш пример: 8 * 4</i></p> <p><i>Введите ответ: 42</i></p> <p><i>Неверно! Верный ответ: 32!</i></p>
--	----------------	--

ВАРИАНТ 4

1	<p>Пользователь вводит два числа. Программа вычисляет, разность между большим и меньшим числом</p>	<p><i>Введите два числа: 25 5</i></p> <p><i>Разность первого и второго числа равна 20</i></p> <p><i>*****</i></p> <p><i>Введите два числа: 6 36</i></p> <p><i>Разность второго и первого числа равна 30</i></p>
2	<p>Программа сравнивает, учебник по какому предмету служба доставки книжного магазина доставит быстрее всего. Пользователь вводит 3 товара и сроки их доставки. Программа выдает ответ.</p>	<p><i>Введите наименование первого учебника и срок его доставки: Math 3</i></p> <p><i>Введите наименование второго учебника и срок его доставки: English 5</i></p> <p><i>Введите наименование третьего учебника и срок его доставки: Physics 7</i></p> <p><i>Быстрее всего доставят учебник Math</i></p>
3	<p>Расстояние от Москвы до Турции 2000 км. Напишите программу, которая рассчитывает, какая скорость должна быть у машины, чтобы добраться от Москвы до Турции за определённое время (вводит пользователь).</p>	<p><i>Эта программа рассчитывает, какая скорость должна быть у машины, чтобы доехать от Москвы до Турции.</i></p> <p><i>Введите расчетное время в часах: 20</i></p> <p><i>-----Считаем-----</i></p> <p><i>Скорость машины должны быть 100 км/ч</i></p>
4	<p>Напишите программу, которая рассчитывает, сколько клиентов может обслужить один кассир за свою рабочую смену. Время обслуживания одного клиента складывается из времени, за которое клиент делает заказ, и</p>	<p><i>Эта программа рассчитывает, сколько клиентов успеет обслужить кассир за смену.</i></p> <p><i>Введите длительность смены в часах: 8</i></p> <p><i>Сколько минут клиент делает заказ? 2</i></p>

	времени, которое кассир собирает и отдает заказ. Время рабочей смены вводится в часах, но время заказа и сборки – в минутах. Все расчеты также производятся в минутах	<i>Сколько минут кассир собирает заказ? 4 -----Считаем----- За смену кассир успеет обслужить 80 клиентов.</i>
5	Напишите программу, которая напоминает о домашнем задании для детей 1-4 класса по двум предметам: English и Math(английский/математика). По Английскому нужно выполнить упражнение 1 для 1 класса, упражнение 2 для 2ого класса, упражнение 3 для 3его, упражнение 4 для 4го. По математике то же самое, но с примерами (1, 2, 3, 4). В случае ввода других классов и предметов выводилось «ошибка». Сделайте так, чтобы math, МATH и Math (и то же самое с английским) считались как одно и то же слово	<i>Введите свой класс: 1 Введите предмет: math Ваше домашнее задание: пример 1 ***** Введите свой класс: 2 Введите предмет: English Ваше домашнее задание: упражнение 2</i>

5. ПРОМЕЖУТОЧНАЯ ИТОГОВАЯ РАБОТА ПО ТЕМАМ «ТИПЫ ПЕРЕМЕННЫХ. ОПЕРАТОРЫ. УСЛОВИЯ. ЦИКЛЫ»

Инструкция по организации выполнения контрольных заданий

Задание содержит 10 вопросов открытого типа, которые учащиеся выполняют в любом порядке, записывая ответы в бланке. Время выполнения 10 вопросов – 15 минут.

Далее учащиеся переходят к 11 заданию, где они последовательно выполняют на компьютере задачи, которые в устной форме им сообщает педагог. Переход к выполнению следующей задачи осуществляется только после того, как выполнена предыдущая задача. Время выполнения задач – 20 минут.

Общее время выполнения – 40 минут (35 минут задания + 5 минут на включение компьютера и создание проектов)

Содержание задания

1. Перечислите знакомые вам типы переменных и их вид:

№	Название типа	Вид
1		
2		
3		
4		

2. Укажите перевод ключевых слов языка C++ и как (где/для чего) они используются:

№	Ключевое слово	Перевод	Назначение
1	if		
2	while		
3	for		
4	else		
5	break		

3. Укажите перевод и назначение следующих команд:

№	Ключевое слово	Перевод	Назначение
1	include iostream	include – iostream –	
2	using namespace std	using – namespace – std –	

4. Перечислите, как называются (за что отвечают) три части стандартного цикла for:

```
for (inti = 0; i<10; i++)  
{
```

```
}
```

1 (inti = 0) –

2 (i < 10) -

3 (i ++)-

5. Объявите переменную типа `char`, равную символу `f`:

6. Запишите операторы и укажите их вид: А – арифметический, Л – логический:

№	Оператор	Как выглядит	Вид: А/Л
1	Больше или равно		
2	Вычитание		
3	Умножение		
4	Остаток от деления		
5	Или		
6	Неравенство		
7	Декремент		
8	Равенство		
9	Деление		
10	Инкремент		
11	Меньше		
12	Сложение		
13	И		

7. Запишите вид полной и сокращенной формы условной конструкции, вместо кода (чисел/переменных) используя слова «условие» и «действие». Не забудьте про знаки препинания.

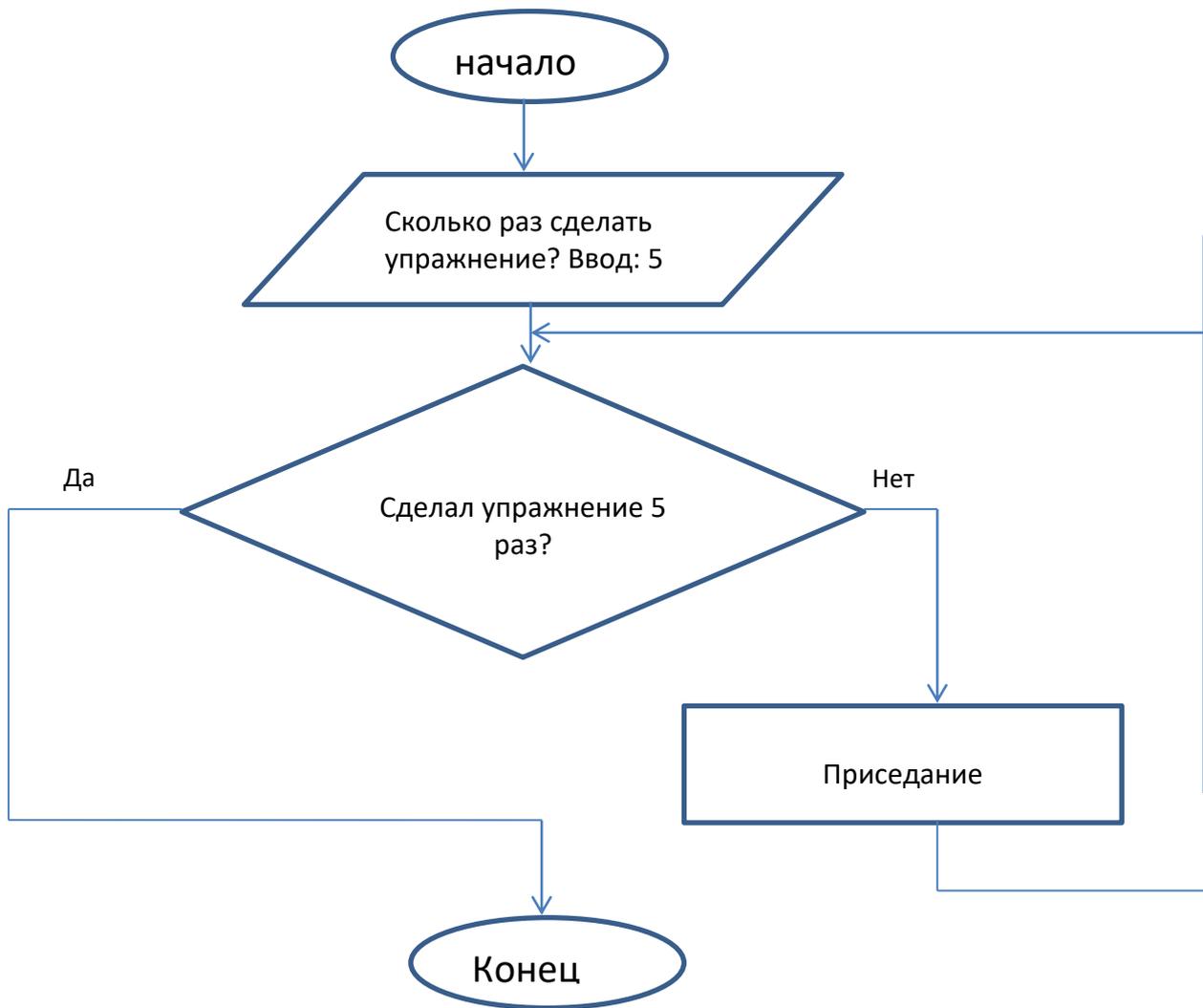
Полная форма условной конструкции	Сокращенная форма условной конструкции

8. Запишите вид циклов `while`, `dowhile`, вместо кода (чисел/переменных) используя слова «условие» и «действие». Не забудьте про знаки препинания.

Цикл	Запись
<code>while</code>	
<code>do while</code>	

9. Объявите числовую переменную, равную `0,5`:

10. Напишите цикл for для следующей блок-схемы:



Записать здесь:

11. Решение задач

Уровни задач (выполнение отмечает преподаватель):

Числа	Строки	Условные конструкции	Циклы	Комплекс

Задачи

Уровень 1. Числа.

Пользователь вводит 2 числа, найти сумму чисел

Уровень 2. Строки.

Пользователь вводит имя пользователя, страну назначения и номер рейса.

Программа выводит на экран: «Пассажир _____ летит в _____ рейсом номер _____».

Уровень 3. Условные конструкции.

Пользователь вводит 3 числа. Найти максимум из трех чисел.

Уровень 4. Циклы.

Пользователь вводит любое слово, а также количество раз, сколько это слово нужно повторить. На экран выводится слово в столбик столько раз, сколько указал пользователь, с нумерацией от 1 до указанного числа.

Уровень 5. Комплекс.

Создать программу с циклом, которая спрашивает имя пользователя и сколько ему лет, а затем выводит на экран «Через 5 лет пользователю _____ будет _____ лет» (к указанному возрасту прибавляется 5). Далее программа спрашивает, хочет ли пользователь продолжить. При ответе «да» вопросы повторяются, при ответе «нет» программа завершается.

6. КОНТРОЛЬНЫЕ ЗАДАЧИ ПО ТЕМЕ «РАБОТА С СИМВОЛАМИ»

Инструкция по организации выполнения контрольных заданий

Задание содержит 2 варианта контрольных заданий по 4 задачи в каждом. Задачи разбиты по уровням: первый уровень – самый простой, четвертый – самый сложный.

Задача учащегося – «прокачать» себя до 4 уровня (который назначается после выполнения 4ой задачи). Переход на следующий уровень осуществляется только после того, как учащийся получил предыдущий уровень.

Задания выполняются на компьютере. Педагог контролирует правильность выполнения.

Время выполнения - 2 учебных часа, после чего подводится итог, каких уровней достигли учащиеся.

Содержание задания

Задача. Вариант 1. Уровень 1.

Пользователь вводит строку – любое слово/набор символов без пробелов. Программа выдает, сколько букв в данном наборе символов.

Задача. Вариант 2. Уровень 1.

Пользователь вводит строку – любое слово/набор символов без пробелов. Программа выдает, что введено больше или меньше 10 символов.

Задача. Вариант 1. Уровень 2.

Пользователь вводит слово. Затем выводит символ (букву). Программа заменяет первую букву в слове на введенный символ и выводит на экран новое слово. После вывода программа спрашивает, хочет ли пользователь продолжить? Если да – снова спрашивает слово, если нет – завершение программы.

Например:

Введите слово: cat

Введите букву: t

cat

Хотите продолжить: y/n? y

Введите слово:

Задача. Вариант 2. Уровень 2.

Пользователь вводит слово. Затем выводит символ (букву). Программа заменяет вторую букву в слове на введенный символ и выводит на экран новое слово. После вывода программа спрашивает, хочет ли пользователь продолжить? Если да – снова спрашивает слово, если нет – завершение программы.

Например:

Введите слово: cat

Введите букву: u

cut

Хотите продолжить: y/n? y

Введите слово:

Задача. Вариант 1. Уровень 3.

Пользователь вводит пример на сложение или вычитание в формате: число – знак – число (в виде трёх символов).

Программа определяет, являются ли первый и третий символ числами, а второй символом знаком + или -. Если всё корректно, выдается ответ. Если некорректно, выдается сообщение, что запись некорректная.

Задача. Вариант 2. Уровень 3.

Пользователь вводит пример на умножение или деление в формате: число – знак – число (в виде трёх символов).

Программа определяет, являются ли первый и третий символ числами, а второй символ знаком * или /. Если всё корректно, выдается ответ. Если некорректно, выдается сообщение, что запись некорректная.

Задача. Вариант 1. Уровень 4.

Проверка корректности записи даты рождения. Дата должна быть введена в формате 01.01.1999 в виде строки

Проверка проходит по параметрам:

- количество символов – 10
- все символы, кроме Зего и бого – цифры
- Зий и бой символ – точки
- Месяцев не может быть больше 12, дней – больше 31.

Тем, что в некоторых месяцах кол-во дней – 30 или 28 – пренебрегаем.

Задача. Вариант 2. Уровень 4.

Проверка корректности записи даты рождения. Дата должна быть введена в формате 1999/01/30 в виде строки (год – месяц - день)

Проверка проходит по параметрам:

- количество символов – 10
- все символы, кроме 5ого и 8ого – цифры
- 8ой и 5ый символ – слеш
- Месяцев не может быть больше 12, дней – больше 31.

Тем, что в некоторых месяцах кол-во дней – 30 или 28 – пренебрегаем.

Ключи (код)

Задача. Вариант 1. Уровень 1.

```
#include<iostream>
using namespace std;
```

```
int main() {
string word;
```

```
cout<< "Введите слово: ";
cin>>word;
cout<< "В этом слове " <<word.length() << " символов";
}
```

Задача. Вариант 2. Уровень 1.

```
#include<iostream>
using namespace std;
```

```

int main() {
string word;

cout<< "Введите слово: ";
cin>> word;
if (word.length() > 10){
cout<< "В этом слове больше 10 символов";
}
else
{
cout<< "В этом слове меньше 10 символов";
}

}

```

Задача. Вариант 1. Уровень 2.

```

#include<iostream>
using namespace std;

int main() {
string word, answer;
char letter;
while (true){
cout<< "Введите слово: ";
cin>> word;
cout<< "Введите символ: ";
cin>> letter;
word[0] = letter;
cout<< word <<endl;
cout<< "Хотите продолжить? у/н ";
cin>> answer;
if (answer == "n"){
break;
}
}
}

```

Задача. Вариант 2. Уровень 2.

```

#include<iostream>
using namespace std;

int main() {
string word, answer;
char letter;

```

```
while (true){
cout<< "Введите слово: ";
cin>> word;
cout<< "Введите символ: ";
cin>> letter;
word[1] = letter;
cout<< word <<endl;
cout<< "Хотите продолжить? у/n ";
cin>> answer;
if (answer == "n"){
break;
}
}
}
```

7. ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ТЕМЕ «ИГРОВЫЕ СКРИПТЫ». ПРОЕКТИРОВАНИЕ МИНИ-ИГРЫ «ЛАБИРИНТ» В UNITY.

Инструкция по организации выполнения контрольных заданий

Учащиеся обобщают полученные знания по теме «Игровые скрипты» и по работе в движке Unity.

Задача учащихся – создать рабочий вариант мини-игры лабиринт, где игровой объект (кубик) ищет выход из лабиринта, передвигаясь вперед-назад и разворачиваясь вправо-влево при помощи клавиш «стрелки» на клавиатуре.

Последовательность работы прописана в файле. Учащимся нужно выполнить работу по этапам, указанным в файле. Правильность каждого этапа контролируется педагогом.

В процессе создания скриптов учащимся необходимо исправить ошибки, которые допущены в тексте скриптов в файле.

Время выполнения – 2 учебных часа.

Содержание задания

Задача: на основе имеющихся знаний спроектировать мини-игру в Unity, где игровой объект (куб) перемещается пользователем по лабиринту из точки входа к выходу. Игрок управляет кубом при помощи стрелок вверх, вниз, вправо, влево (либо W, A, S, D), где стрелки вверх-вниз – это движение вперед-назад, а вправо-влево – разворот в соответствующие стороны.

Работу выполняйте поэтапно. После каждого этапа – контроль педагога, который разрешает переход к следующему этапу.

Этап 1. Настройка сцены

- Откройте Unity, создайте проект.
- Создайте игровой объект Куб:(*GameObject – 3dobject – cube*). При помощи инструмента масштабирования (*Scaletool*) сделайте из него платформу, на которой будет располагаться будущий лабиринт.
- Настройте материал (цвет) вашей платформе. Для этого правой кнопкой клик в пространстве *Assets– Create– Material*. Цвет меняется в поле «*Albedo*». Созданный материал перетащить мышкой на платформу.
- Настройте камеру таким образом, чтобы в окне Gameвы видели куб почти что сверху (допускается небольшой наклон). Платформа будет выглядеть как плоский квадрат. Нужно, чтобы весь ваш будущий лабиринт хорошо просматривался на этой платформе. Камера настраивается путем ее перемещения и вращения (*MovetoolandRotatetool*).

Этап 2. Создание стен и препятствий лабиринта.

- Стены создаются также из кубов, которых нужно замасштабировать и разнести в разные места платформы (см Этап 1). Имейте в виду, что лабиринт должен иметь проход из точки входа к выходу. Желательно, чтобы этот проход был для игрока сложным.
- Настройте материал на стены (другой цвет, отличный от платформы, см Этап 1).

Этап 3. Создание игрока.

- Игрок – куб. Это нужно создать, замасштабировать **ПРОПОРЦИОНАЛЬНО** (куб не должен превратиться в прямоугольник) настолько, чтобы он помещался в проходы ваших стен и препятствий.
- Настройте материал на игрока (другой цвет, отличный от платформы, см Этап 1).

Этап 4. Написание начала скрипта

- Создайте C# скрипт с названием *move* (правой кнопкой клик в пространстве *Assets – Create – C# script*). Откройте его.
- Над блоком *Start* пропишите переменные на игровой объект, скорость линейную и скорость вращения (разворота).

ВНИМАНИЕ!!!!!! В приведенных ниже скриптах допущены синтаксические ошибки (пропуски знаков препинания) и некорректные типы переменных (тех, что вы уже знаете). Вам нужно исправить ошибки и написать правильный скрипт.

```
publicGameObjectplayer //переменная игрового объекта
publicstringspeed//переменная линейной скорости
publicfloatspeed_r//переменная скорости вращения
```

Этап 5. Определение игрока

- Пропишите, какой объект будет игроком (player). Нам нужно указать, что игрок – это тот объект, на котором висит скрипт (знаки препинания пропущены). Разместите эту строчку в нужном методе (Startили Update)

```
player = (GameObject)this.gameObject
```

Этап 6. Написание скрипта на движение вперед-назад

- Прописываем движение вперед-назад по двум кнопкам. Прописана только одна кнопка, вторую вы копируете и изменяете самостоятельно. В скрипте пропущены знаки препинания. Также пропущена нужная переменная (подчеркнутое слово). Разместите эти строчки в нужном методе (Startили Update):

```
if(Input.GetKey(KeyCode.UpArrow)
```

```
player.transform.position += player.transform.forward * линейнуюскорость * Time.deltaTime
```

- Примените скрипт к игровому объекту

Этап 7. Написание скрипта на разворот вправо-влево

- Прописываем движение вперед-назад по двум кнопкам. Прописана только одна кнопка, вторую вы копируете и изменяете самостоятельно. В скрипте пропущены знаки препинания. Также пропущена нужная переменная (подчеркнутое слово). Разместите эти строчки в нужном методе (Startили Update):

```
if(Input.GetKey(KeyCode.RightArrow)
```

```
player.transform.Rotate(Vector3.down * скорость разворота)
```

```
// down – вниз up – вверх right – правый left – левый
```

Этап 8. Проверка прохождения лабиринта

- Проконтролируйте, что ваш объект перемещается так, как нужно, и ему достаточно места для прохождения в проходах между препятствиями.

- Да, ваш куб проходит сквозь стены. Это нормально. Это происходит потому, что сейчас для Unity эти объекты – это не физические тела. Они не обладают никакими физическими свойствами, как притяжение, масса и т.п.

Этап 9. Что сделать, чтобы объекты не проходили сквозь друг друга

- Нужно, чтобы объекты (все, кроме платформы) стали физическими телами. Для этого нужно применить к этим объектам компонент Rigidbody(перевод – твердое тело).

компонент Rigidbody – Это по сути тоже скрипт, просто он встроен в Unity, и вам не нужно видеть его код. Как его применить?

Выделите нужный объект. Мышкой заходим в верхнем меню: Component – Physics–Rigidbody.

- Примените этот компонент на все объекты, кроме платформы.

- Проверьте.

Куб толкает и сбивает стены препятствий. Это происходит потому, что объекты стали физическими телами, и сейчас у них одинаковая масса. Мы можем увеличить массу стен, чтобы куб не сбивал стены.

- Выделите стену. Найдите в поле инспектора справа компонент Rigidbody, который висит на выделенном объекте. Найдите параметр Mass. Сейчас он у всех одинаковый – 1 кг. Измените массу стены на 100 кг.

- Измените массу всех стен.

Этап 10. Финальное тестирование игры

Игрок должен пройти из точки входа к выходу

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Орленко, П.А. С++ на примерах. Практика, практика и только практика. / П.А. Орленко, П.В. Евдокимов. – СПб. : Наука и техника, 2022. – 288 с.
2. Доусон, М. Изучаем С++ через программирование игр. / Майкл Доусон; пер. с англ. Е. Зазноба, О. Сивченко – СПб.: Питер, 2022. – 352 с.