Содержание

[ОБЩАЯ ИНФОРМАЦИЯ 2](#_heading=h.gjdgxs)

[Тема занятия](#_heading=h.30j0zll) 2

[Цели и задачи 2](#_heading=h.1fob9te)

[Ожидаемый результат 2](#_heading=h.3znysh7)

[Структура занятия 3](#_heading=h.tyjcwt)

[ОПИСАНИЕ ЗАНЯТИЯ 4](#_heading=h.3dy6vkm)

# ОБЩАЯ ИНФОРМАЦИЯ

### Тема занятия

### CLI. Случайные числа.

**Цели и задачи:**

* Рассказать о принципе работы со случайными числами
* Научиться работать с библиотекой cstdlib
* Научиться работать с файловыми потоками
* Научиться работать с ошибками, обрабатывая их с помощью изученных операторов

**По результатам занятия слушатель будет знать:**

* Как использовать механизмы исключений
* Как работать с посевом генератора случайных чисел

### Структура занятия

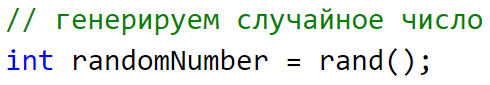
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тайминг занятия** | |  |  |  | Таблица 1 |
| № | Этапы | Что делает преподаватель | Что делает Слушатель | Время | Общее время |
| 1 | Приветственное слово преподавателя |  |  | 5 мин. | 5 мин. |
| 2 | Повторение пройденного |  |  | 5 мин. | 40 мин. |
| 3 | Теоретическая часть |  |  | 30 мин. |
| 4 | Вопросы по теоретической части |  |  | 5 мин. |
|  | *Перерыв* |  |  | 15 мин. | 15 мин. |
| 5 | Практическая часть |  |  | 40 мин. | 45 мин. |
| 6 | Рефлексия и вопросы |  |  | 5 мин. |

# ОПИСАНИЕ ЗАНЯТИЯ

**Теоретическая часть**

****

**Функция** - это блок кода, который срабатывает и возвращает значение. Для вызова функции пишется ее имя, за которым следует пара круглых скобок. Если функция вернет значение, то это значение можно присвоить переменной. Значение, возвращенное функцией rand() (то есть случайное число), присвоится переменной randomNumber



Функции могут принимать и такие значения, которые затем будут использовать в ходе работы. Вы указываете эти значения в круглых скобках после имени функции, разделяя значения запятыми. Эти значения называются аргументами функции. Указывая их, вы передаете (сообщаете) их функции. Я не передаю никакого значения функции rand( ), так как она не принимает аргументов.

**Посев генератора случайных чисел**

Компьютеры генерируют псевдослучайные числа, то есть числа, не являющиеся абсолютно случайными, руководствуясь при этом специальной формулой. Можно сказать, что компьютер читает значения в огромной книге заранее подготовленных чисел. При этом создается впечатление, что компьютер выдает последовательность абсолютно случайных чисел.

Однако здесь возникает проблема: дело в том, что компьютер всегда начинает читать эту книгу с начала. Поэтому он выдает в программе все время одну и ту же серию •случайных• чисел. В играх нас это не устраивает. Например, мы бы не хотели получать одинаковые последовательности бросков кости.

Чтобы устранить эту проблему, запрограммируем компьютер так: пусть он начинает •читать• книгу чисел с произвольного места. Указание этого места называется посевом генератора случайных чисел. Разработчики игр дают генератору случайных чисел исходное число, называемое зерном. Зерно соответствует начальной позиции в последовательности псевдослучайных чисел. Следующий код выполняет посев генератора случайных чисел:

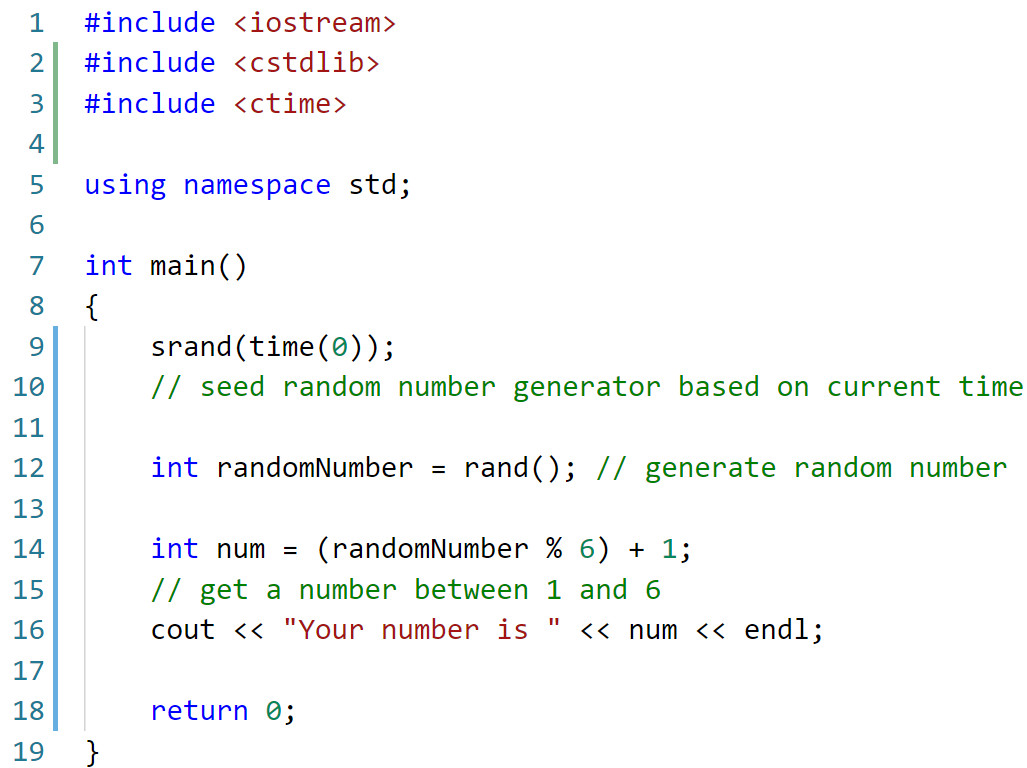


Расчет числа в заданном диапазоне

**Сгенерировав случайное** число, мы записываем в переменную randomNumber числовое значение из диапазона от О до 32 767 (именно таковы верхний и нижний пределы диапазона целых чисел в той реализации языка С++). Но мне требуется число в диапазоне от 1 до 6, поэтому далее стоит применить:



Любое положительное число, деленное на 6, даст остаток в диапазоне от О до 5. В приведенной строке кода берётся этот остаток и прибавляю к нему 1. В результате имеем положительное число в диапазоне от 1 до 6



Понятие об иrровом цикле

Игровой цикл - это обобщенное представление потока событий, происходящих в игре. В сумме эти события повторяются, именно поэтому мы и говорим здесь о цикле. Хотя реализация цикла от игры к игре может существенно различаться, фундаментальная структура цикла является практически одинаковой во всех играх любых жанров. Идет ли речь о тривиальном космическом шутере или о сложной ролевой игре (RPG), игровой процесс обычно можно разбить на одинаковые повторяющиеся компоненты, образующие игровой цикл.



О Установка параметров. На данном этапе игра принимает исходные параметры или загружает игровые ресурсы, например звуки, музыку и графику. Кроме того, пользователю могут быть изложены сюжет игры и ее цели.

О Получение пользовательского ввода. Программа принимает пользовательский ввод через то или иное устройство ввода: клавиатуру, мышь, джойстик, трекбол т. п.

О Обновление внутреннего состояния игры. В игровом мире срабатывают игровая логика и правила, при этом учитывается полученный ранее пользовательский ввод. В частности, речь может идти о системе игровой физики, определяющей правила взаимодействия объектов на экране, либо о вычислениях, связанных с искусственным интеллектом виртуального врага.

О Обновление изображения на экране. В большинстве компьютерных игр именно этот процесс является наиболее затратным для аппаратного обеспечения, поскольку связан с отрисовкой графики. Однако этот процесс может быть и тривиальным - сводиться к отображению текста.

О Проверка того, не окончена ли игра. Если игра не окончена (персонаж еще жив или пользователь пока не справился с задачей), процесс управления вновь откатывается на этап пользовательского ввода. Если игра окончена, цикл переходит к завершающей стадии.

О Завершение. На этом этапе игра окончена. Зачастую пользователю выдается какая-нибудь итоговая информация, например набранные им очки. При необходимости программа высвобождает все ресурсы и завершается.

**Практическая часть**

**Guess Му Number.**

Это классическая игра по угадыванию чисел. Компьютер выбирает случайное число, допустим, в диапазоне от 1 до 100, а игрок пытается угадать это число с наименьшего количества попыток. Всякий раз, когда пользователь вводит вариант, компьютер сообщает ему, насколько близко это число расположено к загаданному: слишком много, слишком мало или почти-почти. Как только пользователь угадает число, игра закончится.

