



Урок 32. C++: случайные числа и ветвление

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 32 из 34 · 45 минут · Модуль «Марафон: проверяемая учебная игра»

Уровень: начинающий программист; C++17.

Программа: <https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

При каждом испытании секрет меняется, и ошибка не повторяется. Как оставить режим псевдослучайной игры и одновременно проверять каждую ветвь?

Результат урока: Построить однопопытную учебную игру с явным seed и фиксированными секретами для тестов.

01. Проблемная ситуация

Секрет 5, догадки 2,5,8. За 5 минут запишите три ожидаемых ответа higher,win,lower и объясните, почему случайные повторы могут не проверить равенство. Игра без ставок, регистрации и личных данных.

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Генератор, распределение и независимый тестовый секрет

std::mt19937 — псевдослучайный движок, его начальное состояние задаёт seed.

uniform_int_distribution<int>(1,8) переводит значения движка в целый замкнутый диапазон 1...8.

Оператор остатка вместо распределения здесь не используется: мы не объявляем любой % равномерным генератором.

Ввод: seed целое от 0 до 1000000, forced от 0 до 8, guess от 1 до 8. forced=0 выбирает генерацию, forced=1...8 задаёт фиксированный секрет. Для отладки печатается secret=число; это учебный прозрачный режим, не защита секрета. Один запуск принимает одну догадку.

Одинаковый seed в одной используемой реализации и одинаковая последовательность вызовов воспроизводят результат. Мы не обещаем одинаковые значения распределения у любых стандартных библиотек. Для точных переносимых проверок ветвей используется forced. В фиксированном режиме генератор не выбирает секрет. Псевдослучайность не равна криптографической защите.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Для ввода 42 5 2 выводятся secret=5 и higher. Для 42 5 5 — secret=5 и win. Для 42 5 8 — secret=5 и lower. Во всех трёх seed одинаков, но секрет задан прямо, поэтому ожидания не зависят от реализации распределения.

```
#include <iostream>
#include <random>

int main() {
    unsigned seed;
    int forced, guess;
    if (!(std::cin >> seed >> forced >> guess) || seed > 1000000
        || forced < 0 || forced > 8 || guess < 1 || guess > 8) {
        std::cout << "error\n"; return 0;
    }
    std::mt19937 gen(seed);
    std::uniform_int_distribution<int> pick(1, 8);
    int secret = forced;
    if (secret == 0) secret = pick(gen);
    std::cout << "secret=" << secret << "\n";
    if (guess < secret) std::cout << "higher\n";
    else if (guess > secret) std::cout << "lower\n";
    else std::cout << "win\n";
}
```



C++: случайные числа и ветвление

Проверка входа	seed0...10 ⁶ ; forced0...8; guess1...8
Выбор секрета	forced=0 → pick(gen)
Тестовый режим	forced>0 → secret=forced
Ветви	guess<secret / == / >

forced=0 выбирает генератор; forced=1...8 задаёт тестовый секрет. Ниже, равно, выше дают higher, win, lower.

03. Практика 1. Проверяем материал

Три ветви и обе границы диапазона

14–25 минут. Реализуйте игру по контракту и таблице. Секрет не скрывается при отладке.

seed forced guess	Ожидаемые строки
42 5 2	secret=5; higher
42 5 5	secret=5; win
42 5 8	secret=5; lower
0 1 8	secret=1; lower
0 8 1	secret=8; higher
42 9 5	error

2 минуты — прогноз; 4 — код; 3 — фиксированные тесты и два запуска 42 0 4; 2 — сравнение повторов. Продукт: исходник и журнал. Критерии: все три ветви, обе границы, отказ вне диапазона, совпадение повторных запусков в данной среде.

04. Практика 2. Перенос способа

Расширение диапазона без несогласованных проверок

25–33 минуты. Новая версия использует диапазон 1...12 и тот же seed. За 2 минуты перечислите все места изменения; за 4 реализуйте; за 2 проверьте 1 12 12, 1 12 11, 1 1 12, 1 13 5 и 1 0 12. Критерии: обе проверки границ и распределение согласованы, три ветви доступны.

05. Обсуждение и границы вывода

33–40 минут. Что доказывают сто случайных запусков, а чего они не гарантируют? Сравните их с полным набором фиксированных пар forced/guess 1...8. Назовите отдельную проверку генерации.

06. Индивидуальная проверка

40–45 минут. Для forced=3 запишите подсказки для guess=1,3,8. Объясните роль forced=0 и назовите три верхние границы при переходе к 12. Четыре критерия.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Точные ответы фиксированных тестов даны в таблице. Для 42 0 4 проверяют secret в 1...8, согласованность подсказки и повторяемость, а не заранее придуманный номер. Ноль допустим только для forced, не для guess. Код ниже полный.

```
#include <iostream>
#include <random>

int main() {
    unsigned seed;
    int forced, guess;
    if (!(std::cin >> seed >> forced >> guess) || seed > 1000000
        || forced < 0 || forced > 8 || guess < 1 || guess > 8) {
        std::cout << "error\n"; return 0;
    }
    std::mt19937 gen(seed);
    std::uniform_int_distribution<int> pick(1, 8);
    int secret = forced;
    if (secret == 0) secret = pick(gen);
    std::cout << "secret=" << secret << "\n";
    if (guess < secret) std::cout << "higher\n";
    else if (guess > secret) std::cout << "lower\n";
    else std::cout << "win\n";
}
```

Практика 2: разбор и критерии

Фиксированные ответы: secret=12/win; secret=12/higher; secret=1/lower; error. Для случайного режима секрет должен лежать 1...12. Меняются верхние границы forced, guess и pick. Замена одного pick не делает тестовый ввод 12 допустимым.

```
#include <iostream>
#include <random>

int main() {
    unsigned seed;
    int forced, guess;
    if (!(std::cin >> seed >> forced >> guess) || seed > 1000000
        || forced < 0 || forced > 12 || guess < 1 || guess > 12) {
        std::cout << "error\n"; return 0;
    }
    std::mt19937 gen(seed);
    std::uniform_int_distribution<int> pick(1, 12);
    int secret = forced;
    if (secret == 0) secret = pick(gen);
    std::cout << "secret=" << secret << "\n";
    if (guess < secret) std::cout << "higher\n";
    else if (guess > secret) std::cout << "lower\n";
    else std::cout << "win\n";
}
```

Обсуждение: возможный вывод

Все 64 фиксированные пары покрывают отношения <,=,> и края, но не испытывают само преобразование движка. Повторные seed и проверка диапазона дополняют их. Ни один малый набор не доказывает безопасность или качество случайности для иных задач.

Индивидуальная проверка: ответ и критерии

higher, win, lower; forced=0 включает генерацию; меняют проверки forced и guess, а также границу распределения. Три подсказки вместе — 1 балл, остальные элементы — ещё 3.



Учителю: подготовка и план на 45 минут

Подготовьте компьютер, рабочую папку с правом записи, редактор, распечатку ученической части PDF, бумагу и ручку. Локальный компилятор G++ с C++17 или настроенная учителем IDE; сборка `g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson`. В Windows задайте `lesson.exe`. Запуск `./lesson` в Linux/macOS, `.\lesson.exe` в Windows. Для урока 31 дополнительно обычный браузер: открывать локальный SVG, не C++-исходник. Все входы и ожидаемые результаты даны. Установку и проверку запуска учитель выполняет до занятия; детские аккаунты и интернет не нужны.

Уровень — начинающий программист, 7 класс. Предпосылки: `if/else`, диапазоны и функции библиотеки. В паре автор вводит и объясняет, проверяющий сопоставляет с прогнозом, затем роли меняются. Без компьютера трассируйте полный код на бумаге, отмечайте координаты по схеме и проверяйте готовые результаты; учитель демонстрирует заранее собранный пример. Бумажная трассировка не выдаётся за реальную компиляцию или запуск ученика. План 45 минут включает чтение, ввод, сборку и обсуждение. В графических задачах сначала прогноз, затем сравнение. Матрица программы предназначена учителю.

Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ

07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Случайные и псевдослучайные числа. Условный оператор	Построить однопопытную учебную игру с явным seed и фиксированными секретами для тестов.	Самостоятельный продукт и тесты, описанные в практиках
Составлять и отлаживать простые диалоговые программы	Сверяет прогноз с фактическим запуском; объясняет ошибочный вариант	Журнал тестов и индивидуальный ответ по критериям

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

C++: равномерное целочисленное распределение

<https://eel.is/c++draft/rand.dist.uni.int>

C++: движок Mersenne Twister

<https://eel.is/c++draft/rand.eng.mers>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 17.10.2026.