

1. Тема урока/занятия. **«Сравнительный анализ эффективности алгоритмов Евклида нахождения наибольшего общего делителя двух целых чисел»**
2. Цель как формулировка конечного результата урока/занятия. **Углубить и усовершенствовать знания учащихся по математике и программированию посредством реализации алгоритмов нахождения наибольшего общего делителя двух чисел и проведения сравнительного анализа результатов работы рассматриваемых алгоритмов.**
3. Задачи как пути реализации цели урока/занятия. 1) **Изучение и реализация алгоритмов Евклида нахождения наибольшего общего делителя двух чисел**
4. Планируемые результаты. **Изучение алгоритмов, которые позволят расширить знания по программированию. (По итогам двух уроков и выполнения домашнего задания лучшие работы были отправлены на Всероссийский Конкурс, двое учеников получили дипломы 2ой и 3ей степени в номинации «IT-планета», методические разработки по этой теме были отправлены в качестве работы на тот же конкурс, за что был получен диплом I-ой степени в номинации «Творческие и педагогические работы»)**

5. Перечень оборудования кабинета к уроку/занятию, включая мультимедийное.

Персональный компьютер, проектор, экран для проектора, персональные компьютеры в количестве равном количеству учеников в классе (в группе/подгруппе)

6. Структура урока/занятия (план, отражающий этапы урока/занятия).

Данные разработки были применены в контексте спаренных уроков (2 академических часа подряд (40мин+40мин)):

В рамках апробации разрабатываемых ФГОС СПО по специальностям ТОП-50 был проведён урок с применением новых модулей «Разработка прикладных программных приложений:

5 минут — организационные моменты,

10 минут — проверка домашнего задания,

8 минут — исторические сведения

47 минут — основная часть урока

10 минут - рекомендации по выполнению домашнего задания

Исторические справки:

Евклид или **Эвклид** (др.-греч. Εὐκλείδης, от «добрая слава»^[1], время расцвета — около 300 года до н. э.) — древнегреческий математик, автор первого из дошедших до нас теоретических трактатов по математике.

Евклид — первый математик Александрийской школы. Его главная работа «Начала» (Στοιχεῖα, в латинизированной форме — «Элементы») содержит изложение планиметрии, стереометрии и ряда вопросов теории чисел.

Евклиду удалось систематизировать и доработать все знания по математике, накопленные к третьему веку до нашей эры. В своих трудах он изложил информацию таким образом, что стала видна их логическая связанность: для этого он включал недостающие фрагменты и самостоятельно разрабатывал доказательства. Его работа настолько обширна, что долгое время считалось, что такого человека и вовсе не существовало, а за его именем скрывается несколько мудрецов.

При создании Неевклидовой геометрии Лобачевский опирался на труды Евклида. Начала считаются одним из самых популярных памятников Древности.



Основная часть урока:

Наибольшим общим делителем (НОД) для двух целых чисел m и n называется наибольший из их общих делителей. Пример: для чисел 70 и 105 наибольший общий делитель равен 35. Наибольший общий делитель существует и однозначно определён, если хотя бы одно из чисел m или n не равно нулю.

Итак, в 6-ом классе мы изучали правило: чтобы найти НОД двух чисел, необходимо

- 1) разложить делители чисел на простые множители (вычисления удобно записывать с помощью вертикальной черты; слева от черты записываем делимое, справа — делитель.)
- 2) Далее в левом столбце записываем значения частных.

Нахождение НОД посредством разложения чисел на множители

Наибольший общий делитель может быть найден по разложениям чисел на простые множители. НОД двух целых положительных чисел a и b равен произведению всех общих простых множителей, находящихся в разложениях чисел a и b на простые множители.

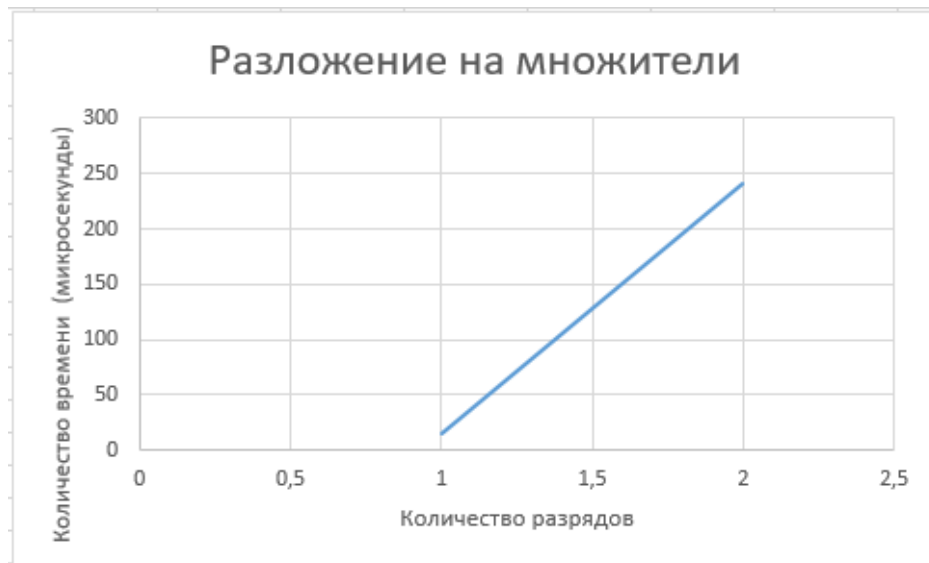
Если разложить числа a и b на простые множители и найти произведение всех их общих множителей, то этим будет найден наибольший общий делитель чисел a и b .

$$a = m_1 * m_2 * m_3 * \dots * m_N$$

m – простые числа

```
long gcd03(long a, long b) {
long nod = 1L;
if (a > b) {
long tmp = a;
a = b;
b = tmp;
}
while (a > 1L && b > 1L) {
for (long i = 2; i <= a; i++) {
if (a % i == 0 && b % i == 0) {
nod *= i;
a /= i;
b /= i;
break;
}
if (a % i == 0) {
a /= i;
break;
}
if (b % i == 0) {
b /= i;
break;
}
}
}
```

```
}  
return nod;  
}
```



Программа компилировалась под MS Visual Studio 2013 и TDM-GCC 4.8.1 в режиме 64-bit Release. Время работы программы при выборе 5 разрядов – 257,4 микросекунды. 10 разрядов – 219,5 микросекунд. 15 разрядов – 240,1 микросекунд.

Однако на практике вышеописанный метод используется редко, так как вычислительная сложность данного алгоритма слишком велика. Для нахождения НОД были разработаны более эффективные алгоритмы, вычислительная сложность которых существенно ниже.

Рассмотрим наиболее часто используемые алгоритмы нахождения НОД:

Перебор от минимального числа

Метод перебора делителей меньшего числа для нахождения наибольшего общего делителя (НОД) натуральных чисел.

1. Найти делители меньшего из данных чисел.
2. Найти, начиная с большего, тот из выписанных делителей, который является также делителем другого числа.
3. Записать найденное число – НОД.

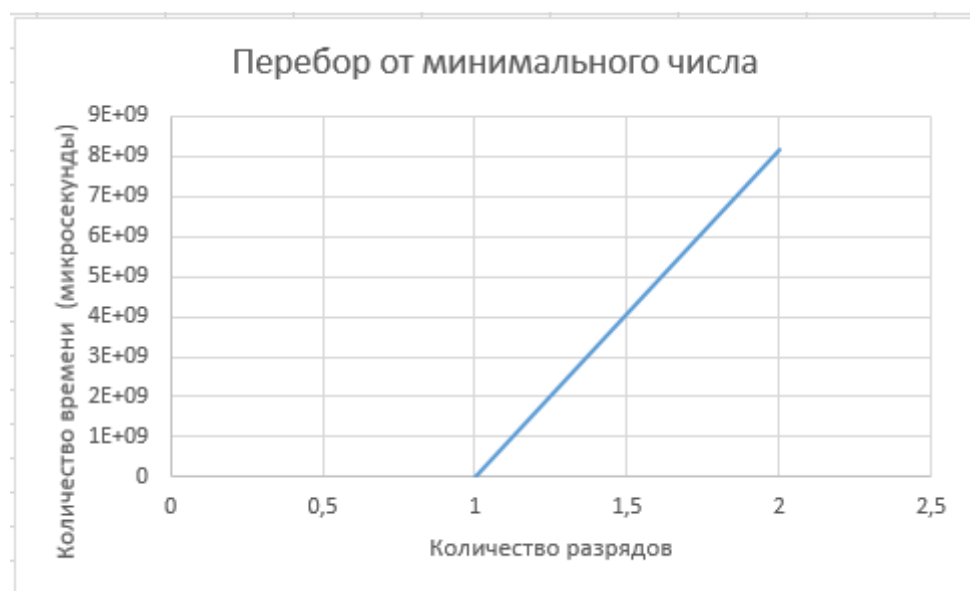
Например: Найти НОД(12;32).

$D(12) = \{1; 2; 3; 4; 6; 12\}$. 12 не является делителем числа 32; 6 не является делителем числа 32; 4- делитель 32. НОД(12; 32) = 4.

Этот алгоритм легко реализуется, но его недостатком является то, что необходимо проверять много вариантов.

```
long min(long a, long b) {  
    return a > b ? b : a;  
}  
  
long gcd02(long a, long b) {  
    long nod = 1L;  
    for (long i = min(a, b); i > 0; i--) {  
        if (a % i == 0 && b % i == 0) {  
            nod = i;  
            break;  
        }  
    }  
    return nod;  
}
```

Для вычисления минимального числа для пары чисел нужно добавить функцию и инициализировать переменную цикла меньшим из двух чисел.



Программа компилировалась под MS Visual Studio 2013 и TDM-GCC 4.8.1 в режиме 64-bit Release. Время работы программы при выборе 5 разрядов – 283,6 микросекунд. 10 разрядов – 720349815 микросекунд. 15 разрядов – 8165887655 микросекунд.

Алгоритм Евклида (итерационный)

Алгоритм Евклида — эффективный алгоритм для нахождения наибольшего общего делителя двух целых чисел (или общей меры двух отрезков). Алгоритм назван в честь греческого математика Евклида, который впервые описал его в VII и X книгах «Начал». В самом простом случае алгоритм Евклида применяется к паре положительных целых чисел и формирует новую пару, которая состоит из меньшего числа и разницы между большим и меньшим числом. Процесс повторяется, пока числа не станут равными. Найденное число и есть наибольший общий делитель исходной пары.

В цикле, пока первое из чисел не обнулится, делается следующее: если первое из чисел больше второго, они меняются местами; затем из второго вычитается первое. Когда цикл завершится, первое число обратится в нуль, а второе и будет искомым НОД, оно и возвращается из процедуры.

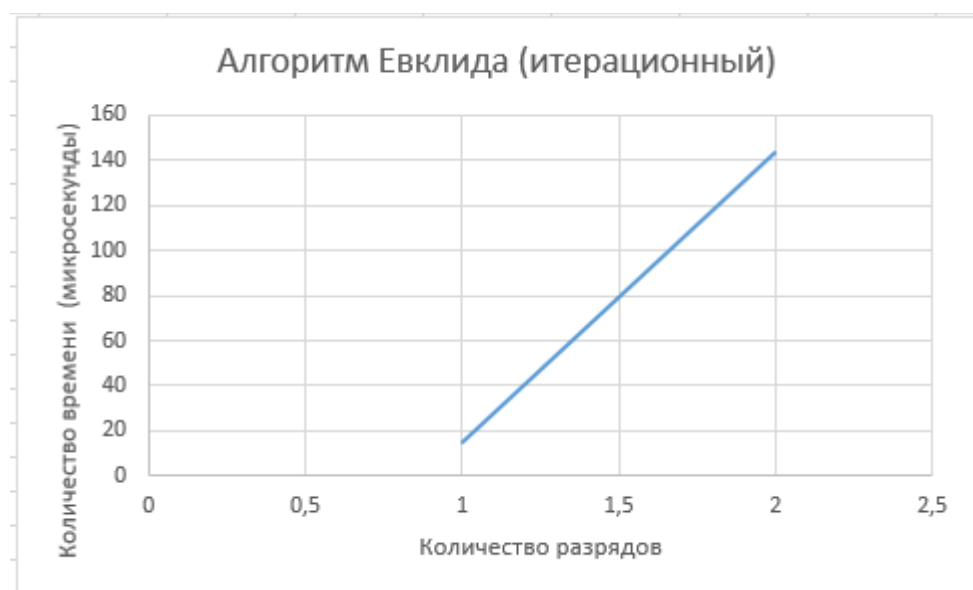
Итерационная версия:

```
long gcd07(long a, long b) {
    long nod = 1L;
    long tmp;
    if (a == 0L)
        return b;
    if (b == 0L)
        return a;
    if (a == b)
        return a;
    if (a == 1L || b == 1L)
        return 1L;
    while (a != 0 && b != 0) {
        if (a % 2L == 0L && b % 2L == 0L) {
            nod *= 2L;
            a /= 2L;
            b /= 2L;
            continue;
        }
        if (a % 2L == 0L && b % 2L != 0L) {
```

```

a /= 2L;
continue;
}
if (a % 2L != 0L && b % 2L == 0L) {
b /= 2L;
continue;
}
if (a > b) {
tmp = a;
a = b;
b = tmp;
}
tmp = a;
a = (b - a) / 2L;
b = tmp;
}
if (a == 0)
return nod * b;
else
return nod * a;
}

```



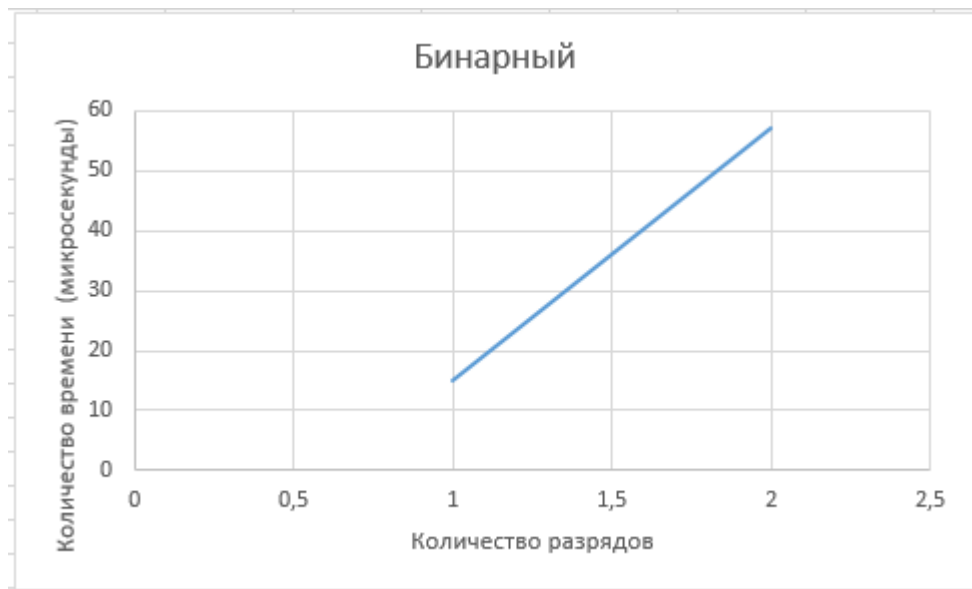
Программа компилировалась под MS Visual Studio 2013 и TDM-GCC 4.8.1 в режиме 64-bit Release. Время работы программы при выборе 5 разрядов – 135,8 микросекунд. 10 разрядов – 142,8 микросекунд. 15 разрядов – 143,2 микросекунд.

Бинарный алгоритм Евклида

Отличительная особенность бинарного алгоритма — возможность использования битовых сдвигов вместо медленных операций деления и умножения.

В этой версии функции операции деления и умножения на 2 заменены арифметическими сдвигами, а проверка на чётность – на проверку младшего бита числа.

```
long gcd06(long a, long b) {
if (a == 0L)
return b;
if (b == 0L)
return a;
if (a == b)
return a;
if (a == 1L || b == 1L)
return 1L;
if (a % 2L == 0L && b % 2L == 0L)
return 2L * gcd06(a / 2L, b / 2L);
if (a % 2L == 0L && b % 2L != 0L)
return gcd06(a / 2L, b);
if (a % 2L != 0L && b % 2L == 0L)
return gcd06(a, b / 2L);
if (a < b)
return gcd06((b - a) / 2L, a);
else
return gcd06((a - b) / 2L, b);
}
```

Программа компилировалась под MS Visual Studio 2013 и TDM-GCC 4.8.1 в режиме 64-bit Release. Время работы программы при выборе 5 разрядов – 12 микросекунды. 10 разрядов – 20 микросекунд. 15 разрядов – 57 микросекунд.

7. Содержание этапов урока/занятия (конспект) с указанием способов применения электронного контента (электронных учебников, электронных приложений, интерактивных игр, интерактивных учебных курсов, презентаций, видеороликов, мультфильмов, слайд-шоу и др.) и конкретных электронных/мультимедийных технических средств на уроке/занятии.

Теоретическая информация, представленная в ходе уроков по данной теме, воспроизводится исключительно с помощью проектора и экрана, что позволяет показывать особенности реализации алгоритмов на языке C# интерактивно, при этом возможно объяснять принципы алгоритмов и сразу же показывать их реализацию, пользуясь современными инструментами программирования.

8. Инструкции к выполнению домашнего задания, в т.ч. с применением компьютера и других электронных средств.

На предыдущих уроках мы разбирали способы реализации рекурсивных функций посредством использования средств C#. Мы реализовали рекурсивные алгоритмы для вычисления очередного числа ряда Фибоначчи, факториала заданного числа, функцию возведения в степень. В качестве домашнего задания На основе полученных знаний реализовать Алгоритм Евклида (рекурсивный).

В самом простом случае алгоритм Евклида применяется к паре положительных целых чисел и формирует новую пару, которая состоит из меньшего числа и разницы между

большим и меньшим числом. Процесс повторяется, пока числа не станут равными.

Найденное число и есть наибольший общий делитель исходной пары.

Для выполнения домашнего задания потребуется ПК и программное обеспечение Visual Studio.

9. К конспекту прилагается описание достигнутых результатов по каждому уроку/курсу.

<http://www.талантум.рф/itogi8>

Результаты: Ахметшин Нияз - 2 место (номинация: IT-планета), Узлов Даниэль (номинация: IT-планета)- 3 место, Коростелева Диана - 1 место (номинация: Творческие и педагогические работы)

10. К конспекту прилагается список информационных источников, использованных при подготовке и во время проведения урока/занятия

Клейнберг Дж., Тардос Е. Алгоритмы: разработка и применение.

Павловская Т.А.С#. Программирование на языке высокого уровня.

11. К конспекту прилагается рефлексия педагога по каждому уроку/занятию (что удалось и что не удалось при внедрении информационно-коммуникационных технологий, причины неудач, что хотелось бы изменить).

Каждая программа по реализации алгоритмов нахождения НОД была оценена, итоговые оценки были выставлены в журнал. По результатам опроса, 100% учащихся отметили, что урок был полезен и были углублены знания как в области программирования, так и в области IT. Посредством реализации данных алгоритмов также были изучены средства языка С#: использование циклов, особенности реализации функций (в том числе рекурсивных функций), особенности передачи переменных в функцию в качестве параметров, вызов функции, ввод и вывод информации с помощью консоли.