

Давай, жми клавишу!

Зайцев Никита Алексеевич

создатель онлайн антивируса [Meteor Security](#)



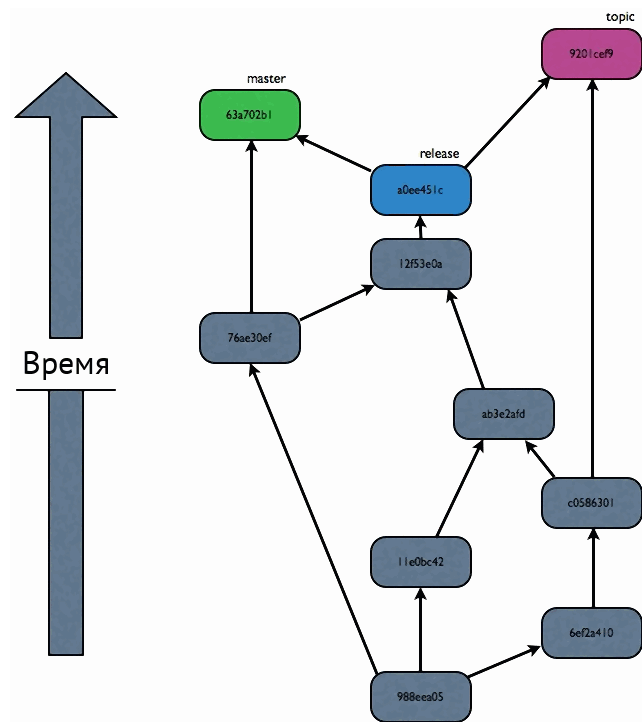
Основная проблема - передача кода

Система контроля версий

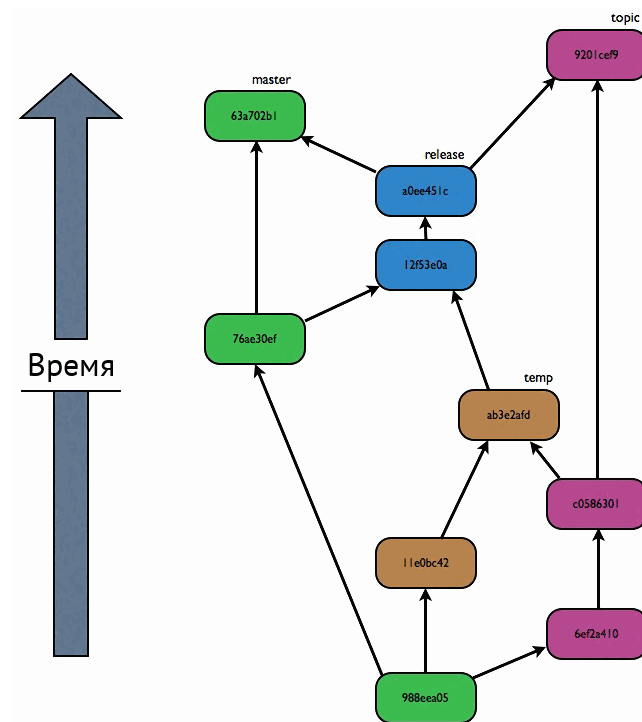
Системы контроля версий

- **Git**
- **Mercurial**
- **Subversion**

В чем различия?



git



mercurial

Что будем использовать мы?



Почему именно git?

Git это:

- **Стильно**
- **Модно**
- **Молодежно**
- **Меня заставили**

Как это работает?



Как установить?

- <https://desktop.github.com/> с GUI от GitHub
- <http://git-scm.com/download/win> официальный GIT

Хостинги для git репозиториев

- Bitbucket
- Github
- Sourceforge
- GitLab
- Google Code
- etc...

Популярные хостинги

- **Bitbucket**
- **Github**

В чем отличия?



VS



GitHub (<https://github.com>)

- **Известность**
- **Удобные тарифные планы**
- **Современность**
- **Удобство**

BitBucket (<https://bitbucket.org>)

- Бесплатность
- Безлимитные приватные репозитории
- Сложность интерфейса
- Принадлежит Atlassian

Что выбираем мы?

Select Repository Host

(Next step: Select repository)

GitHub

Connect your GitHub account



Bitbucket

Connect your Bitbucket account





Хорошо, решили. А как работать то?

- Каждое изменение должно сопровождаться "пушом"
- Не забывайте получать изменения в коде, сделанные вашими товарищами
- Что-то не получается? Открывай документацию и учи!

Основные команды

- `git init` - инициализировать готовый репозиторий
- `git push` - отправить изменения
- `git pull` - получить изменения
- `git checkout -b "имя ветки"` - создать ветку и перейти к ней
- `git merge "название ветки для слияния"` - слияние веток

Что такое ветка?

Для чего она нужна?

Практика

- **Регистрация в GitHub**
- **Создание первого репозитория**
- **Создание первого кода**
- **Отправка первого кода**

**Алгоритмы ветвления и оператор
выбора, циклы, обработка строк,
массивов и файлов**

Алгоритмы ветвления

```
if [( )<логическое выражение>( )] [then]  
  <оператор 1>  
[else <оператор 2>]
```

Оператор выбора

«На вход подается цифра, а на выходе должно быть название этой цифры».

Решение:

```
readln(a);
if a = 0 then writeln('ноль') else
  if a = 1 then writeln('один') else
    if a = 2 then writeln('два') else
      if a = 3 then writeln('три') else
        if a = 4 then writeln('четыре') else
          if a = 5 then writeln('пять') else
            if a = 6 then writeln('шесть') else
              if a = 7 then writeln('семь') else
                if a = 8 then writeln('восемь') else
                  if a = 9 then writeln('девять');
```

Оператор выбора

```
case <выражение> of  
  Значение1 [, значение11]: оператор1;  
  [Значение2 [, значение21]: оператор2;]  
  [Значение3..значение5]: оператор3;  
  . . .  
  [else: операторElse1;]  
end;
```

Верное решение

```
Readln(a);  
Case a of  
0: writeln( 'ноль' );  
1: writeln( 'один' );  
2: writeln( 'два' );  
3: writeln( 'три' );  
4: writeln( 'четыре' );  
5: writeln( 'пять' );  
6: writeln( 'шесть' );  
7: writeln( 'семь' );  
8: writeln( 'восемь' );  
9: writeln( 'девять' );  
end;
```

Циклы

Циклы

- С предусловием (while)
- С постусловием (until)
- Со счётчиком (for)
- Совместный цикл (for)
- С выходом из середины (break)

Цикл с предусловием

```
while <условие> do  
begin  
    <тело цикла>  
end;
```


Цикл с постусловием

```
repeat  
    <тело цикла>  
until <условие выхода>
```

Цикл со счётчиком

```
for i:=1 to 9 do  
begin  
    //тело цикла  
end;
```

Совместный цикл

```
for item in [1..100] do  
begin  
    //тело цикла  
end;
```

Цикл с выходом из середины

```
while(n div 10 <> 0) do begin
    if (N mod 10 = 0) then
        break else
            writeln('Do it again!');
end;

writeln(N);
```

Строки

- тип данных, значениями которого является произвольная последовательность (строка) символов алфавита. Каждая переменная такого типа (строковая переменная) может быть представлена фиксированным количеством байтов либо иметь произвольную длину.
- неограниченный массив символьного типа

Строковые функции

Функции преобразования

- `IntToStr(N: Integer): String` - преобразование числа в строку
- `StrToInt(S: String): Integer` - преобразование строки в число
- `FloatToStr(X: Extended): String` - преобразование числа с плавающей точкой в строку
- `StrToFloat(S: String): Extended` - преобразует строку в число с плавающей точкой

Функции преобразования

- `AnsiLowerCase(const S: String): String` - преобразование строки к нижнему регистру
- `AnsiUpperCase(const S: String): String` - преобразование строки к верхнему регистру
- `Length(const S: String): Integer` - измеряет длину строки
- `Trim(const S: String): String` - удаляет из строки начальные и завершающие пробелы и **управляющие символы**
- `TrimLeft(const S: String): String` - удаляет из строки **начальные** пробелы и управляющие символы
- `TrimRight(const S: String): String` удаляет из строки **завершающие** пробелы и управляющие символы

Функции сравнения

- `AnsiCompareStr(const S1, S2: String): Integer` -
Сравнивает две строки S1 и S2 с учётом регистра символов. Возвращает значение < 0 если $S1 < S2$, 0 если $S1=S2$, > 0 если $S1 > S2$
- `AnsiCompareText(const S1, S2: String): Integer` -
Сравнивает две строки S1 и S2 без учёта регистра символов. Возвращает значение < 0 если $S1 < S2$, 0 если $S1=S2$, >0 если $S1 > S2$

Прочие функции

- `Pos(Substr: String; Str: String): Integer` - поиск первого вхождения символа или строки.
- `Insert(Source: String; var S: String; Index: Integer): Integer` - вставка строки(символа) в строку начиная с **N** элемента
- `Delete(var S: String; Index, Count: Integer)` - удаление из строки **COUNT** символов начиная с **INDEX** позиции
- `Copy(S: String; Index, Count: Integer): String` - копирует из строки подстроку начиная с **INDEX** номера на **COUNT** количество символов

Задача 1:

Создать приложение, предлагающее пользователю ввести строку, определяющее количество символов в ней и выводящее результат, согласуя окончания различных падежей слова "символ" с числом символов, например: "В тексте 1 символ", "В тексте 32 символа", "В тексте 47 символов".

Примерный интерфейс

Form1

— □ ×

Рассчитать

Результат: число

Код программы

```
begin
N:=Length(Edit1.Text); //определить число СИМВОЛОВ
Label2.Caption = 'В тексте ';
if N>20 then N:=N mod 10; //Вытащить последнюю цифру

case N of
    1 : S:=' СИМВОЛ';
    2..4 : S:=' СИМВОЛА';
    0,5..20 : S:=' СИМВОЛОВ';
end;
Label2.Caption:=Label2.Caption
+ IntToStr(Length(Edit1.Text))+s;
end;
```

Задача 2

Создать приложение, которое предлагает ввести строку, а затем предлагает выбрать, какие символы будут заменены. Вариант по замене символов должен быть в виде CheckBox'ов, то есть если CheckBox в положение Checked(стоит галочка) значит условие обрабатывается, иначе игнорировать. А также вывести количество измененных символов

Примерный интерфейс

Делать его мне было лень, поэтому он будет нарисован на доске

Код программы

```
var S: string; //наша строка
    N: integer; //количество измененных символов
begin
N := 0; //Всегда обнуляй переменные!!
Edit3.Text := '';
Edit2.Text := '';
S := Edit1.Text; //Запоминаем строку
If CheckBox2.Checked then
while Pos('a', S) > 0 do //пока в тексте есть символ 'a'
begin
    N:=N+1; //Добавляем изменение
    S[Pos('a', S)] := 'б'; //Заменить символ 'a' на 'б'
end;
if CheckBox3.Checked then
while Pos(' ', S) > 0 do //Пока в строке есть пробелы
begin
    N:=N+1;
    S[Pos(' ', S)] := '-'; //заменить пробел на -
end;
if CheckBox1.Checked then Edit3.Text:=IntToStr(N); //Вывести
Edit2.Text:=S; //вывести результат
end;
```


На следующем занятии

- **Массивы**
- **Файлы**
- **Обработка строк в виде массива**

Самое важное за сегодня!

- Весь код должен быть в системе контроля версий
- Все вопросы вы можете задавать в личные сообщения или на email nickzaytsew@gmail.com
- Помощь с кодом только в системе контроля версий!!

FINISH

Всем спасибо, пока!

