

Давай, жми клавишу!

Зайцев Никита Алексеевич

создатель онлайн антивируса [Meteor Security](#)



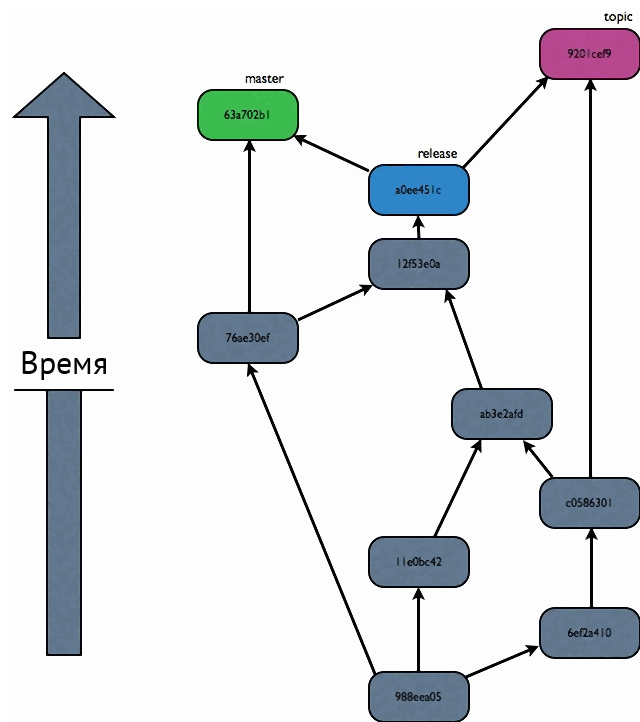
Основная проблема - передача кода

Система контроля версий

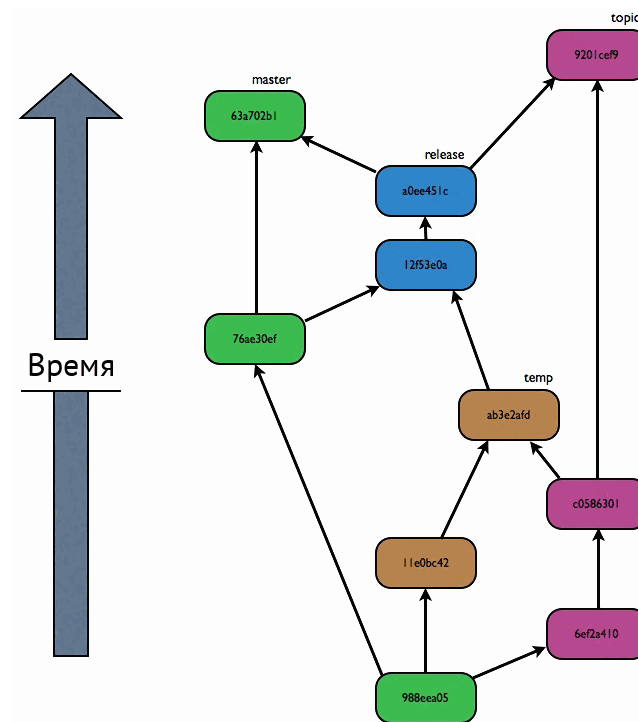
Системы контроля версий

- **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 $S1 > S2$

ISO 8859-5 Cyrillic

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 000	STX 001	SOT 002	ETX 003	EOT 004	ENQ 005	ACK 006	BEL 007	BS 008	HT 009	LF 010	VT 011	FF 012	CR 013	SO 014	SI 015
10	DLE 016	DC1 017	DC2 018	DC3 019	DC4 020	NAK 021	SYN 022	ETB 023	CAN 024	EM 025	SUB 026	ESC 027	FS 028	GS 029	RS 030	US 031
20	SP 032	!		#	\$	%	&	()	*	+	,	-	.	/	
30	0 048	1 049	2 050	3 051	4 052	5 053	6 054	7 055	8 056	9 057	:	;	<	=	>	? 063
40	@ 064	A 065	B 066	C 067	D 068	E 069	F 070	G 071	H 072	I 073	J 074	K 075	L 076	M 077	N 078	O 079
50	P 080	Q 081	R 082	S 083	T 084	U 085	V 086	W 087	X 088	Y 089	Z 090	[091	\ 092] 093		
60		a 096	b 097	c 098	d 099	e 100	f 101	g 102	h 103	i 104	j 105	k 106	l 107	m 108	n 109	o 110
70	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	{ 123	 124	}	~ 126	
80																
90																
A0		Ё 160	Ђ 161	Ѓ 162	Є 163	Ѕ 164	І 165	Ї 166	Ї 167	Ї 168	Ї 169	Ї 170	Ї 171	Ї 172	Ї 173	Ї 174
B0		А 176	Б 177	В 178	Г 179	Д 180	Е 181	Ж 182	З 183	И 184	Й 185	К 186	Л 187	М 188	Н 189	О 190
C0		Р 192	С 193	Т 194	У 195	Ф 196	Х 197	Ц 198	Ч 199	Ш 200	Щ 201	Ъ 202	Ы 203	Ь 204	Э 205	Ю 206
D0		а 208	б 209	в 210	г 211	д 212	е 213	ж 214	з 215	и 216	й 217	к 218	л 219	м 220	н 221	о 222
E0		р 224	с 225	т 226	у 227	ф 228	х 229	ц 230	ч 231	ш 232	щ 233	ъ 234	ы 235	ь 236	э 237	ю 238
F0		ё 240	ђ 241	ѓ 242	є 243	ѕ 244	і 245	ї 246	ї 247	ї 248	ї 249	ї 250	ї 251	ї 252	ї 253	ї 254

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!

