

# Раздел 1: «Общие навыки»

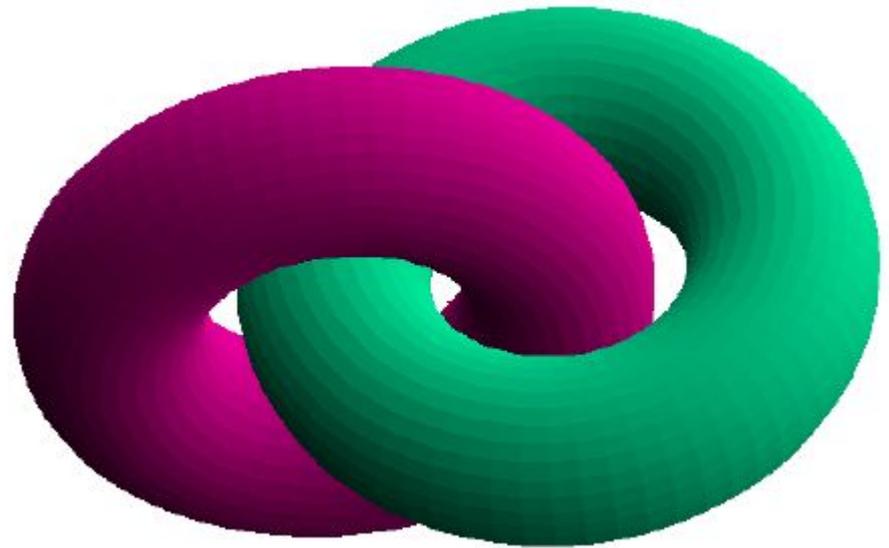
## Тема 2

### «Функциональное и доменное тестирование»

**ИЛИ**

**профессиональный подход  
к выбору стратегии и  
средств**

**Функциональное  
тестирование и его связь с  
другими видами  
тестирования**



Ранее мы определили функциональное тестирование (functional testing) как «проверку на соответствие требованиям и спецификациям».



Теперь мы посмотрим на этот вопрос более пристально.



**Функциональное тестирование** (functional testing) – проверка на соответствие требованиям и спецификациям.

Может быть **ручным** (manual testing) или **автоматизированным** (automated testing).

## Функциональное тестирование

(functional testing) – вид тестирования, направленный на исследование отдельных (изолированных) функций приложения.



Здесь нет противоречия с предыдущим определением. Мы лишь конкретизировали область применения функционального тестирования в чистом виде.

# Тогда что же относится к нефункциональному тестированию?

Те виды тестирования, которые ставят во главу угла проверку некоторых **нефункциональных требований**.

Однако, очень часто **нефункциональные требования** невозможно проверить в отрыве от выполнения функции.



# Посмотрим на нефункциональные виды тестирования...

Вид тестирования	«Нуждается» в ФТ
Доменное тестирование	Очень сильно
Тестирование на основе спецификаций	Очень сильно
Тестирование на основе рисков	Очень сильно
Стрессовое тестирование	Очень сильно
Исследовательское тестирование	Очень сильно
Сценарное тестирование	Очень сильно
Тестирование интернационализации	Сильно
Тестирование локализации	Сильно
И т.д. 😊	Сильно 😊

# Какие бывают тесты

## Основные виды тестов:

- позитивные;
- негативные.

## Направления тестирования:

- статическое;
- динамическое.

## Методы тестирования:

- чёрный ящик;
- белый ящик;
- серый ящик.

## Виды тестирования:

- инсталляционное;
- регрессионное;
- нового функционала;
- конфигурационное;
- совместимости;
- удобство использования;
- интернационализации;
- локализации;
- исследовательское.

Подписаться на рассылку

Подписаться

**Тогда зачем всё это!?**

**Каждый вид/метод...**

**Хорош для своих целей**

**Пропускает часть ошибок**

**Не может заменить собой другие**

**Опытный тестировщик**

**Может использовать разные виды/методы**

**Может выбирать лучший вид/метод**

# К чему мы готовимся?

## Типичные вопросы на собеседовании:

- Назовите плюсы и минусы функционального тестирования.
- Что такое классы эквивалентности и граничные условия?
- Назовите и объясните типичные тесты для следующих полей: текстовое, числовое, даты, двух связанных дат и т.п.
- Определите переменные, представленные в требовании или на рисунке. Определите, какие из них являются независимыми, а какие – связанными. Сформируйте чек-лист.

# Итак, о функциональном тестировании

## Основная цель

тестировать  
каждую функцию в  
отдельности

## Источники информации для тестов

экран приложения,  
каждое отдельное  
поле

меню, каждый  
пункт в  
отдельности

# «+» и «-» функционального тестирования

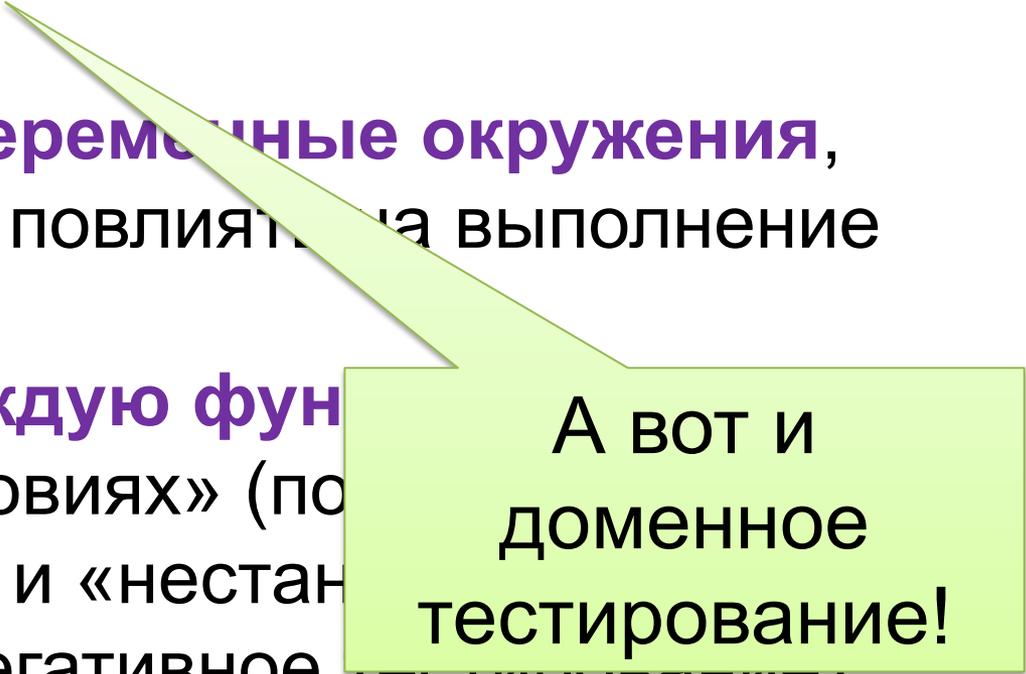
Глубокий анализ каждой отдельной функции.

Нет учёта взаимозависимостей.

Слабая ориентация на исследование основных преимуществ программы.

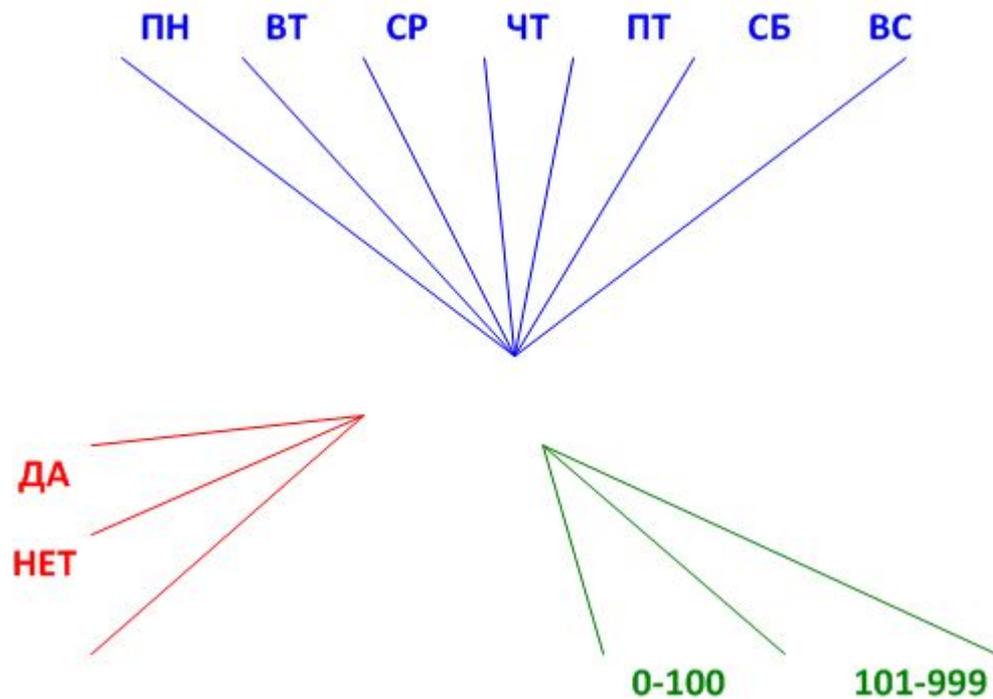
# Задачи функционального тестирования

- Определить **набор функций**.
- Определить **переменные и граничные условия**.
- Определить **переменные окружения**, которые могут повлиять на выполнение функции.
- Проверить **каждую функцию** «обычных условиях» (положительное тестирование) и «нестандартных ситуациях» (негативное тестирование).



А вот и  
доменное  
тестирование!

# Доменное тестирование

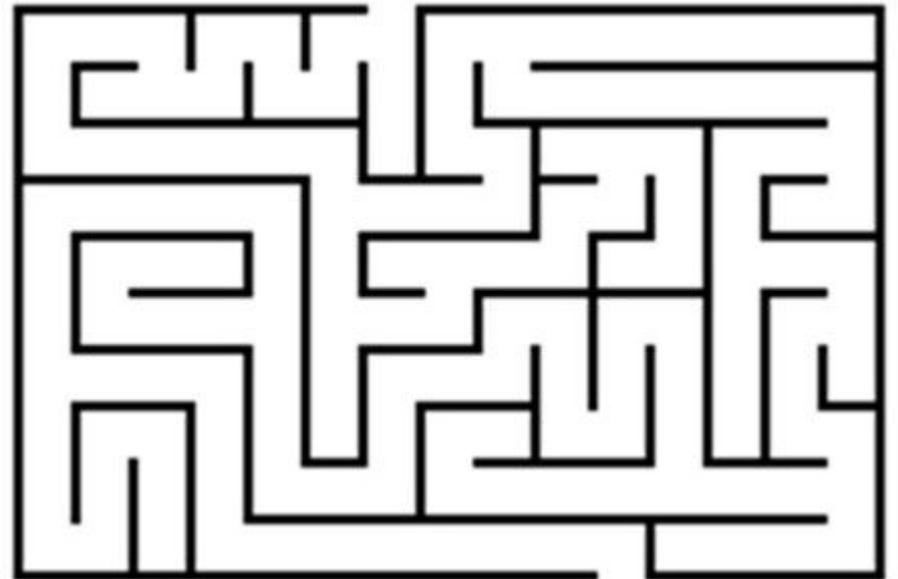


**Доменное тестирование** (domain testing) – вид тестирования, направленный на анализ **показательных значений и взаимосвязи элементов.**

... также известно как:

- «тестирование разделением» (partitioning testing);
- «анализ эквивалентности» (equivalence analysis);
- «анализ граничных условий» (boundary analysis).

В доменном тестировании **входные данные считаются эквивалентными**, если программа проходит один и тот же путь выполнения для их обработки.



**Основная цель доменного тестирования:** предоставить стратегию по выбору **минимального набора показательных тестов.**

Должен остаться  
только один...



## Классический подход к достижению цели:

- Разделить пространство значений на группы.
- Выбрать значения, представляющие каждую группу.
- Особое внимание обратить на граничные значения групп.
- Сформировать конечный набор «наиболее показательных» значений и провести тесты с их использованием.

## Пример!

*«В конце рабочего дня клерк заносит в БД информацию о количестве обработанных заказов. Если количество превышает 100, система запрашивает подтверждение «Вы уверены?»»*

# Решение примера

1. Разделить пространство значений на группы.

Количество заказов:

- MinInt - -1
- 0 - 100
- 101 - MaxInt
- Нецелые числа и не числа

*Из условия задачи автоматически следует, что данное поле принимает только целые числа (при условии, что фиксируется количество полностью выполненных заказов).*

## Решение примера

2. Выбрать значения, представляющие каждую группу.
3. Особое внимание обратить на граничные значения групп.

Выбранные значения:

- 0, 50, 100, 101, 500, MaxInt
- -1, -200, 1.98
- [empty], буквы, пробелы, спецсимволы

# Решение примера

4. Сформировать **конечный набор** «наиболее показательных» значений и провести тесты с их использованием.

Значения:

- 0, 100, 101, MaxInt
- -1, 1.7
- [empty], «абв», пробел(ы), !#@%

## Расширенный подход к достижению цели:

- Скомбинировать наиболее показательные тесты, полученные в результате «классического подхода».
- Уменьшить полученный набор, определяя наиболее показательные комбинации.

Применение этого подхода  
рассмотрим в контексте  
дальнейшего материала...

## «+» и «-» доменного тестирования

Обнаружение ошибок при минимальном количестве тестов.

Интуитивно понятный, универсальный подход.

Низкая вероятность обнаружения ошибок НЕ на граничных условиях.

Низкая вероятность обнаружения ошибок в сложных взаимодействиях.

Пространство значений часто бывает сложно формализовать.



## Ключевые действия при доменном тестировании:

- Определение классов эквивалентности.
- Определение наиболее показательных значений.
- Создание таблиц эквивалентности.
- Комбинирование нескольких тестов.
- Определение взаимозависимостей.

Частые возражения: «Мы тестируем на реальных пользовательских данных. Нам всё это не нужно!»



- Конкретные ограничения часто являются не более, чем предположениями.
- Баги живут годами.
- Пользовательские данные – лишь пример, но не полный перечень.
- Осознанный подход приносит больше пользы.
- Изучим сейчас – быстро применим, когда понадобится.

# Полезные навыки

- Определение **неоднозначности** (неполноты) в требованиях.
- Определение **диапазонов** значений.
- Определение **«многомерных» значений**, эквивалентных в одном контексте и не эквивалентных в другом.
- Определение **переменных** экране.



# Определение неоднозначности (неполноты) в требованиях

Игрок вводит целое число.

Компьютер отвечает:

- «Меньше», если введено число большее, чем загаданное.
- «Больше», если введено число меньшее, чем загаданное.
- «Угадал», если введено число, равное загаданному.

Что здесь «не так»?

# Определение диапазонов значений

Программа позволяет использовать любое допустимое целое число.

Как в таком случае  
узнать диапазон?

# Определение «многомерных» значений

«Измерение»	Классы эквивалентности	Примеры значений
Длина	min-max, внутри диапазона	min, max
	0, min-1	0, min-1
	max+1, бесконечность	max +1, очень большое число
Символы	алфавитно-цифровые	123abc
	спецсимволы	!@#\$%^&*()_+ =\\{}[]:~';'<>?.,/
	разные языки	ÀÇÈÌÑÒÙàçèìíñò
	разные кодировки (вкл. UTF)	学間気汽
Пробелы	нет пробелов	a
	пробелы в начале, в конце	_a, _a_, a_
	пробелы в середине	a_a
Уникальность, Регистр	разные значения	aaa / bbb
	одинаковые значения	aaa / aaa
	различия в регистре	aaa / aAa
Способ заполнения	набор с клавиатуры	набрать текст
	вставка	вставить текст

# Определение переменных на экране

Проблемы:

- Некоторые переменные выглядят как простые надписи.
- Некоторые поля не заполнены и/или не видны.
- Некоторые переменные связаны друг с другом и должны быть протестированы совместно.



# Где здесь переменные?

EPAM PROJECT MANAGEMENT CENTER Logged in as Svyatoslav Kulikov (Testing Team Lead) | [Logout](#) | [Contact Help Desk](#)

[Main](#) | [Communication & Support](#) | [Summary](#) | [Reporting](#) | [Project Info](#) | [Settings](#) | [Help](#) Project: **EPM-FARM**

[Time Journal](#) | [Time Report](#) | [Locked Time Periods](#)

### Time Journal for Svyatoslav Kulikov

[Data Export](#) | [Help](#)

[Save Changes](#) | [Reset Changes](#)  [Search >>](#)

Filter: All   Period: 7/11/11 - 7/17/11

[< Previous Week](#) | [Current Week](#) | [Next Week >](#)

	Mo	Tu	We	Th	Fr	Sa	Su
<b>Total 0 (of 20)</b>							
<b>Non-Project Activities</b>							
<input type="checkbox"/> EDU - Education & Trainings							
<input type="checkbox"/> ENV - Environment							
<input type="checkbox"/> MAN - Management							
<input type="checkbox"/> NBA - Bench Time							
<input type="checkbox"/> ORG - Org and Resource Administration							
<input type="checkbox"/> OTH - Other							
<input type="checkbox"/> QMS - Quality Management System							
<b>Vacation Info</b>							
EXV - Exceptional Vacation							
ILL - Illness							
OVT - Overtime Vacation							
POV - Leave Without Pay							
VAC - Regular Vacation							
<b>Total 0 (of 20)</b>							
<a href="#">&lt; Previous Week</a>   <a href="#">Current Week</a>   <a href="#">Next Week &gt;</a>							

# А вот они! И это ещё не все...

EPAM PROJECT MANAGEMENT CENTER

Logged in as **Svyatoslav Kulikov** (resting Team Lead) | Logout | Contact Help Desk

Main | Communication & Support | Summary | Reporting | Project Info | Settings | Help

Time Journal | Time Report | Locked Time Periods

Project: **EPM-FARM**

### Time Journal for **Svyatoslav Kulikov**

Save Changes | Reset Changes

Filter: All | Period: 7/11/11 - 7/17/11

◀ Previous Week | Current Week | Next Week ▶

	Mo	Tu	We	Th	Fr	Sa	Su
<b>Total 0 (of 20)</b>							
<b>Non-Project Activities</b>							
EDU - Education & Trainings							
ENV - Environment							
MAN - Management							
NBA - Bench Time							
ORG - Org and Resource Administration							
OTH - Other							
QMS - Quality Management System							
<b>Vacation Info</b>							
EXV - Exceptional Vacation							
ILL - Illness							
OVT - Overtime Vacation							
POV - Leave Without Pay							
VAC - Regular Vacation							
<b>Total 0 (of 20)</b>							

◀ Previous Week | Current Week | Next Week ▶

# Здесь – списки!

- Project Team
- Aleh Drazdov
  - Alexsei Mordas
  - Aleksey Margolin
  - Alena Hrashchanka
  - Alesya Bondarenko
  - Alexandra Samoylenko
  - Aliaksandr Antaniuk
  - Aliaksandr Baranouski
  - Aliaksandr Chaiko
  - Aliaksandr Deshkevich
  - Aliaksandr Ihnatsiuk
  - Aliaksandr Kazelka
  - Aliaksandr Khvesiuk
  - Aliaksandr Pliust
  - Aliaksandr Senko
  - Aliaksandr Sheliutsin
  - Aliaksei Biaskosty
  - Aliaksei Haurylau
  - Aliaksei Kandratovich
  - Aliaksei Kuliashou
  - Aliaksei Neuski
  - Aliaksei Piatrovich
  - Aliaksei Tatarovichyk
  - Anastasia Okhrimenko
  - Anatol Bludau
  - Andrei Bushyk
  - Andrei Halalobau
  - Andrei Harbunou
  - Andrei Karzhou
  - Andrei Leonau
  - Andrei Man

## Time Journal for Svyatoslav Kulikov

[Data Export](#) | [Help](#)

[Save Changes](#) | [Reset Changes](#)

Filter: All | Period: 7/11/11 - 7/17/11

[Previous Week](#) | [Current Week](#) | [Next Week](#)

Total 0 (of 20)

	Mo	Tu	We	Th	Fr	Sa	Su
<b>Non-Project Activities</b>							
EDU - Education & Trainings							
ENV - Environment							
MAN - Management							
NBA - Bench Time							
ORG - Org and Resource Administration							
OTH - Other							
QMS - Quality Management System							
<b>Vacation Info</b>							
EXV - Exceptional Vacation							
ILL - Illness							
OVT - Overtime Vacation							
POV - Leave Without Pay							
VAC - Regular Vacation							
Total 0 (of 20)							

[Previous Week](#) | [Current Week](#) | [Next Week](#)

# Здесь – связанные переменные!

EPAM PROJECT MANAGEMENT CENTER

Logged in as Svyatoslav Kulikov (Testing Team Lead) | Logout | Contact Help Desk

Main | Communication & Support | Summary | Reporting | Project Info | Settings | Help

Project: EPM-FARM

Time Journal | Time Report | Locked Time Periods

### Time Journal for Svyatoslav Kulikov

Save Changes | Reset Changes

Filter: All | Period: 7/11/11 - 7/17/11

Previous Week | Current Week | Next Week

	Mo	Tu	We	Th	Fr	Sa	Su
Total 0 (of 20)							
Non-Project Activities							
EDU - Education & Trainings							
ENV - Environment							
MAN - Management							
NBA - Bench Time							
ORG - Org and Resource Administration							
OTH - Other							
QMS - Quality Management System							
Vacation Info							
EXV - Exceptional Vacation							
ILL - Illness							
OVT - Overtime Vacation							
POV - Leave Without Pay							
Total 0 (of 20)							

Имя сотрудника, роли, вид календаря.

Поля даты.

Проект, список сотрудников, задачи проекта.

Суммы времени, поля времени.

# Простой вопрос:

- Не нашли или нашли мало ошибок.

**Плохое тестирование?**

**Хорошее качество?**

- А если это последний релиз-кандидат?
- А если это приемочный тест?

# Простой вопрос:

- Нашли много или очень много ошибок.

**Хорошее тестирование?**

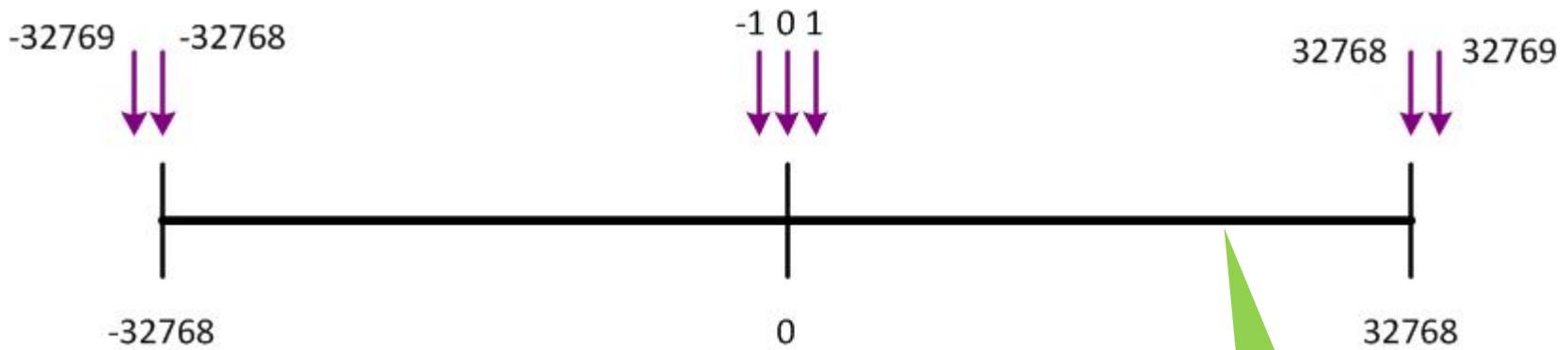
**Плохое качество?**

- Получается, что разработка – плохая?
- А если раз за разом?
- А если добавлено много нового функционала?

Потренируемся в работе с комбинациями!

$$A * B = C$$

Итак, у нас есть три переменные...



[7 значений для A] \*  
[7 значений для B] =  
**49 тестов**

Сколько тестов нужно  
провести?

Почему здесь именно  
такие числа? И всё ли с  
этими числами в  
порядке?

<b>Тип данных</b>	<b>Байты</b>	<b>Биты</b>	<b>Min</b>	<b>Max</b>
signed char	1	8	- 128	127
unsigned char	1	8	0	255
signed short	2	16	-32768	32767
enum	2	16	-32768	32767
unsigned short	2	16	0	65535
signed int	2	16	-32768	32767
unsigned int	2	16	0	65535
signed long	4	32	-2147483648	2147483647
unsigned long	4	32	0	4294967295

Здесь также многое зависит от реализации. По крайней мере, для Borland C++ 4.5, основные характеристики целочисленных типов выглядят следующим образом

[http://citforum.ru/programming/cpp\\_march/cpp\\_017.shtml](http://citforum.ru/programming/cpp_march/cpp_017.shtml) <http://tinyurl.com/7utf88x>

# Идеальный программист

1. Изучает проблемы (найденные самостоятельно или их обнаружили тестировщики)
2. Обобщает проблемы (подброшенные идеи)
3. Исследует ПО на выявление и исправление таких же или подобных ошибок



# Итак, тестов всегда много, а времени всегда недостаточно!

Что же делать?! Что нас может спасти???!?

Эвристические стратегии:

- Тестирование экстремальных точек (граничных значений / особых значений)
- Тестирование комбинаций (типична, популярна, плохо управляема)
- Число тестов растет экспоненциально с увеличением количества входных переменных



# Эвристическая стратегия

Предельные значения = 26 тестов

		A						
		0	1	-1	MIN	MIN-1	MAX	MAX+1
B	0							
	1							
	-1							
	MIN							
	MIN-1							
	MAX							
	MAX+1							

# «1x1, слабая стратегия» (Бейзер)

1 тест на границе, 1 вне = 16 тестов

		A						
		0	1	-1	MIN	MIN-1	MAX	MAX+1
B	0							
	1							
	-1							
	MIN							
	MIN-1							
	MAX							
	MAX+1							

# «1x1xN, сильная стратегия» (Бейзер)

1 тест на границе, 1 вне, N между ними  
= 24 теста

		A							X
		0	1	-1	MIN	MIN-1	MAX	MAX+1	
B	0								
	1								
	-1								
	MIN								
	MIN-1								
	MAX								
	MAX+1								
	Y								

# Выводы:

- Всегда используйте **графическое представление**.
- Помните о **значениях**:
  - **входных**;
  - **выходных**;
  - **промежуточных**.
- **Лишние тесты следует убрать**.



# Проблемы с переменными



Увы, часто в требованиях не указаны длины, допустимые символы, форматы и т.п.

## Проблема:

- Разработчики принимают свои решения и могут отклонять отчёты о существующих ошибках.
- Приложение может работать некорректно.
- Сбои при нетипичном поведении пользователя могут остаться незамеченными.

# Решение проблемы:

- Создать **список переменных** (не полей!)
- Сделать **разумные допущения** о
  - допустимых значениях;
  - недопустимых значениях и реакции приложения;
- Согласовать полученное **с разработчиками**.
- Согласовать полученное **с заказчиком**.



# Решение проблемы, пример:

<b>Пере- менная</b>	<b>Тип</b>	<b>Обяз</b>	<b>Уник</b>	<b>Мин</b>	<b>Макс</b>	<b>Дополнительно</b>
Логин	Строка	Да	Да	1	100	Любые символы UTF-32, НЕчувствителен к регистру
Пароль (ввод)	Строка	Да	Нет	6	50	Любые символы UTF-32, чувствителен к регистру
Пароль (хранение)	Строка	н/а	Нет	40	40	40

# Затем создаём чек-лист:

Экран	Переменная	Идеи тестов	Да/Нет
Создание пользователя	Логин	Корректный несуществующий логин	Да
		Оставить пустым	Нет
		Минимальная длина -1	Нет
		Максимальная длина +1	Нет
		Спецсимволы - !@#\$%^&*()_+ -=\{}[]:~<>,./?	Да
		Существующий логин	Нет
		Существующий логин в другом регистре	Нет

# Полезные трюки



# Делим или умножаем на два

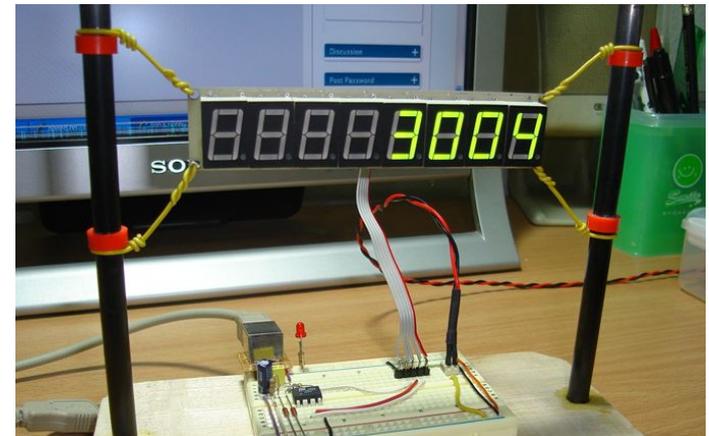
- Вводим в поле, не принимающее спецсимволы, **полный набор** таковых. Если что-то пошло не так, вводим **половину**, потом **половину половины** и т.д.
- Если надо выяснить максимальную длину принимаемого текста, каждый раз **удваиваем** её. Когда нашли проблему, **добавляем по  $\frac{1}{4}$**  от предыдущего успешного результата.



# Считаем слова/символы автоматически

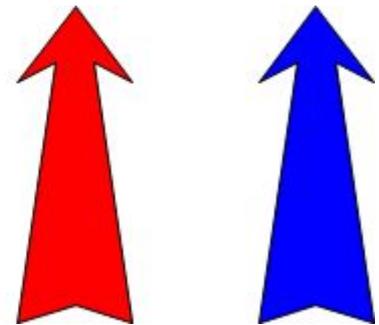
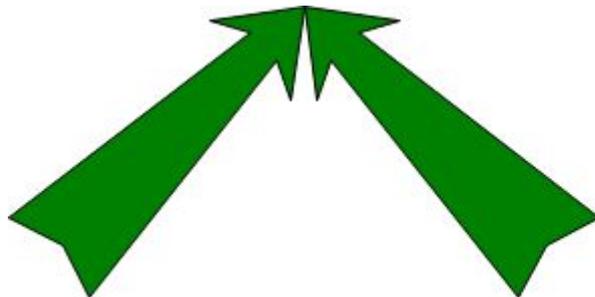
- Если у вас под рукой нет удобного средства анализа данных, можно воспользоваться word'ом для подсчёта длин.
- Если хочется «всего и сразу», можно написать программу, которая будет выдавать вам полную информацию об анализируемом тексте.

**Какую информацию о  
тексте имеет смысл  
собрать?  
Ваши идеи!**



# Позитивное вместе, негативное отдельно

- Несколько **ПОЗИТИВНЫХ ТЕСТОВ** (например, ввод данных в несколько полей) **МОЖНО** провести за раз.
- При этом **негативные тесты так комбинировать нельзя**, т.к. мы должны убедиться, что программа корректно отслеживает проблемы с каждым из полей.



# Используем готовые чек-листы

- В большинстве случаев **кто-то уже сталкивался** с тем, с чем сейчас столкнулись мы.
- Спрашиваем коллег, ищем в **корпоративной библиотеке**.



<epam>

«EPAM IS NOT  
A RELIGION»

EPAM POLICY 34:18

TIME FOR EVIL



# Есть вопросы? Давайте обсудим!



*При подготовке презентации  
использованы материалы тренингов  
Eduard Shymkus (EPAM)*