Стандартные проверки на поля экранных форм:

1. Текстовое поле (все)
2. Числовое поле (кроме пробелов и спецсимволов)
3. Выпадающий список (без рукописного ввода)
4. Дата (ДД.ММ.ГГГГ)



Стандартные проверки GUI:

1. Состав элементов управления
2. Взаимное расположение элементов управления
3. Условное отображение элементов управления
4. Порядка обхода элементов управления по Таб
5. Значения по умолчанию в полях ввода
6. Состав значений полей списочных типов
7. Доступность полей для редактирования
8. Условная доступность полей для редактирования
9. Обязательность заполнения полей
10. Условная обязательность заполнения полей
11. Ограничения на форматы значения полей
12. Ограничения на диапазоны значений полей (граничные значения)
13. Сложные ограничения на значения полей (вычисляемые и условные ограничения)
14. Расчет вычисляемых значений полей на типичных данных
15. Сохранение значений полей в системе
16. Отображение значений полей из системы
17. Функциональности кнопок и других активных элементов управления
18. Нестандартные элементы управления и сложная презентационная логика (диаграммы, мультимедиа и т.д.)

Техники, базирующиеся на спецификации:



Множество входных данных программного продукта как правило практически бесконечно. Выбор конечного подмножества, достаточного для проведения исчерпывающего тестирования, является ключевой задачей тестировщика.

Классы эквивалентности

Два теста считаются эквивалентными, если:

* Они тестируют одну и ту же вещь
* Если один из тестов ловит ошибку, то другой скорее всего тоже ее поймает
* Если один из тестов не ловит ошибку, то другой скорее всего тоже не поймает

Класс эквивалентности – это набор значений переменной, который считается эквивалентным.

Эквивалентное разбиение – это разработка тестов методом черного ящика в которой тестовые сценарии создаются для проверки элементов эквивалентной области. Как правило, тестовые сценарии разрабатываются для покрытия каждой области как минимум один раз.

Алгоритм использования классов эквивалентности

* Определить классы эквивалентности
* Выбрать одного представителя от каждого класса
* Выполнить тесты

Советы при определении классов эквивалентности

* Не забывайте о классах, охватывающих заведомо неверные или недопустимые входные данные
* Организуйте формируемый перечень классов в виде таблицы или плана
* Определите диапазоны числовых значений
* Для полей или параметров, принимающих фиксированные перечни значений, выясните, какие из значений входят в перечень
* Проанализируйте возможные результаты выбора из списков и меню
* Поищите переменные, значения которых должны быть равными
* Поищите классы значений, зависящих от времени
* Выявите группы переменных, совместно участвующих в определенных вычислениях, результат которых ограничивается конкретным набором или диапазоном значений
* Посмотрите, на какие действия программа отвечает эквивалентными событиями
* Продумайте варианты операционного окружения

Анализ граничных значений

Тесты строятся с ориентацией на использование тех величин, которые определяют предельные характеристики тестируемое системы.

1. Выделить классы эквивалентности
2. Определить граничные значения этих классов
3. Понять, к какому классу будет относиться каждая граница
4. Для каждой границы нам нужно провести тесты по проверке значения до границы, на границе, и сразу после границы.

Кол-во тестов = кол-во границ \* 3

Предположение ошибок

* Составить список, который перечисляет возможные ошибки и ситуации, в которых эти ошибки могли проявиться.
* Составить тесты на основе списка

План:

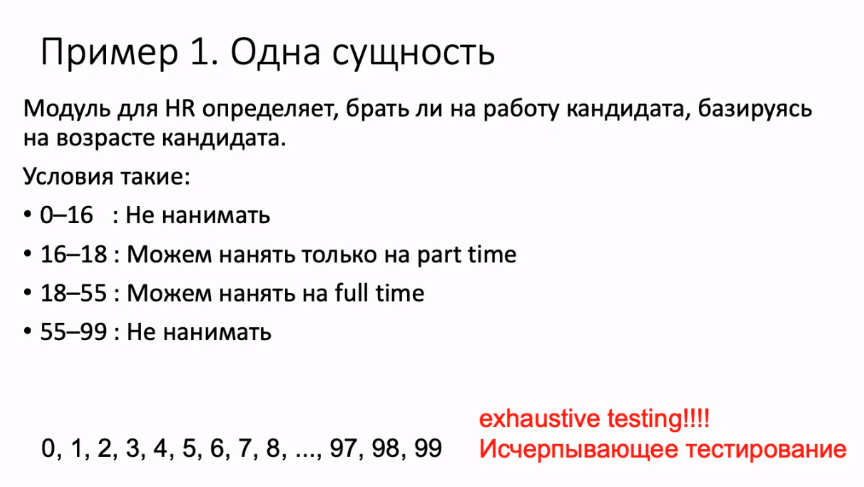
Виды полей

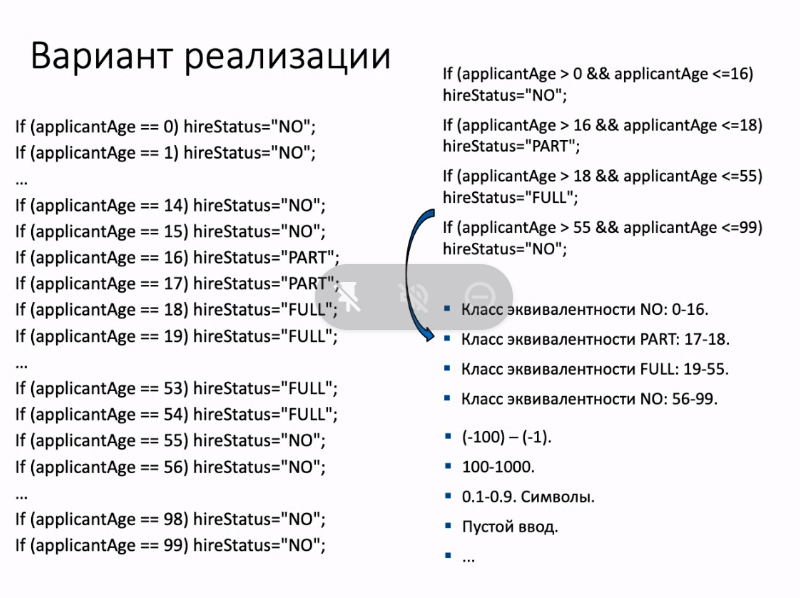
Проверки ГУИ

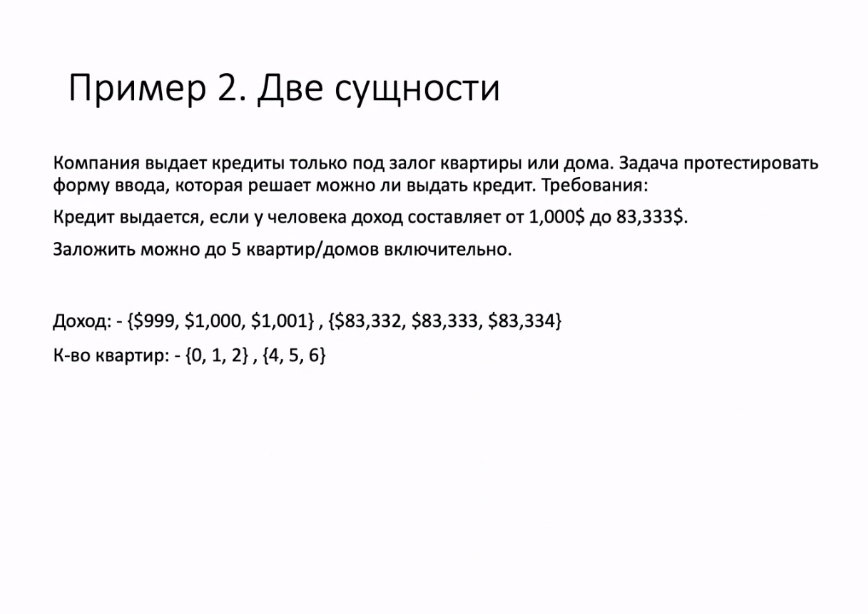
Классы эквивалентности

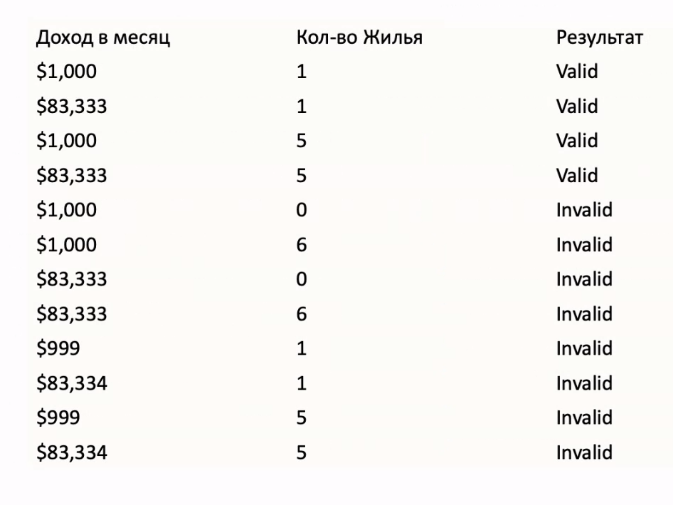
Анализ граничных значений

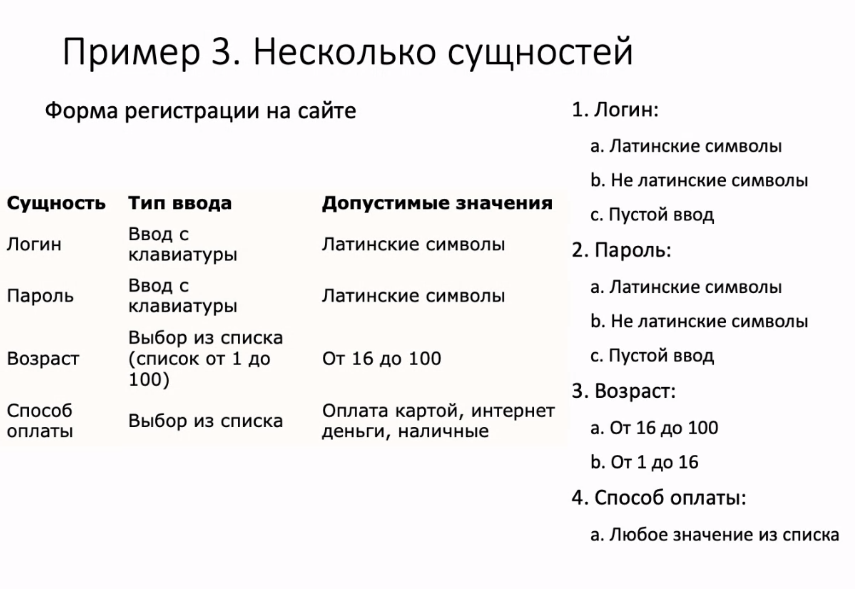
Предположение ошибок

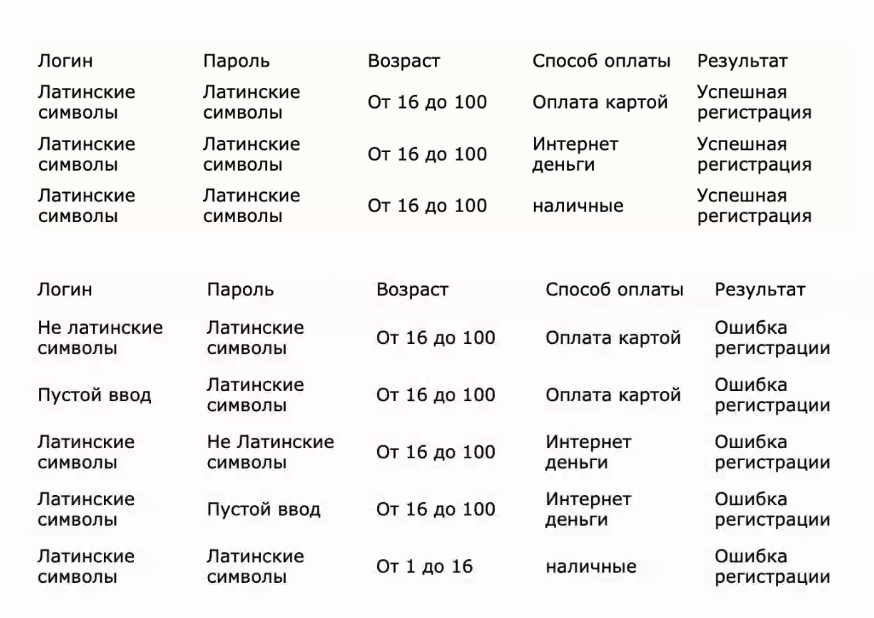












Матрица принятия решений

Описывает логику приложения основываясь на сущностях состояния системы. Каждая decision table должна описывать 1 состояние системы.

