

Практическая работа №7,8

Интеграционное тестирование

Цель работы: изучить Интеграционное тестирование

Многие современные ИТ-системы взаимодействуют с другими системами и модулями, поэтому крайне

важно иметь представление об их взаимосвязи и проверять их работоспособность. Интеграционное тестирование позволяет обнаружить дефекты, возникающие при взаимодействии систем. Для этого могут быть использованы как ручные, так и автоматизированные тесты.

Интеграционное тестирование рекомендуется проводить перед началом системного тестирования. Данный вид тестирования следует проводить как можно раньше, поскольку дефекты интеграции, как правило, имеют архитектурный характер, их исправление на поздних стадиях разработки является рискованным и может обойтись значительно дороже. Для ускорения начала процесса тестирования рекомендуется воспользоваться разработкой эмуляторов внешних систем.

В рамках интеграционного тестирования также может проводиться регрессионное тестирование с целью проверки сделанных в приложении или окружающей среде изменений и работоспособности унаследованной функциональности.

Интеграционное тестирование позволяет имитировать действия пользователей и быстро получать подтверждение, что программный продукт успешно взаимодействует с другими системами. Такой подход гарантирует сразу несколько преимуществ:

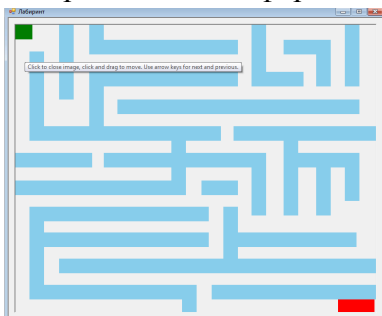
- Предотвращение появления критичных ошибок в опытно-промышленной эксплуатации;
- Снижение влияния человеческого фактора;
- Экономия затрат на исправление дефектов.

Главной задачей интеграционного тестирования является поиск ошибок, связанных с взаимодействием модулей системы или нескольких систем. В результате все смежные системы и модули одной системы должны работать согласованно.

Способы проведения интеграционного тестирования подбираются в зависимости от интеграционных решений.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ И ФОРМА ОТЧЕТНОСТИ:

1. Создаем проект Windows Form.
2. Задаем размер формы (любой, но чем больше, тем можно сложнее и интереснее создать лабиринт).
3. Прописываем название формы.
4. Перемещаем на форму Panel, растягиваем по краям так, чтобы был небольшой зазор.
5. В свойствах FormBorderStyle выставляем значение Fixed3D (чтобы пользователи не могли растягивать форму).
6. Отключаем кнопку развернуть в заголовке окна. MaximizeBox= false.
7. Для Panel также выставляем значение Fixed3D.
8. Теперь создаем на форме лабиринт из элементов Label.



Выделяем Label:

Задаем свойству AutoSize значение false.

Выберите какой-нибудь цвет.

Удалите значение Text.

9. Добавляем Label Finish, в свойствах присваиваем свойству Name – имя, Text- Финиш, цвет на ваш выбор. Так же поступаем с Label Start.
10. При старте программы, курсор должен находиться на Label Start, если курсор касается стенки лабиринта, то его перекидывает в начало.
11. Для этого выделяем все стенки(Label) и создаем обработчик для события MouseEnter.
12. Так же для Panel проставляем свойство MouseLeave указываем MouseEnter.
13. Создаем обработчик для формы, нужно чтобы в момент загрузки формы выставить позицию курсора. Свойство Load событие Form1_Load
14. Сам код:

```
3 using System.Windows.Forms;
4
5 namespace Labyrinth
6 {
7     public partial class Form1 : Form
8     {
9         public Form1()
10         {
11             InitializeComponent();
12         }
13         //При касании любой стены лабиринта, позиция будет обновляться до стартового.
14         private void label71_MouseEnter(object sender, EventArgs e)
15         {
16             Cursor.Position = new Point(this.Location.X + 55, this.Location.Y + 60);
17         }
18         //Выставляем стартовую позицию курсора.
19         // +55 и +60 это позиция labelStart относительно позиции формы на рабочем столе.
20         private void Form1_Load(object sender, EventArgs e)
21         {
22             Cursor.Position = new Point(this.Location.X + 55, this.Location.Y + 60);
23         }
24     }
25 }
```

Задание 1. Когда курсор достигнет финиша, должно выводиться сообщение о Победе.

Задание 2. Установить препятствия, и когда на них наводится курсор, всплывает окошко с каким-нибудь сообщением (например Error)

Задание 3. Вместо курсора использовать клавиатуру и какой либо объект (например картинку Расман).

Задание 4. Придумать свой вариант лабиринта с какой-либо особенностью.