

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Физические методы контроля»

# ГРАФИЧЕСКАЯ СРЕДА И СРЕДСТВА ПРОГРАММИРОВАНИЯ ДЛЯ НЕРАЗРУШАЮЩЕГО КОНТРОЛЯ

*Методические рекомендации к лабораторным работам  
для студентов направления подготовки  
12.03.01 «Приборостроение» очной формы обучения*



Могилев 2023

УДК 620.179:004.4  
ББК 34.9:32.973-018  
Г35

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Физические методы контроля» «01» сентября 2023 г.,  
протокол № 1

Составитель канд. техн. наук, доц. А. В. Кушнер

Рецензент канд. техн. наук, доц. В. В. Кутузов

В методических рекомендациях кратко изложены теоретические сведения, необходимые для выполнения лабораторных работ, и требования к оформлению. Методические рекомендации составлены в соответствии с рабочей программой по дисциплине «Графическая среда и средства программирования для неразрушающего контроля».

Учебное издание

## ГРАФИЧЕСКАЯ СРЕДА И СРЕДСТВА ПРОГРАММИРОВАНИЯ ДЛЯ НЕРАЗРУШАЮЩЕГО КОНТРОЛЯ

Ответственный за выпуск	А. В. Хомченко
Корректор	А. Т. Червинская
Компьютерная верстка	Е. В. Ковалевская

Подписано в печать . Формат 60x84/16. Бумага офсетная. Гарнитура Таймс.  
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 26 экз. Заказ №

Издатель и полиграфическое исполнение:  
Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет».  
Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий  
№ 1/156 от 07.03.2019.  
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский  
университет, 2023

## Содержание

1 Лабораторная работа № 1. Определение требований проектированию пользовательских интерфейсов.....	4
2 Лабораторная работа № 2. Прототипирование пользовательских интерфейсов .....	6
3 Лабораторная работа № 3. Моделирование вариантов использования, пользовательских историй.....	9
4 Лабораторная работа № 4. Исследование сред взаимодействия.....	12
5 Лабораторная работа № 5. Анализ задач и создание модели среды – отображение структур, процессов, объектов .....	15
6 Лабораторная работа № 6. Разработка модели пользователя.....	16
7 Лабораторная работа № 7. Проектирование графа диалога с использованием сетей Петри.....	20
8 Лабораторная работа № 8. Проектирование сценария текстового диалога.....	22
9 Лабораторная работа № 9. Разработки модели оценки интерфейса на этапе проектирования, используя формальные методы анализа диалога на тупики .....	23
10 Лабораторная работа № 10. Разработка многооконных интерфейсов информационной системы в инструментальной среде разработки для использования в системах неразрушающего контроля.....	25
11 Лабораторная работа № 11. Разработка пользовательского интерфейса прямого манипулирования для использования в системах неразрушающего контроля.....	26
12 Лабораторная работа № 12. Применение технологии Drag and Drop при создании интерфейса для использования в системах неразрушающего контроля .....	29
Список литературы .....	30

# 1 Лабораторная работа № 1. Определение требований проектированию пользовательских интерфейсов

**Цель работы:** научиться определять требования для проектирования пользовательских интерфейсов.

## *1.1 Теоретические сведения*

Основным источником требований к информационной системе, безусловно, являются соображения, высказанные представителями Заказчика.

В соответствии с иерархической моделью требований данная информация структурируется как минимум на два уровня: бизнес-требования и требования пользователей.

Проблема состоит в том, что требования формулируются к создаваемой, еще не существующей системе, т. е. по сути решается начальная подзадача задачи проектирования АИС, а представители Заказчика далеко не всегда бывают компетентны в данном вопросе. Поэтому, наряду с требованиями, высказанными Заказчиком, целесообразно собирать и требования от других совладельцев системы: сотрудников аналитической группы исполнителя, внешних экспертов и т. д.

Результирующий, часто достаточно сырой материал рассматривается, как документ «Требования совладельцев». На требования совладельцев обычно не накладывается никаких специальных ограничений.

Другим важным источником информации, помимо выявления требований, являются артефакты, описывающие предметную область. Это могут быть документы с описанием бизнес-процессов предприятия, выполненные консалтинговым агентством, либо просто документы (должностные инструкции, распоряжения, своды бизнес-правил), принятые на предприятии. Одной из немногих методологий, в которой специально выделяется рабочий поток делового моделирования, является Rational Unified Process.

Еще одна альтернатива, используемая при выявлении требований, – так называемые «лучшие практики», широко применяемые в настоящее время в бизнес-консалтинге и при внедрении корпоративных информационных систем.

Лучшие практики представляют собой описания моделей деятельности успешных компаний отрасли, используемые длительное время в сотнях и тысячах компаний по всему миру.

Таким образом, основными источниками, образующими «вход» процесса выявления требований, являются требования, высказанные совладельцами, как таковые или (и) артефакты, описывающие объект исследования. Однако это достаточно упрощенный взгляд: чтобы данные поступили «на вход», аналитики требований должны проделать немалую работу, связанную с подбором респондентов и информационных материалов, организацией интервью и т. д.

## ***1.2 Порядок выполнения работы***

1 Составить документ «Профиль клиента», в котором кратко описать информацию о компании (1–2 страницы).

2 Разработать видение ИС (видение выполнения проекта и границы проекта – документ, который кратко описывает, в каких подразделениях и в какой функциональности будет внедряться ИС), в том числе:

– проанализировать проблемную ситуацию, определить позицию разрабатываемой АИС;

– сформулировать краткое описание АИС, определить его возможности;

– выработать и описать прочие требования к АИС.

Шаблон документа «Видение» содержит следующие основные разделы.

1 Введение.

2 Позиционирование.

3 Описания совладельцев и пользователей.

4 Краткий обзор изделия.

5 Возможности продукта.

6 Ограничения.

7 Показатели качества.

8 Старшинство и приоритеты.

9 Другие требования к изделию.

10 Требования к документации.

11 Приложение.

Во введении описываются цель документа, его контекст (связь и взаимовлияние с различными проектами), определения, сокращения, ссылки на другие документы, краткое содержание.

В разделе «Позиционирование» помещается определение решаемой проблемы (проблем), указывается целевой заказчик и исследуются деловые преимущества изделия перед аналогичными на рынке.

В описании совладельцев и пользователей, помимо собственно описания этих двух групп, исследуется демография рынка: целевые рыночные сегменты, размер и темпы роста рынка, существующие конкурентные предложения на рынке, репутация Разработчика на рынке.

Краткий обзор изделий содержит резюме изделия, описание его перспектив и ключевых возможностей, предположения и зависимости, указывается стоимость и ее калькуляция, рассматриваются вопросы лицензирования и инсталляции.

В разделе, посвященном возможностям продукта, они описываются более подробно, каждая – в отдельном параграфе.

В раздел «Ограничения» следует выносить существующие технические, технологические и другие обстоятельства, которые необходимо учитывать на данной стадии.

Раздел «Показатели качества» содержит описание наиболее существенных нефункциональных требований к системе (эффективности, надежности, отказоустойчивости и др.).

Раздел «Старшинство и приоритеты» ранжирует сформулированные ранее требования и возможности системы по степени важности, очередности реализации и т. п.

Раздел «Другие требования к изделию» описывает применяемые стандарты, системные требования, эксплуатационные требования, требования к окружающей среде.

В требованиях к документации приводятся ключевые характеристики руководства пользователя, интерактивной справки, руководства по установке и конфигурированию, файла Read Me.

В приложение выносятся атрибуты возможностей. RUP рекомендует следующий набор атрибутов: статус, выгода, объем работ, риск, стабильность, целевой выпуск, назначение, причина.

3 Оформить работу.

4 Осуществить защиту работы.

### ***Контрольные вопросы***

1 Что такое интерфейс?

2 Какие типы пользовательских интерфейсов существуют?

3 Перечислите этапы разработки пользовательских интерфейсов.

4 К какому типу интерфейсов будет относиться интерфейс, разработанный в данной лабораторной работе?

5 Какие модели интерфейсов существуют?

6 Какая модель интерфейса будет использована в данной работе?

7 Что такое диалог?

8 Какие типы диалогов существуют?

9 Какие формы диалога вы знаете?

10 Какой тип диалога и какая форма диалога будут использованы в данной работе?

## **2 Лабораторная работа № 2. Прототипирование пользовательских интерфейсов**

**Цель работы:** научиться прототипировать пользовательские интерфейсы.

### ***2.1 Теоретические сведения***

При построении интерфейса важно придерживаться определенных принципов, способствующих обеспечению его унификации и дружелюбности:

– контроль пользователем интерфейса. У пользователя создается субъективное ощущение управления системой, делающей его взаимодействие с системой более комфортным;

– уменьшение загрузки памяти пользователя. Элементы, редко или вовсе не используемые пользователем, должны располагаться «на заднем плане» или быть скрыты от пользователя, уступая место наиболее актуальным, часто используемым элементам управления;

– последовательность пользовательского интерфейса. Все элементы управления в своей совокупности должны «читаться»: у пользователя не должно возникать сомнения, в каком порядке ему следует использовать те или иные элементы.

Поэтому важнейшим этапом разработки пользовательского интерфейса является проектирование его прототипа. Решение этой задачи не всегда однозначно – обилие доступных элементов управления затрудняет выбор каждого конкретного интерфейсного элемента, предназначенного для решения той или иной задачи.

Еще одна особенность, которая должна быть учтена при проектировании интерфейса, состоит в обеспечении его объектности, заключающейся в возможности сопоставления объектов и субъектов предметной области отдельным элементам управления.

Решение этих задач носит вполне формализованный характер и базируется на определении объектов и субъектов разрабатываемой системы в контексте доступных элементов управления.

На начальном этапе разработки прототипа интерфейса должен быть составлен перечень объектов и данных, которые должны быть задействованы в данном проекте. Для каждого объекта указывается его обобщенный тип (например, данные или устройство), позволяющий заранее выделить необходимые классы объектов. При этом каждый объект разрабатываемого прототипа будет представлять собой экземпляр одного из выделенных классов.

Следующий этап разработки прототипа связан с определением взаимосвязей между описанными ранее объектами. Эти взаимосвязи носят характер передачи потоков информации от одного объекта к другому или передачи одного объекта другому в форме потока. Решение данной задачи позволит заранее спланировать воздействие одного объекта на другой и при их последующей практической реализации сформировать логику функционального описания этих объектов.

Последний этап создания проекта прототипа пользовательского интерфейса предполагает агрегирование данных, полученных на двух предшествующих этапах, в форме матрицы прямого манипулирования объектами. Эта матрица представляет собой двумерную таблицу, строки и столбцы которой описывают исходные и конечные объекты взаимодействия экземпляров классов, выявленных на предыдущих этапах. Для того чтобы описать все возможные варианты взаимодействия, необходимо и в столбцах, и в строках таблицы указать все выявленные на первом этапе проектирования объекты и данные. На пересечении строк и столбцов таблицы должны быть заданы связывающие их действия.

Полученная матрица манипулирования позволяет из множества доступных элементов управления выбрать те, что наиболее эффективно реализуют заданные объекты и их взаимодействие.

Как известно, проектирование интерфейса является итерационным процессом. Негативная оценка проекта пользователями обуславливает необходимость его корректировки, т. е. возвращение на предыдущий этап разработки. Поэтому еще на этапе проектирования должны быть предусмотрены альтернативные варианты интерфейсного представления всех заявленных объектов, которые могут быть использованы в случае корректировки проекта прототипа интерфейса.

Такие элементы должны быть подобны в плане их функциональной нагрузки, однако их внешнее представление может коренным образом отличаться.

## ***2.2 Порядок выполнения работы***

1 На основании системного проекта и графа диалога выделить основные объекты и действия, которые должны участвовать в процессе функционирования разрабатываемой системы.

2 Определить взаимосвязи между выделенными объектами.

3 Составить матрицу прямого манипулирования объектами.

4 На основе функциональной модели разработать визуальное представление разрабатываемой системы, определить основные компоненты интерфейса будущей системы.

5 Обосновать выбор используемых элементов управления и предусмотреть альтернативные варианты.

6 Организовать возможность получения информации в режиме диалога. Для этого следует использовать разработанные в лабораторной работе вопросы к пользователю.

7 Предусмотреть в проекте интерфейса возможность вывода результата пользователю на основе исходных данных, полученных от пользователя.

8 При проектировании предусмотреть согласование с созданной в лабораторной работе № 1 функциональной моделью. В случае необходимости, доработать свой проект в соответствии с моделью.

9 Оформить отчет. Отчет должен содержать:

- титульный лист, название и цель работы;
- вариант задания и постановку задачи;
- таблицу 1 объектов и действий;
- схему взаимодействия объектов;
- матрицу прямого манипулирования;
- перечисление элементов интерфейса и обоснование их выбора;
- подробное описание функциональной нагрузки каждого выбранного элемента интерфейса;
- выводы.

Таблица 1 – Объекты и действия

Объекты и данные	Тип объекта	Действия

### ***Контрольные вопросы***

1 Перечислите основные принципы проектирования пользовательского интерфейса. Какие типы объектов вы знаете?

2 Какие типы данных циркулируют между объектами? Как они выражаются?

3 Какие виды взаимосвязей объектов вы знаете?

4 Что представляет собой матрица прямого манипулирования?

## **3 Лабораторная работа № 3. Моделирование вариантов использования, пользовательских историй**

**Цель работы:** научиться моделировать различные варианты использования, пользовательских историй.

### ***3.1 Теоретические сведения***

#### **Сценарии использования.**

Сценарий – один из способов описания структуры задачи. Это повествовательный рассказ о совершаемых действиях, это история, эпизод, происходящий в данных временных рамках и в данном контексте. Различные формы сценариев широко применяются при разработке программного обеспечения.

Сценарии задач и взаимодействий обычно богаты характеристиками и обладают высокой реалистичностью.

Сценарии при разработке пользовательского интерфейса описывают взаимодействие между пользователем (или типом пользователей) и системой. Обыкновенные сценарии обладают некоторыми серьезными ограничениями при попытке использовать их для проектирования пользовательского интерфейса. В них делается упор на реалистичность и детали, при этом на серьезные проблемы и общую организацию обращается недостаточно внимания. Сценарии включают в себя правдоподобные описания комбинаций отдельных действий и задач, поэтому часто бывает тяжело выделить и понять основную суть взаимодействия.

#### **Модели use case.**

Элемент use case – это ситуация, вариант использования, т. е. некоторый случай применения системы. По сути, use case – это:

- обеспечение функциональности;
- сугубо внешняя точка зрения (принцип «черного ящика»);
- повествовательное описание;

- описание взаимодействия между пользователем (в какой-то роли) и системой;

- завершенное и понятное пользователю применение системы.

Каждый элемент use case описывает в повествовательной форме завершенное, хорошо определенное взаимодействие, имеющее ясную цель с точки зрения пользователя. При объектно-ориентированном подходе элементы use case могут описывать взаимодействие с другими системами и оборудованием, а не только с живыми пользователями. Тем не менее, когда целью является разработка пользовательского интерфейса, можно совершенно спокойно ограничиться рассмотрением только тех элементов use case, которые относятся к взаимоотношениям человека и системы.

### **Сущностные элементы use case.**

Сущностный элемент use case – это структурированное повествование, выраженное на языке данной прикладной области и пользователей системы и содержащее упрощенное, обобщенное, абстрактное, не зависящее от технологии и реализации описание одной завершенной, наполненной смыслом и хорошо определенной с точки зрения пользователей задачи или взаимодействия. Предполагается, что пользователь играет определенную роль по отношению к системе, а в описании воплощаются цели и замыслы лежащего в его основе взаимодействия.

Сущностные элементы use case строятся на основе целей и задач пользователя, а не на основе каких-то конкретных механизмов или этапов, ведущих к достижению этих целей. Некоторым кажется значимым включение целей пользователей в модели use case, но это не должно быть связано с упрощениями, присущими сущностному моделированию.

При применении подхода, ориентированного на удобство использования, в сущностных элементах use case, являющихся структурированным описанием, можно выделить три части: изложение общих устремлений пользователя, выраженное в элементе use case, плюс состоящее из двух частей описание, включающее в себя модель пользовательских устремлений и модель обязательств системы. Сущностные элементы use case именуются, причем при помощи этих имен стараются выразить пользовательские намерения в условиях данного варианта использования. В соответствии с соглашением, предложенным Якобсоном, элемент use case изображается в виде эллипса с именем элемента.

### **Описание вариантов использования.**

В контексте процесса управления требованиями варианты использования трактуются следующим образом (согласно Коберну):

- вариант использования фиксирует соглашение между участниками проекта относительно поведения системы;

- вариант использования описывает поведение системы при различных условиях, когда система отвечает на запрос одного из участников, называемого основным действующим лицом;

– основное действующее лицо инициирует взаимодействие с системой, чтобы добиться некоторой цели. Система отвечает, соблюдая интересы всех участников.

Варианты использования – это вид документации, применяемый, когда требуется сконцентрировать усилия на обсуждении принципиальных требований к разрабатываемой системе, а не на подробном их описании. Стиль их написания зависит от масштаба, количества участников и критичности проекта. В общем случае рекомендуется придерживаться следующих правил:

– названия вариантов использования должны быть деловыми (нетехническими) терминами, имеющими значение для заказчика;

– каждый вариант использования должен представлять собой завершённую транзакцию между пользователем и системой, представляющую для первого некоторую ценность;

– хорошо написанный вариант использования легко читается и состоит из предложений, написанных в единой грамматической форме. На обучение чтению варианта использования не должно уходить больше нескольких минут.

Формат описания варианта использования (по Коберну).

1 Имя – цель в виде краткой активной глагольной фразы.

2 Контекст использования – более длинное описание цели.

3 Область действия.

4 Уровень точности.

5 Основное действующее лицо.

6 Другие участники и их интересы.

7 Предусловие (определяет, выполнение какого условия гарантирует система перед тем, как разрешить запуск варианта использования).

8 Минимальные гарантии (наименьшие обещания системы участникам, в частности, когда цель основного действующего лица не может быть достигнута).

9 Гарантии успеха (или постусловие – *postcondition* – устанавливает, что интересы участников удовлетворяются при успешном завершении варианта использования в конце основного сценария).

10 Триггер (событие, которое запускает вариант использования).

11 Основной сценарий или поток (простой для понимания типичный сценарий, в котором достигается цель основного действующего лица и удовлетворяются интересы всех участников). Каждый шаг основного сценария описывает: взаимодействие двух действующих лиц («Клиент вводит адрес»); шаг подтверждения для защиты интереса участника («Система подтверждает PIN-код»); внутреннее изменение для удовлетворения интереса участника («Система выводит сумму из баланса»).

12 Расширения (запускаются при возникновении определенного условия, содержат последовательность шагов, описывающих, что происходит при этом условии, и заканчивается достижением цели или отказом от неё).

13 Список изменений в технологии и данных.

14 Вспомогательная информация.

### ***3.2 Порядок выполнения работы***

- 1 Провести анализ предметной области в соответствии с выбранным заданием.
- 2 Составить пять сценариев использования программного обеспечения пользователем согласно формату описания Коберна.
- 3 Оформить отчет.
- 4 Осуществить защиту работы.

### ***Контрольные вопросы***

- 1 Каким образом производится моделирование задач?
- 2 Что такое сценарий использования?
- 3 Что такое элемент use case?
- 4 Что такое сущностные элементы use case?
- 5 Чем отличаются сценарии использования от модели use case?
- 6 Каким образом можно описать варианты использования?
- 7 Приведите пример описания варианта использования по Коберну?

## **4 Лабораторная работа № 4. Исследование сред взаимодействия**

**Цель работы:** научиться и провести исследование сред взаимодействия.

### ***4.1 Теоретические сведения***

#### **Карты элементов use case.**

Элементы use case не существуют отдельно от внешнего мира. Полноценная программная система должна обеспечивать поддержку десятков, а то и сотен элементов use case, причем, внутри каким-то образом связанного с ней приложения должна существовать связь между этими элементами. Отображение взаимосвязи между приложениями дает возможность описать общую структуру задачи, решаемой приложением и его интерфейсом. Карта элементов use case для данной задачи разбивает все функциональные возможности системы на множество взаимосвязанных сущностных элементов use case. Выделив все различающиеся и важные взаимодействия и показав отношения между ними, можно создать упрощенную общую модель задач, решаемых системой, и возможностей, которые она обязана предоставить.

Полноценная модель use case представляет собой множество описаний, определяющих суть всех элементов use case, и карту этих элементов, показывающую отношения между ними. Между сущностными элементами use case могут существовать отношения разных типов, включая специализацию, расширение, композицию, а также сходство. Знание этих отношений позволяет аналитику или разработчику выделить общие элементы задачи и в итоге создать более простую модель задачи.

### **Специализация.**

Некоторые элементы use case могут являться специализированными версиями других элементов. Например, при разработке приложения «банкомат» элементы use case «получение Денег», «размещение Средств» и «запрос Состояния» являются субклассами, или специализированными вариантами абстрактного класса взаимодействий, который может быть назван «использование Банкомата». Что касается отношения между элементами «получение Денег» и «использование Банкомата», то его можно охарактеризовать как классификацию, или специализацию. Такой тип отношения означает, что один элемент use case «является» («ts-a») специализацией другого. В объектно-ориентированном анализе и проектировании такое отношение соответствует отношению класс/подкласс.

Специализация дает возможность упростить общую модель use case путем отделения общих или универсальных форм взаимодействия от специфических форм, адаптированных для более узкого применения. Таким образом, нет необходимости переписывать заново самые общие паттерны применения. Достаточно написать их один раз, а затем лишь «повторно использовать» («reuse»), ссылаясь на них. Для отображения отношения специализации используется двойная стрелка. Рядом со стрелкой можно встретить подпись «is-a» или «specialize» в зависимости от контекста.

### **Расширение.**

Одной из инноваций в объектно-ориентированной программной инженерии, навеянных идеями Якобсона, стало признание расширения одним из возможных отношений между элементами use case. Говорят, что один элемент «расширяет» другой, когда он содержит вставляемые или альтернативные паттерны взаимодействия, которые войдут в расширяемый элемент. Например, при отработке элемента use case для изменения внешнего вида какой-то части экрана пользователю необходимо заниматься поиском по всей системе файла, содержащего нужную картинку или значок. Нормальное, или ожидаемое, развитие событий (обеспечивается базисом, или базисным элементом use case) тем не менее вовсе не подразумевает поиск каких-то дополнительных графических файлов.

Расширение – это удачная концепция, позволяющая значительно упростить сущностные модели use case. На карте элементов use case отношение расширения изображается в виде пунктирной линии со стрелкой и имеет подпись extend. Если для расширения дается дополнительное описание, в него может быть включено примечание, показывающее, какие элементы use case расширяются.

### **Композиция.**

Элементы use case можно декомпозировать на составные части, или подэлементы, являющиеся подчиненными или включенными паттернами взаимодействия. Отношение композиции обозначается на карте элементов use case пунктирной стрелкой, указывающей на подэлемент use case и имеющей метку include.

Взаимодействие, описываемое суперэлементом use case, осуществляется при помощи взаимодействий, входящих в подэлемент или подэлементы. причем описание суперэлемента будет ссылаться на все используемые подэлементы. Например, элемент use case под названием «начало Протоколирования Задачи», созданный для программы, отслеживающей ход выполнения задач, может использовать элементы «авторизация Доступа» и «ввод Параметров Задачи». Такой способ моделирования взаимодействий позволяет разделить независимые и почти никак не связанные между собой подзадачи «авторизации Доступа» и «ввода Параметров Задачи».

### **Диаграммы деятельности.**

Для описания функциональных требований помимо диаграмм вариантов использования применяются диаграммы деятельности:

- для описания поведения, включающего большое количество параллельных процессов;
- для анализа варианта использования (описывают последовательность действий и их взаимосвязь);
- для анализа потоков работ (workflow) в различных вариантах использования.

Когда варианты использования взаимодействуют друг с другом, диаграммы деятельности являются средством представления и анализа их поведения.

## **4.2 Порядок выполнения работы**

- 1 Изучить основы построения use case-диаграмм и диаграмм деятельности.
- 2 Построить use case-диаграммы.
- 3 Построить диаграммы деятельности для каждого варианта использования.
- 4 Оформить отчет.
- 5 Осуществить защиту работы.

### **Контрольные вопросы**

- 1 Что такое карта элементов use case?
- 2 Что означает роль на use case-диаграмме?
- 3 В чем заключается суть отношения специализации? Приведите пример.
- 4 В чем заключается суть отношения расширения? Приведите пример.
- 5 В чем заключается суть отношения композиции? Приведите пример.
- 6 Чем отличается отношение специализации от расширения?
- 7 Что собой представляет диаграмма деятельности?
- 8 В чем заключаются отличия use case-диаграммы от диаграммы деятельности?

## 5 Лабораторная работа № 5. Анализ задач и создание модели среды – отображение структур, процессов, объектов

**Цель работы:** научиться анализировать задачи и создавать модели среды – отображение структур, процессов, объектов.

### 5.1 Теоретические сведения

Классы в UML изображаются на диаграммах классов, которые позволяют описать систему в статическом состоянии – определить типы объектов системы и различного рода статические связи между ними.

Классы отображают типы объектов системы.

Между классами возможны различные отношения:

- зависимости, которые описывают существующие между классами отношения использования;
- обобщения, связывающие обобщенные классы со специализированными;
- ассоциации, отражающие структурные отношения между объектами классов.

Зависимостью называется отношение использования, согласно которому изменение в спецификации одного элемента (например, класса «товар») может повлиять на использующий его элемент (класс «строка заказа»). Часто зависимости показывают, что один класс использует другой в качестве аргумента.

Обобщение – это отношение между общей сущностью (родителем – класс «клиент») и ее конкретным воплощением (потомком – классы «корпоративный клиент» или «частный клиент»). Объекты класса-потомка могут использоваться всюду, где встречаются объекты класса-родителя, но не наоборот. При этом он наследует свойства родителя (его атрибуты и операции). Операция потомка с той же сигнатурой, что и у родителя, замещает операцию родителя; это свойство называют полиморфизмом. Класс, у которого нет родителей, но есть потомки, называется корневым. Класс, у которого нет потомков – листовым.

Ассоциация – это отношение, показывающее, что объекты одного типа неким образом связаны с объектами другого типа («клиент» может сделать «заказ»). Если между двумя классами определена ассоциация, то можно перемещаться от объектов одного класса к объектам другого. При необходимости направление навигации может задаваться стрелкой. Допускается задание ассоциаций на одном классе. В этом случае оба конца ассоциации относятся к одному и тому же классу. Это означает, что с объектом некоторого класса можно связать другие объекты из того же класса. Ассоциации может быть присвоено имя, описывающее семантику отношений. Каждая ассоциация имеет две роли, которые могут быть отражены на диаграмме. Роль ассоциации обладает свойством множественности, которое показывает, сколько соответствующих объектов может участвовать в данной связи.

Если приходится моделировать отношение типа «часть – целое», то используется специальный тип ассоциации – агрегирование. В такой ассоциации один из классов имеет более высокий ранг (целое – класс «заказ») и состоит из нескольких меньших по рангу классов (частей – класс «строка заказа»).

В UML используется и более сильная разновидность агрегации – композиция, в которой «объект – часть» может принадлежать только единственному целому. В композиции жизненный цикл частей и целого совпадают, любое удаление целого обязательно захватывает и его части.

Для ассоциаций можно задавать атрибуты и операции, создавая по обычным правилам UML классы ассоциаций.

## **5.2 Порядок выполнения работы**

1 Изучить основы построения диаграмм классов. Построить диаграммы классов.

2 Оформить отчет.

3 Осуществить защиту работы.

### **Контрольные вопросы**

1 Что такое класс?

2 Назначение диаграммы классов

3 Опишите отношение ассоциации. Приведите пример.

4 Опишите отношение обобщения. Приведите пример.

5 Опишите отношение композиции. Приведите пример.

6 Опишите отношение агрегации. Приведите пример.

## **6 Лабораторная работа № 6. Разработка модели пользователя**

**Цель работы:** научиться разрабатывать модели пользователя.

### **6.1 Теоретические сведения**

Целевая аудитория – группа пользователей, на которую сфокусировано содержание программного продукта. Целевые пользователи точно знают, в получении какой информации они заинтересованы и какой именно товар или услугу желают приобрести.

Выявление целевой аудитории поможет:

– понять, чем именно конкурирующие продукты привлекают ваших потенциальных пользователей;

– не создать «Франкенштейна»: продукт, в котором так много функций «для всех», что им не может пользоваться никто;

– сфокусироваться на главном, сэкономив средства на разработку;

- повысить эффективность работы пользователей и их удовлетворенность;
- обнаружить эффективные маркетинговые каналы для вашей компании.

Изучение целевой аудитории необходимо для проектирования взаимодействия в целом и корректировки элементов интерфейса. Это в конечном итоге направлено на удержание текущих пользователей и привлечение потенциальных пользователей.

В итоговом документе, составленном по результатам исследования целевой аудитории, приводятся следующие данные:

- 1) описание аудитории – общий взгляд на целевую аудиторию;
- 2) структура аудитории – сегментирование целевой аудитории;
- 3) объем аудитории – количество людей в составе целевой аудитории;
- 4) предпочтения – какие предпочтения свойственны целевой аудитории в целом и ее отдельным группам в частности;
- 5) факторы принятия решения о покупке – какие факторы влияют на принятие решения о покупке товара и какие моменты являются определяющими;
- 6) социально-демографические характеристики – каковы ключевые характеристики целевой аудитории: пол, возраст, социальный статус и т. п.;
- 7) тенденции – как будет изменяться объем, состав и предпочтения целевой аудитории в ближайшем будущем.

Подробные способы получения указанной информации приведены далее.

При исследовании целевой аудитории на первом этапе необходимо составить ее общий портрет, учитывая разные характеристики.

*Социально-демографические характеристики* (основополагающая информация): пол, возраст, образование, уровень дохода, род занятий. Например, сайт для подростков 15–18 лет будет отличаться от сайта для пожилых людей в возрасте 60+ лет.

*Психологические характеристики*: стиль жизни, особенности личности, черты характера, жизненная позиция, система ценностей. Более ценная информация для проектирования, чем первая группа критериев. Например, если известно, что целевая аудитория больше всего ценит время, можно спроектировать простой интерфейс и дать возможность получать не весь контент, а самое ценное для конкретной целевой группы, или даже дать инструменты персонализации каждого человека.

*Поведенческие характеристики*: повод для регистрации, искомые выгоды, частота посещаемости конкурентов, степень готовности к переходу на другой продукт, отношение к проекту (если он не новый) и т. д. Эта группа показателей одна из самых важных для проектирования. В то же время собрать эти данные будет очень сложно. Эта информация может быть у заказчика, если проектируется новая версия уже существующего проекта, у конкурента, или её нужно будет собирать по крупицам через опросы целевой аудитории или составление карт эмпатии.

*Географические характеристики*: страна, город, район. В общем случае это маловажный критерий, однако если стоит задача по проектированию национальных продуктов или продуктов с геолокацией, то важность данного критерия резко возрастает. Кроме того, если есть географическая привязка,

это может повлиять на контент, о котором тоже нужно думать при проектировании.

Для получения данных, составляющих общий портрет целевой аудитории, используют различные способы сбора информации.

1 Данные счетчика посещений (позволяют изучить все действия пользователей на сайте и конкретизировать распределение аудитории сайта по регионам, по времени и др.);

2 Статистика запросов поисковых систем. Оценить величину целевой аудитории можно по числу поисковых запросов. Подобные сервисы имеют, например, поисковые системы Яндекс (Яндекс Статистика поиска), Рамблер (Adslat Rambler) и Google (Google Статистика поиска).

3 Опросы аудитории (анкетирование пользователей, использование опросной формы или регистрации на сайте);

4 Системы аудита и традиционные опросы исследовательских компаний (агентств).

5 Анализ данных из социальных сетей и других публичных источников.

6 Составление карты эмпатии. Эмпатия – это психологический термин, который отражает способность понимать чувства и настроения других людей. Это не просто эмоциональное сопереживание, а умение поставить себя на место другого и увидеть мир его глазами.

Карта эмпатии – это метод исследования целевой аудитории. Из потребительских сегментов выбирают конкретных потребителей, с которыми проводится кропотливая работа: группа заполняет анкеты и проходит тестирование. При этом важно наблюдать за поведением каждого человека в группе и анализировать его слова и поступки, чтобы на основании всей информации составить полную картину. Главная задача карты эмпатии – составить подробный портрет типичного потребителя для конкретного продукта.

7 Разработка персонажей. Персонаж – это реалистичный собирательный образ пользователя, представляющий один сегмент целевой аудитории. Персонажи отличаются моделью поведения, целями, задачами, потребностями и ценностями.

Характеристики персонажа, кроме портрета, должны быть синтезированными. Это означает, что они должны совмещать в себе все значимые характеристики представителя группы целевой аудитории с добавлением специфических черт для придания персонажу реалистичности.

Обобщённый профиль представителя целевой аудитории, содержит в себе информацию о:

- демографических и психографических особенностях пользователей;
- задачах или целях пользователя;
- контекстах (среде), в которой пользователь работает (рабочее место, операционная система, браузер, даже освещение на рабочем месте может учитываться).

## 6.2 Порядок выполнения работы

1 Сегментировать целевую аудиторию в зависимости от параметров, наиболее влияющих на проектируемую систему (пол, возраст, род занятий, уровень компьютерной грамотности и т. д.).

2 В каждом сегменте выделить типичную персону.

3 Разработать карту эмпатии для этой персоны на основании рисунка 6.1.

<i>&lt;Категория целевой аудитории&gt;</i>	
<i>&lt;Роль персоны&gt;</i>	
	<b>Описание</b> <i>&lt;В описании указываются следующие характеристики: ФИО, пол, возраст, род занятий, семейное положение, образование, увлечения, социальный статус, место работы&gt;</i>
<b>Фотография</b>	
<b>Личные характеристики</b>	
<b>Цели:</b> <i>&lt;Перечень целей, которые пользователь стремится достичь во время использования системы&gt;</i>	
<b>Взаимодействие с продуктом</b> <i>&lt;Рабочий процесс и контекст (окружение)&gt;</i>	
<b>Неудовлетворенности и ожидания</b> <i>&lt;описание исключительных ситуаций и вытекающих из них проблем, описание дополнительных возможностей программного продукта&gt;</i>	

Рисунок 6.1 – Профиль клиента

4 Разработать профиль «ключевого персонажа» и «второстепенного персонажа» согласно рисунку 6.1.

5 Оформить отчет.

6 Осуществить защиту работы.

### Контрольные вопросы

1 Что такое целевая аудитория?

2 Для чего необходимо изучать целевую аудиторию?

3 Какие существуют способы сбора информации о целевой аудитории.

4 Какие данные о сегментах целевой аудитории необходимо указывать после проведения исследования?

5 Что такое эмпатия? Что такое карта эмпатии?

6 Из каких характеристик состоит профиль персонажа?

7 Какой персонаж называют ключевым, а какой второстепенным?

## 7 Лабораторная работа № 7. Проектирование графа диалога с использованием сетей Петри

**Цель работы:** научиться проектировать графы диалога с использованием сетей Петри.

### 7.1 Теоретические сведения

Отличительной особенностью информационного взаимодействия между пользователем и компьютером является его диалоговый характер. Каждая итерация этого взаимодействия предполагает поочередный обмен информацией между его участниками.

Для того чтобы диалог был конструктивным, должны соблюдаться следующие правила:

- участники диалога должны понимать язык друг друга;
- участники диалога не должны говорить одновременно;
- очередное высказывание должно учитываться как общий контекст диалога, так и последнюю информацию, полученную от собеседника.

При проектировании диалога необходимо определить:

- а) структуру диалога;
- б) возможный сценарий развития диалога;
- в) содержание управляющих сообщений и данных, которыми могут обмениваться человек и приложение (семантику сообщений);
- г) визуальные атрибуты отображаемой информации (синтаксис сообщений).

Общая структура диалога определяется его типом. Выделяют четыре типа диалога.

Диалог на основе командного языка. Исторически это первая из реализованных структур диалога. При такой организации диалога система не выводит ничего, кроме постоянной подсказки (приглашения на ввод команды), которая означает готовность системы к работе. Каждую команду вводят с новой строки и обычно заканчивают нажатием клавиши «ввод». Ответственность за правильность задаваемых команд ложится на пользователя. Система информирует о невозможности выполнения неверной команды, не поясняя, как правило, причин.

Диалог типа «вопрос – ответ». Структура диалога типа «вопрос – ответ» (Q&A) основана на аналогии с обычным интервью. Система берет на себя роль интервьюера и получает информацию от пользователя в виде ответов на вопросы. Это наиболее известная структура диалога; все диалоги, управляемые компьютером, в той или иной степени состоят из вопросов, на которые пользователь отвечает, однако в структуре Q&A этот процесс выражен явно. В каждой точке диалога система выводит в качестве подсказки один вопрос, на который пользователь дает один ответ. В зависимости от полученного ответа

система может решить, какой следующий вопрос задавать. Такую структуру диалога обычно применяют при проектировании экспертных систем.

**Диалог на основе меню.** Меню является наиболее популярным вариантом организации запросов на ввод данных во время диалога, управляемого компьютером. Структура типа «меню» является наиболее естественным механизмом для работы с устройствами указания и выбора: меню представляет собой изображение тех объектов, которые выбираются пользователем. Меню можно с равным успехом применять для ввода как управляющих сообщений, так и данных. Приемлемая структура меню зависит от его размера и организации, от способа выбора пунктов меню и реальной потребности пользователя в поддержке со стороны меню.

**Диалог на основе экранных форм.** Диалог на основе экранных форм допускает обработку на одном шаге диалога нескольких ответов. На практике формы используются там, где учет какой-либо деятельности требует ввода стандартного набора данных. Человек работает с формой до тех пор, пока не заполнит ее полностью и не передаст системе. Система может проверять каждый ответ непосредственно при вводе или по окончании заполнения всей формы. Такая структура позволяет повысить скорость ввода данных по сравнению со структурой типа «вопрос – ответ» и манипулировать более широким диапазоном входных данных, нежели меню; кроме того, с ней могут работать пользователи любой квалификации.

## ***7.2 Порядок выполнения работы***

1 На основании разработанных функциональной и информационной моделей (лабораторная работа № 1) граф диалога предусмотреть возможность выбора не менее 10 параметров.

2 Разработать граф состояний и переходов интерфейса проектируемой системы. Выявить типовые ситуации и описать способы их устранения.

3 Оформить работу.

4 Осуществить защиту работы.

### ***Контрольные вопросы***

1 Что такое диалог? Какие типы диалога вы знаете?

2 Что такое граф диалога? Какова его роль в процессе проектирования интерфейса?

3 Что такое граф состояний и переходов интерфейса? Каковы его составляющие?

4 Приведите примеры классов моделей, которые могут быть использованы для построения графа диалога.

5 Какие типы состояний предусмотрены в графе состояний и переходов интерфейса?

## 8 Лабораторная работа № 8. Проектирование сценария текстового диалога

**Цель работы:** научиться проектировать сценарии текстового диалога.

### 8.1 Теоретические сведения

Сценарии поведения – это ситуация взаимодействия персонажа с продуктом. Сценарий описывает действия персонажа, цели, которые он хочет достичь, и ожидания от производимых действий.

Сценарии помогают нам смоделировать максимально короткий и эффективный способ достижения персонажем своих целей на сайте и, что также важно, предлагать пользователю контекстную информацию на его пути следования по сценарию. Основная польза сценариев в том, что они помогают избегать проектирования под исключительные ситуации.

Сценарий основан на цели/потребности пользователя и поэтому не позволяет от неё отклоняться при проектировании.

При составлении сценария необходимо учитывать следующее.

1 Эффективность сценария определяется его охватом, а не глубиной. Глубина определяется, когда моделируется поведение персонажа.

2 Сценарий нужно описать полностью: от начала до логического завершения.

3 Сценарий должен быть максимально правдоподобным.

Виды сценариев:

– контекстные – описывают контекст использования информационной системы пользователем, как правило, это описание на значительном промежутке времени – например, в течение одного рабочего дня. Такие сценарии обычно создаются для программного обеспечения или для сайта, который служит рабочим инструментом;

– рабочие – сценарии, которые детализируют использование продукта и описывают конкретные операции. Рабочие сценарии в свою очередь делятся на три типа:

1) типичные – наиболее часто повторяющиеся, такие сценарии должны быть в дизайне на самом видном месте, этих сценариев редко бывает больше трёх;

2) исключительные – реализуются исключительно редко, для таких сценариев взаимодействие можно проектировать без особого внимания к деталям;

3) обязательные – без них невозможно решить некоторые задачи сайта.

Дополнительная функция сценариев, помимо описания ситуаций, – уточнение и проверка целей персонажа. Сформулировать все цели персонажа максимально чётко при его описании довольно трудно.

Особое внимание необходимо обращать на контекстные (если они есть) и типичные сценарии. Они составляют основу взаимодействия пользователя и системы.

Обязательные, но не типичные сценарии, также важны, но они, как правило, реализуются достаточно редко.

## 8.2 Порядок выполнения работы

1 Разработать сценарий взаимодействия персонажа, выделенного в лабораторной работе № 6, с программным продуктом.

2 Составить карту сценариев на основании рисунка 8.1.

Шаг:	<название шага 1>
Вопросы:	<перечислить все вопросы, которые могут возникнуть у персонажа при работе с продуктом на данном шаге>
Пожелания:	<перечислить пожелания, которые могут возникнуть у персонажа при работе с продуктом на данном шаге>
Эмоции:	<какие эмоции возникнут у персонажа от взаимодействия с продуктом>
Комментарии:	<описание окружения, исходных данных>

Рисунок 8.1 – Карта сценариев

3 Оформить работу.

4 Осуществить защиту работы.

### Контрольные вопросы

1 Что такое сценарий поведения?

2 Какие существуют виды сценариев?

## 9 Лабораторная работа № 9. Разработки модели оценки интерфейса на этапе проектирования, используя формальные методы анализа диалога на тупики

**Цель работы:** научиться разрабатывать модели оценок интерфейса на этапе проектирования, используя формальные методы анализа диалога на тупики.

### 9.1 Теоретические сведения

Ошибки в программном обеспечении, ухудшающие его эксплуатационные характеристики, приводят к убыткам как потребителей, так и разработчиков.

Однако эти характеристики зависят и от множества различных факторов, и одним из самых значительных из них является качество пользовательского интерфейса.

Ошибки в интерфейсе наиболее критичны для таких свойств продукта, как производительность пользователей системы (этот показатель определяется из расчета, сколько единиц продукции в час может создать пользователь), количество человеческих ошибок (как часто пользователь ошибается при выполнении определенной операции), скорость обучения работе с системой и субъективное удовлетворение пользователей. Удобство применения является «клеем», который скрепляет все части, которые должны соединиться вместе, чтобы составить какой-либо продукт.

Согласно определению международной организации стандартизации (ISO), удобство применения – это эффективность, рентабельность и удовлетворение, с которым пользователи смогут выполнить те или иные задачи в заданной среде.

Существуют следующие способы проведения тестирования:

- наблюдение;
- проведение опросов и исследований;
- контекстуальные опросы;
- эвристические оценки;
- работа с выделенными группами;
- лабораторное тестирование.

Каждый из этих способов характеризуется различной степенью сложности и ориентирован на конкретные категории пользователей.

Важной частью любого тестирования является правильный подбор пользователей (и достаточного их числа). Участники теста должны быть типичными представителями пользователей данного продукта. Их количество зависит от многих факторов, в том числе времени, ресурсов, проекта теста, типа тестируемых задач и вида статистического анализа, который вы планируете провести по полученным результатам.

Прежде чем планировать и проводить тестирование на удобство применения, следует четко определить цели и задачи, стоящие перед ним. Эта ответственность лежит на владельцах продукта, специалистах, составляющих планы, проектировщиках и разработчиках. Цель – обеспечение преимуществ продукта перед конкурирующими в области простоты изучения, эффективности, гибкости и т. д. Для большей точности и достоверности каждая цель делится на подзадачи.

## ***9.2 Порядок выполнения работы***

1 Спроектировать тесты для оценки удобства применения разработанного интерфейса.

2 Выполнить спроектированные тесты с привлечением сторонних пользователей.

3 Проанализировать полученные результаты.

4 Оформить отчет. Отчет должен содержать следующее:

- титульный лист, название и цель работы;
- вариант задания и постановку задачи;
- таблицу результатов выполнения тестов (таблица 9.1).

Таблица 9.1 – Результаты выполнения тестов

Краткое описание тестируемого действия	Ожидаемый результат	Фактический результат

5 Осуществить защиту работы.

### ***Контрольные вопросы***

- 1 Что такое тестирование на удобство применения?
- 2 Какие требования предъявляются к тестированию?
- 3 Какие государственные стандарты регламентируют проведение тестирования?

## **10 Лабораторная работа № 10. Разработка многооконных интерфейсов информационной системы в инструментальной среде разработки для использования в системах неразрушающего контроля**

**Цель работы:** научиться разрабатывать многооконные интерфейсы и информационной системы в инструментальной среде разработки для использования в системах неразрушающего контроля.

### ***10.1 Теоретические сведения***

Традиционно существуют три разновидности приложений, которые могут создаваться MS:

1) приложения, основанные на диалоговом окне. Такие приложения предоставляются пользователям в виде единого диалогового окна, с помощью которого может быть осуществлен доступ ко всем функциональным возможностям;

2) однодокументные интерфейсы (Single Document Interface, SDI). Такие приложения предоставляются пользователям в виде меню, одной или нескольких линеек инструментов и одного окна, в котором пользователь может выполнять определенные действия;

3) многодокументные интерфейсы (Multi-Document Interface, MDI). Такие приложения предоставляются пользователям в таком же виде, что SDI-приложения, однако обладают способностью одновременно поддерживать несколько открытых окон.

Приложения, основанные на диалоговом окне, обычно представляют собой небольшие одноцелевые приложения, которые ориентированы либо для решения конкретной задачи, требующей ввода небольшого количества данных, либо для работы с какими-то необычными типами данных. В качестве примера такого приложения можно привести Calculator (калькулятор), поставляемый вместе с MS Windows.

Однодокументные интерфейсы, как правило, предназначены для решения какой-то одной конкретной задачи, при этом они позволяют пользователю загружать в приложение единственный документ, с которым он и будет вести работу. Эта задача предполагает выполнение пользователем большого количества действий, и зачастую пользователю могут потребоваться возможности, позволяющие сохранять или загружать плоды своего труда. Хорошим примером SDI-приложений могут служить MS Paint и WordPad, также поставляемые совместно с MS Windows.

Многодокументные интерфейсы почти полностью аналогичны SDI-приложениям, за исключением того, что они обладают возможностью поддерживать более одного открытого документа в различных окнах, которые могут быть открыты одновременно. Одним из простых признаков MDI-приложения является наличие пункта Window на правой стороне линейки меню перед пунктом Help. Примерами MDI-приложений служат Adobe Acrobat Reader и MS Word 97.

Четвертый тип приложений представлен MS Office 2000. Этот тип является смесью SDI- и MDI-приложений: окна, предоставляемые пользователю, имеют различное местоположение, и каждое окно отображается в линейке заданий. Такое приложение представляет собой несколько MDI-приложений, поскольку основное приложение не будет закрыто до тех пор, пока не будут закрыты все окна, а с помощью пункта меню Windows можно выбирать, какой именно из открытых документов будет просматриваться, хотя при этом собственно пользовательский интерфейс представляет собой SDI-интерфейс.

### ***10.2 Порядок выполнения работы***

- 1 Разработать MDI-приложение в среде программирования Visual Studio.
- 2 Оформить отчет.
- 3 Осуществить защиту работы.

### ***Контрольные вопросы***

- 1 Опишите структуру SDI-приложения.
- 2 Опишите структуру MDI-приложения.
- 3 Опишите технологию создания MDI-приложения в Visual Studio.

## **11 Лабораторная работа № 11. Разработка пользовательского интерфейса прямого манипулирования для использования в системах неразрушающего контроля**

**Цель работы:** научиться разрабатывать пользовательские интерфейсы прямого манипулирования для использования в системах неразрушающего контроля.

## *11.1 Теоретические сведения*

Возможность прямого манипулирования, предусмотренная в WIMP-интерфейсах, позволяет разрабатывать для приложений объектно-ориентированные интерфейсы прямого манипулирования.

Интерфейсы данного типа на внешнем уровне используют директивную форму диалога: ввод команды осуществляется при выполнении определенных действий с пиктограммой объекта мышью. Основными элементами этих интерфейсов являются метафоры, объекты, представления объектов и технология Drag and Drop («перетащил и бросил»).

Метафора – мысленный перенос свойств или признаков одного объекта на другой, чем-то аналогичный первому. Использование метафор в интерфейсах предполагает активизацию имеющегося у пользователя опыта (ментальных моделей выполнения аналогичных действий в повседневной жизни или на рабочем месте).

Интерфейс прямого манипулирования должен обеспечивать пользователю среду, содержащую знакомые ему элементы, с которыми пользователь не раз встречался в профессиональной деятельности или в быту, и предоставлять ему возможность манипулирования отдельными объектами. Наличие метафор упрощает для пользователя процесс освоения интерфейса. Например, метафора «Выбрасывание мусора», которую использует Windows для удаления файлов, облегчает пользователю усвоение этой операции.

Использовать метафоры надо очень аккуратно, т. к. при этом смысл придается всем элементам интерфейса, например, похожие элементы должны вести себя похожим образом, а элементы, выделенные одним цветом, должны находиться в определенной связи друг с другом. Семантическое несоответствие между элементами интерфейса тем, что от них ожидают, и тем, что они на самом деле выполняют, раздражает и дезориентирует пользователей.

Следует также учитывать, что полное соответствие может обмануть ожидание пользователя, т. к. все-таки он оперирует не реальными предметами, а их моделями, а значит, его возможности ограничены, о чем необходимо напоминать.

При реализации метафор все большая роль уделяется средствам мультимедиа, в основном анимации. Движение привлекает внимание, а резкая смена кадров требует некоторого времени на определения связи данного кадра с предыдущим и на изучение этого кадра. Следовательно, используя мультипликацию, можно не только развлекать пользователя, но и «готовить» его к смене кадров, сокращая время, необходимое на адаптацию к изменившейся ситуации.

Однако, решая проблемы, связанные с особенностями восприятия человека, анимационные интерфейсы создают дополнительные проблемы для разработчиков и программистов. К ставшим привычными функциональному и интерфейсному уровням программы добавляется еще и визуальный уровень. Программа, реализующая такой интерфейс, никогда не простаивает,

т. к. во время ожидания ввода команды пользователя она продолжает отображать соответствующие кадры. В основе таких программ лежит временное программирование. В отличие от событийного программирования, которое позволяет связывать изображение на экране с внешними и внутренними событиями в системе, временное программирование обеспечивает изменение проектируемой последовательности кадров в зависимости от состояния моделируемых процессов и действий пользователя.

Объекты интерфейса прямого манипулирования и их представления.

Существуют три основных типа объектов интерфейсов прямого манипулирования: объекты-данные, объекты-контейнеры и объекты-устройства.

Объекты-данные снабжают пользователя информацией. Это могут быть тексты, изображения, электронные таблицы, музыка, видео и т. п., а также любая их комбинация. В рамках операционной системы таким объектам соответствуют приложения, которые запускаются при раскрытии объекта. В масштабе приложения объекту соответствует одна или несколько форм, в которых содержимое объекта представляется в разных видах. Операции с содержимым объекта реализуются обработчиками событий формы.

Объекты-контейнеры могут манипулировать своими внутренними объектами, в том числе и другими контейнерами, например, копировать их или сортировать в любом порядке. К типичным контейнерам относятся папки, корзины и т. п. При раскрытии контейнера демонстрируются сохраняемые им компоненты и появляется возможность ими манипулировать. Компоненты при этом могут обозначаться пиктограммами или представляться в виде таблицы.

Объекты-устройства часто представляют устройства, существующие в реальном мире: телефоны, факсы, принтеры и т. д. Их используют для обозначения этих устройств в абстрактном мире интерфейса. При раскрытии такого объекта, как правило, можно увидеть его настройки.

Следует иметь в виду, что различие между типами объектов является условным, т. к. один и тот же объект в разных ситуациях может вести себя то как объект-данные, то как объект-устройство, то как объект-контейнер.

Например, принтер обычно рассматривают как объект-устройство, но он может обладать и свойствами объекта-контейнера, т. к. может содержать объекты данные в очереди на печать. Соответственно, в Windows объект контейнер/устройство принтер имеет, помимо пиктограммы, еще два представления: окно очереди на печать и окно настроек.

## ***11.2 Порядок выполнения работы***

1 Добавить в программу, разработанную в предыдущих лабораторных работах, объекты интерфейса прямого манипулирования.

2 Оформить отчет.

3 Осуществить защиту работы.

### ***Контрольные вопросы***

- 1 Что такое метафора?
- 2 Назовите основные типы объектов интерфейсов прямого манипулирования. Охарактеризуйте каждый из них.
- 3 В каких случаях предпочтительно использовать элементы анимации?

## **12 Лабораторная работа № 12. Применение технологии Drag and Drop при создании интерфейса для использования в системах неразрушающего контроля**

**Цель работы:** научиться применять технологию Drag and Drop при создании интерфейса для использования в системах неразрушающего контроля.

### ***12.1 Теоретические сведения***

Технология Drag and Drop («перетащил и бросил») определяет основные принципы прямого манипулирования, описанные в руководстве по разработке пользовательских интерфейсов фирмы IBM (CUA- Common User Access):

- результат перемещения объекта должен соответствовать ожиданиям пользователя;
- пользователи не должны неожиданно терять информацию;
- пользователь должен иметь возможность отменить неправильное действие.

Эта технология также определяет основные принципы визуализации операции прямого манипулирования:

- исходное выделение – используется в качестве обратной связи пользователю, чтобы сообщить ему, что объект захвачен, в Windows с этой целью применяется выделение цветом;
- визуализация перемещения используется для идентификации выполняемого действия;
- целевое выделение используется для идентификации пункта назначения, показывая, таким образом, куда «упадет» объект, если его отпустить в текущий момент времени;
- визуализация действия используется для обозначения времени ожидания завершения операции, обычно с этой целью применяют анимацию или изменение формы курсора на «песочные часы».

### ***12.2 Порядок выполнения работы***

- 1 Добавить в программу (изменить часть программы), разработанную в предыдущих лабораторных работах, возможность выполнять отдельные операции по технологии drag-and-drop.
- 2 Оформить отчет.
- 3 Осуществить защиту работы.

### ***Контрольные вопросы***

- 1 Что такое метафора?
- 2 Назовите основные типы объектов интерфейсов прямого манипулирования. Охарактеризуйте каждый из них.
- 3 В каких случаях предпочтительно использовать элементы анимации?

### **Список литературы**

- 1 **Терещенко, П. В.** Интерфейсы информационных систем / П. В. Терещенко, В. А. Астапчук – Новосибирск : НГТУ, 2012. – 67 с.
- 2 **Овчеренко, В. А.** Периферийные устройства информационных систем. Физические принципы организации и интерфейсы ввода-вывода : учебное пособие / В. А. Овчеренко, В. Г. Токарев. – Новосибирск: НГТУ, 2018. – 75 с.
- 3 **Ткаченко, О. Н.** Взаимодействие пользователей с интерфейсами информационных систем для мобильных устройств: исследование опыта : учебное пособие / О. Н. Ткаченко – Москва: Магистр; ИНФРА-М, 2020. – 152 с.
- 4 **Гаврилова, И. В.** Разработка приложений : учебное пособие / И. В. Гаврилова. – 3-е изд., стер. – Москва : ФЛИНТА, 2017. – 242 с.