

## Лабораторная работа №8

### Применение отладочных классов в проекте

Краткие теоретические и учебно-методические материалы по теме практической работы

Сущность объектно-ориентированного подхода к программированию заключается в том, что основные идеи объектно-ориентированного подхода опираются на следующие положения:

- программа представляет собой модель некоторого реального процесса, части реального мира;
- модель реального мира или его части может быть описана как совокупность взаимодействующих между собой объектов;
- объект описывается набором параметров, значения которых определяют состояние объекта, и набором операций (действий), которые может выполнять объект;
- взаимодействие между объектами осуществляется посылкой специальных сообщений от одного объекта к другому. Сообщение, полученное объектом, может потребовать выполнения определенных действий, например, изменения состояния объекта;
- объекты, описанные одним и тем же набором параметров и способные выполнять один и тот же набор действий представляют собой класс однотипных объектов.

С точки зрения языка программирования класс объектов можно рассматривать как тип данного, а отдельный объект - как данное этого типа. Определение программистом собственных классов объектов для конкретного набора задач должно позволить описывать отдельные задачи в терминах самого класса задач (при соответствующем выборе имен типов и имен объектов, их параметров и выполняемых действий).

Таким образом, объектно-ориентированный подход предполагает, что при разработке программы должны быть определены классы используемых в программе объектов и построены их описания, затем созданы экземпляры необходимых объектов и определено взаимодействие между ними.

Классы объектов часто удобно строить так, чтобы они образовывали иерархическую структуру. Например, класс «Студент», описывающий абстрактного студента, может служить основой для построения классов «Студент 1 курса», «Студент 2 курса» и т.д., которые обладают всеми свойствами студента вообще и некоторыми дополнительными свойствами, характеризующими студента конкретного курса. При разработке интерфейса с пользователем программы могут использовать объекты общего класса «Окно» и объекты классов специальных окон, например, окон информационных сообщений, окон ввода данных и т.п. В таких иерархических структурах один класс может рассматриваться как базовый для других, производных от него классов. Объект производного класса обладает всеми свойствами базового класса и некоторыми собственными свойствами, он может реагировать на те же типы сообщений от других объектов, что и объект базового класса и на сообщения, имеющие смысл только для производного класса. Обычно говорят, что объект производного класса наследует все свойства своего базового класса.

Некоторые параметры объекта могут быть локализованы внутри объекта и недоступны для прямого воздействия извне объекта. Например, во время движения объекта-автомобиля объект-водитель может воздействовать только на ограниченный набор органов управления (рулевое колесо, педали газа, сцепления и тормоза, рычаг переключения передач) и ему

недоступен целый ряд параметров, характеризующих состояние двигателя и автомобиля в целом.

Существует ряд техник, предназначенных для обеспечения качества кода, выполняемых по мере его конструирования.

Основные техники обеспечения качества, используемые в процессе конструирования, включают:

1. рефакторинг;
2. модульное (unit) и интеграционное (integration) тестирование;
3. разработка с первичностью тестов (test-first development - тесты пишутся до конструирования кода);
4. пошаговое кодирование (деятельность по конструированию кода разбивается на мелкие шаги, только после тестирования результатов которых производится переход к следующему шагу кодирования; известен также как итеративное кодирование с тестированием);
5. использование процедур утверждений (assertion);
6. отладка (в привычном понимании — debugging);
7. технические обзоры и оценки (review);
8. статический анализ.

Выбор и использование конкретных техник часто диктуется стандартами (внутренними и внешними), используемыми проектной командой, а также зависят от опыта и подготовленности специалистов, занимающихся конструированием кода.

Ключевым аспектом защитного программирования является использование отладчиков.

Отладчик (debugger) — программа, предназначенная для поиска ошибок в других программах и позволяет выполнять трассировку, отслеживать, устанавливать или изменять значения переменных в процессе выполнения кода, устанавливать и удалять контрольные точки или условия остановки и т. д.

При программировании важно:

1. Не использовать ограничения промышленной версии в отладочной версии.
2. Внедрять поддержку отладки как можно раньше.
3. Использовать наступательное программирование:
  - реализовать так, чтобы все утверждения завершали работу программы;
  - заполнять всю выделенную память, для обнаружения ошибок выделения памяти;
  - заполнять все файлы и потоки;
  - при попадании в операторе case в ветви default или else программа прекращает работу;
  - заполнять объекты мусором перед их удалением;
  - настроить отправку журналов ошибок по электронной почте.
4. Запланировать удаление отладочных средств. Для реализации этого использовать средства автоматизации контроля версий и сборки программ (ant, maven, make), встроенный или собственный препроцессор, отладочные заглушки.

В промышленной версии желательно придерживаться следующих рекомендаций:

1. Оставить код, который проверяет только существенные ошибки.
2. Удалить код, проверяющий незначительные ошибки.
3. Удалить код, приводящий к прекращению работы программы.
4. Оставить код, который позволяет аккуратно завершить программу.
5. Регистрировать ошибки для технической поддержки.
6. Оставленные сообщения об ошибках должны быть дружелюбны.

#### Задания для выполнения:

1. Создать основной класс. Создать 3 класса наследника. Создать в основном классе 5 методов и 3 свойства, связанных с его назначением, отличных от методов и свойств других подгрупп.
2. Придумать и создать, как минимум, по 2 метода в каждом классе-наследнике (всего не меньше 6) и хотя бы по 2 свойства (всего не меньше 6). Методы и свойства должны быть связаны с особенностями класса-наследника.
3. Сделать, как минимум, 3 метода основного класса переопределёнными в классах наследника. Сделать, как минимум, 1 метод, который нельзя переопределить.
4. Скомпилировать программу, демонстрирующую работу каждого из методов каждого из классов.

#### Варианты на лабораторную работу

В ходе лабораторной работы нужно разработать программу.

- 1,11,21. Основной класс «Автомобили», 3 наследника «Грузовики», «Легковые» и «Автобусы».
- 2,12,22. Основной класс «Одежда», 3 наследника «Куртки», «Шубы» и «Пуховики».
- 3,13,23. Основной класс «Еда», 3 наследника «Супы», «Закуски» и «Напитки».
- 4,14,24. Основной класс «Гаджеты», 3 наследника «Планшеты», «Смартфоны» и «Нетбуки».
- 5,15,25. Основной класс «Овощи», 3 наследника «Морковь», «Свёкла» и «Лук».
- 6,16,26. Основной класс «Песни», 3 наследника «Частушки», «Баллады» и «Романсы».
- 7,17,27. Основной класс «Оружие», 3 наследника «Пистолеты», «Автоматы» и «Пулеметы».
- 8,18,28. Основной класс «Документы», 3 наследника «Справки», «Приказы» и «Заявления».
- 9,19,29. Основной класс «Погода», 3 наследника «Пасмурно», «Ясно» и «Ураган».
- 10,20,30. Основной класс «Звери», 3 наследника «Зайцы», «Волки» и «Лисы».

5. Оформить отчет.

6. Ответить на контрольные вопросы

Вопросы для закрепления теоретического материала к практическому занятию:

1. В чем заключается сущность объектно-ориентированного подхода при разработке программного продукта?
2. На что направлен объектно-ориентированный анализ?
3. Перечислите основные достоинства объектно-ориентированной методологии по сравнению со структурными методами.
4. Перечислите принципы объектного подхода. Дайте им краткие характеристики
5. Назовите основные методики объектно-ориентированного анализа.
6. Какие понятия, необходимые для проектирования системы включает концептуальная модель?

Порядок выполнения отчета по практической работе

Защита отчета по практической работе заключается в предъявлении преподавателю полученных результатов, демонстрации полученных навыков и ответах на вопросы преподавателя.

