

СОВРЕМЕННЫЕ АНТИПАТТЕРНЫ*БНТУ, г. Минск**Научный руководитель: канд. техн. наук, доцент Дробыш А. А.*

В настоящее время, в эпоху информационного общества, а также повсеместного внедрения программного обеспечения, все больше внимание отводится процессу программирования и собственно разработчикам. В процессе создания новых приложений очень часто программисты используют определенные шаблоны, которые были придуманы до них, и лишь в нужном русле их применяют (копируют). Так вот данные шаблоны называются паттерны. Однако, существуют примеры неудачного применения данных шаблонов и тогда паттерн переходит в состояние анти-паттерна, то есть шаблон ошибки, который часто встречаются при достижении поставленной цели. И задача любого программиста избегать данных ошибок при разработке ПО. Давайте рассмотрим несколько наиболее распространенных примеров данных анти-паттернов.

Во-первых, Copy and Paste Programming (программирование копированием и вставкой). Обычно при решении нескольких однотипных задач программист использует определенные заготовки или примеры уже решенных задач и просто путем обычного копирования и вставки пытается их решить. Однако, существует несколько сложностей и опасностей, которые могут подстергать его:

Проблема 1 – при бездумном копировании кода возможны варианты, когда программист забывает внести определенные изменения под свой исходных код, под свою задачу, и тогда, это может приводить к неверному результату или неработоспособности всего кода.

Проблема 2 – часто программист не проверяет работоспособность копируемого кода и не догадывается о возможных

ошибках и багах в нем. Вследствие чего программисту потом приходится часами сидеть и разбираться в том, что он скопировал для того, чтобы решить проблему или просто приходится самому набирать заново этот участок кода.

Почему же программисты используют бездумное копирование? Основная причина – это отсутствие опыта и нежелание разбираться в процессе решения задачи.

Во-вторых, Spaghetti code (Спагетти-код). Это пример многократного использования копирования и вставки, а также запутанного и слабо разбираемого кода, который может содержать огромное количество ненужного строк и объектов, никакого отношения к решению поставленной задачи не имеющих. Причинами получения спагетти-кода являются:

- Недостаток опыта в разработке программного обеспечения;
- Удаленная работа нескольких программистов;
- Нежелание разбираться в том участке кода, который был скопирован из других источников.

Полученный код может представлять собой разветвление самых разнообразных циклов и подпрограмм, которые в последствии не разберет даже сам разработчик. Как же решить данную проблему? Переписать программу или сесть и хорошенько подумать, что Вам надо для решения поставленной цели, а что нет, иначе данное «спагетти» будет преследовать Вас на протяжении всей вашей карьеры.

В-третьих, Golden hammer (Золотой молоток). Это применение одного решения как универсального средства для решения всех задач. Обычно данное мышление присуще начинающим программистам, которые знают один алгоритм решения и придерживаются его в любом случае, какая бы задача перед ним не стояла. Причина проявления данного антипаттерна – это некомпетентность или недостаточные знания программиста в данной области. Как решить данную проблему? Необходимо осознавать, что для любой задачи существует ее особенное оптимальное решение, потому что в большин-

стве случае «универсальный код» очень громоздок и разветвлен и требует много ресурсов для реализации.

Следующий анти-паттерн – Magic numbers (Магические числа). Это использование в коде констант, значение которых не указано в комментариях и не несет явного влияния на процесс работы программы. Данные числа усложняют процесс понимания программы и собственно сам код. Иногда программисту просто лень писать несколько десятков строчек в коде программы, и он задает какой-то переменной числовое значение, которое потом постоянно вызывает, но если данный код попадет в руки другому человеку не факт, что он сможет разобраться с этим «числом». Да и сам разработчик через некоторое время может не помнить, зачем он вообще вводил эту константу. Как же не допускать этого? Для этого необходимо еще в начале программы задавать все константы, необходимые для работы программы.

И еще один популярный анти-паттерн, Reinventing the wheel (Повторное изобретение колеса). Данный анти-паттерн говорит о том, что разработчик придумывает свое собственное решение для задачи, для которой уже существует решение, при этом существующее решение является более оптимальным, чем то, что придумал программист. В большинстве случаев это приводит к потере времени и все равно в последующем к замене своего придуманного решения на ранее известное и оптимальное. Для избегания данного анти-паттерна необходимо увеличивать свои знания при работе с определенными типами задачи.

И последний популярный анти-паттерн – это бездумное комментирование. Суть его заключается в том, что программист добавляет комментарии, где это нужно и не нужно и иногда количество комментариев превосходит размер кода программы. Данный инструмент необходимо использовать для пояснения сложных для понимания моментов в програм-

ме, а не для диалога с пользователем или другим разработчиком, который будет потом использовать этот код.

Итак, анти-паттерны – это то, через что, наверное, проходит каждый разработчик, но некоторые проходят этот этап, усваивают то, что их надо обходить стороной, и продолжают свой путь к вершинам программирования. Для того, чтобы избегать использование анти-паттернов в своей работе необходимо усвоить несколько простых правил: увеличивать кругозор знаний в тех языках программирования, на которых работаете; использовать принцип «Поспешешь – людей насмешешь»; при использовании кода, разработанного до Вас, понимать его смысл и каждую операцию в нем, а не просто бездумно вставлять его, просто потому, что он работает, а Ваш – нет. При усвоении этих правил программист будет огражден от большинства анти-паттернов и данная проблема не будет портить его жизнь.

УДК 621.762.4

Руйчева А. П.

ИНВЕСТИЦИИ В ДРАГОЦЕННЫЕ МЕТАЛЛЫ

БНТУ, г. Минск

Научный руководитель: канд. техн. наук, доцент Дробыш А. А.

У каждого человека, владеющего свободным денежным капиталом, возникает вопрос: куда и как инвестировать финансовое состояние в часы экономической нестабильности, получив приумножение денег и обеспечив им сохранность.

В сегодняшние дни во всех информационных источниках: газетах, журналах, радио, телевидении, Всемирной Сети – обилие рекламных сообщений с разнообразными предложениями для частных вложений. К сожалению, предложенная информация крайне противоречива, и ее недостаточно для принятия решения. Ни один из вариантов вложений не исключает риски и не гарантирует высокую доходность.