Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Казанский (Приволжский) федеральный университет»

Институт вычислительной математики и информационных технологий

**ОТЧЕТ**

**по технологической практике**

|  |  |
| --- | --- |
| Обучающийся Еникеева А.И. 09-116  (ФИО, группа) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Руководитель практики от КФУ

старший преподаватель кафедры анализа данных и технологий программирования Матренина Ольга Михайловна

(должность, ФИО)

Оценка за практику \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя практики)

Дата сдачи отчета 30.12.2022

Казань – 2022

Содержание

[Введение 3](#_Toc122971250)

[Анализ предметной области и обоснование актуальности поставленной задачи 4](#_Toc122971251)

[Заключение 9](#_Toc122971252)

[Список использованных источников 10](#_Toc122971253)

# Введение

Технологическая практика проходила на кафедре анализа данных и технологий программирования ИВМиИТ. Индивидуальное задание было получено в процессе коммуникации с руководителем практики.

Актуальность работы состоит в важности закрепления полученных знаний, полученных в процессе обучения в высшем учебном заведении.

Целью технологической практики является анализ предметной области и обоснование актуальности создания интегрированной среды разработки для построения информационно-расчетных приложений.

Для достижения данной цели потребовалось решить следующие задачи:

* Изучить материалы в сфере создания интегрированной среды разработки для построения информационно-расчетных приложений;
* Проанализировать предметную область;
* Обосновать актуальность поставленной задачи.

# Анализ предметной области и обоснование актуальности поставленной задачи

Одной из характерных особенностей развития средств проектирования и разработки программных комплексов на современном этапе является появление разнообразных эффективно используемых средств автоматизации создания программных приложений.

Среди них можно выделить работы по созданию:

* CASE технологий [1];
* паттернов проектирования [2];
* работ по объектно-ориентированным средствам описания программных систем [4, 5].

Большая часть этих работ имеет универсальную направленность и, как правило, не учитывает или недостаточно учитывает специфику предметной области, приводя к использованию громоздких и неадекватных инструментальных средств. С другой стороны, слишком узкая специализация технологических средств обречена в большинстве случаев на одноразовое применение. Отсюда следует необходимость использования подходов, предусматривающих гибкое сочетание универсальности со специализацией на основе выбора класса, объединяющего множество подобных специализированных задач, и построения специализированной интегрированной среды разработки соответствующих приложений.

Интегрированная среда разработки определяется как совокупность средств и методов, предназначенных для автоматизации процессов проектирования и разработки программных приложений. Понятие интегрированной среды, появившееся еще в процедурных системах программирования, впоследствии оказалась очень важным и неотъемлемым атрибутом технологии объектно-ориентированного программирования. Интегрированная среда разработки традиционно включает в себя средства описания и реализации многоуровневых моделей программных систем, различные библиотеки классов, инструментальные средства программирования и генераторы программ.

В данной работе рассматривается один из таких подходов, ориентированный на создание специализированных средств, обеспечивающих автоматизацию проектирования и разработки так называемых информационно-расчетных задач, к которым относятся задачи компьютерной бухгалтерии, банковской деятельности, статистики, информационного поиска и тому подобное.

Основной особенностью упомянутого класса задач является относительно простые структуры данных, в большинстве случаев, адекватно представляемых аппаратом реляционной алгебры. Однако, такое важное понятие, как иерархия абстракций, определяющее один из основных принципов объектно-ориентированного подхода, в рамках реляционной модели, как уже упоминалось, представляется далеко неадекватными способами, что может существенно отразиться на эффективности функционирования соответствующей интегрированной среды разработки программных приложений. Особенностью предлагаемого подхода является расширение реляционной модели путем включения аппарата фрейма с целью гибкого сочетания простоты реляционной модели со средствами эффективного представлений – иерархии абстракций. Основное предназначение включаемого фрейма – это построение специализированной среды разработки, обеспечивающей возможность эффективного создания различных приложений для автоматизации решения информационно-расчетных задач, связанных с иерархической организацией данных и процессов их обработки.

Одним из ключевых принципов создания подобной интегрированной среды, основанной на объектно-ориентированной модели, является возможность повторного использования различных компонент из разрабатываемых ранее программных средств (так называемых паттернов проектирования). Объектно-ориентированный фрейм разработки реализуется в виде множества абстрактных классов, образующих функциональный базис, а также множества классов, специфичных для соответствующей предметной области.

Расширение реляционной модели путем включения аппарата фреймов обеспечивает процесс эффективного построения и упорядочения иерархии абстракций при проектировании и разработке приложений из области информационно-расчетных задач. Уровни абстракций, получаемые при этом, соответствуют степеням специализации разрабатываемого приложения и позволяет разработчику адекватно выделать обобщающие аспекты различных частей приложения на каждом уровне. Такой подход уменьшает вероятность появления в разрабатываемом проекте дублирующих компонент и исключает в идеале необоснованную избыточность разрабатываемого проекта. С другой стороны, этот подход, предусматривающий адекватную структуризацию процесса разработки и проектирования приложения, оказывается полезным для повторного использования созданных при этом компонент в практической разработке других подобных приложений.

Большая часть повторно-используемого программного обеспечения фокусируется на повторно-используемых кодах, таких, как повторно-используемые части существующих приложений, повторно-используемые библиотечные функции или повторно-используемые заготовочные компоненты. В последних работах, посвященных паттернам проектирования и объектно-ориентированным фреймам разработки, интерес смещается от непосредственно повторно-используемых кодов к повторному использованию ранее выполненных разработок.

Паттерн проектирования представляет собой некоторую абстрактную модифицируемую заготовку, многократно используемую в процессах проектирования, разработки и реализации различных программных приложений. Иными словами, паттерн идентифицирует и абстрагирует ключевые аспекты структуры общего решения определенной задачи. Паттерны проектирования обеспечивает повторное использование тех частей проекта разработки, которые могут неоднократно повторяться в данном проекте или других проектах по разработке программного обеспечения. В данном контексте объектно-ориентированный фрейм разработки можно рассматривать как повторно-используемый проект некоторой системы или подсистемы. Объектно-ориентированный фрейм разработки организуется, как набор абстрактных классов, которые определяют основные функциональные возможности фрейма разработки наряду с конкретными классами для специфических приложений, включенными для полноты. Пользователи фрейма разработки завершают или расширяют фрейм разработки добавлением специальных компонентов или функций с целью создания соответствующих приложений.

Проектирование фрейма разработки отличается от проектирования отдельного приложения, по крайней мере, в двух аспектах. Во-первых, различен уровень абстракции. Фреймы разработки предназначены для обеспечения общего решения, множества подобных задач или связанных задач или задач в одной области, в то время- как приложения обеспечивают определенное решение для специальной задачи. Во-вторых, фреймы разработки в связи со своей природой являются неполными (незавершенными) в том смысле, что именно посредством заполнения (завершения) фрейма осуществляется создание соответствующих приложений из фрейма.

Фрейм разработки не обеспечивает функциональную полноту, требуемую для специальной области, но вместо этого, обобщает функции, необходимые для определенного класса приложений, включая эти функции в

общий проект и предоставляя пользователю фрейма разработки возможность

детализации параметров в процессе построения соответствующих специализированных приложений. В связи с этим, основная задача проектирования фрейма разработки фокусируется на обеспечении средств, обеспечивающих гибкость обобщений и эффективность адаптации фреймов, к решению специализированных проблем. Объектно-ориентированные технологии представляют естественную среду для фреймов- разработки. Аналогично тому, что одни классы могут представляться как. экземпляры других более абстрактных классов, каждое приложение также можно рассматривать как результат специализации некоторого абстрактного фрейма

разработки.

# Заключение

Таким образом, проанализировав предметную область и обосновав актуальность поставленной задачи, можно сделать вывод, что создание интегрированной среды разработки для построения информационно-расчетных приложений актуально, поскольку позволит автоматизировать процессы создания приложений из области информационно-расчетных задач.

Все поставленные цели и задачи выполнены.

# Список использованных источников

1. Вендеров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М. “Финансы и статистика” – 1998 г. – 176c.
2. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования – СПб: Питер, 2004 г. – 366с.
3. Гради Буч, Джеймс Рамбо, Ивар Джекбсон – Язык UML. Руководство пользователя. Издательство ДМК Пресс, 2001 г. – 496с.
4. Элиенс А. Принципы объектно-ориентированной разработки программ. 2-ое изд. Изд. Дом “Вильямс”. Москва – С.-Петербург – Киев, 2002 г. – 495с.