

Рекомендации к Методике оценки трудоемкости разработки программных средств

1. НОРМАТИВНЫЕ ССЫЛКИ

При разработке настоящего документа использованы:

ГОСТ Р ИСО/МЭК 12207-99	Информационная технология. Процессы жизненного цикла программных средств.
ГОСТ Р ИСО/МЭК 14764–2002	Информационная технология. Сопровождение программных средств.
ГОСТ 19.101–77	Виды программ и программных документов.
ГОСТ 19781–90	Обеспечение систем обработки информации программное. Термины и определения.
ГОСТ 28806–90	Качество программных средств. Термины и определения.
ГОСТ 34.003–90	Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.
ГОСТ 34.601–90	Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
ГОСТ 20886-85	Организация данных в системах обработки данных. Термины и определения.

2. ОБЩИЕ ПОЛОЖЕНИЯ

2.1. Методика применяется для расчета трудоемкости разработки **программных средств**.

Программное средство (по ГОСТ 28806-90) – объект, состоящий из программ, процедур, правил, а также, если предусмотрено, сопутствующих им документации и данных, относящихся к функционированию системы обработки информации.

Допускается использование Методики для оценки трудоемкости разработки отдельных компонентов (программных средств) автоматизированной системы, если такие компоненты можно классифицировать как программные средства. Точность расчета

трудоемкости при этом зависит от того, насколько глубоко интегрирована компонента в систему: чем меньше связей – тем выше точность.

Декомпозиция АС должна производиться с учетом функциональных особенностей составных частей АС, возможности их независимой разработки, ввода в действие (см. Приложение 10).

2.2. Не допускается оценка трудоемкости разработки АС как единого программного средства. Такой подход приведет к занижению трудоемкости за счет того, что не будут учитываться элементы, являющиеся общими для нескольких компонентов АС.

2.3. На трудоемкость разработки ПС оказывает влияние ряд объективных и субъективных факторов. Для учета влияния этих факторов в Методике введены соответствующие поправочные коэффициенты. Если определить значение какого-либо поправочного коэффициента не представляется возможным, то его значение следует принять равным 1.00.

Таблица 3.1. Соответствие средств разработки

Средство разработки в соответствии с прежней версией Методики	Средство разработки в соответствии с Методикой 2010 года
Языки низкого уровня (Assembler)	Языки низкого уровня (Assembler)
Процедурные языки высокого уровня (Pascal, C)	Pascal
C#	C#
C++	C++
Microsoft Visual C++	C++
Borland C++	C++
PowerBuilder	VBA
Языки 4GL (Visual Basic, VBA, Delphi), 1C, Uniface	VBA
Visual FoxPro	Clarion
Системы программирования на основе СУБД типа FoxPro (Clipper, Clarion)	Clarion
Встроенные в ПО макроязыки	Встроенные в ПО макроязыки (1C)
Системы программирования на основе СУБД типа Oracle, SqlServer, Informix, Paradox	PL/SQL
Командные скриптовые языки	Командные скриптовые языки
HTML/CSS (без использования редакторов)	HTML/CSS
HTML/CSS (с использованием редакторов)	HTML/CSS
XML/XSLT	XML/XSLT
Java	Java
JavaScript	JavaScript
Скриптовые языки для WEB (ASP, PHP, Perl)	ASP
SQL	SQL

3. РАСЧЕТ ТРУДОЕМКОСТИ РАЗРАБОТКИ ПРОГРАММНЫХ СРЕДСТВ С ИСПОЛЬЗОВАНИЕМ *КАТАЛОГА ФУНКЦИЙ*

3.1. Оценка объема ПС

3.1.1. Оценка объема разрабатываемого ПС выполняется с использованием Каталога функций программных средств (далее Каталог), приведенного в таблице 1.1 Приложения 1 Методики.

3.1.2. Состав функций разрабатываемого ПС формируется на основе Технического задания (ТЗ) на разработку ПС. Программному средству устанавливается в соответствие набор функций из Каталога. Данный набор функций должен соответствовать следующим основным требованиям:

- охватывать полностью все возможности ПС, описанные в ТЗ и пояснительной записке;

– не иметь пересечений: разные функции не должны выполнять одни и те же действия для решения одной задачи;

– состоять из функций с наибольшей детализацией.

Правила применения Каталога представлены в пункте 3.2 Методики.

3.1.3. Каталог содержит функции трех типов:

1 тип. Атомарные функции (обозначены в Каталоге символом *). Например, функция 1.5.4 «Формирование запросов к аналитическим БД», объем которой задан исходя из выполнения **одного** запроса. Количество реализаций этой функции будет определяться числом различных выполняемых запросов.

2 тип. Композитные функции и функции, реализующие классы задач (обозначены в Каталоге символом **). Например, функция 1.5.11 «Прикладное администрирование аналитических БД». Функции этого типа составлены из других функций Каталога в рамках того же класса функций или представляют собой реализацию значительной части программного средства. Например, приведенная функция включает в себя контроль целостности и восстановление БД, функции настройки и ведение журнала операций с БД (функции 1.5.5, 1.5.8, 1.5.10 Каталога). Объем таких функций примерно соответствует объему входящих в нее функций с учетом доли участия каждой из них в составе композитной функции (см. таблицу 2 Приложения 1 Рекомендаций).

Функция не может применяться одновременно с соответствующей композитной функцией, в состав которой она входит, для решения одной задачи. Например, Если для работы с базой данных выбрана функция «Прикладное администрирование аналитических БД», то недопустимо для работы с этой же БД использовать функцию «Контроль целостности и восстановление БД».

В этой связи применение композитных функций и функций, реализующих классы задач, допускается в следующих случаях:

- если для ПС затруднительно далее провести детализацию функций (например, недостаточно детально изложено техническое задание);
- для крупных ПС, требующих представления набора функций на более высоком уровне.

Допускается использование не более одной функции данного типа в рамках расчета. Для повышения точности рекомендуется использовать более детальные функции из Каталога.

Детальными по отношению к композитным функциям являются функции первого и третьего типа.

3 тип. Функции, реализующие отдельную задачу. К ним относятся все остальные функции Каталога, не относящиеся к двум первым типам. Для таких функций допускается повторное использование.

3.1.4. Для определения объема разрабатываемого ПС необходимо определить объем каждой отдельной функции, входящей в набор функций данного ПС. Объем функции определяется одним из способов, приведенных в Методике в пункте 3.2.1.

3.1.5. Объемы функций в Каталоге представлены в виде диапазона значений. Нижняя граница объема функции из Каталога соответствует предоставлению минимально возможного набора операций для реализации задачи функции, что вполне достаточно для реализации ПС среднего уровня.

Если значение объема функции на языке разработки (например, в случае расчета по написанному коду ПС) находится ниже или выше рекомендуемого диапазона, то в качестве значения объема функции следует принимать нижнюю или верхнюю границу допустимого диапазона значений соответственно.

Выбор объема функции с верхней границей объема должен обязательно обосновываться разработчиком.

В случае невозможности оценки объема функции следует выбирать величину объема в соответствии с рекомендованным значением в Приложении 1 Рекомендаций.

3.1.6. Если функция входит в состав готового программного модуля или библиотеки (например, разработчик использует готовый исходный код или реализацию функции из заимствованного программного обеспечения), то на трудозатраты она никак не влияет и степень ее повторного использования при этом равна 0 (полностью используется повторно). Например, степень повторного использования для любой функции из заимствованного программного модуля, включаемого в разрабатываемое программное средство, равна 0.

Если в рамках программного средства или компонента программного средства некоторая функция используется повторно (исходный код или реализация функции дописывается разработчиком, или исходный код готовой функции разработчиком частично модифицируется), то величина коэффициента повторного использования будет равна доле изменений объема функции, используемой повторно. Например, в расчете необходимо использовать чтение данных XML формата. При этом используется готовая библиотека,

встроенная в операционную систему. Для полного достижения необходимой функциональности (если, например, после чтения XML необходимо определенным образом обработать данные) необходимо написать еще 500 строк кода. Объем функции «Загрузка файлов», с помощью которой моделируется чтение (загрузка) XML формата, составляет по каталогу 1250 строк кода. Таким образом, степень повторного использования функции составит $k_i = 500 / 1250 = 0.4$.

В случае, когда функция реализуется неоднократно (число реализаций функции задается больше 1), величина коэффициента ее повторного использования (для заданного числа реализаций) определяется как доля изменений объема всех реализаций. Например, для функции «Ведение журналов» с объемом 2750 строк кода задано 4 реализации ($r_i = 4$) по одной на тип журнала. Объем всех реализаций функции без учета прочих факторов составит $2750 * 4 = 11000$ строк кода. При этом 40% кода каждой функции ($2750 * 0.4 = 1100$ строк кода) одинаковы (используется повторно). Объем повторно используемого кода составит $1100 * 4 = 4400$ строк кода. Таким образом, степень повторного использования составит $k_i = (11000 - 4400) / 11000 = 0,6$ (60%).

3.1.7. Объем программного средства в строках исходного кода с учетом его структуры рассчитывается в соответствии с пунктом 3.2.5 Методики.

3.2. Оценка трудоемкости разработки программного средства

3.2.1. Расчет трудоемкости разработки программных средств выполняется Разработчиком ПС (опорные объекты информатизации Банка России или сторонние организации).

3.2.2. Оценка трудоемкости разработки программного средства выполняется в два этапа:

- расчет базовой трудоемкости разработки ПС;
- расчет трудоемкости отдельных подпроцессов разработки;

Расчет базовой трудоемкости разработки выполняется в соответствии с пунктом 3.2.6 Методики.

Расчет трудоемкости отдельных подпроцессов разработки ПС выполняется строго в соответствии с пунктами 3.2.8–3.2.14 Методики. Поскольку модель расчета трудоемкости разработки предполагает разбиение базового значения трудоемкости по этапам, то итоговая величина трудоемкости разработки считается путем суммирования ее по всем этапам (пункт 3.2.14 Методики).

Расчет трудоемкости разработки с учетом влияния сроков работ выполняется в соответствии с пунктом 3.2.15 Методики.

3.2.3. Если при разработке ПС используются несколько средств разработки, то это программное средство разбивается на функции таким образом, чтобы каждая функция была реализована с использованием одного средства разработки. Различные функции одного ПС могут быть реализованы с использованием различных средств разработки.

3.2.4. Если в процессе разработки программного средства появляется необходимость реализовать дополнительно одну или несколько функций (задач), то следует оценивать величину трудоемкости доработок следующим образом:

- рассчитать величину трудоемкости T_2 с учетом реализации дополнительных функций (задач) для всего программного средства;
- найти величину трудоемкости доработок, которая будет составлять $\Delta T = T_2 - T_1$, где T_1 – величина трудоемкости разрабатываемого ПС без дополнительных функций (задач).

3.2.5. В случае, когда программное средство уже было разработано, и возникает необходимость реализовать дополнительно одну или несколько функций (задач), или в случае, когда изменение требований нормативных документов Банка России предполагает существенного изменения функциональных возможностей ПС, трудоемкость доработок (развития) рекомендуется рассчитывать одним из следующих способов:

- 1) Рассчитать величину трудоемкости разработки T_1 для исходного ПС так, как если бы оно разрабатывалось заново, или в качестве T_1 использовать ранее рассчитанную величину трудоемкости, если для данного ПС такое вычисление проводилось. Затем выполнить все шаги п. 4.2.4 Рекомендаций.
- 2) Выполнить следующие шаги:
 - а) оценить трудоемкость необходимых доработок в соответствии с разделом 3.2 Методики, получив величину T_3 . При этом в расчет включаются только те функции (задачи), которые необходимо реализовать дополнительно;
 - б) так как Методика предполагает оценку трудоемкости самостоятельного и полнофункционального ПС, а включаемые в расчет функции (задачи) образуют дорабатываемый функционал ПС, полученный результат необходимо скорректировать по формуле: $\Delta T = K \cdot T_3$. Значение коэффициента K выбирается по следующей таблице:

Отношение объема доработок к объему ПС, r , %	Значение K
$r < 10$	0,2
$10 \leq r < 30$	0,4
$30 \leq r < 60$	0,5
$60 \leq r < 90$	0,6
$R \geq 90$	0,7

3.2.6. Для примерной оценки величины трудоемкости разработки АС необходимо:

- 1) Представить АС в виде компонентов (см. Приложение 10 Рекомендаций);
- 2) Оценить величину трудоемкости каждого компонента АС;
- 3) Рассчитать трудоемкость T^* для АС суммированием трудоемкости разработки ее каждого компонента;
- 4) Уменьшить величину T^* экспертным образом, вычитая из нее трудоемкости многократно учитываемых факторов, с учетом того, что:

- подпроцесс «анализ требований» должен выполняться один раз для всей АС;

- подпроцесс «ввод в действие» выполняется для всей АС, а не для каждого компонента;

- вспомогательные и организационные процессы, величина трудоемкости для которых уже входит в трудоемкость отдельных подпроцессов, должны быть учтены один раз для всей АС;

- разработка документации выполняется для всей АС, а не для каждого компонента.

3.3. Правила применения коэффициентов

3.3.1. Настоящий раздел регламентирует правила выбора уровней поправочных коэффициентов для модели расчета трудоемкости разработки ПС согласно порядку расчета, представленному в Методике.

3.3.2. Значение коэффициента сложности функции ($K_{сложн}$) при расчете объема ПС (пункт 3.2.4 Методики) выбирается согласно уровню сложности функции соответствующего типа операций.

Каждая функция из Каталога соответствует одному из пяти типов операций:

- управляющие операции;

- вычислительные операции;
- операции, зависящие от аппаратуры;
- операции управления данными;
- операции управления пользовательского интерфейса.

Для определения уровня сложности функции необходимо выполнить следующие шаги:

- 1 шаг.** Определить, к какому типу операции относится функция. Тип операции определяется по Каталогу функций (таблица 1.1 Приложения 1 Методики).
- 2 шаг.** Определить уровень сложности функции по одной из таблиц 2.1–2.5 Приложения 2 Методики в зависимости от типа операции.
- 3 шаг.** Найденному номеру уровня L_v сопоставить значение из таблицы 2.6 Приложения 2 Методики.

3.3.3. Коэффициент, учитывающий конкретные условия и средства разработки ПС ($K_{ср.разр}$) при расчете объема ПС (пункт 3.2.4 Методики) выбирается в соответствии с используемым для реализации функции средством разработки.

В случае если необходимое средство разработки в приведенной таблице отсутствует, то, согласно Методике, необходимо самостоятельно отнести его к какому либо средству, наиболее близкому по уровню используемого языка. Аналогично следует поступать при пересчете объема кода согласно пунктам 3.2.1 и 3.2.2 Методики для различных средств разработки с применением переводных коэффициентов (таблица 1.2 Приложения 1 Методики).

3.3.4. Значения коэффициентов, учитывающих уровень требований к показателям качества ПС (требования к надежности ($K_{над}$), производительности ПС ($K_{произв}$) и уровню информативности документации ($K_{докум}$) на этапах жизненного цикла) при расчете величины базовой трудоемкости разработки ПС (пункт 3.2.6 Методики) выбирается **с учетом требований заказчика** к ожидаемому качеству программного средства.

Допустим, если выполняется разработка АРМ, то уровень требования к надежности можно установить средним, поскольку сбои в работе подобного ПС не приведут к серьезным последствиям. Требования к надежности выше среднего устанавливаются для ПС, результатом неправильной работы которых может являться возможная потеря важной информации и т.п.

3.3.5. Уровень коэффициента, учитывающего степень новизны ПС (K_n) при расчете базовой трудоемкости разработки ПС (пункт 3.2.6 Методики), выбирается в соответствии с таблицей 3.1 Приложения 3 Методики.

Программное средство является принципиально новым в случае, если не имеется доступных аналогов данного ПС в ФАП Банка России.

Программное средство является развитием параметрического ряда ПС в следующих случаях:

- если выполняется разработка новой версии уже существующего программного средства;
- если осуществляется доработка комплекса задач, задачи, функций и создаются новые компоненты, программные модули ПС;

Учет новизны типа ТС и ОС осуществляется следующим образом: если тип ТС/ОС, на котором предполагается функционирование ПС, не использовался в подразделениях Банка России, то он считается новым.

Тип ТС и ОС устанавливается в ТЗ. Выбор значения коэффициента должен соответствовать требованиям ТЗ.

3.3.6. Значение коэффициента, учитывающего использование технологий разработки ($K_{тех}$) при расчете величины базовой трудоемкости разработки ПС (пункты 3.2.6–3.2.7 Методики) устанавливается в соответствии со структурой ПС и используемыми технологиями взаимодействия компонентов ПС.

Структура ПС и набор технологий взаимодействия компонентов устанавливаются в соответствии с техническим заданием на разработку программного средства. Технологии взаимодействия компонентов могут выбираться для любой структуры программного средства, кроме структуры «Монолитное ПС». Одна или несколько технологий взаимодействия компонентов выбираются, если они реализуются в программном средстве.

Уровни значений коэффициента, учитывающего используемые средства проектирования ($K_{ср.пр}$) при расчете трудоемкости подпроцесса «Проектирование» (пункт 3.2.10 Методики), соответствуют классам применяемых средств проектирования. Разработчик должен отнести используемые им средства проектирования к одной из предлагаемых групп, наиболее близких по своим возможностям для применяемых средств.

Аналогично устанавливается значение коэффициента, учитывающего используемые среды разработки (K_{cp}) при расчете трудоемкости подпроцесса «Программирование» (пункт 3.2.11 Методики).

3.3.7. Коэффициент, учитывающий используемые средства тестирования ($K_{cp.mc}$) при расчете трудоемкости подпроцесса «Тестирование» (пункт 3.2.12 Методики), отражает влияние на значение трудоемкости наличия средств автоматизированного тестирования. Значение коэффициента устанавливается в соответствии с описаниями его уровней.

3.3.8. Поправочный коэффициент, учитывающий размер тестовой БД ($K_{cp.mc}$) при расчете трудоемкости подпроцесса «Тестирование» (пункт 3.2.12 Методики), отражает дополнительные трудозатраты на сбор данных для завершения процесса тестирования. Значение этого коэффициента определяется отношением размера (объема) БД в байтах к количеству строк кода отдельных ПС или комплекса ПС.

Объем данных в БД рассчитывается исходя из размера одной записи и количества записей. Размер одной записи рассчитывается как сумма размеров полей, входящих в запись. Например, тестовая БД состоит из одной таблицы; запись таблицы тестовой БД состоит из двух полей. Первое поле типа INT (размер поля 4 байта), второе – типа VARCHAR (размер поля 256 байтов). В таком случае размер одной записи составит 260 байтов. Допустим, таблица тестовой БД содержит 100 записей, тогда размер тестовой БД составит $(260 \cdot 100) = 26000$.

3.3.9. Значение коэффициента, учитывающего опыт работы программистов со средством разработки ($K_{опыт}$) при расчете объема ПС (пункт 3.2.4 Методики), выбирается с учетом числа разработанных ПС (компонентов АС) коллективом разработчиков с использованием данного средства разработки по аналогичной тематике. Оценка проводится для всего коллектива разработчиков, участвующего в разработке ПС.

Аналогично устанавливается значение коэффициента, учитывающего опыт разработки программных средств подобного типа ($K_{оп}$), при расчете базовой трудоемкости разработки ПС (пункт 3.2.6 Методики).

Значение коэффициента, учитывающего опыт работы аналитиков в данной предметной области ($K_{опыт.ан}$), при расчете трудоемкости подпроцесса «Анализ требований к ПС» (пункт 3.2.9 Методики) и значение коэффициента, учитывающего опыт проектировщиков в данной предметной области ($K_{опыт.пр}$), при расчете трудоемкости

подпроцесса «Проектирование» (пункт 3.2.10 Методики) выбирается с учетом числа разработанных ПС (компонентов АС), относящихся к данной предметной области.

3.3.10. Оценку уровня квалификации персонала, задействованного в выполнении отдельных подпроцессов, следует производить для всего коллектива, занятого в этом подпроцессе. Значение коэффициента, учитывающего уровень квалификации аналитиков ($K_{квал.ан}$) при расчете трудоемкости подпроцесса «Анализ требований к ПС» (пункт 3.2.9 Методики) устанавливается согласно квалификации персонала.

Аналогично устанавливается значение коэффициента, учитывающего уровень квалификации проектировщиков в данной предметной области ($K_{квал.пр}$), при расчете трудоемкости подпроцесса «Проектирование» (пункт 3.2.10 Методики), значение коэффициента, учитывающего уровень квалификации программистов ($K_{квал.прог}$), при расчете трудоемкости подпроцесса «Программирование» (пункт 3.2.11 Методики), значение коэффициента, учитывающего уровень квалификации персонала, осуществляющего тестирование ($K_{квал.тест}$), при расчете трудоемкости подпроцесса «Тестирование» (пункт 3.2.12 Методики), значение коэффициента, учитывающего уровень квалификации персонала, осуществляющего ввод в действие ПС ($K_{квал.вн}$), при расчете трудоемкости подпроцесса «Ввод в действие» (пункт 3.2.13 Методики).

Оставляя право формирования коллектива разработчиков, участвующих в разработке ПС для Банка России, за руководством компании-разработчика, рекомендуется относить коллектив разработчиков к имеющим очень высокий уровень опыта в случаях, когда компания сотрудничает с Банком России более 4-х лет и имеет более 5-ти разработанных ПС (компонентов АС) для Банка России или по аналогичной тематике. Кроме того, если компания-разработчик разрабатывает (развивает) отдельные компоненты к ранее разработанному ПС (АС), то уровень квалификации коллектива разработчиков считается как минимум высоким.

3.4. Особенности расчета трудоемкости разработки макета программного средства

3.4.1. Данная методика может применяться для расчета макета программного средства. Под макетом понимается совокупность программных элементов, отражающая основные функциональные возможности разрабатываемого программного средства. Это означает, что принцип расчета этапов, модели и дискретный уровень коэффициентов

остаются теми же самыми, что и для расчета ПС, однако выбор качественных и количественных характеристик для макета отличается следующим образом:

- требования к производительности – коэффициент устанавливается на уровень «Требования к производительности ПС не установлены (однако производительность ПС должна обеспечивать приемлемое время отклика при работе пользователя в интерактивном режиме)»;

- информативность документации – коэффициент устанавливается на низкий уровень – «Не учтены некоторые потребности жизненного цикла»;

3.4.2. Функциональный состав макета состоит из базовых функций, которые позволяют решить поставленные в ТЗ задачи, т.е. макет не обязан реализовывать все функции как полноценное ПС. Большая часть внимания, как правило, уделяется алгоритмическим функциям, нежели интерфейсным и транспортным. Кроме того, для функций, не являющихся основными для данного ПС (реализация интерфейса и т.п.), рекомендуется выбирать минимальное значение объема.

4. РАСЧЕТ ТРУДОЕМКОСТИ РАЗРАБОТКИ ПРОГРАММНЫХ СРЕДСТВ С ИСПОЛЬЗОВАНИЕМ МЕТОДА ФУНКЦИОНАЛЬНОЙ БАЛЛЬНОЙ ОЦЕНКИ

4.1. Оценка объема ПС

4.1.1. Метод ФБО является альтернативным способом оценки объема ПС. Оценка объема ПС с использованием метода ФБО не связана с субъективной оценкой объема каждой отдельной функции, что имеет место при использовании Каталога функций. В отличие от Каталога функций, данный метод оценивает объем исходя как из логической структуры (набор программных модулей, их структура и состав), так и из функционального состава ПС (набор функций и количество используемых ими элементов данных).

4.1.2. Использование метода ФБО не исключает использование Каталога функций: для оценки объема ПС может выбираться любой из методов в зависимости от имеющихся исходных данных и их полноты. Если данных для использования этого метода недостаточно или из имеющихся документов невозможно достоверно установить одновременно логическую структуру и функциональный состав ПС, то используется Каталог функций.

При использовании метода ФБО необходимо, чтобы техническое задание на разработку содержало:

- UML-диаграммы классов, используемых для определения логических файлов;
- UML-диаграммы вариантов использования, определяющих транзакционные функции;
- ER-диаграммы, задающие структуру используемых баз данных.

4.1.3. Определение границы программного средства и границы расчета выполняется в соответствии с правилами пунктов 4.2.2–4.2.3 Методики. Данные процессы строятся на понятии «Логика работы программы». Необходимо считать, что элемент модели (класс, вариант использования) определяет логику работы программы, если его исключение или модификация приводит к нарушению либо изменению алгоритма решения основной задачи (задачи, повлекшей разработку этого программного средства).

Если исключение или модификация элемента модели не влияет на алгоритм решения основной задачи, а влияет на такие свойства как вариант реализации (в виде сервиса, исполняемого модуля, библиотеки и др.), общую структуру, возможности пользовательского интерфейса и т.п., то этот элемент не задает логику работы программного средства, а задает

особенности реализации. Следовательно, в границу расчета такой элемент не входит. Он включается только в границу программного средства.

4.1.4. Необходимо учитывать, что граница программного средства разделяет не физически различные объекты в рамках одного программного средства (например, клиентская программа и используемая ей база данных – 2 объекта), а наборы объектов, относящихся к различным программным средствам. При этом совокупность объектов одного ПС (например, приложения с набором библиотек) полностью входит в границу этого ПС.


4.1.5. В UML-диаграмме классов под элементами данных следует понимать атрибуты классов с учетом требований правил пункта 4.2.6 Методики. Число записей логических файлов определяется в соответствии пунктом 4.2.6 Методики. Для классов, не связанных отношением композиции или обобщения, число записей каждого логического файла всегда равно 1.


4.1.6. В ER диаграмме, описывающей структуру базы данных, под элементами данных следует понимать атрибуты сущности, представляющей таблицу БД во второй или третьей нормальной форме (в реализации базы данных атрибуты – это поля таблицы) с учетом требований правил пункта 4.2.6 Методики. Число записей для любого такого логического файла всегда равно 1.


4.1.7. Элемент данных может быть представлен не только в виде одиночного атрибута, но и совокупностью атрибутов. Например, если дата представлена как набор из трех атрибутов (число, месяц, год), то все три атрибута считаются как один элемент данных. Аналогично и для других подобных групп (время, адрес, ФИО и др.)

4.1.8. Если на диаграмме представлен набор атрибутов для повторного ввода значения одной величины (например, 12 атрибутов для ввода значения ежемесячного дохода по одному на каждый месяц), то этот набор учитывается как один элемент данных. Другими словами, важно наличие самой величины, а не число ее экземпляров.

4.1.9. В UML-диаграмме вариантов использования под транзакционной функцией следует понимать элемент структуры, называемый «Вариант использования». При этом вариант использования считается транзакционной функцией, если он реализует одну задачу, а не комплекс задач (в этой ситуации его необходимо разбивать на более мелкие единицы).

4.1.10. Если вариант использования на UML-диаграмме, определенный как транзакционная функция, объединен отношением обобщения () либо расширения

`<<extend>>`
() с другими вариантами использования, то обобщаемый либо расширяемый вариант использования не считается транзакционной функцией.

4.1.11. Если вариант использования на UML-диаграмме, определенный как транзакционная функция, связан отношением включения (`<<include>>` ) с другими вариантами использования, отражающими на этой диаграмме более детальные действия, то включаемый вариант использования не считается транзакционной функцией.

4.1.12. При выделении транзакционных функций необходимо обязательно учитывать правила, приведенные в пункте 4.2.8 Методики.

4.1.13. Варианты использования, учитываемые как транзакционные функции, имеют аналогичные (или подобные) представленным в техническом задании на разработку названиям выполняемых программой функций (см. пример расчета). Этим же определяется уровень детализации, на котором необходимо рассматривать диаграмму вариантов использования.

Если вариант использования не определен как транзакционная функция, то далее в расчете он не учитывается.

4.2. Оценка трудоемкости разработки программного средства

4.2.1. Согласно Методике, расчет трудоемкости разработки с использованием метода функциональной балльной оценки выполняется аналогично тому, как это делается при использовании Каталога функций. Для этого предварительно необходимо пересчитать объем ПС в функциональных точках в строки исходного кода. Пересчет выполняется в соответствии с пунктом 4.2.15 Методики.

4.2.2. Правила выбора значений коэффициентов аналогичны тем, которые указаны для расчета разработки ПС с использованием Каталога функций.

5. (9) КОММЕНТАРИИ К ОЦЕНКЕ ЗАТРАТ ВРЕМЕНИ И ЧИСЛЕННОСТИ СПЕЦИАЛИСТОВ

5.1. Трудоемкость разработки ПС рассчитывается для всех подпроцессов разработки. Трудоемкость сопровождения и эксплуатации может рассчитываться для всех или некоторых (выполняемых) работ процесса.

5.2. В качестве значения фонда времени (число рабочих дней в месяце) в расчетах и при переводе трудоемкости из чел-дней в чел-месяцы рекомендуется величина 21 (день/мес.). Для получения более точной величины необходимо выполнить следующие шаги :

1 шаг. Подсчитать количество рабочих дней n_1 за период выполнения работ по разработке, сопровождению и эксплуатации ПС.

2 шаг. Подсчитать количество рабочих дней n в соответствующем году.

3 шаг. Определить уточненный срок выполнения работ в месяцах по формуле:

$$t = 12 \cdot \frac{n_1}{n} \text{ [мес.]} \quad (8.1)$$

4 шаг. Определить среднее количество рабочих дней в месяце D для расчета трудоемкости в рассматриваемом периоде (с округлением до целого числа):

6. (10) АНАЛИЗ ПОЛУЧАЕМЫХ РЕЗУЛЬТАТОВ

6.1. Общие сведения

6.1.1. Анализ получаемых результатов позволяет оценить степень приемлемости получаемых результатов при оценке трудоемкости разработки ПС, выявить причины получения неадекватных результатов, а также определить направления совершенствования рекомендаций по применению Методики (дополнительная калибровка модели, пополнение состава используемых факторов, влияющих на трудоемкость и др.).

6.1.2. В основе предлагаемого метода анализа лежит определение численного значения допустимой ошибки вычисления, в пределах которого расчет можно признать достоверным. Это значение определяется исходя из характеристик самой модели и статистической выборки, используемой для построения модели. Поэтому данный метод позволяет оценить достоверность модели расчета относительно фактических данных и определить пути корректировки модели.

6.1.3. Сравнительный анализ значений трудоемкости производится на основе сопоставления планируемых и расчетных величин.

Планируемое значение трудоемкости определяется экспертным путем с учетом имеющегося опыта или накопленной статистики для разработанных ПС (компонентов АС) подобного типа.

Расчетное значение трудоемкости находится для разработки и сопровождения ПС (в соответствии с пунктами 3.2 или 4.2 рекомендаций по применению Методики).

6.1.4. Сопоставление планируемых и расчетных значений производится путем расчета относительной ошибки вычисления трудоемкости. Относительная ошибка находится по формуле:

$$\delta = \frac{|T_{\text{факт}} - T_{\text{расч}}|}{|T_{\text{факт}}|}, \quad (9.1)$$

где $T_{\text{факт}}$ – планируемое значение трудоемкости, определяемое экспертным путем;

$T_{\text{расч}}$ – расчетное значение трудоемкости разработки, рассчитываемое в соответствии с пунктом 3.2 или 4.2 рекомендаций по применению Методики.

Относительная ошибка вычисления трудоемкости рассчитывается для каждого подпроцесса разработки.

6.1.5. Полученная величина относительной ошибки оценивается по шкале допустимой величины. Такая шкала построена для каждого подпроцесса разработки с учетом информации о расчетах, заложенных в модель расчета. Чем ближе относительная погрешность стремится к нулю, тем более точно расчетное значение трудоемкости совпадает с фактическим.

Шкалы для оценки точности проведенного расчета трудоемкости приведены в **Приложении 9** Рекомендаций.

6.1.6. Шкала состоит из трех зон:

- $(0 - \delta_1)$ – допустимая ошибка (зеленая зона); можем считать оценку трудоемкости выполненной с достаточной точностью;
- $(\delta_1 - \delta_2)$ – условно допустимая ошибка (желтая зона); полученная величина трудоемкости объяснима моделью, однако такой расчет все же дает большие отклонения;
- больше δ_2 – недопустимая ошибка (красная зона); означает большую неточность при выполнении расчетов.

При попадании величины ошибки в желтую или красную зоны необходимо выполнить проверку выбранных значений коэффициентов.

6.2. Порядок расчета

6.2.1. Расчет трудоемкости разработки ПС выполняется в соответствии с пунктами 3.2 или 4.2 рекомендаций по применению Методики.

6.2.2. Планируемое значение трудоемкости определяется экспертным путем с учетом имеющегося опыта либо накопленной статистики для разработанных ПС (компонентов АС) подобного типа.

6.2.3. Вычисляется величина относительной ошибки расчета по формуле (9.1) для каждого подпроцесса разработки ПС.

6.2.4. Полученные величины относительных погрешностей оцениваются по шкале (**Приложение 9**), приведенной в виде таблиц. В зависимости от подпроцесса (либо работы) и порядка величины трудоемкости (планируемой) выбирается соответствующая строка таблицы.

Если величины относительных погрешностей для расчетных значений всех подпроцессов разработки или работ сопровождения рассматриваемого программного

средства попадут в зеленую зону шкалы, то полученные расчетные значения трудоемкостей рассматриваемых работ можно считать достоверными.

Для всех значений трудоемкости, попавших в желтую или красную зону шкалы, необходимо:

- 1) Проверить правильность выбора уровней коэффициентов с учетом требований Методики расчета.
- 2) Проверить достоверность параметров модели (величина объема ПС, документации, доработок, количество установленных экземпляров ПС, число пользователей) с учетом требований Методики расчета.

ПРИЛОЖЕНИЕ 1
КАТАЛОГ ФУНКЦИЙ ПРОГРАММНЫХ СРЕДСТВ, СОДЕРЖАЩИЙ
РЕКОМЕНДУЕМЫЕ ОБЪЕМЫ ФУНКЦИЙ

Таблица 1. Каталог функций

№	Наименование типа, класса; наименование (содержание) функции	Комментарий (Обоснование выбора)	Рекомендуемый объем функции ПС (строк исходного текста на условном языке)
1.	Управляющие операции		
1.1.	<i>Информационная безопасность</i>		
1.1.1.	Реализация криптографических алгоритмов (<i>один алгоритм</i>)	Функции, обеспечивающие обратимый способ преобразования информации с целью ее защиты.	550*
1.1.2.	Обеспечение безопасности передачи сообщений и обмена данными (шифрование)	Функции, обеспечивающие обратимый способ преобразования информации с целью ее защиты в канале передачи данных.	850
1.1.3.	Организация криптозащиты с помощью ПТК Криптоцентр авизо		5500
1.1.4.	Контроль целостности ПО	Функции, обеспечивающие контроль и защиту программных файлов от модификации в процессе эксплуатации ПС.	3000
1.1.5.	Реализация базовых функций информационной безопасности в соответствии с нормативными документами Банка России		10500**
1.1.6.	Поддержка дискреционного контроля доступа	Реализация и обеспечение дискреционного контроля доступа - разграничение доступа между поименованными субъектами и поименованными объектами доступа.	3500

1.1.7.	Поддержка мандатного контроля доступа	Реализация и обеспечение мандатного контроля доступа - разграничение доступа субъектов к объектам, основанное на характеризуемой меткой конфиденциальности информации, содержащейся в объектах, и официальном разрешении (допуске) субъектов обращаться к информации такого уровня конфиденциальности.	3500
1.1.8.	Создание, редактирование, удаление пользователей (группы пользователей)		3250
1.1.9.	Установка прав доступа к объекту	Функции, предоставляющие разграничение прав доступа к объекту.	1500
1.1.10.	Ведение протокола доступа		3000
1.1.11.	Настройка политики безопасности	Настройка параметров процедур и интерфейсов задач, связанных с обеспечением информационной безопасности.	1500
1.1.12.	Ведение журналов отказа		1250
1.1.13.	Взаимодействие с ПО Верба-OW, САЭД	Взаимодействие с ПО Верба – OW, САЭД Интеграция СКЗИ Верба – OW или САЭД в прикладную программу в соответствии с требованиями, указанными в документации на данные СКЗИ.	350
1.1.14.	Взаимодействие с ПО СКАД «Сигнатура»		600
1.2.	<i>Реализация интерфейсов между программными средствами</i>		
1.2.1.	Взаимодействие с другим ПО	Процессы, которые посылают данные или управляющую информацию за пределы приложения.	750
1.2.2.	Взаимодействие с транспортной (почтовой) системой	Функции, обеспечивающие прием и передачу информации через почтовую систему.	2500
1.2.3.	Реализация ПО обмена информацией между ЦОИ и клиентами (WEB-сервер)		9500**
1.2.4.	Организация доступа к динамическим информационным ресурсам через Intranet Банка России (в т.ч. через КПИ Банка России)	Организация работы сотрудников ЦА и ТУ Банка России через Корпоративный портал Интранет Банка России.	12500**

1.2.5.	Организация регулярно актуализируемого статистического контента для доступа через Intranet Банка России (в т.ч. через КПИ Банка России)	Реализация процедур организации доступа пользователей к сформированному регулярно актуализируемому статистическому контенту через Intranet Банка России.	12500**
1.3.	<i>Выполнение регламентных операций</i>		
1.3.1.	Пусковое решение		750
1.3.2.	Регламентные операции: открытие, закрытие операционного дня в регионе, УБР, ответисполнителя (день, месяц, год)	Обработка информации; запрет ввода новых данных в БД	4500
1.3.3.	Операционный день		12500
1.3.4.	Реализация процедур обновления версий ПО		1500
1.3.5.	Управление режимами взаимодействия клиентов с КЦОИ		
1.3.6.	Управление режимами работы счета клиента в КЦОИ		
1.3.7.	Проведение автоматизированных процедур обработки учетно-операционной и расчетной информации ТУ в соответствии с регламентом функционирования ТУ		
1.3.8.	Управление автоматическими процедурами обработки, проводимыми по установленному регламенту, либо определяемых параметрами функционирования ТУ		
1.3.9.	Проведение сеанса урегулирования		
1.3.10.	Проведение регулярного сеанса		
1.3.11.	Проведение сеанса завершения обработки		
1.4.	<i>Реализация взаимосвязи программных средств и компонентов</i>		
1.4.1.	Сетевая передача команд и сообщений	Передача команд и сообщений между модулями ПС по сети.	550
1.4.2.	Реализация связи между распределенными приложениями с использованием стандартных транспортных средств	Реализация функций, позволяющие использовать для передачи информации стандартные транспортные средства, такие как MSMQ	550
1.4.3.	Реализация связи между распределенными приложениями на основе сетевых интерфейсов низкого уровня	Реализация функций, позволяющие использовать для передачи информации сетевые интерфейсы низкого уровня, такие как sockets	350
1.4.4.	Удаленная доставка информации с подтверждением получения		350

1.4.5.	Вызов удаленных процедур (единичный вызов)	Использование сервиса RPC	750
1.4.6.	Управление файлами, доступом к файлам и передачей файлов между удаленными и разнородными файловыми системами	Реализация функций для разграничения доступа к файлам и их перемещению в рамках сети.	350
1.4.7.	Обработка сообщений	Обработка отдельных сообщений, передаваемых данному компоненту ПС от другого компонента.	105
1.4.8.	Обработка распределенных транзакций	Обработка сообщений компонентом ПС в рамках транзакции.	155
1.4.9.	Реализация средства контроля состояния распределенной сети однородных компонент	Реализация функций, позволяющих описывать и управлять состоянием компонентов распределенного в сети ПС.	150
1.4.10.	Доступ к общей памяти : - в рамках одной машины - в рамках вычислительной сети	Распределение общей памяти (оперативной, виртуальной).	30 105
1.4.11.	Ведение журнала обращений к распределенному ПС		255
1.5.	<i>Создание и поддержка аналитических БД***</i>		
1.5.1.	Формирование физической структуры аналитических БД (на одну БД)	Проектирование структуры таблиц, формирование логической и концептуальной моделей данных, формирование и заполнение справочников.	4500
1.5.2.	Формирование многомерной структуры аналитических БД (на одну БД)	Формирование многомерной модели данных, формирование и заполнение репозитория системы.	4500
1.5.3.	Описание связей между аналитическими базами данных	Описание связи аналитических БД в части исходных данных и аналитических показателей.	1750
1.5.4.	Формирование запросов к аналитическим БД (один запрос)	Формирование многомерных и реляционных запросов к БД.	45*
1.5.5.	Ведение журнала операций с аналитическими БД	Формирование журнала регистрации операций с многомерными структурами БД.	1500
1.5.6.	Формирование ETL-процедур	Формирование процедур преобразования и загрузки информации, приведения информации к общей структуре БД и агрегация данных по всем видам мониторинга предприятий.	8500
1.5.7.	Протоколирование ETL-процедур		3750

1.5.8.	Функции настройки аналитических БД	Настройка окружения работы аналитической БД (настройка соединений, пересоздание представлений, синонимов, процедур, планировщиков задач).	1750
1.5.9.	Модификация аналитических БД (<i>один запрос</i>)	Модификация БД со справочной информацией в связи с добавлением новой информации.	225*
1.5.10.	Контроль целостности и восстановление аналитических БД	Функции поддержки баз данных в актуальном состоянии, резервного копирования, восстановления данных в случае сбоев.	750
1.5.11.	Прикладное администрирование аналитических БД	Организация среды выполнения процедур прикладного администрирования с службой эксплуатации для наполнения и ведения аналитической БД.	3000**
1.5.12.	Поиск, представление и вывод информации	Организация поиска информации в базах данных, вывода информации на экранных формах	3500
1.5.13.	Обработка записей базы данных		1750
2.	Вычислительные операции		
2.1.	Ввод и обработка данных		
2.1.1.	Ввод данных первичных документов в интерактивном режиме	Регистрация документов, расчет значений реквизитов, ввод дополнительной информации, обработка входной информации.	5500
2.1.2.	Ввод данных по нарушениям в интерактивном режиме		3000
2.1.3.	Ввод и первичный контроль документов из файлов	Ввод, контроль полноты и корректности введенных данных.	3750
2.1.4.	Логический, синтаксический и номенклатурный контроль данных		3750
2.1.5.	Расчет алгебраических выражений	Выполнение арифметических вычислений	550
2.1.6.	Статистическая обработка данных	Реализация методов статистической обработки информации	3500
2.1.7.	Обработка документов на исключение из картотеки в ЦОИ		2250
2.1.8.	Обработка документов ответисполнителей	Преобразование данных из таблиц БД в файлы; формирование промежуточных данных для документирования; изменение признаков состояния	1. 5000
2.1.9.	Обработка документов ответного авизования		6500

2.1.10.	Обработка документов по принудительной квитантке		1750
2.1.11.	Обработка и учет заявок сопровождаемых регионов		3000
2.1.12.	Обработка протоколов		2250
2.1.13.	Обработка документов начального авизования		8500
2.1.14.	Обработка документов МЭР в ЦОИ		7500
2.1.15.	Ввод прочих данных	Чтение исходной информации из различных источников (либо ее интерактивный ввод) с целью получения данных, пригодных для дальнейшей обработки.	2500
2.1.16.	Обработка входной информации (прочих данных)	Первоначальная обработка данных с целью преобразования ее к виду, пригодному для дальнейшего использования (с статистическая обработка, выполнение вычислений по данным и т.п.)	3750
2.1.17.	Оплата платежей по МЭР/по расчетам с ПУ ТУ		
2.1.18.	Обработка ошибочных платежей		
2.1.19.	Реализация в ПС УБР функций режима резервного ввода		
2.1.20.	Ввод реквизитов расчетных документов на бумажных носителях		
2.1.21.	Ввод документов для учета операций по внебалансовым счетам		
2.1.22.	Ввод документов для учета операций по внутрибанковским и кассовым операциям		
2.2.	<i>Составление и анализ балансов</i>		
2.2.1.	Составление балансов		750**

2.2.2.	Анализ балансов	Включает в себя проведение проверок: -корсчета банка (сравнение валютного корсчета с нормативом; проверка наличия средств на рублевом корсчете); -кассы (проверка наличия валюты и рублей); - вкладов; - кредитов; - депозитов юридических лиц; - остатков на расчетных счетах клиентов; - собственных акций банков; - уставного капитала; -резервов в ЦБ; - накопительных счетов при выпуске акций; - просроченные кредиты; просроченные проценты по кредитам; - межбанковские кредиты; - прочие активы (средства, вложенные в уставные капиталы предприятия; основные средства; МБП) и т.п.	2250**
2.2.3.	Выполнение баланса и оборотов КО за месяц		3000**
2.2.4.	Выполнение ежедневного баланса		1750**
2.2.5.	Корректировка балансов		350**
2.2.6.	Пересчет балансов		2500**
2.3.	<i>Проведение экономического анализа</i>		
2.3.1.	Анализ безналичных расчетов в платежных системах		3000**
2.3.2.	Анализ денежной массы в обращении по районам	Проведение количественного (объем) и качественного (покупорный состав) анализов денежной массы.	5500**
2.3.3.	Анализ курсов валют, устанавливаемых ЦБ	Контроль разницы курсов валют и проведение анализа прибыли.	4000**
2.3.4.	Анализ курсов основной валюты в обменных пунктах КО		7500**
2.3.5.	Анализ событий нефинансового характера		700**
2.4.	<i>Выполнение задач планирования</i>		
2.4.1.	Планирование инспекторских проверок		1750**
2.4.2.	Планирование эмиссии	Организация выдачи собственных акций и векселей.	4500**
2.4.3.	Планирование прочих задач		2400**
2.5.	<i>Мониторинг</i>		

2.5.1.	Мониторинг отчетности КО	Организация наблюдения за показателями, характеризующими деятельность КО, позволяющее оперативно выявлять существенные изменения, прослеживать интенсивность процессов, происходящих в КО.	6500**
2.5.2.	Мониторинг отчетности ТУ	Организация наблюдения за показателями, характеризующими деятельность ТУ, позволяющее оперативно выявлять существенные изменения, прослеживать интенсивность процессов, происходящих в ТУ.	6500**
2.6.	<i>Формирование и обработка аналитических показателей</i>		
2.6.1.	Расчет экономических показателей	Расчет синтетических (производных, интегральных, агрегированных) экономических показателей на основе первичных показателей.	10000**
2.6.2.	Экономический и финансовый анализ	Проведение различных видов исследований финансовых характеристик объекта. Выявление тенденций и основных факторов, влияющих на развитие различных экономических объектов или протекание экономических процессов.	15000**
2.6.3.	Моделирование и прогнозирование	Формирование различных видов экономико-математических моделей объектов для объяснения поведения объекта, проведения прогнозных вариантных расчетов либо поиска оптимального управления объектом.	25000**
2.6.4.	Ведение показателей системы, определяющих ее штатное состояние функционирования		
3.	Операции, зависящие от аппаратуры		
3.1.	<i>Прием документов от различных источников</i>		

3.1.1.	Прием файлов, первичный контроль	Контроль полноты и корректности информации, содержащейся в исходных файлах (функция может использоваться для различных форматов файлов).	900
3.1.2.	Прием информации из ЦОИ		5000
3.1.3.	Прием данных, содержащихся в DBF файлах	Вариант функции 3.1.1 для файлов формата DBF.	700
3.1.4.	Прием описи к конвертам с документами		1750
3.1.5.	Прием текстовых файлов	Вариант функции 3.1.1 для текстовых файлов.	550
3.2.	<i>Регистрация входных документов</i>		
3.2.1.	Регистрация входных документов		750
3.2.2.	Регистрация исходящих авизо		2750
3.2.3.	Регистрация телеграфных и почтовых авизо		1250
3.3.	<i>Работа с файлами и форматами</i>		
3.3.1.	Загрузка файлов	Чтение файла во внутренние структуры данных	1250
3.3.2.	Преобразование формата данных		2750
3.3.3.	Формирование файлов	Запись внутренних структур данных в файл	1750
3.4.	<i>Ведение архива и копирование информации</i>		
3.4.1.	Создание архива		1500
3.4.2.	Добавление данных в архив		750
3.4.3.	Извлечение данных из архива		750
3.4.4.	Поиск данных в архиве		1000
3.4.5.	Проверка целостности архива		1000
3.4.6.	Задачи ведения электронного архива	Состоит из всех задач ведения и копирования информации (3.4.1-3.4.5)	5000**
4.	<i>Операции управления данными</i>		
4.1.	<i>Ведение журналов</i>		
4.1.1.	Ведение журналов (на один журнал)		2750*
4.1.2.	Ведение журнала регистрации запросов и подтверждений по начальным (ответным) авизо		3500
4.1.3.	Вывод журналов приема-передачи		650
4.1.4.	Контроль и журнализация доступа к защищенным ресурсам		1750
4.2.	<i>Работа со справочниками</i>		
4.2.1.	Ведение справочников (на один справочник)		1750*

4.2.2.	Ведение книги регистрации открытых лицевых счетов		10400
4.3.	<i>Контроль информации документов</i>		
4.3.1.	Контроль входной информации документов	Контроль правильности документов при вводе	1250
4.3.2.	Контроль информации при вводе данных документа (<i>на форму ввода</i>)	Контроль правильности вводимой пользователем информации	350*
4.3.3.	Контроль МЭР/расчетов с ПУ ТУ		
4.4.	<i>Настройка ПС</i>		
4.4.1.	Разработка и ввод метаданных	Организация работ с репозиторием системы, обеспечивающего ведение метаданных, т.е. всех необходимых описаний входной и выходной информации, алгоритмов преобразования; а также другой декларативной и процедурной информации об объектах системы.	4500
4.5.	<i>Ведение и обработка протоколов</i>		
4.5.1.	Протоколирование работы	Протоколирование работы включает в себя: -протоколирование выполняемых действий; -протоколирование возникающих ошибок.	3000
4.5.2.	Ведение протокола выполнения расчетов		600
4.5.3.	Формирование протокола проводок		1100
4.5.4.	Обработка протоколов		1500
4.6.	<i>Формирование отчетов</i>		
4.6.1.	Формирование отчетов (<i>на отчет</i>)	Представление результатов в табличном и графическом виде через систему аналитических отчетов.	2050*
4.6.2.	Просмотр и редактирование отчетных форм	Организация просмотра данных отчетных форм; возможность редактирования данных.	750
4.6.3.	Подготовка форм отчетности (<i>на одну форму</i>)		2000*
4.7.	<i>Распределенная обработка данных</i>		
4.7.1.	Создание одного объекта на базе технологии CORBA		50*
4.7.2.	Создание одного объекта на базе технологии COM		60*
4.7.3.	Вызов метода CORBA-объекта (<i>один вызов</i>)		2*
4.7.4.	Вызов метода COM-объекта (<i>один вызов</i>)		1*

4.7.5.	Реализация сетевого взаимодействия на базе средств Sockets API (серверная сторона, один сервер)		40*
4.7.6.	Реализация сетевого взаимодействия на базе средств Sockets API (клиентская сторона, один клиент)		20*
4.8.	<i>Ведение очереди отложенных платежей</i>		
4.8.1.	Централизованное ведение очереди отложенных платежей		
4.8.2.	Управление очередью отложенных платежей		
4.9.	<i>Ведение картотек расчетных документов</i>		
4.9.1.	Ведение картотеки расчетных документов, ожидающих акцепта для оплаты		
4.9.2.	Ведение картотеки расчетных документов, не оплаченных в срок		
4.9.3.	Прием ЭСИС – запросов УБР для получения информации об актуальном состоянии картотек		
4.9.4.	Ведение картотек расчетных документов	Состоит из всех функций ведения картотек расчетных документов (4.9.1-4.9.3)	**
4.10.	<i>Ведение синтетического и аналитического учета</i>		
4.10.1.	Отражение операций по лицевым счетам аналитического учета		28000
4.10.2.	Ведение синтетического учета совершаемых операций		16875
5.	<i>Операции управления пользовательского интерфейса</i>		
5.1.	<i>Настройка ПС на условия применения</i>		
5.1.1.	Настройка параметров	Настройка параметров процедур и интерфейсов комплекса задач.	750
5.1.2.	Настройка аналитических таблиц	Настройка аналитических и служебных таблиц.	1500
5.2.	<i>Реализация пользовательского интерфейса¹</i>		
5.2.1.	Реализация интерфейса пользователя	Организация диалога с пользователем для обеспечения работ по выполнению функций комплексов задач. Реализует одну из функций 5.2.2-5.2.8	2510

5.2.2.	Реализация стандартного графического пользовательского интерфейса (однооконное приложение)	Функция 5.2.1 для однооконного приложения	1150
5.2.3.	Реализация стандартного графического пользовательского интерфейса (диалоговое приложение)	Функция 5.2.1 для диалогового приложения	510
5.2.4.	Реализация стандартного графического пользовательского интерфейса (многооконное приложение)	Функция 5.2.1 для многооконного приложения	3750
5.2.5.	Реализация стандартного графического пользовательского интерфейса (однооконное приложение) с использованием: – API – MFC/OWL – VCL	Функция 5.2.2, если известны используемые средства (библиотеки, интерфейсы)	1500
5.2.6.			750
5.2.7.			500
5.2.8.	Реализация стандартного графического пользовательского интерфейса (диалоговое приложение) с использованием: – API – MFC/OWL – VCL	Функция 5.2.3, если известны используемые средства (библиотеки, интерфейсы)	750
5.2.9.			70
5.2.10.			60
5.2.11.	Реализация стандартного графического пользовательского интерфейса (многооконное приложение) с использованием: – API – MFC/OWL – VCL	Функция 5.2.4, если известны используемые средства (библиотеки, интерфейсы)	3500
5.2.12.			1050
5.2.13.			1000
5.2.14.	Реализация стандартного графического пользовательского интерфейса (однооконное приложение) с использованием WEB-технологий	Функция 5.2.1 при использовании WEB-технологий	6150
5.3.	Реализация машинной графики		
5.3.1.	Реализация машинной графики для отображения состояния ПС в статике	Графическое представление статических моделей	1000
5.3.2.	Реализация машинной графики для отображения состояния ПС в динамике	Графическое представление динамических моделей	1750

* Функции выполняют единичную операцию и могут иметь несколько реализаций как на уровне ПС в целом, так и на уровне компонентов ПС. Каждая реализация должна характеризоваться коэффициентом степени повторного использования.

** Функции являются композитными (либо реализующими классы задач) и включают в себя другие функции из Каталога. Их использование допускается только для крупных ПС, а также для прочих ПС, в которых трудно детализировать набор функций. Такие функции не могут реализовываться более одного раза в расчете. Список функций, которые могут включаться в композитную функцию, приведен в таблице 2 данного приложения.

Все остальные функции являются структурными единицами ПС. Могут иметь несколько реализаций на уровне компонентов ПС и одну реализацию на уровне ПС. Каждая реализация должна характеризоваться коэффициентом степени повторного использования.

*** Допускается применение функций данной группы к базам данных, не являющихся аналитическими

Каталог содержит рекомендуемые значения объема функций программных средств Банка России, используемые в случае, когда не представляется возможным определить истинное значение объема функции.

¹ Под одним пользовательским интерфейсом следует понимать всю совокупность окон, элементов управления и прочих элементов, реализующих интерфейс одного ПС. Таким образом, одно ПС может реализовывать только одну функцию из группы не более одного раза.

Таблица 2. Состав композитных функций.

Композитная функция	Функциональный состав
<p>1.1.5 Реализация базовых функций информационной безопасности в соответствии с нормативными документами Банка России [6000-15000]</p>	<ul style="list-style-type: none"> – 1.1.2 Обеспечение безопасности передачи сообщений и обмена данными [200-500] – 1.1.3 Организация криптозащиты с помощью ПТК Криптоцентр авизо[5000-6000] – 1.1.6 Поддержка дискреционного контроля доступа [2000-5000] – 1.1.7 Поддержка мандатного контроля доступа [2000-5000] – 1.1.8 Создание, редактирование, удаление пользователей (группы пользователей) [1500-5000] – 1.1.11 Настройка политики безопасности[1000-2000] – 1.1.12 Ведение журналов отказа [1000-1500]
<p>1.2.3 Реализация ПО обмена информацией между ЦОИ и клиентами (WEB-сервер) [9000-10000]</p>	<ul style="list-style-type: none"> – 1.1.1 Реализация криптографических алгоритмов [100-1000] – 1.1.2 Обеспечение безопасности передачи сообщений и обмена данными [200-500] – 1.1.12 Ведение журналов отказа [1000-1500] – 1.2.1 Взаимодействие с другим ПО [500-1000] – 1.2.2 Взаимодействие с транспортной (почтовой) системой [2000-3000] – 1.4.1 Сетевая передача команд и сообщений [90] – 1.4.4 Удаленная доставка информации с подтверждением получения [200-500] – 1.4.6 Управление файлами, доступом к файлам и передачей файлов между удаленными и разнородными файловыми системами [200-500] – 1.4.9 Реализация средства контроля состояния распределенной сети однородных компонент [100-200]

<p>1.2.4 Организация доступа к динамическим информационным ресурсам через Intranet Банка России (в т.ч. через КПИ Банка России) [10000-15000]</p>	<ul style="list-style-type: none"> – 1.1.1 Реализация криптографических алгоритмов [100-1000] – 1.1.2 Обеспечение безопасности передачи сообщений и обмена данными [200-500] – 1.1.12 Ведение журналов отказа [1000-1500] – 1.2.1 Взаимодействие с другим ПО [500-1000] – 1.2.2 Взаимодействие с транспортной (почтовой) системой [2000-3000] – 1.4.1 Сетевая передача команд и сообщений [90] – 1.4.4 Удаленная доставка информации с подтверждением получения [200-500] – 1.4.6 Управление файлами, доступом к файлам и передачей файлов между удаленными и разнородными файловыми системами [200-500] – 1.4.9 Реализация средства контроля состояния распределенной сети однородных компонент [100-200]
<p>1.2.5 Организация регулярно актуализируемого статистического контента для доступа через Intranet Банка России (в т.ч. через КПИ Банка России) [10000-15000]</p>	<ul style="list-style-type: none"> – 1.1.1 Реализация криптографических алгоритмов [100-1000] – 1.1.2 Обеспечение безопасности передачи сообщений и обмена данными [200-500] – 1.1.12 Ведение журналов отказа [1000-1500] – 1.2.1 Взаимодействие с другим ПО [500-1000] – 1.2.2 Взаимодействие с транспортной (почтовой) системой [2000-3000] – 1.4.1 Сетевая передача команд и сообщений [90] – 1.4.4 Удаленная доставка информации с подтверждением получения [200-500] – 1.4.6 Управление файлами, доступом к файлам и передачей файлов между удаленными и разнородными файловыми системами [200-500] – 1.4.9 Реализация средства контроля состояния распределенной сети однородных компонент [100-200]

1.5.11 Прикладное администрирование аналитических БД [1000-5000]	<ul style="list-style-type: none"> – 1.5.5 Ведение журнала операций с аналитическими БД [1000-2000] – 1.5.8 Функции настройки аналитических БД [1500-2000] – 1.5.10 Контроль целостности и восстановление аналитических БД [500-1000]
2.2.1 Составление балансов [500-1000]	<ul style="list-style-type: none"> – 2.1.5 Расчет алгебраических выражений [100-1000] – 4.6.1 Формирование отчетов [100-4000]
2.2.2 Анализ балансов [1000-3500]	<ul style="list-style-type: none"> – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]
2.2.3 Выполнение баланса и оборотов КО за месяц [2500-3500]	<ul style="list-style-type: none"> – 2.1.5 Расчет алгебраических выражений [100-1000] – 4.6.1 Формирование отчетов [100-4000]
2.2.4 Выполнение ежедневного баланса [1500-2000]	<ul style="list-style-type: none"> – 2.1.5 Расчет алгебраических выражений [100-1000] – 4.6.1 Формирование отчетов [100-4000]
2.2.5 Корректировка балансов [200-500]	<ul style="list-style-type: none"> – 2.1.5 Расчет алгебраических выражений [100-1000] – 4.6.2 Просмотр и редактирование отчетных форм [500-1000]
2.2.6 Пересчет балансов [2000-3000]	<ul style="list-style-type: none"> – 2.1.5 Расчет алгебраических выражений [100-1000] – 4.6.2 Просмотр и редактирование отчетных форм [500-1000]
2.3.1 Анализ безналичных расчетов в платежных системах [2500-3500]	<ul style="list-style-type: none"> – 1.5.4 Формирование запросов к аналитическим БД [10-80] – 1.5.12 Поиск, предоставление и вывод информации [1000-6000] – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]
2.3.2 Анализ денежной массы в обращении по районам [5000-6000]	<ul style="list-style-type: none"> – 1.5.4 Формирование запросов к аналитическим БД [10-80] – 1.5.12 Поиск, предоставление и вывод информации [1000-6000] – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]

2.3.3 Анализ курсов валют, устанавливаемых ЦБ [3500-4500]	<ul style="list-style-type: none"> – 1.5.4 Формирование запросов к аналитическим БД [10-80] – 1.5.12 Поиск, предоставление и вывод информации [1000-6000] – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]
2.3.4 Анализ курсов основной валюты в обменных пунктах КО [7000-8000]	<ul style="list-style-type: none"> – 1.5.4 Формирование запросов к аналитическим БД [10-80] – 1.5.12 Поиск, предоставление и вывод информации [1000-6000] – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]
2.3.5 Анализ событий нефинансового характера [300-1100]	<ul style="list-style-type: none"> – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]
2.4.1 Планирование инспекторских проверок [1500-2000]	<ul style="list-style-type: none"> – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]
2.4.2 Планирование эмиссии [4000-5000]	<ul style="list-style-type: none"> – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]
2.4.3 Планирование прочих задач [800-4000]	<ul style="list-style-type: none"> – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]
2.5.1 Мониторинг отчетности КО [6000-7000]	<ul style="list-style-type: none"> – 1.5.12 Поиск, предоставление и вывод информации [1000-6000] – 1.5.13 Обработка записей базы данных [500-3000] – 4.6.3 Подготовка форм отчетности [1000-3000] – 5.3.1 Реализация машинной графики для отображения состояния ПС в статике [1000]

2.5.2 Мониторинг отчетности ТУ [6000-7000]	<ul style="list-style-type: none"> – 1.5.12 Поиск, предоставление и вывод информации [1000-6000] – 1.5.13 Обработка записей базы данных [500-3000] – 4.6.3 Подготовка форм отчетности [1000-3000] – 5.3.1 Реализация машинной графики для отображения состояния ПС в статике [1000]
2.6.1 Расчет экономических показателей [5000 – 15000]	<ul style="list-style-type: none"> – 2.1.1 Ввод данных первичных документов в интерактивном режиме [2000-9000] – 2.1.4 Логический, синтаксический и номенклатурный контроль данных [1500-6000] – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]
2.6.2 Экономический и финансовый анализ [10000 – 20000]	<ul style="list-style-type: none"> – 2.1.1 Ввод данных первичных документов в интерактивном режиме [2000-9000] – 2.1.4 Логический, синтаксический и номенклатурный контроль данных [1500-6000] – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]
2.6.3 Моделирование и прогнозирование [20000-30000]	<ul style="list-style-type: none"> – 2.1.1 Ввод данных первичных документов в интерактивном режиме [2000-9000] – 2.1.4 Логический, синтаксический и номенклатурный контроль данных [1500-6000] – 2.1.5 Расчет алгебраических выражений [100-1000] – 2.1.6 Статистическая обработка данных [2000-5000] – 4.6.1 Формирование отчетов [100-4000]

<p>3.4.1 Задачи ведения электронного архива [2500-7500]</p>	<ul style="list-style-type: none"> – 3.4.1 Создание архива [500-2500] – 3.4.2 Добавление данных в архив [500-1000] – 3.4.3 Извлечение данных из архива [500-1000] – 3.4.4 Поиск данных в архиве [500-1500] – 3.4.5 Проверка целостности архива [500-1500]
<p>4.9.4 Ведение картотек расчетных документов</p>	<ul style="list-style-type: none"> – 4.9.1 Ведение картотеки расчетных документов, ожидающих акцепта для оплаты – 4.9.2 Ведение картотеки расчетных документов, не оплаченных в срок – 4.9.3 Прием ЭСИС – запросов УБР для получения информации об актуальном состоянии картотек

ПРИЛОЖЕНИЕ 2

ПРИМЕР ТЕХНИЧЕСКОГО ЗАДАНИЯ НА РАЗРАБОТКУ ПС

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

1.1. Назначение работы

Программное средство мониторинга финансовых операций, выполняемых через банкомат, предназначено для автоматизации:

- учета выданных карт клиентам банка;
- учета операций по снятию наличных в банкоматах Банка;
- учета операций по оплате услуг оператора сотовой связи через банкоматы;
- составления отчета о выполненных операциях;
- анализа выполненных операций для прогнозирования необходимого количества наличных денег для банкоматов и сроков их пополнения.

Программное средство должно обеспечить возможность:

- автоматизированной обработки информации о проведенных операциях и состоянии личного счета клиентов Банка, использующих возможность выполнения финансовых операций по снятию наличных и оплате услуг оператора сотовой связи через банкоматы Банка;
- составления отчета по выполненным финансовым операциям и состоянию счета клиентов;
- анализа информации, полученной от банкоматов и планирования их обслуживания.

1.2. Область применения

Программное средство, разрабатываемое в соответствии с данным техническим заданием, предназначено для применения в подразделениях Банка России, осуществляющих контроль финансовых операций. ПС создается с целью автоматизации учета финансовых операций, выполняемых клиентами Банка с использованием кредитных карт.

2. ОСНОВНЫЕ ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

2.1. Требования к структуре ПС

Необходимая для функционирования программного средства информация поступает на вход ПС из следующих источников:

- банкомат (передает информацию о выполнении финансовых операций);
- оператор ПС (выполняет ввод, получение и удаление информации о клиентах Банка, которым были выданы кредитные карты);

– администратор ПС (выполняет ввод, получение и удаление информации о новых банкоматах и операторах сотовой связи, вводе данных о пополнении денежных средств банкоматов).

В результате работы программное средство формирует:

– отчет о выполненных операциях;
– результаты анализа наличия денежных средств в банкоматах и примерные сроки их пополнения.

Каждая функция программного средства предполагает возврат сообщения об ошибках.

Программное средство должно состоять из следующих компонентов:

– «АРМ Оператора», предназначенного для работы оператора с программным средством;

– «АРМ Администратора», предназначенного для работы администратора с программным средством;

– «База Данных», предназначенного для хранения информации о выданных кредитных картах, операций по снятию наличных, оплаты услуг сотовой связи;

– «Отчет», предназначенного для составления отчета о выполняемых операциях.

– «Анализ», предназначенного для прогнозирования сроков пополнения и количества необходимых наличных денег для банкоматов.

Основными источниками данных для программного средства являются:

– база данных Банка, содержащая информацию о выданных клиентам кредитных картах и проводимых по ним операциям;

– данные, получаемые от банкоматов Банка.

2.2. Требования к функциональным возможностям ПС

Программное средство должно обеспечивать выполнение следующих функций:

– добавление оператором ПС информации о клиенте Банка в базу данных при получении клиентом кредитной карты;

– получение/удаление оператором ПС информации о клиенте Банка из базы данных;

– добавление администратором ПС информации о новом банкомате;

– получение/ удаление администратором ПС информации о банкомате;

– добавление администратором ПС информации об операторе сотовой связи;

– получение/ удаление администратором ПС информации об операторе сотовой связи;

– выполнении администратором ПС операции ввода информации о пополнении денежных средств банкоматов;

- выполнение оператором ПС или клиентом Банка операции получения состояния счета Клиента Банка;
- выполнение оператором ПС или клиентом Банка операции снятия наличных средств со счета клиента Банка;
- выполнение оператором ПС или клиентом Банка операции перевода денежных средств клиента со счета в Банке при оплате услуг;
- выполнение оператором ПС операции перевода денежных средств на счет клиента Банка;
- формирование оператором ПС отчета о выполненных операциях;
- получение администратором ПС анализа о выполненных операциях для прогнозирования необходимого количества наличных денег для банкоматов и сроков их пополнения.

2.3. Технические и системные требования к ПС

Программное средство мониторинга финансовых операций, выполняемых через банкомат, должно функционировать на ТС со следующими минимальными требованиями по аппаратному и программному обеспечению:

- CPU P3, RAM 128 MB;
- 50 MB дискового пространства;
- ОС Windows XP;
- СУБД SQL Server 6.0.

2.4. Требования к надежности

Программное средство мониторинга финансовых операций, выполняемых через банкомат, должно обладать надежностью, обеспечивающей работу ПС в рабочем режиме и оперативное восстановление работоспособности при сбоях. ПС должно иметь необходимый высокий уровень защиты информации в БД и поступающей в программное средство от банкоматов Банка.

В ПС должны быть предусмотрены:

- контроль за целостностью данных на уровне СУБД;
- сохранение целостности данных в БД при нештатном завершении программы;
- сохранение работоспособности ПС при некорректных действиях администратора или оператора ПС.

2.5. Требования к эргономике

Программное средство должно использовать средства построения графического

интерфейса пользователя, обеспечиваемых операционной системой Windows.

Сообщения, выдаваемые программным средством, должны выдаваться на русском языке за исключением стандартных системных сообщений.

3. ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

3.1. Состав и содержание документации должны соответствовать требованиям ГОСТ 19.101–77.

3.2. Перечень документов должен включать:

- 1) Техническое задание на разработку ПС.
- 2) Пояснительная записка;
- 3) Описание информационного обеспечения программы;
- 4) Спецификация;
- 5) Формуляр;
- 6) Программа и методика предварительных испытаний;
- 7) Программа и методика приемочных испытаний;
- 8) Руководство пользователя.

ПРИЛОЖЕНИЕ 3

РАСЧЕТ ТРУДОЕМКОСТИ РАЗРАБОТКИ МАКЕТА ПС

1. Функциональный состав ПС

Определяем функциональный состав макета программного средства исходя из технического задания на разработку ПС. При этом необходимо отметить:

– Компоненты «АРМ оператора» и «АРМ администратора» реализуют минимально необходимый интерфейс. Компонент «АРМ администратора» не реализует функций по получению и удалению информации о банкоматах и операторах сотовой связи, а реализует только ввод этих данных. Для макета также не обязательна полноценная реализация функций обеспечения безопасности информации, хотя в минимальном варианте данная возможность должна присутствовать. Таким образом, программное средство реализует следующие функции:

1) Компонент «АРМ оператора»:

– добавление, получение и удаление информации о клиенте Банка в базе данных;
– выполнение запросов к базе данных для получения состояния счета, снятия наличных средств со счета, перевода денежных средств клиента со счета в Банке при оплате услуг или перевода денежных средств на счет клиента Банка;

2) Компонент «АРМ администратора»:

– добавление информации о банкомате, операторе сотовой связи;
– выполнение запросов к базе данных для ввода информации о пополнении денежных средств банкоматов.

3) Компонент «База данных»

– формирование структуры базы данных;
– обеспечение безопасности передачи данных при работе с БД.

4) Компонент «Анализ»:

– проведение анализа о выполненных операциях для прогнозирования необходимого количества наличных денег для банкоматов и сроков их пополнения;
– формирование отчета по анализу.

3) Компонент «Отчет»:

– формирование отчета о выполненных операциях.

2. Расчет трудоемкости разработки ПС

Определяем трудоемкость разработки и сроки разработки макета программного средства. Ниже представлен отчет по расчету трудоемкости.

ПРИЛОЖЕНИЕ 4

ПРИМЕР РАСЧЕТА ТРУДОЕМКОСТИ РАЗРАБОТКИ ПС С ИСПОЛЬЗОВАНИЕМ КАТАЛОГА ФУНКЦИЙ

1. Функциональный состав ПС

Определяем функциональный состав программного средства исходя из технического задания на разработку ПС:

1) Компонент «АРМ оператора»:

- добавление, получение и удаление информации о клиенте Банка в базе данных;
- выполнение запросов к базе данных для получения состояния счета, снятия наличных средств со счета, перевода денежных средств клиента со счета в Банке при оплате услуг или перевода денежных средств на счет клиента Банка;

2) Компонент «АРМ администратора»:

- добавление, получение и удаление информации о банкомате;
- добавление, получение и удаление информации об операторе сотовой связи;
- выполнение запросов к базе данных для ввода информации о пополнении денежных средств банкоматов.

3) Компонент «База данных»

- формирование структуры базы данных;
- обеспечение безопасности передачи данных при работе с БД.

4) Компонент «Анализ»:

- проведение анализа о выполненных операциях для прогнозирования необходимого количества наличных денег для банкоматов и сроков их пополнения;

- формирование отчета по анализу.

3) Компонент «Отчет»:

- формирование отчета о выполненных операциях.

2. Расчет трудоемкости разработки ПС

Определяем трудоемкость разработки и сроки разработки программного средства. Ниже представлен отчет по расчету трудоемкости.

РАСЧЕТ ТРУДОЕМКОСТИ С ИСПОЛЬЗОВАНИЕМ ПО АРМ

Расчет трудоемкости разработки ПС (Каталог)

Параметры расчета

Параметр	Значение
Имя расчета	Программное средство мониторинга финансовых операций, выполняемых через банкомат
Тип расчета	Расчет трудоемкости разработки ПС (Каталог)
Версия модели	2.5.1.0
Фонд времени разработчика [дней/мес.]	21
Тип ограничения	По продолжительности этапа [мес.]
Тип структуры программного средства	ПС с набором библиотек (=1.00)
Технологии взаимодействия модулей	Оригинальный протокол взаимодействия (sockets) (=0.80)

Коэффициенты уровня расчета

Коэффициент	Уровень значения
Степень новизны ПС	Принципиально новое ПС, не имеющее доступных аналогов на прежнем типе ТС/ОС (=1.06)
Требования к надежности	Сбои ПС приводят к потерям рабочей информации (=1.06)
Требования к производительности	Повышенные требования к производительности (=1.12)
Информативность документации	Повышенный объем для жизненного цикла данного ПС (=1.07)
Опыт разработки ПС подобного типа	Более 5 разработанных ПС (компонентов АС) (=0.81)
Влияние сроков работ на трудоемкость	Равно или более 100 (=1.00)

Коэффициенты уровня подпроцесса

Подпроцесс	Коэффициент	Уровень значения
Анализ требований	Уровень квалификации аналитиков	Высокий (=0.87)
Анализ требований	Опыт работы аналитиков в данной предметной области	5 разработанных ПС (компонентов АС) (=0.87)
Проектирование	Уровень квалификации проектировщиков	Высокий (=0.92)
Проектирование	Опыт работы проектировщиков с используемыми средствами	5 разработанных ПС (компонентов АС) (=0.92)

	проектирования	
Проектирование	Используемые средства проектирования	Средства проектирования БД (BPWIN, ERWIN/ERX, Designer 2000, DataDirect Explorer, Oracle Designer, Silverrun RBM и BPM, Together J) (=1.03)
Программирование	Уровень квалификации программистов	Высокий (=0.95)
Программирование	Используемая среда разработки	Интегрированные среды разработки (Visual Studio) (=1.00)
Тестирование	Уровень квалификации персонала, осуществляющего тестирование	Высокий (=0.89)
Тестирование	Используемые средства тестирования	Автоматизированные средства тестирования использовались (=0.76)
Тестирование	Размер БД	$10 \leq D/P < 100$ (=1.00)
Ввод в действие ПС	Уровень квалификации персонала, осуществляющего ввод в действие ПС	Высокий (=0.94)

Дерево элементов

№	Элемент	Описание	Vi (на условном языке)	Кср.разр.	gi	ki [0;1]	Ксложн.	Копыт
	АРМ Оператора	Компонент предназначен для работы оператора с программным средством						
2.2.5	+Корректировка балансов**	Реализует интерфейс оператора ПС, обеспечивающий работу оператора с программным средством.	500 из [200-500]	C++ (=1.00)	1	0.90	Высокий (=1.10)	Высокий (=0.91)
2.1.15	+Ввод прочих данных	Добавление информации о клиенте Банка в БД программного средства.	3000 из [1000-4000]	C++ (=1.00)	1	0.85	Высокий (=1.10)	Высокий (=0.91)
1.5.12	+Поиск, представление и вывод информации	Получение и удаление информации о клиенте Банка в базе данных, выполнение запросов к базе данных для получения состояния счета, снятия наличных средств со счета, перевода денежных средств клиента со счета в Банке при оплате услуг или перевода денежных сред	3500 из [1000-6000]	C++ (=1.00)	1	0.90	Высокий (=1.10)	Высокий (=0.91)
	АРМ Администратора	Компонент предназначен для работы администратора с программным средством						
2.2.5	+Корректировка балансов**	Реализует интерфейс администратора ПС, обеспечивающий работу администратора с программным средством.	500 из [200-500]	C++ (=1.00)	1	0.90	Высокий (=1.10)	Высокий (=0.91)
2.1.15	+Ввод прочих данных	Добавление информации о банкомате, об операторе сотовой связи, выполнение запросов к базе данных для ввода информации о пополнении денежных средств банкоматов.	3000 из [1000-4000]	C++ (=1.00)	1	0.85	Высокий (=1.10)	Высокий (=0.91)
1.5.12	+Поиск, представление и вывод информации	Получение и удаление информации о банкомате и операторе сотовой связи.	3500 из [1000-6000]	C++ (=1.00)	1	0.90	Высокий (=1.10)	Высокий (=0.91)
	База Данных	Компонент предназначен для хранения информации о выданных кредитных картах, операций по снятию наличных, оплаты услуг сотовой связи.						
1.5.1	+Формирование	Обеспечивает формирование структуры базы данных (для	4500 из	C++	1	1.00	Высокий	Высокий

	физической структуры аналитических БД (на одну БД)	обеспечения ее нормального функционирования с разрабатываемым программным средством).	[1000-8000]	(=1.00)			(=1.10)	(=0.91)
2.1.1	+Ввод данных первичных документов в интерактивном режиме	Обеспечивает безопасность передачи информации из базы данных (согласно требованиям технического задания на разработку ПС).	2000 из [2000-9000]	C++ (=1.00))	1	1.00	Высокий (=1.10)	Высокий (=0.91)
	Отчет	Компонент предназначен для составления отчета о выполняемых операциях (снятие наличных со счета, оплата услуг сотовой связи, получение информации о состоянии счета).						
4.6.1	+Формирование отчетов (на один отчет)*	Формирование отчета о выполненных операциях (снятие наличных со счета, оплата услуг сотовой связи, получение информации о состоянии счета).	2050 из [100-4000]	C++ (=1.00)	1	1.00	Высокий (=1.10)	Высокий (=0.91)
	Анализ	Компонент предназначен для прогнозирования сроков пополнения и количества необходимых наличных денег для банкоматов.						
5.1.2	+Настройка аналитических таблиц	Проведение анализа о выполненных операциях (снятие наличных со счета, оплата услуг сотовой связи, получение информации о состоянии счета) для прогнозирования сроков пополнения и количества необходимых наличных денег для банкоматов.	1000 из [1000-2000]	C++ (=1.00)	1	1.00	Высокий (=1.10)	Высокий (=0.91)
4.6.1	+Формирование отчетов (на один отчет)*	Формирование отчета по анализу выполненных операций (снятие наличных со счета, оплата услуг сотовой связи, получение информации о состоянии счета) через банкоматы Банка.	2050 из [100-4000]	C++ (=1.00)	1	1.00	Высокий (=1.10)	Высокий (=0.91)
	Общий объем ПС:		22010					

Расчет трудоемкости подпроцессов

Подпроцесс	Трудоемкость [чел.дн.]	Кол-во разработчиков [чел]	Продолжительность [мес]/[дн]
Анализ требований	4.89	0.12	2.00/42.00
Проектирование	67.55	1.07	3.00/63.00
Программирование	484.60	5.77	4.00/84.00
Тестирование	30.57	0.73	2.00/42.00
Ввод в действие ПС	6.07	0.29	1.00/21.00
Всего	593.69		12.00/252.00
Базовая трудоемкость	645.71		

ПРИЛОЖЕНИЕ 5

ПРИМЕР РАСЧЕТА ТРУДОЕМКОСТИ РАЗРАБОТКИ ПС С ИСПОЛЬЗОВАНИЕМ МЕТОДА ФУНКЦИОНАЛЬНОЙ БАЛЬНОЙ ОЦЕНКИ

1. Функциональный состав ПС

Программное средство мониторинга финансовых операций, выполняемых через банкомат, решает задачи:

- формирования и ведения базы данных о финансовых операциях, выполняемых клиентами Банка через банкоматы и оператором ПС (запросы о состоянии счета, снятие наличных и оплата услуг оператора сотовой связи);
- учета кредитных карт, выданных клиентам Банка;
- составления отчета о выполненных операциях;
- анализа текущего состояния банкоматов Банка с целью определения сроков обслуживания банкоматов (пополнение наличных денежных средств банкоматов) и количестве необходимых для банкомата наличных денежных средств.

К основным функциям рассматриваемого программного средства относятся следующие:

- добавление информации оператором ПС о клиенте Банка в базу данных при получении клиентом кредитной карты;
- получение/удаление оператором ПС информации о клиенте Банка из базы данных;
- внесение оператором ПС информации о зачислении денежных средств на счет клиента;
- добавление администратором ПС информации о новом банкомате;
- получение/удаление администратором ПС информации о банкомате;
- добавление администратором ПС информации об операторе сотовой связи;
- получение/удаление администратором ПС информации об операторе сотовой связи;
- выполнении администратором ПС операции ввода информации о пополнении денежных средств банкоматов;
- выполнении оператором ПС или клиентом Банка операции получения состояния счета клиента Банка;
- выполнении оператором ПС или клиентом Банка операции снятия наличных средств со счета клиента Банка;

- выполнение оператором ПС или клиентом Банка операции перевода денежных средств клиента со счета в Банке при оплате услуг;
- формирование оператором ПС отчета о выполненных операциях;
- получение администратором ПС анализа о выполненных операциях для прогнозирования необходимого количества наличных денег для банкоматов и сроков их пополнения.

2. Описание базы данных

База данных представляет собой реляционную модель с простыми (атомарными) типами данных, приведенную к 2 нормальной форме (Рисунок 2.1 Структура базы данных). БД состоит из 8 таблиц:

- 1) Клиент – содержит информацию о клиенте банка (фамилия, имя и отчество, адрес, паспортные данные) и договоре на обслуживание кредитной карты, заключенном с ним (номер и дата заключения договора, номер выданной клиенту Банка карты).
- 2) Счет – содержит информацию о состоянии счета клиента банка (сумму денежных средств на карте);
- 3) Банкомат – содержит информацию о банкоматах Банка (номер банкомата, его серийный номер и место расположения), посредством использования которых клиент Банка способен осуществить операции получения баланса персонального счета, снятия наличных со счета и оплаты услуг оператора сотовой связи.
- 4) ДенежныйОстаток – содержит информацию о времени и сумме пополнения денежных средств банкоматов Банка;
- 5) ОператорСотовойСвязи – содержит информацию об операторах сотовой связи (номер счета и название оператора, а также его юридический адрес), услуги которых клиент может оплатить с использованием банкоматов Банка.
- 6) Операция – содержит информацию об операциях (номер операции и банкомата, код типа операции, номер карты, использованной для совершения операции, дату выполнения и результат операции, а также сумму, на которую была совершена операция), выполненных клиентами с использованием банкоматов Банка.
- 7) ОперацияОплатыУслуг – содержит дополнительную информацию (счет и номер абонента) об операциях оплаты услуг оператора сотовой связи.
- 8) Тип операции – содержит информацию о типах операций, которые могут совершаться через банкоматы Банка.

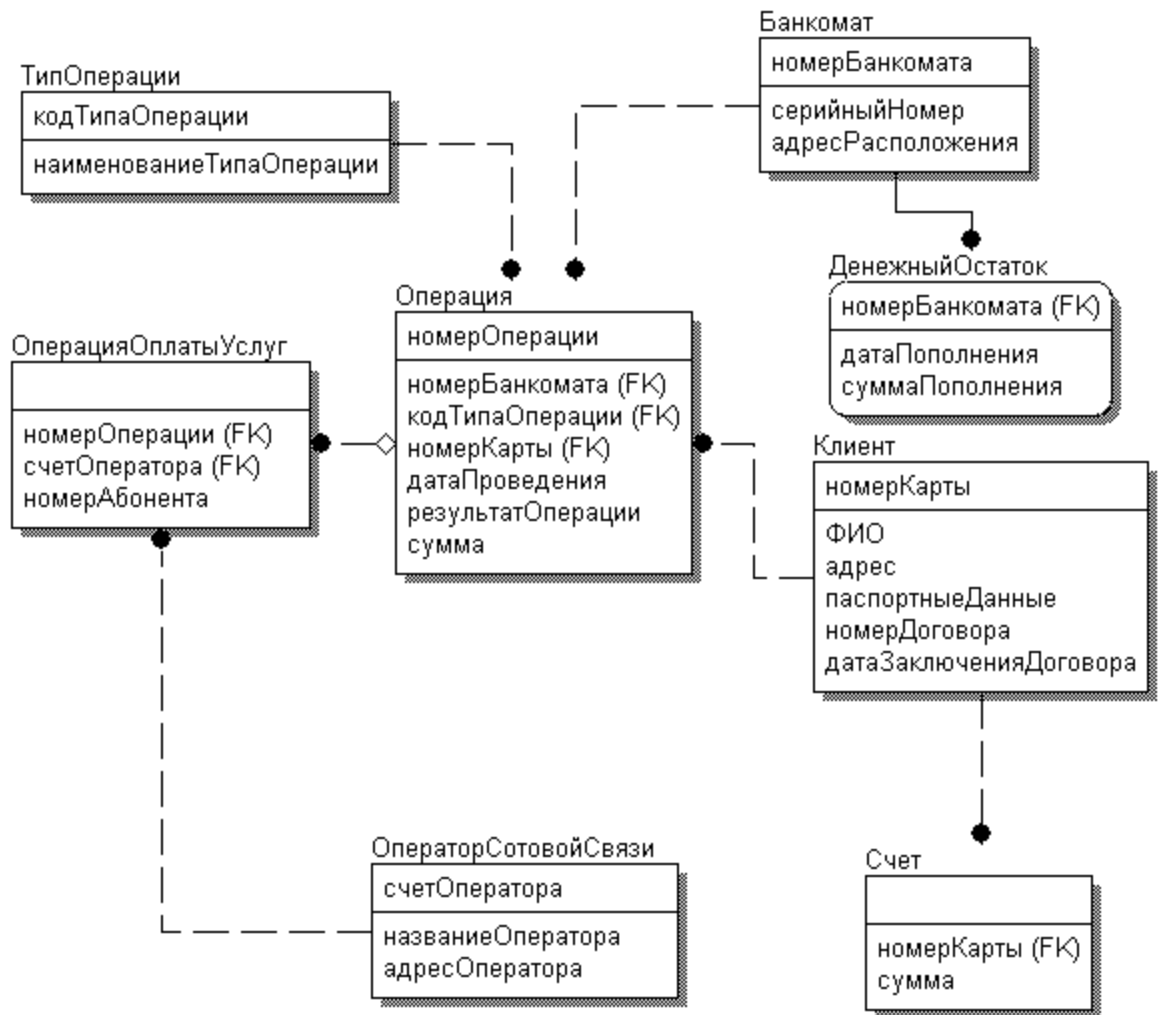


Рисунок 2.1 Структура базы данных

3. Описание архитектуры ПС.

Программное средство мониторинга финансовых операций, выполняемых через банкомат, реализуется в виде набора модулей:

- исполняемый модуль «АРМ Оператора»;
- исполняемый модуль «АРМ Администратора»;
- база данных;
- модуль формирования отчета;
- модуль анализа.

Компонентная структура ПС представлена на рисунке 2.2 в виде UML-диаграммы КОМПОНЕНТОВ.

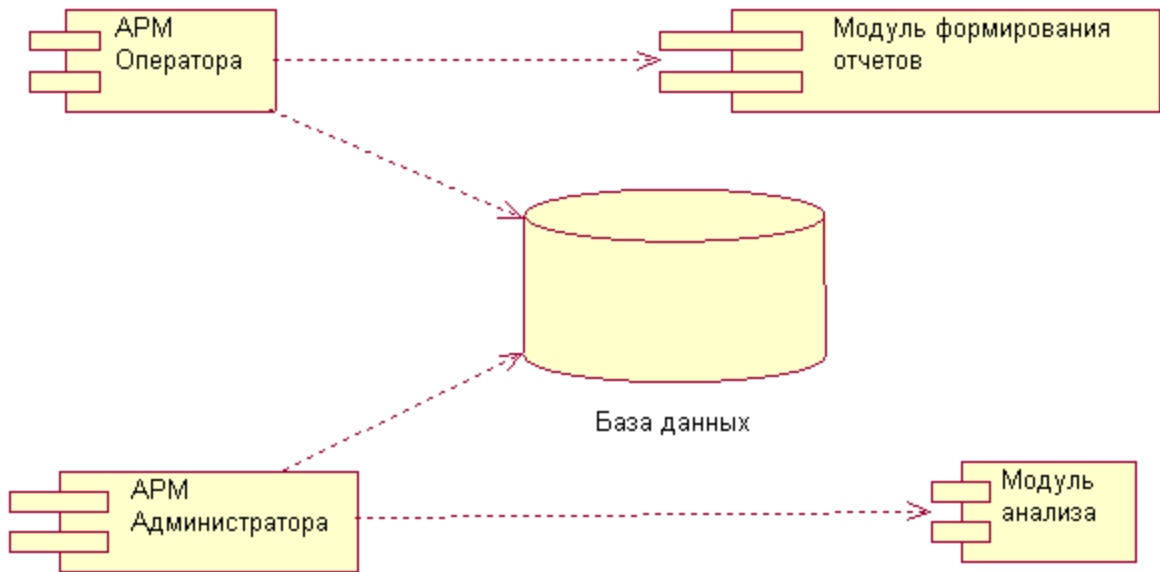


Рисунок 2.2 UML-диаграмма компонентов

Зависимость компонентов в соответствии с требованиями нотации UML показана на диаграмме в виде пунктирных стрелок.

Для описания функциональных возможностей реализуемого ПС составим UML-диаграмму вариантов использования (рис. 2.3). Варианты использования должны соответствовать требуемым функциям программного средства.

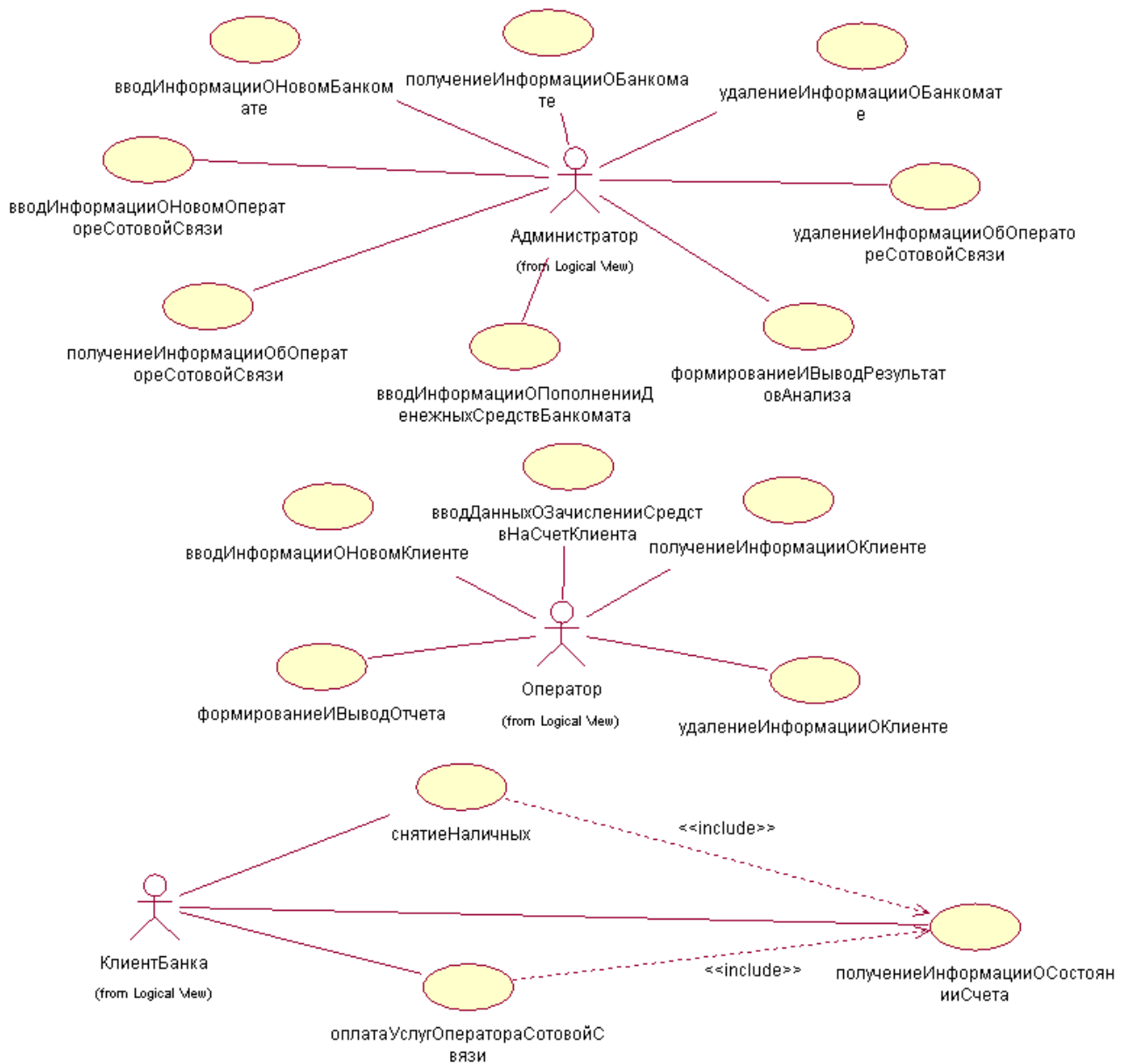


Рисунок 2.3 Диаграмма вариантов использования

Функции, выполнение которых инициируется администратором, оператором ПС или клиентом Банка, отмечены на диаграмме классов (рис. 2.3) в виде ассоциативной связи (сплошная прямая линия) между соответствующими сущностями и вариантами использования.

Для описания внутренней структуры реализуемого ПС построим диаграмму классов, построенную в соответствии с требованиями нотации UML (рис. 2.4).

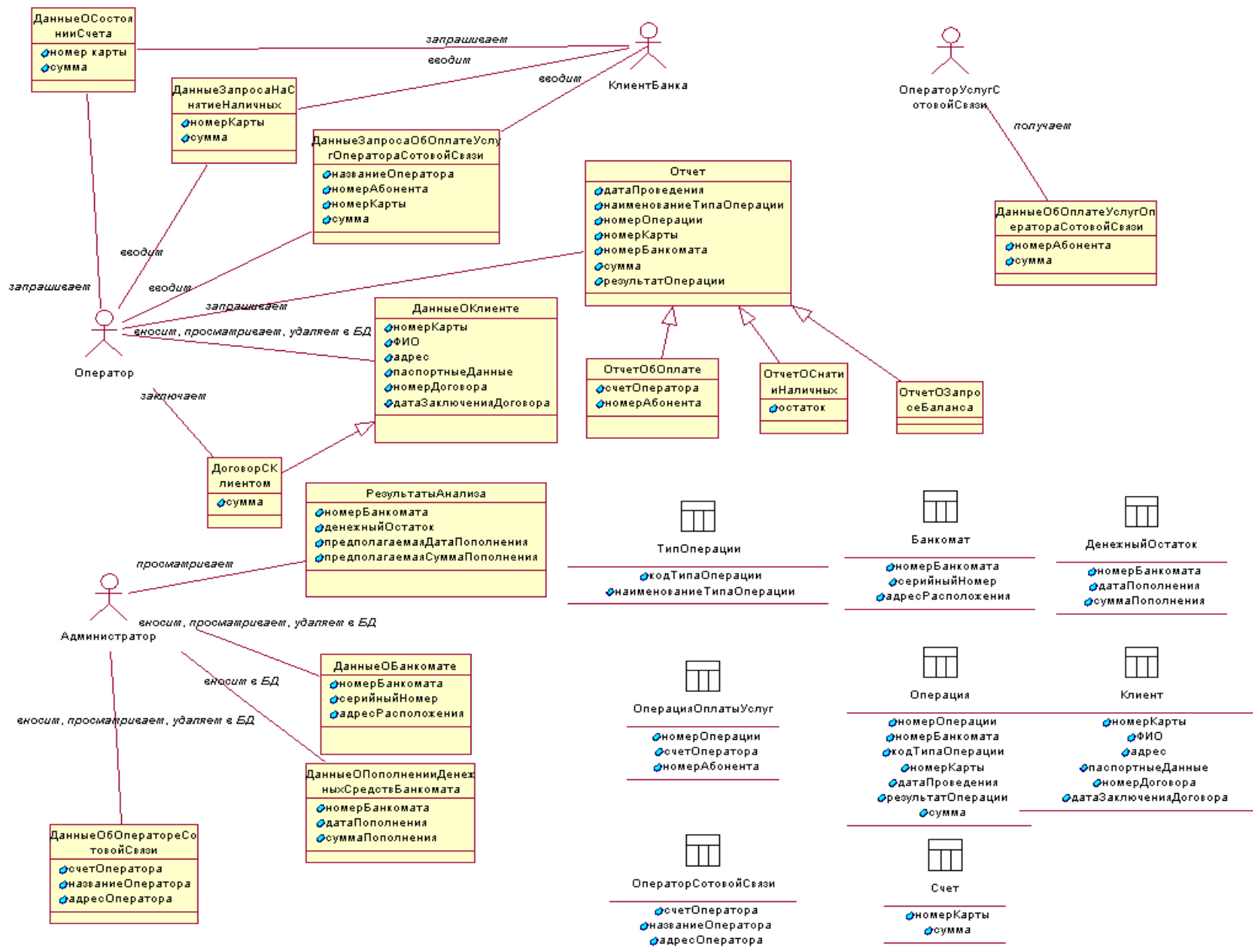


Рисунок 2.4 Диаграмма классов (внутренняя структура ПС)

Классы «ТипОперации», «Банкомат», «ДенежныйОстаток», «ОперацияОплатыУслуг», «Операция», «Клиент», «ОператорСотовойСвязи», «Счет» являются таблицами базы данных, с которыми осуществляется работа программного средства. Для того, чтобы не загромождать диаграмму, в рассматриваемом примере не показаны ассоциативные связи между таблицами (рассмотрены в пункте 2 «Описание базы данных»), а также их связь с остальными классами (учтены в таблице 2.1).

Варианты использования (рис. 2.3) как элементарные функции программного средства предполагают чтение и запись атрибутов определенных классов (рис. 2.4). В таблице 2.1 показаны классы и атрибуты классов для каждого из вариантов использования.

Атрибуты, используемые в рамках процессов, определяемых вариантами использования

Таблица 2.1

Вариант использования	Класс	Атрибут
вводИнформацииОНовомКлиенте	«ДанныеОКлиенте»	номерКарты, ФИО, адрес, паспортныеДанные, номерДоговора, датаЗаклученияДоговора
	«Клиент»	номерКарты, ФИО, адрес, паспортныеДанные, номерДоговора, датаЗаклученияДоговора
вводДанныхОЗачисленииСредствНаСчетКлиента	«ДанныеОКлиенте», «ДоговорСКлиентом»	номерКарты, сумма
	«Счет»	номерКарты, сумма
получениеИнформацииОКлиенте	«Клиент»	номерКарты, ФИО, адрес, паспортныеДанные, номерДоговора, датаЗаклученияДоговора
	«ДанныеОКлиенте»	номерКарты, ФИО, адрес, паспортныеДанные, номерДоговора, датаЗаклученияДоговора
формированиеИВыводОтчета	«Отчет», «ОтчетОбОплате», «ОтчетОСнятииНаличных», «ОтчетОЗапросеБаланса»	датаПроведения, наименованиеТипаОперации, номерОперации, номерКарты, номерБанкомата, сумма, результатОперации, счетОператора, номерАбонента, остаток

	«Операция»	номерОперации, номерБанкомата, код ТипаОперации, номерКарты, датаПроведения, результатОперации, сумма
	«ОперацияОп латыУслуг»	номерОперации, счетОператора, номерАбонента
	«ТипОперации »	код ТипаОперации, наименованиеТипаОперац ии
	«Счет»	номерКарты, сумма
удалениеИнформацииОКлиенте	«ДанныеОКли енте»	номерКарты
вводИнформацииОНовомБанкомате	«Банкомат»	номерБанкомата, серийныйНомер, адресРасположения
	«ДанныеОБанк омате»	номерБанкомата, серийныйНомер, адресРасположения
получениеИнформацииОБанкомате	«Банкомат»	номерБанкомата, серийныйНомер, адресРасположения
	«ДанныеОБанк омате»	номерБанкомата, серийныйНомер, адресРасположения
удалениеИнформацииОБанкомате	«ДанныеОБанк омате»	номерБанкомата
вводИнформацииОНовомОператореСотово йСвязи	«ОператорСот овойСвязи»	счетОператора, названиеОператора, адресОператора
	«ДанныеОбОп ератореСотово йСвязи»	счетОператора, названиеОператора, адресОператора
получениеИнформацииОбОператореСотово йСвязи	«ОператорСот овойСвязи»	счетОператора, названиеОператора, адресОператора
	«ДанныеОбОп ератореСотово йСвязи»	счетОператора, названиеОператора, адресОператора
удалениеИнформацииОбОператореСотовой Связи	«ДанныеОбОп ератореСотово йСвязи»	счетОператора
вводИнформацииОПополненииДенежныхС редствБанкомата	«ДенежныйОс таток»	номерБанкомата, датаПополнения, суммаПополнения

	«Данные О Пополнении Денежных Средств Банкомата»	номер Банкомата, дата Пополнения, сумма Пополнения
формирование И Вывод Результатов Анализа	«Результаты Анализа»	номер Банкомата, денежный Остаток, предполагаемая Дата Пополнения Денежных Средств, предполагаемая Сумма Пополнения Денежных Средств
	«Денежный Остаток»	номер Банкомата, дата Пополнения, сумма Пополнения
	«Операция»	номер Банкомата, код Типа Операции, дата Проведения, результат Операции, сумма
снятие Наличных	«Банкомат»	номер Банкомата, серийный Номер
	«Операция»	номер Операции, номер Банкомата, код Типа Операции, номер Карты, дата Проведения, результат Операции, сумма
	«Клиент»	дата Заключения Договора, номер Карты
	«Счет»	номер Карты, сумма
	«Данные Запроса На Снятие Наличных»	номер Карты, сумма
оплата Услуг Оператора Сотовой Связи	«Банкомат»	номер Банкомата, серийный Номер
	«Операция»	номер Операции, номер Банкомата, код Типа Операции, номер Карты, дата Проведения, результат Операции, сумма
	«Клиент»	дата Заключения Договора, номер Карты
	«Счет»	номер Карты, сумма
	«Оператор Сотовой Связи»	счет Оператора, название Оператора
	«Операция Оплаты Услуг»	номер Операции, счет Оператора, номер Абонента

	«ДанныеЗапросаОбОплатеУслугОператораСотовойСвязи»	названиеОператора, номерАбонента, номерКарты, сумма
	«ДанныеОбОплатеУслугОператораСотовойСвязи»	номерАбонента, сумма
получениеИнформацииОСостоянииСчета	«Банкомат»	номерБанкомата, серийныйНомер
	«Операция»	номерОперации, номерБанкомата, кодТипаОперации, номерКарты, датаПроведения, результатОперации, сумма
	«Клиент»	датаЗаключенияДоговора, номерКарты
	«Счет»	номерКарты, сумма
	«ДанныеОСостоянииСчета»	номерКарты, сумма

4. Расчет трудоемкости разработки ПС

Шаг 1. Определяем границу программного средства. В границу ПС входят:

- класс «Клиент»;
- класс «Банкомат»;
- класс «Операция»;
- класс «ТипОперации»;
- класс «Счет»;
- класс «ДенежныйОстаток»;
- класс «ОперацияОплатыУслуг»;
- класс «ОператорСотовойСвязи»;
- вариант использования «вводИнформацииОНовомБанкомате»;
- вариант использования «получениеИнформацииОБанкомате»;
- вариант использования «удалениеИнформацииОБанкомате»;
- вариант использования «вводИнформацииОНовомОператореСотовойСвязи»;
- вариант использования «получениеИнформацииОбОператореСотовойСвязи»;
- вариант использования «удалениеИнформацииОбОператореСотовойСвязи»;
- вариант использования
«вводИнформацииОПополнениеДенежныхСредствБанкомата»;
- вариант использования «формированиеИВыводРезультатовАнализа»;
- вариант использования «вводИнформацииОНовомКлиенте»;
- вариант использования «получениеИнформацииОКлиенте»;
- вариант использования «удалениеИнформацииОКлиенте»;
- вариант использования «формированиеИВыводОтчета»;
- вариант использования «снятиеНаличных»;
- вариант использования «оплатаУслугОператораСотовойСвязи»;
- вариант использования «получениеИнформацииОСостоянииСчета»;
- вариант использования «вводДанныхОЗачисленииСредствНаСчетКлиента».

В границу ПС не входят:

- класс «ДанныеОбОператореСотовойСвязи»;
- класс «ДанныеОПополненииДенежныхСредствБанкомата»;
- класс «ДанныеОБанкомате»;

- класс «РезультатыАнализа»;
- класс «ДоговорСКлиентом»;
- класс «ДанныеОКлиенте»;
- класс «ДанныеЗапросаОбОплатеУслугОператораСотовойСвязи»;
- класс «ДанныеЗапросаНаСнятиеНаличных»;
- класс «ДанныеОСостоянииСчета»;
- класс «Отчет»;
- класс «ОтчетОбОплате»;
- класс «ОтчетОСнятииНаличных»;
- класс «ОтчетОЗапросеБаланса»;
- класс «ДанныеОбОплатеУслугОператораСотовойСвязи»;

Перечисленные классы с точки зрения логической организации ПС представляют собой формы ввода (все кроме «Отчет» и классов, связанных с ним отношением обобщения, класса «РезультатыАнализа», класса «ДанныеОбОплатеУслугОператораСотовойСвязи») либо представления информации (класс «Отчет» и классы, связанные с ним отношением обобщения, класс «РезультатыАнализа», а также класс «ДанныеОбОплатеУслугОператораСотовойСвязи», который является интерфейсом для выхода посредством сетевого интерфейса данных об оплате услуг оператору сотовой связи).

Шаг 2. Определяем границу расчета. В границу расчета входят:

- класс «Клиент»;
- класс «Банкомат»;
- класс «Операция»;
- класс «ТипОперации»;
- класс «Счет»;
- класс «ДенежныйОстаток»;
- класс «ОперацияОплатыУслуг»;
- класс «ОператорСотовойСвязи»;
- класс «ДанныеОбОператореСотовойСвязи»;
- класс «ДанныеОПополненииДенежныхСредствБанкомата»;
- класс «ДанныеОБанкомате»;
- класс «РезультатыАнализа»;
- класс «ДоговорСКлиентом»;

- класс «ДанныеОКлиенте»;
- класс «ДанныеЗапросаОбОплатеУслугОператораСотовойСвязи»;
- класс «ДанныеЗапросаНаСнятиеНаличных»;
- класс «ДанныеОСостоянииСчета»;
- класс «Отчет»;
- класс «ОтчетОбОплате»;
- класс «ОтчетОСнятииНаличных»;
- класс «ОтчетОЗапросеБаланса»;
- класс «ДанныеОбОплатеУслугОператораСотовойСвязи»;
- вариант использования «вводИнформацииОНовомБанкомате»;
- вариант использования «получениеИнформацииОБанкомате»;
- вариант использования «удалениеИнформацииОБанкомате»;
- вариант использования «вводИнформацииОНовомОператореСотовойСвязи»;
- вариант использования «получениеИнформацииОбОператореСотовойСвязи»;
- вариант использования «удалениеИнформацииОбОператореСотовойСвязи»;
- вариант использования «вводИнформацииОПополненииДенежныхСредствБанкомата»;
- вариант использования «формированиеИВыводРезультатовАнализа»;
- вариант использования «вводИнформацииОНовомКлиенте»;
- вариант использования «получениеИнформацииОКлиенте»;
- вариант использования «удалениеИнформацииОКлиенте»;
- вариант использования «формированиеИВыводОтчета»;
- вариант использования «снятиеНаличных»;
- вариант использования «оплатаУслугОператораСотовойСвязи»;
- вариант использования «получениеИнформацииОСостоянииСчета»;
- вариант использования «вводДанныхОЗачисленииСредствНаСчетКлиента».

Шаг 3. Определяем логические файлы ПС и их сложность. Выделяем на диаграмме классов те, группы классов, которые связаны отношением обобщения (наследования) и композиции или агрегации.

Класс «ДанныеОКлиенте» является обобщением для класса «ДоговорСКлиентом», поэтому вместе они образуют один логический файл.

Класс «Отчет» является обобщением для классов «ОтчетОбОплате», «ОтчетОСнятииНаличных» и «ОтчетОЗапросеБаланса», поэтому вместе они образуют один логический файл.

Классов, имеющих отношение композиции или агрегации на диаграмме нет. Каждый из оставшихся классов является логическим файлом.

Аналогичным образом, необходимо выделить варианты использования, связанные отношением обобщения. В нашем случае таких элементов на диаграмме вариантов использования нет.

В границу ПС согласно шагу 1 входят классы:

- класс «Клиент»;
- класс «Банкомат»;
- класс «Операция»;
- класс «ТипОперации»;
- класс «Счет»;
- класс «ДенежныйОстаток»;
- класс «ОперацияОплатыУслуг»;
- класс «ОператорСотовойСвязи»;

Соответствующие данным классам логические файлы являются внутренними логическими файлами. Все остальные классы, не входящие в границу ПС, но входящие в границу расчета, являются внешними логическими файлами.

Сложность логических файлов зависит от количества их элементов данных и числа записей. Данные значения определяются в соответствии с Методикой. Результаты приведены в таблице 3.1.

Количество элементов данных и записей логических файлов

Таблица 3.1

Логический файл	Количество элементов данных	Количество записей
Внутренний логический файл, состоящий из класса «Клиент»	7 (все атрибуты класса + первичный ключ «номерКарты»)	1
Внутренний логический файл, состоящий из класса «Банкомат»	4 (все атрибуты класса + первичный ключ «номерБанкомата»)	1
Внутренний логический файл, состоящий из класса «ОператорСотовойСвязи»	4 (все атрибуты класса + первичный ключ «счетОператора»)	1
Внутренний логический файл, состоящий из класса «Операция»	8 (все атрибуты класса + первичный ключ «номерОперации»)	1
Внутренний логический файл, состоящий из класса «ОперацияОплатыУслуг»	3 (все атрибуты класса)	1
Внутренний логический файл, состоящий из класса «Счет»	2 (все атрибуты класса)	1
Внутренний логический файл, состоящий из класса «ТипОперации»	3 (все атрибуты класса + первичный ключ «кодТипаОперации»)	1
Внутренний логический файл, состоящий из класса «ДенежныйОстаток»	3 (все атрибуты класса)	1
Внешний логический файл, состоящий из класса «ДанныеОбОператореСотовойСвязи»	3 (все атрибуты класса)	1
Внешний логический файл, состоящий из класса «ДанныеОПополнении ДенежныхСредствБанкомата»	3 (все атрибуты класса)	1
Внешний логический файл, состоящий из класса «ДанныеОБанкомате»	3 (все атрибуты класса)	1
Внешний логический файл, состоящий из класса «РезультатыАнализа»	4 (все атрибуты класса)	1

Внешний логический файл, состоящий из класса «ДанныеОСостоянииСчета»	2 (все атрибуты класса)	1
Внешний логический файл, состоящий из класса «ДанныеЗапросаНаСнятиеНаличных»	2 (все атрибуты класса)	1
Внешний логический файл, состоящий из класса «ДанныеЗапросаОбОплате УслугОператораСотовойСвязи»	4 (все атрибуты класса)	1
Внешний логический файл, состоящий из класса «ДанныеОбОплатеУслуг ОператораСотовойСвязи»	2 (все атрибуты класса)	1
Внутренний логический файл, состоящий из группы классов «Отчет», «ОтчетОбОплате», «ОтчетОСнятииНаличных», «ОтчетОЗапросеБаланса»	10 (все атрибуты классов)	3 (обобщаемые классы – «ОтчетОбОплате», «ОтчетОСнятииНаличных», «ОтчетОЗапросеБаланса»)
Внутренний логический файл, состоящий из группы классов «ДоговорСКлиентом», «ДанныеОКлиенте»	7 (все атрибуты классов)	1 (обобщаемые классы – «ДоговорСКлиентом»)

На основе количества элементов данных и количества записей, входящих в состав каждого логического файла, по таблице 4.1 Методики определяем уровень функциональной сложности для каждого логического файла:

Уровни сложности логических файлов

Таблица 3.2

Логический файл	Уровень сложности
Внутренний логический файл, состоящий из класса «Клиент»	низкий
Внутренний логический файл, состоящий из класса «Банкомат»	низкий
Внутренний логический файл, состоящий из класса «ОператорСотовойСвязи»	низкий
Внутренний логический файл, состоящий из класса «Операция»	низкий
Внутренний логический файл, состоящий из класса «ОперацияОплатыУслуг»	низкий
Внутренний логический файл, состоящий из класса «Счет»	низкий
Внутренний логический файл, состоящий из класса «ТипОперации»	низкий

Внутренний логический файл, состоящий из класса «ДенежныйОстаток»	низкий
Внешний логический файл, состоящий из класса «ДанныеОбОператореСотовойСвязи»	низкий
Внешний логический файл, состоящий из класса «ДанныеОПополнении ДенежныхСредствБанкомата»	низкий
Внешний логический файл, состоящий из класса «ДанныеОБанкомате»	низкий
Внешний логический файл, состоящий из класса «РезультатыАнализа»	низкий
Внешний логический файл, состоящий из класса «ДанныеОСостоянииСчета»	низкий
Внешний логический файл, состоящий из класса «ДанныеЗапросаНаСнятиеНаличных»	низкий
Внешний логический файл, состоящий из класса «ДанныеЗапросаОбОплате УслугОператораСотовойСвязи»	низкий
Внешний логический файл, состоящий из класса «ДанныеОбОплатеУслуг ОператораСотовойСвязи»	низкий
Внутренний логический файл, состоящий из группы классов «Отчет», «ОтчетОбОплате», «ОтчетОСнятииНаличных», «ОтчетОЗапросеБаланса»	низкий
Внутренний логический файл, состоящий из группы классов «ДоговорСКлиентом», «ДанныеОКлиенте»	низкий

Шаг 4. Выделяем транзакционные функции ПС и определяем их сложность.

Согласно Методике, разделяем транзакционные функции на три группы: единицы входной информации, единицы выходной информации и внешние запросы.

1 группа. Единицы входной информации:

- вариант использования «вводИнформацииОНовомКлиенте»;
- вариант использования «вводДанныхОЗачисленииСредствНаСчетКлиента»;
- вариант использования «удалениеИнформацииОКлиенте»;
- вариант использования «вводИнформацииОНовомБанкомате»;
- вариант использования «удалениеИнформацииОБанкомате»;
- вариант использования «вводИнформацииОНовомОператореСотовойСвязи»;
- вариант использования «удалениеИнформацииОбОператореСотовойСвязи»;
- вариант использования «вводИнформацииОПополненииДенежных СредствБанкомата»;
- вариант использования «снятиеНаличных»;

- вариант использования «оплата Услуг Оператора Сотовой Связи».

2 группа. Единицы выходной информации:

- вариант использования «формирование И Вывод Отчета»;
- вариант использования «формирование И Вывод Результатов Анализа»;
- вариант использования «получение Информации О Состоянии Счета».

3 группа. Внешние запросы:

- вариант использования «получение Информации О Клиенте»;
- вариант использования «получение Информации О Банкомате»;
- вариант использования «получение Информации Об Операторе Сотовой Связи».

Определяем характеристики единиц входной информации:

Единицы входной информации

Таблица 3.3

Единица входной информации	Количество элементов данных, пересекающих границу ПС	Количество используемых логических файлов
ввод Информации О Новом Клиенте	7 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	2
ввод Данных О Зачислении Средств На Счет Клиента	3 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	2
удаление Информации О Клиенте	2 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	1

вводИнформацииОНовомБанкомате	4 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	2
удалениеИнформацииОБанкомате	2 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	1
вводИнформацииОНовомОператореСотовойСвязи	4 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	2
удалениеИнформацииОбОператореСотовойСвязи	2 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	1
вводИнформацииОПополненииДенежных СредствБанкомата	4 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	2

<p style="text-align: center;">снятиеНаличных</p>	<p>6 (номерБанкомата, кодТипаОперации, номерКарты, сумма + элемент на возможность вывода транзакционной функцией сообщения об ошибке + элемент на возможность различного способа выполнения элементарного процесса – вызывается клиентом банка через интерфейс банкомата либо оператором ПС)</p>	<p style="text-align: center;">5</p>
<p style="text-align: center;">оплатаУслугОператораСотовойСвязи</p>	<p>7 (номерБанкомата, кодТипаОперации, номерКарты, сумма, номерАбонента + элемент на возможность вывода транзакционной функцией сообщения об ошибке + элемент на возможность различного способа выполнения элементарного процесса – вызывается клиентом банка через интерфейс банкомата либо оператором ПС)</p>	<p style="text-align: center;">8</p>

Определяем уровень сложности единиц входной информации:

Сложность единиц входной информации

Таблица 3.4

Единица входной информации	Уровень сложности
вводИнформацииОНовомКлиенте	средний
вводДанныхОЗачисленииСредствНаСчетКлиента	низкий
удалениеИнформацииОКлиенте	низкий
вводИнформацииОНовомБанкомате	низкий
удалениеИнформацииОБанкомате	низкий
вводИнформацииОНовомОператореСотовойСвязи	низкий
удалениеИнформацииОбОператореСотовойСвязи	низкий
вводИнформацииОПополненииДенежных СредствБанкомата	низкий
снятиеНаличных	высокий
оплатаУслугОператораСотовойСвязи	высокий

Определяем характеристики единиц выходной информации:

Единицы выходной информации

Таблица 3.5

Единица выходной информации	Количество элементов данных, пересекающих границу ПС	Количество используемых логических файлов
формированиеИВыводОтчета	11 (датаПроведения, наименованиеТипаОперации, номерОперации, номерКарты, номерБанкомата, сумма, результатОперации, счетОператора, номерАбонента, остаток+ элемент на возможность вывода транзакционной функцией сообщения об ошибке)	5
формированиеИВывод РезультатовАнализа	5 (номерБанкомата, денежныйОстаток, предполагаемаяДатаПополнения ДенежныхСредств, предполагаемаяСуммаПополнения ДенежныхСредств+ элемент на возможность вывода транзакционной функцией сообщения об ошибке)	3

получениеИнформацииО СостоянииСчета	4 (номерКарты, сумма+ элемент на возможность вывода транзакционной функцией сообщения об ошибке+ элемент на возможность различного способа выполнения элементарного процесса – вызывается клиентом банка через интерфейс банкомата либо оператором ПС)	5
-------------------------------------	--	---

Определяем уровень сложности единиц выходной информации:

Сложность единиц выходной информации

Таблица 3.6

Единица выходной информации	Уровень сложности
формированиеИВыводОтчета	высокий
формированиеИВыводРезультатовАнализа	низкий
получениеИнформацииОСостоянииСчета	средний

Определяем характеристики внешних запросов:

Внешние запросы

Таблица 3.7

Внешние запросы	Количество элементов данных, пересекающих границу ПС	Количество используемых логических файлов
получениеИнформацииОКлиенте	7 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	2
получениеИнформацииОБанкомате	4 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	2

получениеИнформацииОбОператореСотовойСвязи	4 (все атрибуты логического файла + элемент на возможность вывода транзакционной функцией сообщения об ошибке)	2
--	--	---

Определяем уровень сложности внешних запросов :

Уровень сложности внешних запросов

Таблица 3.8

Внешний запрос	Уровень сложности
получениеИнформацииОКлиенте	средний
получениеИнформацииОБанкомате	низкий
получениеИнформацииОбОператореСотовойСвязи	низкий

Шаг 5. Определяем нескорректированный объем ПС в функциональных точках:

Количество функциональных точек, соответствующих элементу данных или транзакционной функции

Таблица 3.9

Элемент данных или транзакционная функция	Количество функциональных точек
Внутренний логический файл, состоящий из класса «Клиент»	7
Внутренний логический файл, состоящий из класса «Банкомат»	7
Внутренний логический файл, состоящий из класса «ОператорСотовойСвязи»	7
Внутренний логический файл, состоящий из класса «Операция»	7
Внутренний логический файл, состоящий из класса «ОперацияОплатыУслуг»	7
Внутренний логический файл, состоящий из класса «Счет»	7
Внутренний логический файл, состоящий из класса «ТипОперации»	7
Внутренний логический файл, состоящий из класса «ДенежныйОстаток»	7
Внешний логический файл, состоящий из класса «ДанныеОбОператореСотовойСвязи»	5
Внешний логический файл, состоящий из класса «ДанныеОПополнении ДенежныхСредствБанкомата»	5
Внешний логический файл, состоящий из класса «ДанныеОБанкомате»	5

Внешний логический файл, состоящий из класса «РезультатыАнализа»	5
Внешний логический файл, состоящий из класса «ДанныеОСостоянииСчета»	5
Внешний логический файл, состоящий из класса «ДанныеЗапросаНаСнятиеНаличных»	5
Внешний логический файл, состоящий из класса «ДанныеЗапросаОбОплате УслугОператораСотовойСвязи»	5
Внешний логический файл, состоящий из класса «ДанныеОбОплатеУслуг ОператораСотовойСвязи»	5
Внутренний логический файл, состоящий из группы классов «Отчет», «ОтчетОбОплате», «ОтчетОСнятииНаличных», «ОтчетОЗапросеБаланса»	5
Внутренний логический файл, состоящий из группы классов «ДоговорСКлиентом», «ДанныеОКлиенте»	5
Единица входной информации «вводИнформацииОНовомКлиенте»	4
Единица входной информации «вводДанныхОЗачисленииСредствНаСчетКлиента»	3
Единица входной информации «удалениеИнформацииОКлиенте»	3
Единица входной информации «вводИнформацииОНовомБанкомате»	3
Единица входной информации «удалениеИнформацииОБанкомате»	3
Единица входной информации «вводИнформацииОНовомОператореСотовойСвязи»	3
Единица входной информации «удалениеИнформацииОбОператореСотовойСвязи»	3
Единица входной информации «вводИнформацииО ПополненииДенежныхСредствБанкомата»	3
Единица входной информации «снятиеНаличных»	6
Единица входной информации «оплатаУслугОператораСотовойСвязи»	6
Единица выходной информации «формированиеИВыводОтчета»	7
Единица выходной информации «формированиеИВыводРезультатов Анализа»	4
Единица выходной информации «получениеИнформацииОСостоянииСчета»	5
Внешний запрос «получениеИнформацииОКлиенте»	4
Внешний запрос «получениеИнформацииОБанкомате»	3
Внешний запрос «получениеИнформацииОбОператореСотовойСвязи»	3

Нескорректированный размер ПС ($V_{\phi m}$) в функциональных точках определяется как сумма количества функциональных точек, соответствующих элементам данных и транзакционным функциям. $V_{\phi m} = 169$.

Шаг 6. Определяем размер ПС в строках исходного кода:

$$V = V_{\phi m} * 125 = 21125$$

Ниже представлен отчет по расчету трудоемкости.

РАСЧЕТ ТРУДОЕМКОСТИ С ИСПОЛЬЗОВАНИЕМ ПО АРМ

Расчет трудоемкости разработки ПС (метод ФБО)

Параметры расчета

Параметр	Значение
Имя расчета	Программное средство мониторинга финансовых операций, выполняемых через банкомат
Тип расчета	Расчет трудоемкости разработки ПС (метод ФБО)
Версия модели	2.5.1.0
Фонд времени разработчика [дней/мес.]	21
Тип ограничения	По продолжительности этапа [мес.]
Тип структуры программного средства	ПС с набором библиотек (=1.00)
Технологии взаимодействия модулей	Оригинальный протокол взаимодействия (sockets) (=0.80)

Коэффициенты уровня расчета

Коэффициент	Уровень значения
Степень новизны ПС	Принципиально новое ПС, не имеющее доступных аналогов на прежнем типе ТС/ОС (=1.06)
Требования к надежности	Сбои ПС приводят к потерям рабочей информации (=1.06)
Требования к производительности	Повышенные требования к производительности (=1.12)
Информативность документации	Повышенный объем для жизненного цикла данного ПС (=1.07)
Опыт разработки ПС подобного типа	Более 5 разработанных ПС (компонентов АС) (=0.81)
Влияние сроков работ на трудоемкость	Равно или более 100 (=1.00)

Коэффициенты уровня подпроцесса

Подпроцесс	Коэффициент	Уровень значения
Анализ требований	Уровень квалификации аналитиков	Высокий (=0.87)
Анализ требований	Опыт работы аналитиков в данной предметной области	5 разработанных ПС (компонентов АС) (=0.87)
Проектирование	Уровень квалификации проектировщиков	Высокий (=0.92)
Проектирование	Опыт работы проектировщиков с используемыми средствами	5 разработанных ПС (компонентов АС) (=0.92)

	проектирования	
Проектирование	Используемые средства проектирования	Средства проектирования БД (BPWIN, ERWIN/ERX, Designer 2000, DataDirect Explorer, Oracle Designer, Silverrun RBM и BPM, Together J) (=1.03)
Программирование	Уровень квалификации программистов	Высокий (=0.95)
Программирование	Используемая среда разработки	Интегрированные среды разработки (Visual Studio) (=1.00)
Тестирование	Уровень квалификации персонала, осуществляющего тестирование	Высокий (=0.89)
Тестирование	Используемые средства тестирования	Автоматизированные средства тестирования использовались (=0.76)
Тестирование	Размер БД	$10 \leq D/P < 100$ (=1.00)
Ввод в действие ПС	Уровень квалификации персонала, осуществляющего ввод в действие ПС	Высокий (=0.94)

Дерево элементов

Элемент	Описание	Тип	Количество элементов данных (DET)	Количество записей (RET)	Объем кода на условном языке [строк]	Объем [ФТ]
Клиент		Внутренний логический файл	7	1	875	7
Банкомат		Внутренний логический файл	4	1	875	7
ОператорСотовойСвязи		Внутренний логический файл	4	1	875	7
Операция		Внутренний логический файл	8	1	875	7
ОперацияОплатыУслуг		Внутренний логический файл	3	1	875	7
Счет		Внутренний логический файл	2	1	875	7
ТипОперации		Внутренний логический файл	3	1	875	7
ДенежныйОстаток		Внутренний логический файл	3	1	875	7
ДанныеОбОператореСотовойСвязи		Внешний логический файл	3	1	625	5
ДанныеОПополненииДенежныхСредствБанкомата		Внешний логический файл	3	1	625	5
ДанныеОБанкомате		Внешний логический файл	3	1	625	5
РезультатыАнализа		Внешний логический файл	4	1	625	5
ДанныеОСостоянииСчета		Внешний	2	1	625	5

		логический файл				
ДанныеЗапросаНаСнятиеНаличных		Внешний логический файл	2	1	625	5
ДанныеЗапросаОбОплатеУслугОператораСотовойСвязи		Внешний логический файл	4	1	625	5
ДанныеОбОплатеУслугОператораСотовойСвязи		Внешний логический файл	2	1	625	5
Отчет, ОтчетОбОплате, ОтчетОСнятииНаличных, ОтчетОЗапросеБаланса		Внешний логический файл	10	3	625	5
ДоговорСКлиентом, ДанныеОКлиенте		Внешний логический файл	7	1	625	5
вводДанныхОЗачисленииСредствНаСчетКлиента		Единица входной информации	3	2	375	3
вводИнформацииОНовомКлиенте		Единица входной информации	7	2	500	4
удалениеИнформацииОКлиенте		Единица входной информации	2	1	375	3
вводИнформацииОНовомБанкомате		Единица входной информации	4	2	375	3
удалениеИнформацииОБанкомате		Единица входной информации	2	1	375	3
вводИнформацииОНовомОператореСотовойСвязи		Единица входной информации	4	2	375	3
удалениеИнформацииОбОператореСотовойСвязи		Единица входной информации	2	1	375	3
вводИнформацииОПополненииДенежныхСредств		Единица входной информации	4	2	375	3
снятиеНаличных		Единица входной информации	6	5	750	6
оплатаУслугОператораСотовойСвязи		Единица входной информации	7	8	750	6

формированиеИВыводОтчета		Единица выходной информации	11	5	875	7
формированиеИВыводРезультатов Анализа		Единица выходной информации	5	3	500	4
получениеИнформацииОСостоянииСчета		Единица выходной информации	4	5	625	5
получениеИнформацииОКлиенте		Внешний запрос	7	2	500	4
получениеИнформацииОБанкомате		Внешний запрос	4	2	375	3
получениеИнформацииОбОператореСотовойСвязи		Внешний запрос	4	2	375	3
Общий объем ПС:					21125	169

Расчет трудоемкости подпроцессов

Подпроцесс	Трудоемкость [чел.дн.]	Кол-во разработчиков [чел]	Продолжительность [мес]/[дн]
Анализ требований	4.48	0.11	2.00/42.00
Проектирование	61.90	0.98	3.00/63.00
Программирование	444.04	5.29	4.00/84.00
Тестирование	28.01	0.67	2.00/42.00
Ввод в действие ПС	5.56	0.26	1.00/21.00
Всего	543.99		12.00/252.00
Базовая трудоемкость	591.66		