**ЛАБОРАТОРНАЯ РАБОТА № 2**

**Тема:** Разработка модульной структуры проекта (диаграммы модулей)

**Цель:** Изучить процесс разработки модульной структуры программного обеспечения, осуществляемого с помощью структурных карт Константайна и Джексона

**Модульная структура проекта**

Структуру программной системы составляют модули, которые в любом языке программирования имеют следующие общие свойства:

* модуль имеет имя, по которому к нему можно обращаться как к единому фрагменту;
* модуль состоит из множества операторов языка программирования, записанных последовательно;
* модуль может принимать и/или передавать данные как параметры в вызывающей последовательности или связывать данные через фиксированные ячейки и общие области.

Среди множества модулей различают:

* головной модуль - управляет запуском программного продукта (существует в единственном числе);
* управляющий модуль - обеспечивает вызов других модулей на обработку;
* рабочие модули - выполняют функции обработки;
* сервисные модули и библиотеки, утилиты - осуществляют обслуживающие функции.

При этом модульная структура программы должна, помимо картинки, включать **спецификацию программного модуля** – правила использования модуля.

***Спецификация программного модуля*** должна содержать:

* синтаксическую спецификацию его входов (имя модуля, типы передаваемых ему параметров, типы возвращаемых результатов, синтаксис обращения к любому ему входов)
* функциональную спецификацию (описание семантики функций, выполняемых модулем по каждому из его входов).

**Структурные** **карты**

Техника **структурных карт** (схем) используется на фазе проектирования для того, чтобы продемонстрировать, каким образом системные требования будут отражаться комбинацией программных структур. При этом наиболее часто применяются две техники:

* структурные карты Константайна (Constantine), предназначенные для описания отношений между модулями;
* структурные карты Джексона (Jackson), предназначенные для описания внутренней структуры модулей.

Структурные **карты Константайна** являются моделью отношений иерархии между программными модулями. Узлы структурных карт соответствуют модулям и областям данных, потоки изображают межмодульные связи. На диаграмме специальными узлами изображаются циклические и условные вызовы модулей, а потоки проходят через эти специальные узлы. Потоки, изображающие межмодульные связи по данным и управлению, также изображаются на диаграмме специальными узлами, а стрелками указываются направления потоков.

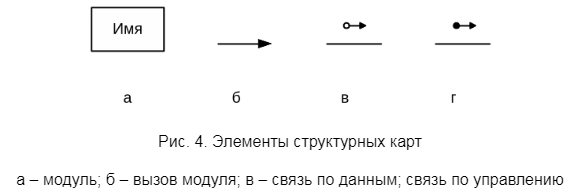


Рис. 1. Элементы структурных карт

а – любой модуль, б – поток - вызов модуля, в – связь по управлению,

г – связь по данным

Модуль является базовым элементом структурной карты. Различают следующие типы модулей (рис. 2):

* модуль (рис. 2, *а*);
* подсистема — детализированный модуль или программа. Может использоваться повторно любое число раз (рис. 2, *б);*
* библиотека — совокупность подпрограмм, размещенных в модуле отдельно от данной системы (рис. 2, *в);*
* область данных — описывает модули, содержащие исключительно области глобальных/распределенных данных (рис. 2, *г).*

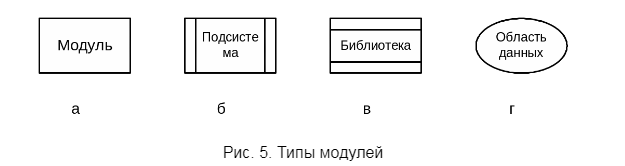
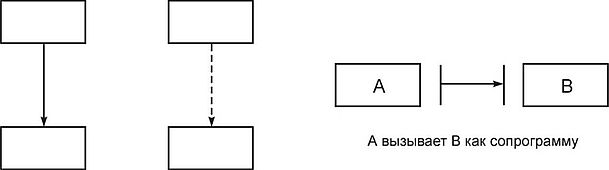


Рис. 2. Типы модулей

Отдельные части программной системы (программы, подпрограммы) могут вызываться последовательно, параллельно или как сопрограммы (рис. 3).

Чаще всего используют *последовательный*вызов, при котором модули, передав управление, ожидают завершения выполнения вызванной программы или подпрограммы, чтобы продолжить прерванную обработку.

Под *параллельным*вызовом понимают распараллеливание вычислений на нескольких вычислителях, когда при активизации другого процесса данный процесс продолжает работу*.*На однопроцессорных компьютерах в мультипрограммных средах в этом случае начинается попеременное выполнение соответствующих программ. Параллельные процессы бывают синхронные и асинхронные. Для синхронных процессов определяют точки синхронизации - моменты времени, когда производится обмен информацией

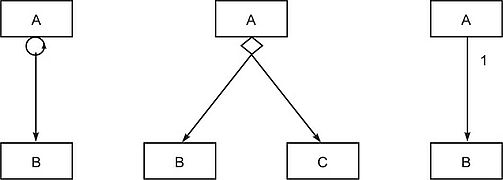


Последовательный вызов Параллельный вызов

Рис. 3. Типы вызовов модулей

Для моделирования условных и циклических вызовов применяются следующие узлы (рис. 4):

* условный узел применяется для моделирования конструкций IF-THEN-ELSE (на диаграмме из узла выходят два потока) и IF-THEN (из узла выходит один поток). Условный узел изображается в виде ромба, потоки — альтернативные вызовы — изображаются выходящими из него;
* итерационный узел используется для того, чтобы показать, что вызываемый модуль выполняется в цикле. Он изображается полуокружностью со стрелкой с выходящими из него потоками.



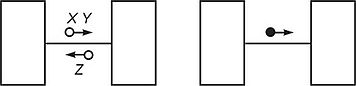
*а б в*

Рис. 4. Условные и циклические вызовы модулей:

*а* — циклический; *б* — условный; *в* — однократный

Если необходимо показать, что подчиненный модуль вызывается однократно, это осуществляется указанием цифры «1» рядом со стрелкой, обозначающей вызов модуля-наследника.

Связи по данным и управлению между модулями (передаваемые как параметры) обозначают стрелками, параллельными дуге вызова, которые показывают направления связей (рис. 5).



*а б*

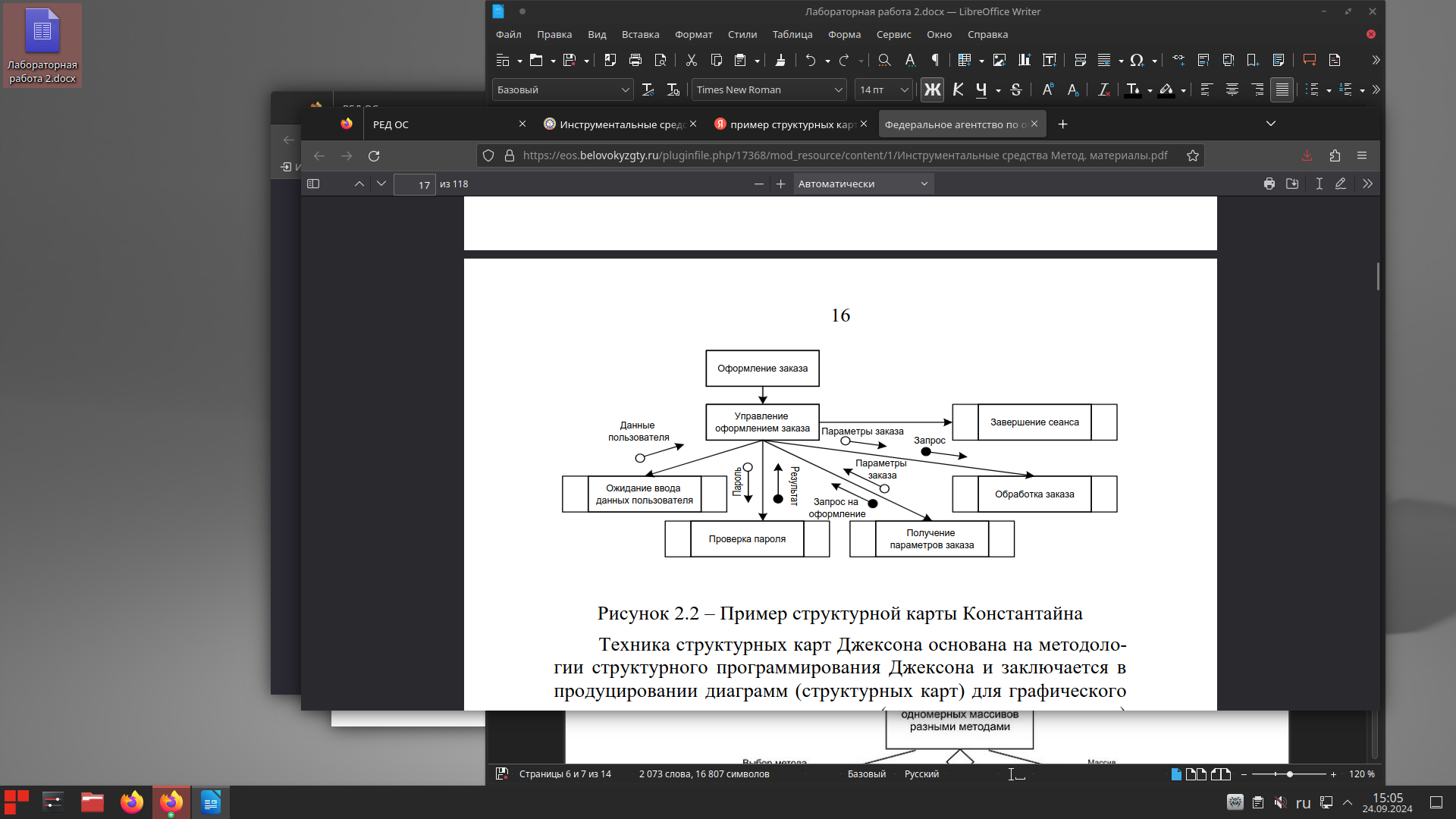
Рис. 5. Связи**:** *а* — по данным; *б* — по управлению

Чтобы добиться декомпозиции на модули максимальной прочности и минимального сцепления, необходимо спроектировать модульную структуру в виде дерева, в том числе и со сросшимися ветвями. В узлах такого дерева размещаются программные модули, а направленные дуги (стрелки) показывают статическую подчинённость модулей, т.е. каждая дуга показывает, что в тексте модуля, из которого она исходит, имеется ссылка на модуль, в который она входит. Взаимодействие между модулями может осуществляться как непосредственно, так и с передачей данных и параметров управления. При графическом представлении карты передача данных изображается в форме стрелки с незакрашенным кружком на конце, а передача управления – в виде стрелки с закрашенным кружком на конце.

В качестве примера рассмотрим структурную карту Константайна, описывающую процесс оформления некоторого заказа (рис. 6).

Базовым модулем является процедура оформления заказа, вызывающая, соответственно, процесс управления оформлением заказа. Из последнего модуля, в свою очередь, последовательно могут быть вызваны последующие модули, описанные как подсистемы. Практически все модули передают или принимают данные или параметры управления. Так, например, процесс «Проверка пароля» принимает на вход данные о введенном пользователем пароле, а в ответ на основании результатов проверки инициируют породивший его процесс управления оформлением заказа, переводя его на новый уровень.

Рис. 6. Пример структурной карты Константайна



Техника **структурных карт Джексона** основана на методологии структурного программирования Джексона и заключается в продуцировании диаграмм (структурных карт) для графического иллюстрирования внутримодульных (а иногда и межмодульных) связей и документирования проекта архитектуры системы ПО. При этом техника позволяет осуществлять проектирование нижнего уровня структуры ПО и на этом этапе является близкой к традиционным блок-схемам.

По аналогии со структурными картами Константайна диаграмма Джексона может включать объекты следующих типов:

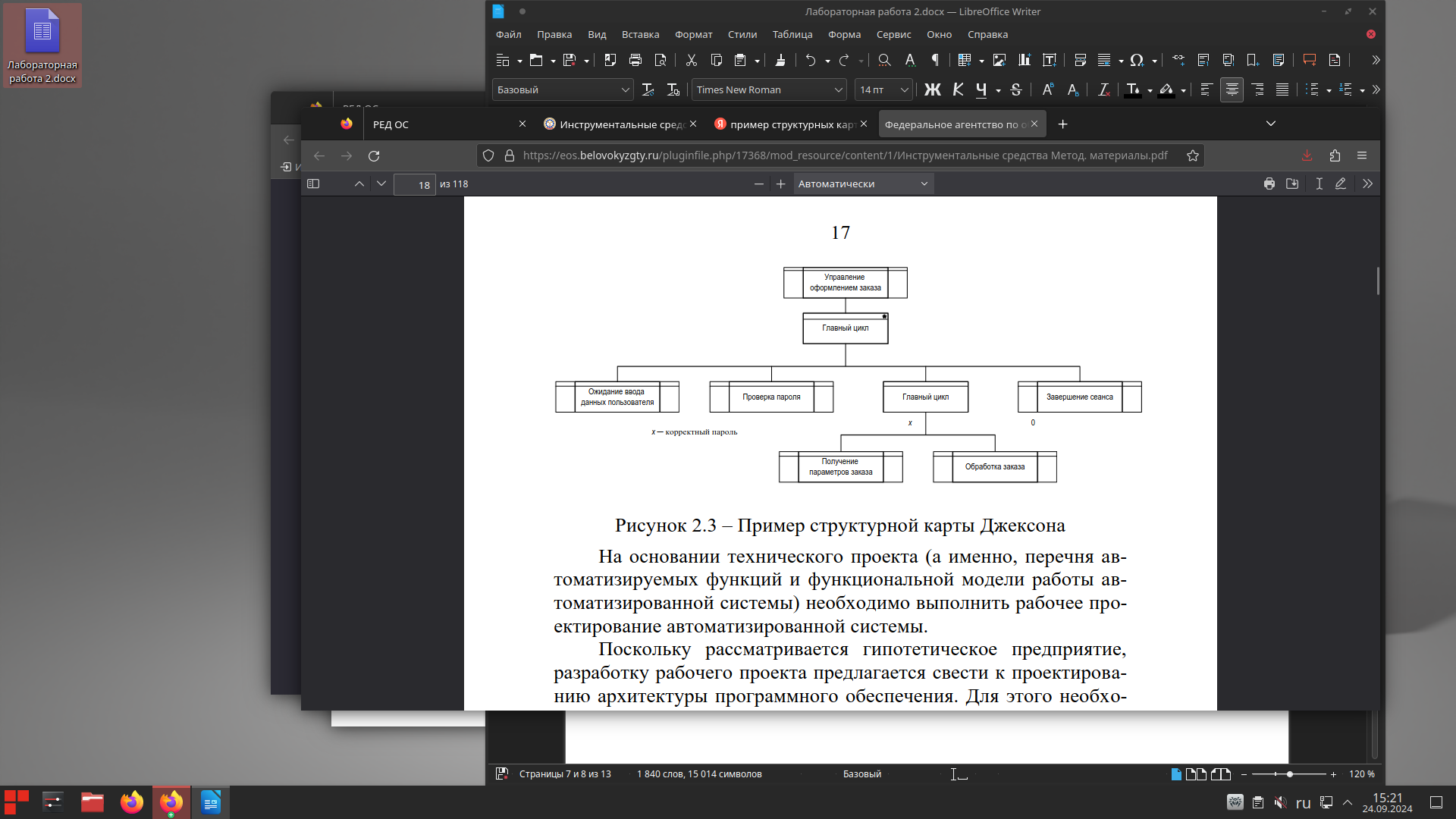
– структурный блок (базовая компонента методологии) представляет частную функцию или блок кодов с одним входом и одним выходом;

– процедурный блок является специальным видом структурного блока, представляющим вызов ранее определенной процедуры;

– библиотечный блок аналогичен процедурному и представляет вызов библиотечного модуля.

Для разделения блоков карт Константайна и Джексона графические блоки последней нотации целесообразно сопроводить горизонтальной линии, размещаемой под верхней гранью блока. Пример использования структурных карт Джексона приведен на рис.7. Он аналогичен рассмотренному выше примеру с картой Константайна (см. рис. 6).

Рис. 7. Пример структурной карты Джексона



**Пример 1.** Система «Кредитный отдел».

В составе программного обеспечения можно выделить следующие программные модули: Головной модуль, Модуль управления устройством считывания кредитной кары, Модуль аутентификации и Модуль получения и обработки запроса на обслуживание. Кроме этого в состав ПО необходимо включить модуль данных кредитной карты.

Основной функцией Головного модуля является организация общего управления поведением подсистемы и выполняет вызов всех остальных программных модулей.

Модуль управления устройством считывания кредитной карты выполняет функции, связанные с обработкой кредитной карты: ввод, считывание хранящейся на ней информации, удаление.

Модуль аутентификации выдает сообщение клиенту на ввод ключевых данных, выполняет получение пароля и проверку его правильности.

Модуль получения и обработки запроса на обслуживание выполняет следующие функции: Получение запроса на обслуживание и проверка возможности его исполнения, Обработка запроса на обслуживание, включающая такие действия как:

* обработка внутренней банковской документации по клиенту;
* распечатка баланса клиента;
* выдача наличных денег и информирование компьютера банка об изъятых из банка деньгах;
* распечатка операции клиента.

Головной модуль обращается к модулям управления устройством считывания кредитной карты, аутентификации и получения и обработки запроса на обслуживание.

Вызов указанных модулей осуществляется согласно внутренней логики головного модуля, реализующей следующий сценарий: при инициации действий со стороны клиента головной модуль, вызывает модуль управления устройством считывания кредитной карты для ее ввода и считывания с нее информации. После завершения считывания управление возвращается головному модулю, который затем обращается к модулю аутентификации.

Модуль аутентификации проверяет подлинность клиента и вместе с результатом этой проверки возвращает управление головному модулю. В зависимости от результатов аутентификации головной модуль либо вызывает модуль управления устройством считывания для удаления кредитной карты, либо обращается к модулю получения и обработки запроса на обслуживание для предоставления требуемого сервиса. Если осуществляется вызов получения и обработки запроса на обслуживание, то после завершения его работы головной модуль обращается к модулю управления устройством считывания для удаления кредитной карты.

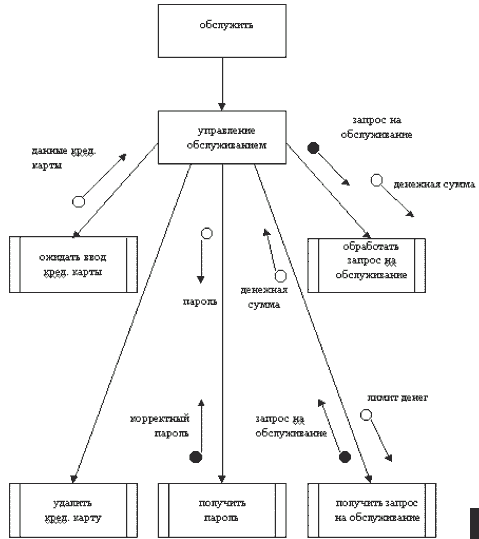


Рис. 8. Фрагмент структурной карты с межмодульными отношениями в банковской системе

Обмен данными между программными модулями осуществляется через общую область памяти, в которую модуль управления устройством считывания помещает данные о пароле (Parol), атрибуты клиента (Client Attributes) и лимит денег на счету (Limit of money).

Модуль аутентификации получает из этой общей области памяти сведения о пароле и возвращает в головной модуль управляющий параметр Autentification flag, содержащий результат аутентификации.

Модуль получения и обработки запроса на обслуживание для своей работы получает из общей области памяти атрибуты клиента и лимит денег на счету.

**Задания -** Разработать модульную структуру заданного программного модуля.

**Технология выполнения:**

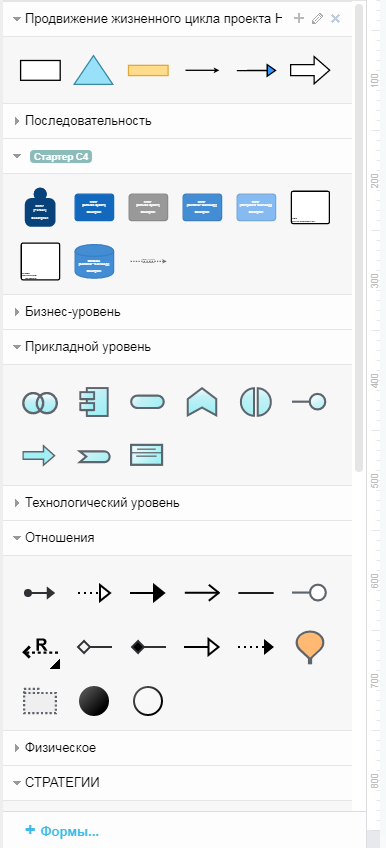
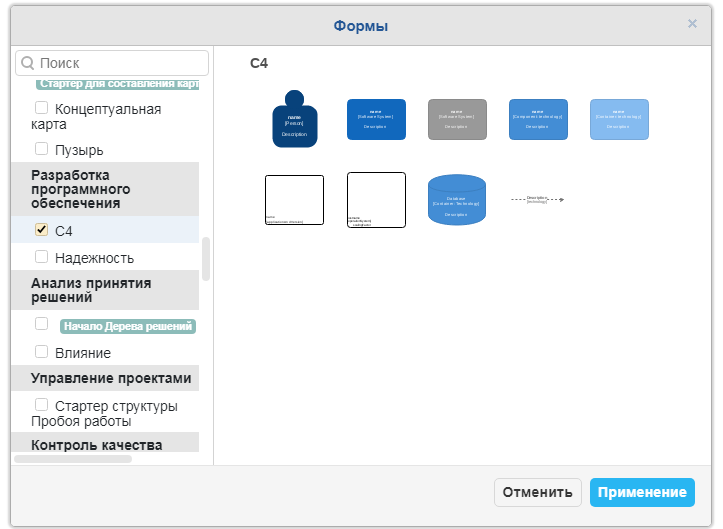
1 Запустить программу Visual Paradigm Online

([https](https://online.visual-paradigm.com/drive/" \l "diagramlist:proj=0&dashboard)://online.visual-paradigm.com/drive/#diagramlist:proj=0&dashboard )

2 Ознакомиться с интерфейсом программы

3 Выбрать шаблоны группы Диаграммы: SDL Diagram, ArchiMate Tutorial   
Starter, Functional Decomposition Diagram

4 Добавить необходимы формы



5 Составить структурные карты Константайна и Джексона

6 Составить спецификацию программного модуля.

**Варианты**

1. Разработать программный модуль «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.
2. Разработать программный модуль «Личные дела студентов». Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.
3. Разработать программный модуль «Решение комбинаторно-оптимизационных задач». Модуль должен содержать алгоритмы поиска цикла минимальной длины (задача коммивояжера), поиска кратчайшего пути и поиска минимального связывающего дерева.

4. Разработать приложение Windows «Органайзер». Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц, и организаций, а также расписания, встреч и др. Приложение предназначено для любых пользователей компьютера.

1. Разработать приложение Windows «Калькулятор». Приложение предназначено для любых пользователей и должно содержать все арифметические операции (с соблюдением приоритетов) и желательно (но не обязательно) несколько математических функций.
2. Разработать программный модуль «Кафедра», содержащий сведения о сотрудниках кафедры (ФИО, должность, ученая степень, дисциплины, нагрузка, общественная работа, совместительство и др.). Модуль предназначен для использования сотрудниками отдела кадров и деканата.
3. Разработать программный модуль «Лаборатория», содержащий сведения о сотрудниках лаборатории (ФИО, пол, возраст, семейное положение, наличие детей, должность, ученая степень). Модуль предназначен для использования сотрудниками профкома и отдела кадров.
4. Разработать программный модуль «Автосервис». При записи на обслуживание заполняется заявка, в которой указываются ФИО владельца, марка автомобиля, вид работы, дата приема заказа и стоимость ремонта. После выполнения работ распечатывается квитанция.
5. Разработать программный модуль «Учет нарушений правил дорожного движения». Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.
6. Разработать программный модуль «Учет успеваемости студентов». В базе содержится информация об обучающемся: год поступления, группа, список изучаемых предметов, успеваемости по каждой дисциплине. Обеспечивать удобный поиск заданного студента по ФИО. Формировать список неуспевающих по каждой дисциплине. Выводить статистические отчеты по итогам сессии. Разработанная программа должна иметь удобный интерфейс, интуитивно-понятные связи между диалогами ввода информации.
7. Разработать программный модуль «Картотека агентства недвижимости», предназначенный для использования работниками агентства. В базе содержатся сведения о квартирах (количество комнат, этаж, метраж и др.). При поступлении заявки на обмен (куплю, продажу) производится поиск подходящего варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.
8. Разработать программный модуль «Картотека абонентов АТС». Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и повременной). Считается, что повременная оплата местных телефонных разговоров уже введена.
9. Разработать программный модуль «Авиакасса», содержащий сведения о наличии свободных мест на авиамаршруты. В базе должны содержаться сведения о номере рейса, экипаже, типе самолета, дате и времени вылета, а также стоимости авиабилетов (разного класса). При поступлении заявки на билеты программа производит поиск подходящего рейса.
10. Разработать программный модуль «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена). Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.
11. Разработать программный модуль «Автостоянка». В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.
12. Разработать программный модуль «Кадровое агентство», содержащий сведения о вакансиях и резюме. Программный модуль предназначен как для поиска сотрудника, отвечающего требованиям руководителей фирмы, так и для поиска подходящей работы.

Примечание. При разработке программы не ограничиваться функциями, приведенными в варианте, добавить несколько своих функций

**Контрольные вопросы:**

1. Что такое рабочее проектирование?
2. На основании результатов каких этапов разработки выполняется рабочее проектирование?
3. Для чего используются структурные карты?
4. Какие методики проектирования структурных карт существуют?
5. Что такое спецификация программного модуля?
6. В чем заключается методика Константайна?
7. Чем отличаются структурные карты Константайна от структурных карт Джексона?