

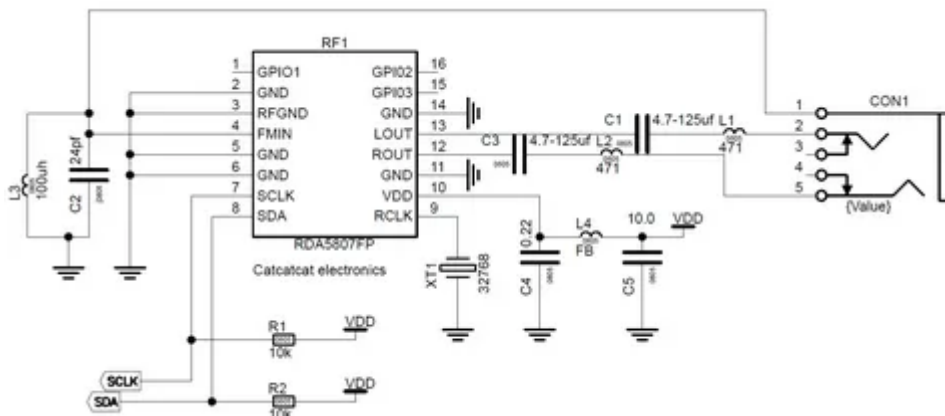
Задание: расписать определения для вариантов записи графической записи

1. схемы ISO 5807:1985

Блок-схема — это диаграмма, на которой обычно представлен процесс, система или компьютерный алгоритм и которая используется для документирования, планирования, уточнения или визуализации многоэтапного рабочего процесса.

Благодаря блок-схемам вы можете определить цели и масштабы рабочего процесса, а также установить необходимые задачи в хронологическом порядке.

Пример:

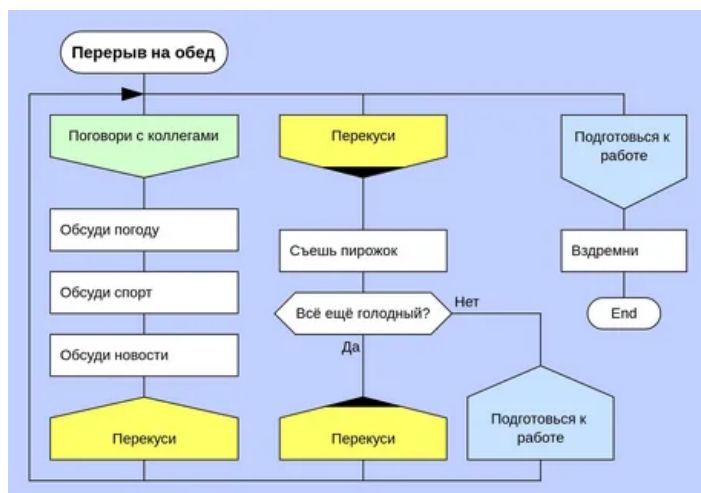


2. Дракон-схемы:

Дракон-схемы есть не что иное, как правильно составленные блок-схемы. Язык ДРАКОН строится на основе блок-схем с целью их улучшения. Использование эргономичных правил ДРАКОНа позволяет упорядочить графический чертёж алгоритма и обеспечить более эффективное восприятие блок-схемы человеком.

3. диаграммы деятельности UML
4. диаграммы потоков данных (DFD).
5. запись алгоритма с помощью Р-схемы.
6. для программирования микроконтроллеров (SFC, FBD).
7. для описания бизнес-процессов (IDEF0, BPMN).

Пример:



3. диаграмма деятельности в UML

По сути, это блок-схема, представляющая поток от одного действия к другому. Эту деятельность можно охарактеризовать как орегация системы. Основная цель диаграмм деятельности — отразить динамическое поведение системы. Ее также называют объектно-ориентированной блок-схемой.

Эта [Диаграмма UML](#) фокусируется на выполнении и потоке поведения системы, а не на реализации. Диаграммы деятельности состоят из действий, состоящих из действий, применимых к технологии поведенческого моделирования.

Компоненты диаграммы деятельности

Активности

Это поведение, которое разделено на одно или несколько действий. Действия представляют собой сеть узлов, соединенных ребрами. Это могут быть узлы действий, узлы управления или узлы объектов. Узлы действий представляют собой некоторые действия. Узлы управления представляют поток управления действием. Узлы объектов используются для описания объектов, используемых внутри действия. Края используются для отображения пути или потока выполнения. Действия начинаются в начальном узле и заканчиваются в конечном узле.

Раздел активности/дорожка для плавания

Раздел действий или дорожка — это высокоуровневая группировка набора связанных действий. Один раздел может относиться ко многим вещам, таким как классы, варианты использования, компоненты или интерфейсы.

Если раздел невозможно отобразить четко, имя раздела пишется поверх названия действия.

Узлы разветвления и соединения

Используя узлы разветвления и соединения, можно создавать параллельные потоки внутри действия. Узел-вилка имеет одно входящее ребро и множество исходящих ребер. Это похоже на слишком много параметров решения. Когда данные достигают входящего ребра, они дублируются и распределяются по множеству исходящих ребер одновременно. Необычно. Один входящий поток делится на несколько параллельных потоков.

Узел соединения противоположен узлу-разветвлению, поскольку он имеет множество входящих ребер и одно выходящее ребро. Он выполняет логическое И органа всех входящих ребрах. Это поможет вам синхронизировать входной поток по одному выходному фронту.

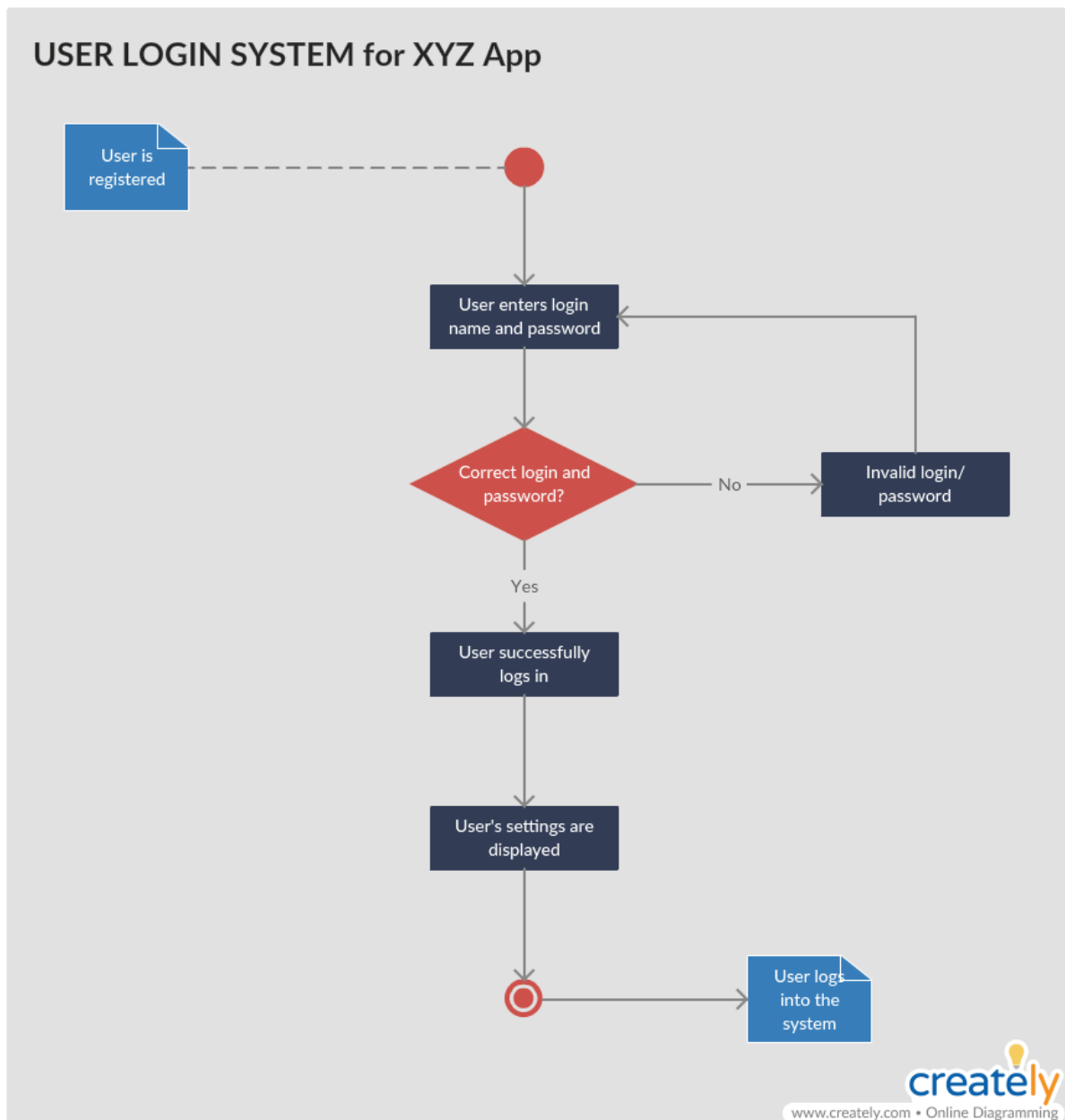
Пальцы

Диаграмма деятельности с множеством потоков становится очень сложной и запутанной.

Булавки используются для прояснения вещей. Он обеспечивает способ управления потоком выполнения действий путем сортировки всех потоков и очистки ненужных деталей. Это узел объекта, который представляет один вход или выход действия.

И входные, и выходные контакты имеют ровно один край.

Пример:



4. DFD (Data Flow Diagram) Диаграммы

Диаграмма отображает потоки данных между системами, базами данных. Ключевыми элементами являются входные/выходные данные, системы, точки хранения и сбора данных.

Выделяют 4 элемента в диаграмме:

1. Процесс.

Процессы, при которых идет изменение потока данных (обработка, трансформация и др. изменения). Процесс как и в других диаграммах обычно прописывается с помощью глагола, например: “Отправка заполненной формы”.

2. Внешняя сущность.

Сущность (объект), которая получает или отправляет данные при взаимодействии с описанным процессом.

3. Хранилище данных.

Все хранилища данных или отдельные файлы, которые хранят исходные или выходные данные, а также все промежуточные хранилища.

4. Поток данных.





Поток данных, который отображает направление и сами данные, которые перемещаются между внешними сущностями и хранилищами данных с помощью процессов.




5. Р-Схемы алгоритмов и программ

Настоящий стандарт распространяется на алгоритмы, программы, данные и процессы для вычислительных машин, комплексов, автоматизированных и систем обработки информации независимо от их назначения области применения.

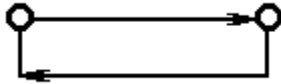
Стандарт устанавливает условные графические обозначения элементов структур Р-схем, а также правила их выполнения автоматическим и (или) ручным способами.

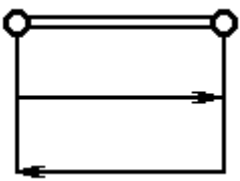
Р-схема (R-chart) - нагруженный по дугам ориентированный граф, изображаемый с помощью вертикальных и горизонтальных линий и состоящий из структур (подграфов), каждая из которых имеет только один вход и один выход.

Наименование	Обозначение и его содержание	Функция
1. Вершина	 Окружность диаметром не менее 2 мм	Вершина Р-схемы
2. Вершина специальная	 Левая и правая круглые скобки на расстоянии не менее 1 мм	Выделение вершины Р-схемы
3. Дуга	а)  б)  Горизонтальная линия со стрелкой а) справа б) слева	Направленное соединение двух вершин Р-схемы

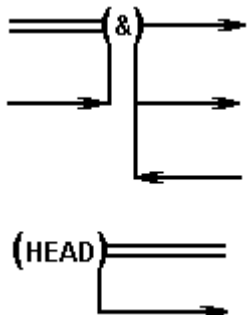
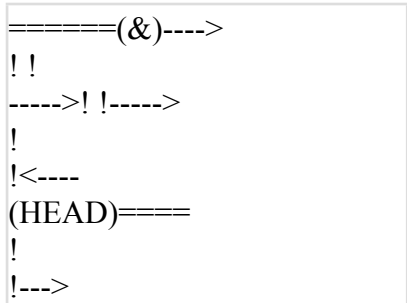
4. Дуга специальная	 Две горизонтальные линии, отстоящие друг от друга на расстоянии 0,8 -- 3 мм	Специальное соединение двух вершин Р-схемы
5. Линия соединительная	 Вертикальная линия	Соединение по вертикали указанных выше элементов Р-схемы
6. Комментарий	 Пунктирная вертикальная (вверх или вниз) и горизонтальная (влево или вправо) линии, оканчивающиеся квадратной скобкой, за которой следует текст комментария.	Связь между элементом Р-схемы и текстом комме

Структура

Наименование	Обозначение и его содержание	Функция
1. Структура базовая	 Две вершины, соединенные одной и более дугами любого направления и в любом сочетании.	Последовательность переходов между вершинами в соответствии с направлениями дуг.

2. Структура специальная	 <p>Две вершины, соединенные специальной дугой или специальной дугой и любым числом дуг любого направления и в любом сочетании.</p>	Последовательность переходов между вершинами, осуществляемых специальным образом.
--------------------------	--	---

Пример

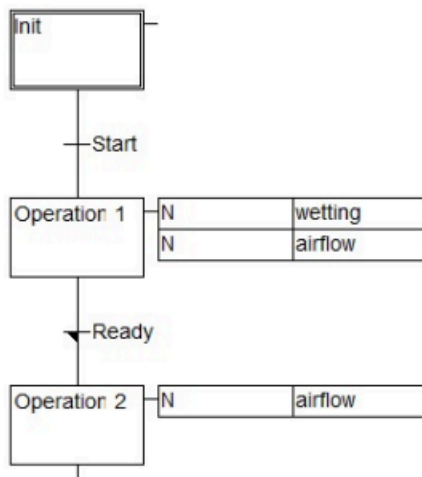
Наименование	Выполнение от руки или автоматизированным способом на графических устройствах ввода-вывода	Выполнение автоматизированным способом на алфавитно-цифровых устройствах ввода-вывода
1. Надпись внутри вершины специальной		

7. для программирования микроконтроллеров (SFC, FBD).

Язык последовательных функциональных схем SFC (Sequential Function Charts) позволяет формулировать логику программы на основе чередующихся процедурных шагов и условных переходов, а также описывать последовательно-параллельные задачи в понятной и наглядной форме [2; 11]. Строго говоря, SFC не является языком программирования. Это средство для структурирования прикладных программ. Он позволяет программировать последовательность выполнения действий системой 60 управления, когда эти действия должны быть выполнены в заданные моменты времени или при наступлении некоторых событий. Язык SFC построен по принципу, близкому к

концепции конечного автомата. Основными элементами программы на языке SFC являются шаги и условия переходов (рисунок 3.8). Шаги в SFC-программе изображаются в виде прямоугольников. Функции (действия), выполняемые шагом, в SFC-программе не отражаются. Они задаются в отдельном окне среды программирования. Назначение шага определяет его название или, если это требуется, краткий комментарий.

Пример:

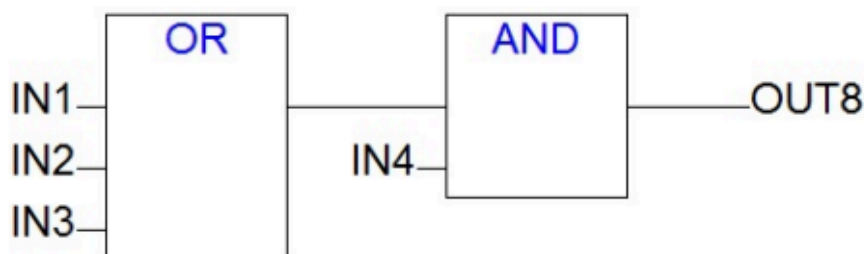


Язык функциональных блочных диаграмм FBD (Function Block Diagram) – это графический язык программирования [2; 11]. Программа, созданная на данном языке, является совокупностью функциональных блоков (ФБ), у которых входы и выходы соединены линиями связи. Связи между входами и выходами ФБ являются, по сути, переменными программы. С их помощью выполняется передача информации между ФБ. ФБ представляет собой программно реализованный компонент, представленный в программе как «черный ящик». Каждый ФБ выполняет определенную функцию, связанную с обработкой данных. В FBD-программе ФБ обычно изображают графически в виде прямоугольников с входами и выходами. Внутри прямоугольников указывают условное обозначение функции, выполняемой ФБ

Используемые символы:

Обозначение	Символ	Действие
ADD	+	Сложение
SUB	-	Вычитание
MUL	*	Умножение
DIV	/	Деление
MOD		Деление по модулю
EXPT		Возведение в степень

Пример:



8. диаграмма Нэсси-Шнейдермана

Схемы Нэсси-Шнейдермана – это схемы, иллюстрирующие структуру передач управления внутри модуля с помощью вложенных друг в друга блоков.

Схемы используются для изображения структурированных схем и позволяют уменьшить громоздкость схем за счёт отсутствия явного указания линий перехода по управлению.

Схемы Нэсси-Шнейдермана называют ещё структурограммами. Изображение основных элементов структурного программирования в схемах Нэсси-Шнейдермана организовано следующим образом. Каждый блок имеет форму прямоугольника и может быть вписан в любой внутренний прямоугольник любого другого блока. Информация в блоках записывается по тем же правилам, что и в структурированных схемах алгоритмов (на естественном языке или языке математических формул).

Пример:

