

**МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН**

**ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АБУ РАЙХАНА БЕРУНИ**

На правах рукописи

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на получение степени магистра
Хасанова Жалола Бахромовича

Разработка алгоритмов и программного обеспечения обработки
биологических сигналов на примере реографии в Delphi

по специальности 5A521512 «Медицинские приборы
и измерительные системы»

Руководитель к.т.н., доц.

Васильева Стелла Александровна

Ташкент 2011 г.

СОДЕРЖАНИЕ

Введение..	
ГЛАВА 1. ОСНОВНЫЕ ПОЛОЖЕНИЯ МЕТОДА РЕОГРАФИИ	
1.1. Общие понятия метода реографии	
1.2. Методики использования реографа при исследовании гемодинамических показателей	
1.3. Методика регистрации реограммы и её анализ.	
1.4. Виды реографии.	
Выводы к главе 1.	
ГЛАВА 2. МАТЕМАТИЧЕСКАЯ ОБРАБОТКА ОБРАБОТКИ КАРДИОГРАФИЧЕСКИХ И РЕОГРАФИЧЕСКИХ ДАННЫХ.	
2.1. Функции нахождения глобального максимума и минимума сигналов	
2.2. Функция пересчёта координат	
2.3. Построение графиков по данным из массива	
2.4. Нахождение R-зубцов на ЭКГ.	
2.5. Определение точек начала изгнания крови из левого желудочка сердца.	
Выводы к главе 2	
ГЛАВА 3. ПРОГРАММНЫЕ МЕТОДЫ ИССЛЕДОВАНИЯ РЕОГРАФИЧЕСКИХ СИГНАЛОВ.	
3.1. Формальное описание программы	
3.2. Работа с файлами	
3.3. Обработка исключительных ситуаций при открытии файла	
3.4. Работа с канвой принтера	
Выводы к главе 3	
Заключение	
Список использованной литературы	
Приложение.	

Введение

В настоящее время одними из самых распространенных являются сердечно-сосудистые заболевания. Поэтому особенно важна ранняя диагностика заболеваний сердечно-сосудистой системы. Особое значение приобретает изучение центральной гемодинамики (кровообращения) при таких экстремальных ситуациях, как гипертонический криз, кардиогенный шок, инфаркт миокарда и т.д. При этом ведущая роль принадлежит гемодинамическим показателям, таким как ударный объем и общее периферическое сопротивление.

Системы регистрации, обработки и анализа физиологической информации человека широко применяются в медицинской практике, это обусловлено их объективностью, информативностью и хорошей воспроизводимостью получаемых результатов. С развитием вычислительной техники, повсеместно осуществляется переход на компьютерную обработку и хранение медицинской информации, совершенствующиеся технологии повышают диагностические возможности медицинской аппаратуры.

Важнейшей задачей системы здравоохранения является сохранение здоровья граждан, причем более эффективно ориентировать систему здравоохранения на профилактику заболеваний, а не на лечение уже развившегося недуга. Поэтому особое значение принимает разработка методик оценки состояния здоровья.

Одним из важных показателей здоровья организма является его способность адаптироваться к изменчивым условиям внешней среды. Любое воздействие среды на организм вызывает ответную реакцию системы регуляции, задача которой за счёт внутренних резервов организма, обеспечить гомеостатический режим его деятельности [1].

Система кровообращения ответственна за адаптацию организма к большому числу различных факторов внешней среды. В большинстве случаев систему кровообращения можно рассматривать как индикатор

состояния функциональных резервов целого организма. Дыхательная система является функционально связанной с системой кровообращения, и, учитывая способность системы регуляции компенсировать нагрузку на связанные системы, совместный анализ кровообращения и дыхания (кардио-респираторной системы) позволяет наиболее полно оценивать состояние организма. При этом важно обеспечить синхронную связь между параллельно регистрируемыми физиологическими сигналами этих систем.

Основной задачей при исследовании работы кровеносной системы человека является определение степени отклонения от нормы. Правильность функционирования системы определяется исходя из сравнения её цикла работы с некоторым нормальным циклом. Существующие системы функциональной диагностики кровообращения основаны на различных физических принципах, используют сигналы различной природы. Электрокардиограф (ЭКГ) записывает изменения электрической активности сердца и позволяет определить моменты деполяризации разных частей сердечной мышцы. Фонокардиограф (ФКГ) фиксирует звуковые колебания и позволяет определить начало и окончание фаз сердечного цикла. Реограф (РГ) записывает сигнал об изменении электрического импеданса исследуемого участка тела переменному току с частотой 100 кГц при прохождении пульсовой волны, который пропорционален изменению объема крови в органе. В зависимости от исследуемого органа различают реоэнцефалограмму (РЭГ), реоплетизмограмму (РПГ), реовазограмму (РВГ). Ультразвуковые доплеровские системы позволяют оценить скорость кровотока в отдельных сосудах. Датчики, использующие лазерное излучение дают информацию о кровотоке в капиллярном отделе системы кровообращения и о насыщенности крови кислородом. Рентгеновские аппараты и методы исследования представляют данные о топологии системы кровообращения. Неинвазивные методы исследования функционирования сердечно-

сосудистой системы остаются более предпочтительными, особенно при необходимости длительного мониторинга.

Обычно, для уточнения диагноза применяют последовательно несколько методов исследования. Используют многоканальную запись сигналов. Что позволяет представить развитие процесса во времени и, оценивая скорость распространения сигналов и степень их искажения в разных каналах, получить данные о структуре и физических свойствах системы кровообращения. Очевидно, одновременное комплексное использование нескольких методов в одной системе исследования кровообращения позволит сократить время исследования и повысить качество диагностики.

Для разработки комплексной системы диагностики необходимо решить задачу комплексирования обработки данных от множества датчиков. Несмотря на различную природу сигналов от датчиков общим для всех методов исследования сердечно-сосудистой системы является сам цикл кровообращения [2]. То есть все сигналы должны соотноситься по времени с периодом сердечных сокращений (рис.1). Поэтому время выбирается в качестве общего параметра при комплексировании обработки сигналов. Одинаковые этапы работы сердца в сигналах от разных датчиков имеют различное представление. Некоторые фазы можно точнее определить с помощью одного из методов. Например, в фонокардиограмме четко определяют моменты закрытия клапанов, а кардиограмма позволяет выявить моменты электрической активности предшествующие сокращению сердца. Реограмма [3] позволяет проследить распространение пульсовой волны от сердца к органам и частям тела, отмечая моменты максимального наполнения сосудов на исследуемом участке (рис.2).

Целью диссертационной работы является разработка алгоритмов и программного обеспечения обработки биологических сигналов на примере реографии в Delphi.

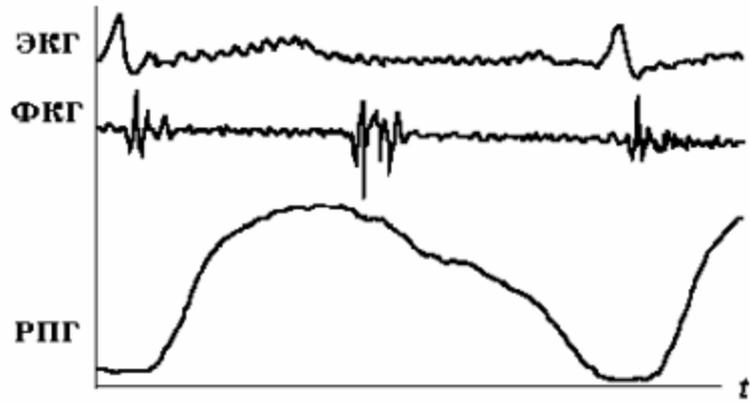


Рис. 1. Одновременная запись ЭКГ, ФКГ и РПГ.

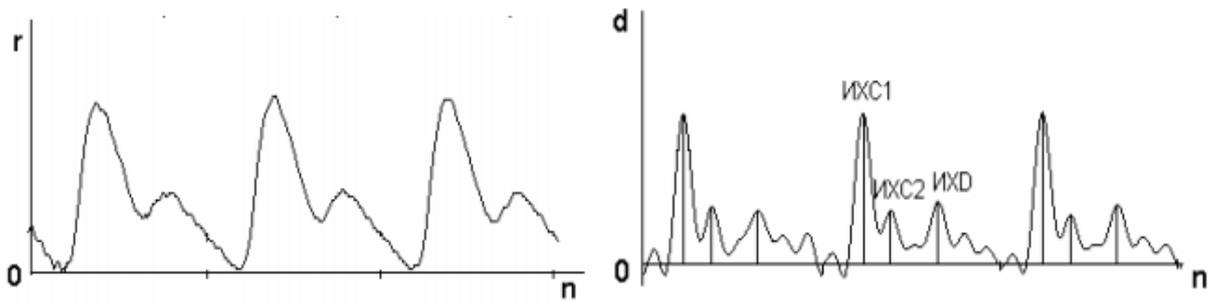


Рис. 2. Реосигнал $r(n)$ и импульсная характеристика $d(n)$.

Задачи исследования: В связи с целью исследования необходимо осуществить

1. Исследование основных гемодинамических и ЭКГ-показателей показателей совместно.
2. Анализ методов обработки реографических и ЭКГ данных.
3. Разработка алгоритма и программного обеспечения для решения задач обработки реографических сигналов.

Степень разработанности проблемы – при значительном развитии информационных технологий и программного обеспечения, в настоящее время не существует решения задачи диагностики болезни сердца при совместном использовании реографических и ЭКГ-данных.

Объектом исследования в работе является гемодинамические показатели в приложении к решению задач определения характерных точек реограммы и ЭКГ.

Научная новизна работы заключается в разработке:

алгоритма решения задач обработки реографических данных с учётом программы, реализующей разработанный алгоритм.

Практическая значимость.

Создан удобный и общедоступный инструмент для решения задач обработки биологических сигналов реографического исследования с использованием средств пакета Delphi.

В главе 1 составлен и проанализирован обзор возможности и важности различных реографических исследований человеческого организма. Показано, что с использованием реографических методов исследований центральной и местной гемодинамики возможно решить широкий круг задач дифференциальной диагностики.

В главе 2 приведены методы математической обработки и разработаны алгоритмы совместного преобразования информативных параметров рео - и кардиограмм.

В третьей главе предлагается программная реализация представления оцифрованных данных географических исследований на языке высокого уровня Delphi.

Объем и структура работы. Магистерская диссертация состоит из введения, трех глав, заключения и приложения, изложена на _____ страницах машинописного текста, содержит _____ рисунков. Список литературы содержит 29 публикаций.

Публикации. По материалам магистерской диссертации опубликована одна печатная работа.

ГЛАВА 1

ОСНОВНЫЕ ПОЛОЖЕНИЯ МЕТОДА РЕОГРАФИИ

1.1. Общие понятия метода реографии

Определение гемодинамических показателей проводится инвазивными и неинвазивными методами. Инвазивный метод связан с нарушением кожного покрова. Он связан с определенным риском для пациента и не всегда приемлем.

При неинвазивных способах нарушения кожного покрова не происходит. Эти способы просты для медицинского персонала и необременительны для пациента, поэтому, по возможности, предпочтительней применять неинвазивные методы исследования. Одним из неинвазивных методов является реография (импедансная плетизмография). Реография - практически единственный метод, обеспечивающий экспресс-диагностику состояния центральной гемодинамики в условиях атравматичного и необременительного для больного исследования.

Реография — неинвазивный метод исследования кровоснабжения органов [4], в основе которого лежит принцип регистрации изменений электрического сопротивления тканей в связи с меняющимся кровенаполнением. Чем больше приток крови к тканям, тем меньше их сопротивление. Для получения реограммы через тело пациента пропускают переменный ток частотой 50-100кГц, малой силы (не более 10 мкА), создаваемый специальным генератором.

Принципиальная разработка реографической методики принадлежит Н. Манну (1937). В дальнейшем методика (электроплетизмография, импеданс-плетизмография) получила развитие в работах А. А. Кедрова и Т. Ю. Либермана (1941— 1949) и др. Детальная разработка и внедрение в клиническую практику метода реографии связано с именами австрийских

исследователей W. Holzer, K. Polzer и A. Marko. Им же принадлежит по существу первая монография (Rheokardiographie, Wien, 1946), в которой авторы не только осветили технические стороны метода (электрические схемы аппарата, варианты генератора переменного тока и другие), но и представили результаты клинического использования реографии при различных заболеваниях сердечно сосудистой системы [4].

Существенный вклад в разработку метода реографии внес Ю.Т. Пушкарь, создавший отечественную конструкцию аппарата и изменивший методику регистрации реограммы (прекардильная реокардиография). В настоящее время доказано клиническое значение применения метода реографии [5-7].

В зависимости от конкретной клинической задачи меняется зона исследования, и соответственно место наложения электродов (рис.3). Поэтому различают реографию легких, сосудов мозга (реоэнцефалография), сосудов конечностей (реовазография) и другие.

Принципиальной основой метода реографии является зависимость изменений сопротивления от изменений кровенаполнения в изучаемом участке тела человека. Другими словами, изучаются пульсовые колебания электрического сопротивления.

Более полное представление о пульсовых колебаниях электрического сопротивления получают при учете (соотношении) базового сопротивления исследуемого участка (то есть суммарного сопротивления тела зондирующему току с частотой 50—100 кГц). Полный импеданс (сопротивление) состоит из двух величин, постоянный или базовый импеданс, обусловленный общим кровенаполнением тканей и их сопротивлением, и переменный или пульсовой импеданс, вызванный колебаниями кровенаполнения во время сердечного цикла. Величина пульсового импеданса ничтожно мала и составляет не более 0,5 % общего

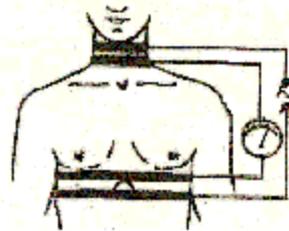
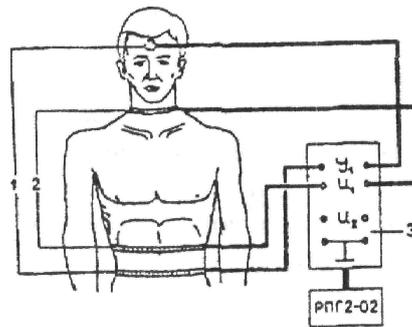


Рис.3. Расположение электродов при проведении реографического исследования.

импеданса. Вместе с тем пульсовой импеданс составляет объект изучения для реографии.

1.2. Методики использования реографа при исследовании гемодинамических показателей

Регистрация реограмм осуществляется с помощью реографов (рис.4). Последние состоят из следующих элементов генератора высокой частоты, преобразователя «импеданс-напряжение», детектора, усилителя, калибровочного устройства, дифференцирующей цепочки.

В зависимости от количества электродов, реализующих один информационный сигнал, различают двухэлектродные (биполярные) и четырехэлектродные (тетраполярные, как на рис. 2) реографы. Последние менее критичны к влиянию контактного сопротивления электрод – кожа [7].

На практике метод реографии адаптирован к исследованию отдельных органов или функциональных систем.

При биполярной методике накладывают 2 электрода, каждый из которых одновременно является токовым и измерительным, электроды фиксируют на соответствующем участке тела. Для снижения контактного сопротивления между электродом и кожей используются те же приемы, что и при записи ЭКГ. При использовании тетраполярной методики участок исследования ограничивают парой измерительных электродов, а возникшее в них напряжение снимают с помощью другой пары электродов, расположенных снаружи по отношению к первой (токовые).

Тetraполярная методика более точна, ибо резко (до минимума) снижается влияние контактного сопротивления (нет необходимости накладывать прокладки, смоченные растворами солей или щелочей, а также пользоваться электродной пастой) и электродной поляризации. Это позволяет с высокой степенью точности измерить импеданс глубинных тканей. Кроме того, достаточно точно получаемые сведения о базисном

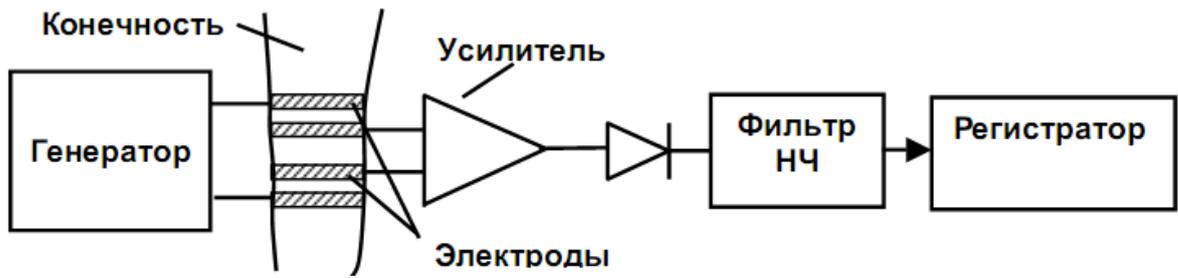


Рис. 4. Общий вид и функциональная схема реографа.

импедансе позволяют дать количественную оценку основным гемодинамическим показателям ударному и минутному объемам кровообращения.

Для кардиолога важнейшее значение имеет измерение ударного выброса крови сердцем. Одним из методов его нахождения является реоплетизмография - графическая регистрация пульсового наполнения.

Структура прибора проста. Через тело пациента пропускается измерительный (зондирующий) ток. Проводимость крови в 2 – 4 раза выше проводимости тканей и изменение кровенаполнения вызывает пульсацию сопротивления. Использование в качестве зондирующего постоянного тока нежелательно из-за имеющих место физиологических противопоказаний и сложной помеховой обстановки [8].

Поэтому используют зондирующий переменный ток. Частота генератора тока выбирается от 40 до 200 кГц. Наложёнными электродами измеряется падение напряжения от зондирующего тока на выбранном участке тела. Рассчитывают полное (базовое) сопротивление между электродами и выделяют пульсовую волну. Эта волна идет в такт с работой сердца.

Базовое сопротивление определяется по закону Ома:

$$R = \rho L/S, \text{ где:}$$

ρ - удельное сопротивление Ом*см (примерно 350 Ом*см, оно изменяется в зависимости от выбранной частоты измерительного тока), S - площадь токопроводящей зоны ткани (обычно 50 см²), L - расстояние между электродами. Для $L=20\text{см}$, $S= 50\text{см}^2$, $\rho=350 \text{ ом*см}$ получим $r=500 \text{ Ом}$. (Для более точного расчета значения R необходимо учитывать форму поля под электродами. Для поверхностных электродов дополнительно приходится учитывать сопротивление электрод - кожа).

С целью получения значения пульсирующего объема модернизируем приведенную формулу. Умножим числитель и знаменатель на L . Тогда:

$R = \rho L^2/V$, где V - находящийся между электродами объем тела. Нас интересует пульсирующий объем ΔV (при этом сопротивление R изменяется от R до $R+\Delta$, где Δ - малое приращение), поэтому проведем не сложные преобразования:

$$\Delta V = \rho L^2 \Delta / R^2 \text{ (или } \Delta V = \Delta S^2 / \rho \text{)}.$$

L, Δ, R непосредственно измеряется, ρ принимается типовым (ρ может быть так же определено измерением, если провести одновременно два замера R для разных L , считая при этом Δ и ΔV одинаковым). ΔV можно отождествлять со сжатием желудочков сердца и ударным выбросом крови.

Электроды выполняются каждый в виде конструктивно объединенных изолятором двух металлических лент (одна для токового электрода, другая для потенциального). При тетраполярной реографии электроды накладываются: один парный на шею, другой парный на бедра. По биполярной методике (Тищенко 1973 г) используется наложение электродов правая рука - правая нога.

Реограмма очень нестабильна, каждое движение тела или мускулатуры искажает график [8, 9]. Поэтому используют дифференцированную реограмму. Она более стабильна. Каждый прибор - реограф имеет каналы полной и дифференцированной реограммы.

Кроме определения параметров центральной гемодинамики (измерение ударного выброса, минутного объема, общего периферического сопротивления), реография успешно применяется для оценки мозгового кровообращения (реоэнцефалография), определения тока крови в конечностях (реовазография) и печени (реогепатография). Наиболее точна, информативна реография в сравнительных вариантах: регистрируется одновременно реограммы правой и левой области, правой и левой конечности.

Одноканальная реография не обладает селективностью по выделению области пульсирующего объема [9]. При сложных случаях заболеваний типа "легочное сердце" и "пульсирующая аорта" дополнительно пульсирующие

объемы нарушают достоверность измерения гемодинамики. Делаются попытки использования одновременно многих электродов и каналов, что потенциально дает возможность точнее локализовать обследуемые объемы. Для выделения сигналов каждого канала используют частотное или временное разделение.

Минимальное число РЕО каналов в приборе два (с развязкой между каналами). Дополнительно должен быть один канал ЭКГ. Частота зондирующего тока от 40 до 200 кГц, величина 0.2- 0.5 мА. Диапазон измерения базового сопротивления от 2 до 600 Ом, пульсирующего сопротивления от 0.002 до 1 Ома. Калибровочный сигнал обычно 0.1 Ома. Приведенный уровень шума не более 0.002 Ома. Полоса пропускания 0.05-30 Гц. Развязка между каналами 40 дБ, не менее.

В настоящее время реография сердца оттеснена ультразвуковым обследованием с прямым измерением сократительной способности миокарда. Однако при обследовании кровотока конечностей, печени, почек, головного мозга реография продолжает оставаться важным методом. Не смотря на сильную конкуренцию УЗИ, реография, при удачном инженерно конструкторском решении, может быть перспективным методом, если достигается упрощение работы мед персонала в процессе обследования: например, при обследовании используются совмещенные с ЭКГ электроды.

1.3. Методика регистрации реограммы и её анализ

Запись реограмм (рис. 5) [7, 10] производится в теплом помещении через 1,5—2 ч после приема пищи или натошак, в положении лежа на спине после 15—20-минутного отдыха. Одновременно с двумя реограммами (основной и дифференциальной) записывается ЭКГ во II стандартном отведении и иногда ФКГ в V точке или над верхушкой на одном из среднечастотных диапазонов. Желательно регистрацию реограммы производить на задержке дыхания при неполном выдохе. Запись производят при скорости движения

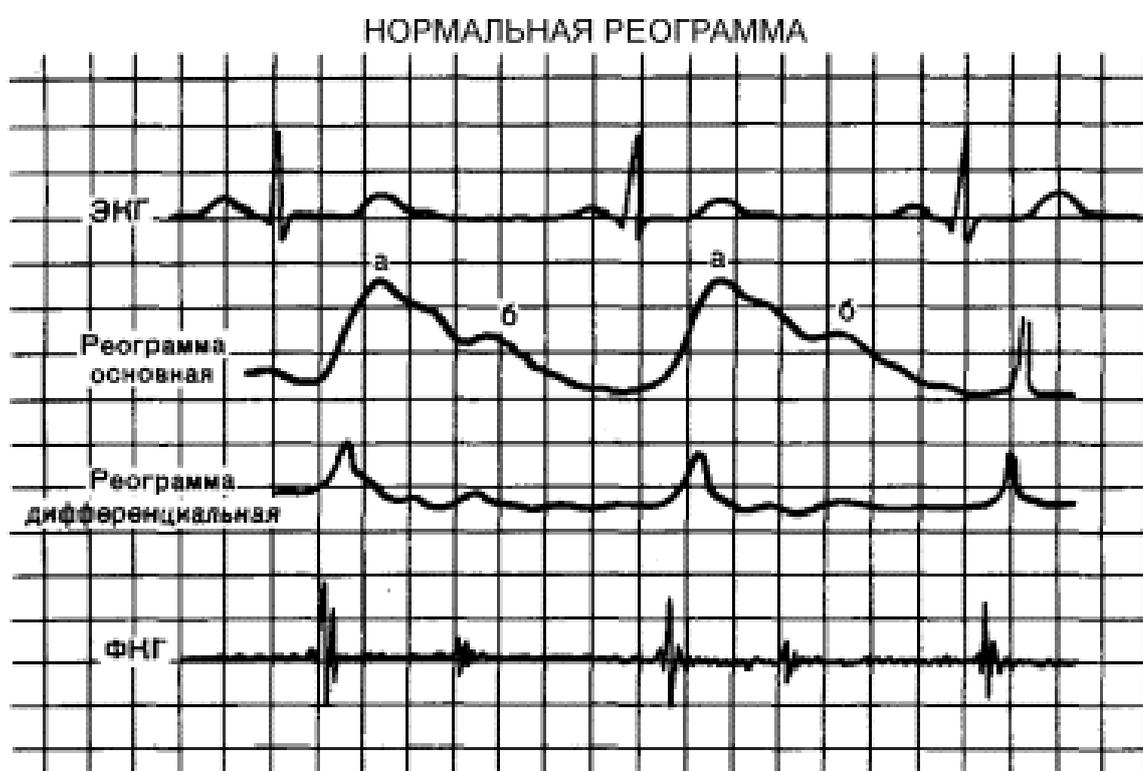


Рис. 5. Изображение нормальной реограммы.

лентопротяжного механизма 25—50 мм/с (реже — 100 мм/с). Необходимо следить за калибровочным сигналом (0,1 Ом=10 мм).

Количественный анализ предусматривает определение следующих показателей (рис.6):

1. Амплитуда систолической волны в миллиметрах измеряется от основания систолической волны до высшей точки реограммы.
2. Амплитуда диастолической волны в миллиметрах измеряется от основания диастолической волны до высшей ее точки.
3. Реографический индекс (систолический - РСИ и диастолический - РДИ) - отношение систолической (диастолической) волны к стандартному калибровочному сигналу (0,1 Ом =10 мм), выражается в относительных единицах. Этот показатель характеризует величину и скорость притока (оттока) крови в исследуемой зоне. Амплитуда кривой измеряется от изолинии до высшей точки волны.
4. Интервал Qa или время распространения пульсовой волны (ВРПВ) на участке «сердце — исследуемый орган» в секундах — соответствует периоду напряжения при фазовом анализе систолы желудочков. Измеряется от начала зубца Q ЭКГ до начала волны реограммы, связанной с данным сердечным циклом. Интервал Qa уменьшается при повышении тонуса или склерозе магистральных сосудов
5. Период или время быстрого наполнения (ВН_{быстр}) - от начала подъема систолической волны реограммы до точки максимальной крутизны на ее восходящем колене (соответствует проекции вершины основного зубца дифференциальной реограммы на восходящее колено объемной реограммы). Этот показатель отражает величину ударного объема и функциональное состояние крупных сосудов.

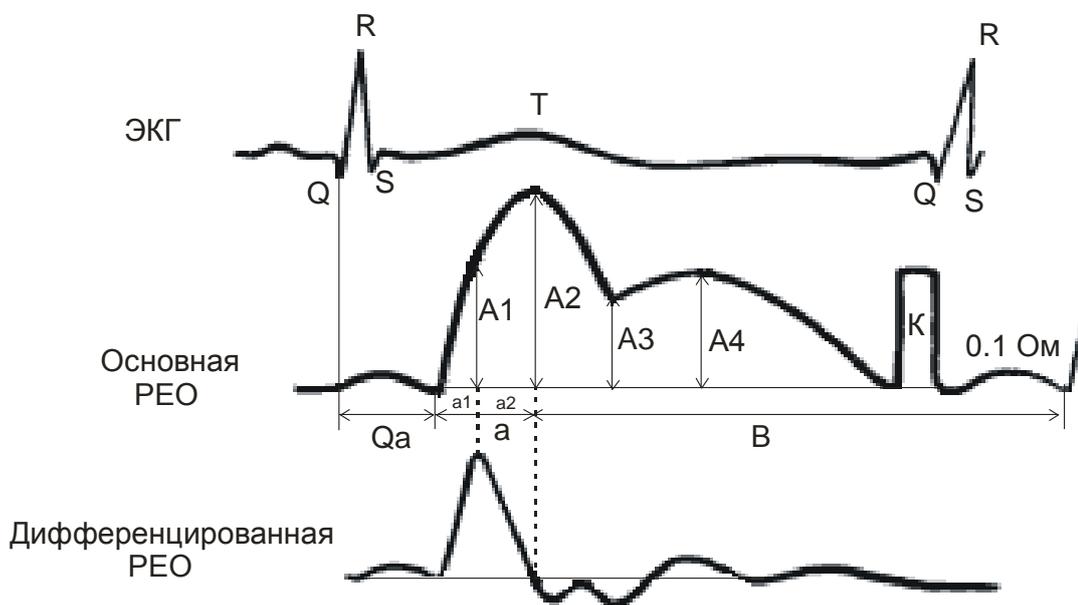


Рис.6. Реографическая кривая с обозначением основных точек:

A1 – максимальная крутизна восходящего колена волны;

A2 – максимальная амплитуда систолической волны;

A3 – нижняя точка инцизуры;

A4 – максимальная амплитуда диастолической волны реограммы;

K – калибровочный сигнал;

B – длительность нисходящей части реограммы;

Qa – время распространения реографической волны на участке

“сердце – исследуемый орган”;

a – время максимального систолического наполнения сосудов;

a1 – время быстрого наполнения;

a2 – время медленного наполнения.

6. Интервал Qa или время распространения пульсовой волны (ВРПВ) на участке «сердце — исследуемый орган» в секундах — соответствует периоду напряжения при фазовом анализе систолы желудочков. Измеряется от начала зубца Q ЭКГ до начала волны реограммы, связанной с данным сердечным циклом. Интервал Qa уменьшается при повышении тонуса или склерозе магистральных сосудов
7. Период или время быстрого наполнения ($VH_{\text{быстр}}$) - от начала подъема систолической волны реограммы до точки максимальной крутизны на ее восходящем колене (соответствует проекции вершины основного зубца дифференциальной реограммы на восходящее колено объемной реограммы). Этот показатель отражает величину ударного объема и функциональное состояние крупных сосудов.
8. Период или время медленного наполнения ($VH_{\text{медл}}$) - от точки максимальной крутизны на восходящем колене реограммы до ее вершины. Этот показатель определяется также как разность между $VH_{\text{макс}}$ и $VH_{\text{быстр}}$ и отражает функциональное состояние сосудов среднего и мелкого калибра.
9. $VH_{\text{быстр}}$ и $VH_{\text{медл}}$ составляют период максимального наполнения - $VH_{\text{макс}}$ который измеряется от начала восходящей части кривой до ее вершины.
10. Амплитудно-частотный показатель (АЧП) — отношение реографического индекса (РИ) к длительности сердечного цикла R-R. АЧП характеризует величину объемного кровотока в исследуемой области в единицу времени.
11. Отношение амплитуд систолической и диастолической волн (A_c/A_d) отражает степень преобладания артериального притока во время систолы над венозным оттоком во время диастолы.
12. Время общего наполнения ($VH_{\text{общ}}$) - интервал от начала подъема реограммы отражает общее время систолического притока крови в данную сосудистую область

13. Продолжительность катакроты в секундах (от высшей точки кривой реограммы до точки пересечения с изолинией)
14. Отношение времени восходящей части к времени нисходящей в процентах.
15. Отношение времени восходящей части реограммы к длительности сердечного цикла или к сумме, как показатель эластичности и тонуса сосудов.
16. Коэффициенты, отражающие отношение времени быстрого наполнения и времени медленного наполнения к общей длительности наполнения $(ВН_{быстр})/(ВН_{общ})$, $(ВН_{медл})/(ВН_{общ})$.

Следует заметить, что в реографии, как ни в одном из методов инструментальной диагностики сердечно-сосудистой системы нет единой методики количественных расчетов и нет единой терминологии. В каждом конкретном случае врач должен определить объем анализируемых показателей, который позволил бы при минимальных расчетах получить оптимальную информацию [4].

1.4. Виды реографии

Реография аорты и легочной артерии (рис.7) широко используется в клинической практике [28]. Для регистрации этих реограмм используют электроды размерами 3X4 см (активный) и 6X10 см. При записи реограммы легочной артерии активный электрод располагают во II межреберье справа по срединно-ключичной линии, а пассивный — в области нижнего угла правой лопатки. При записи реограммы аорты активный электрод фиксируют на груди на уровне II межреберья, а пассивный на спине в области IV — VI грудных позвонков. Такое расположение электродов обеспечивает получение отдельной информации о колебании кровенаполнения левого и правого желудочков сердца и правого легкого. На данных реограммах различают

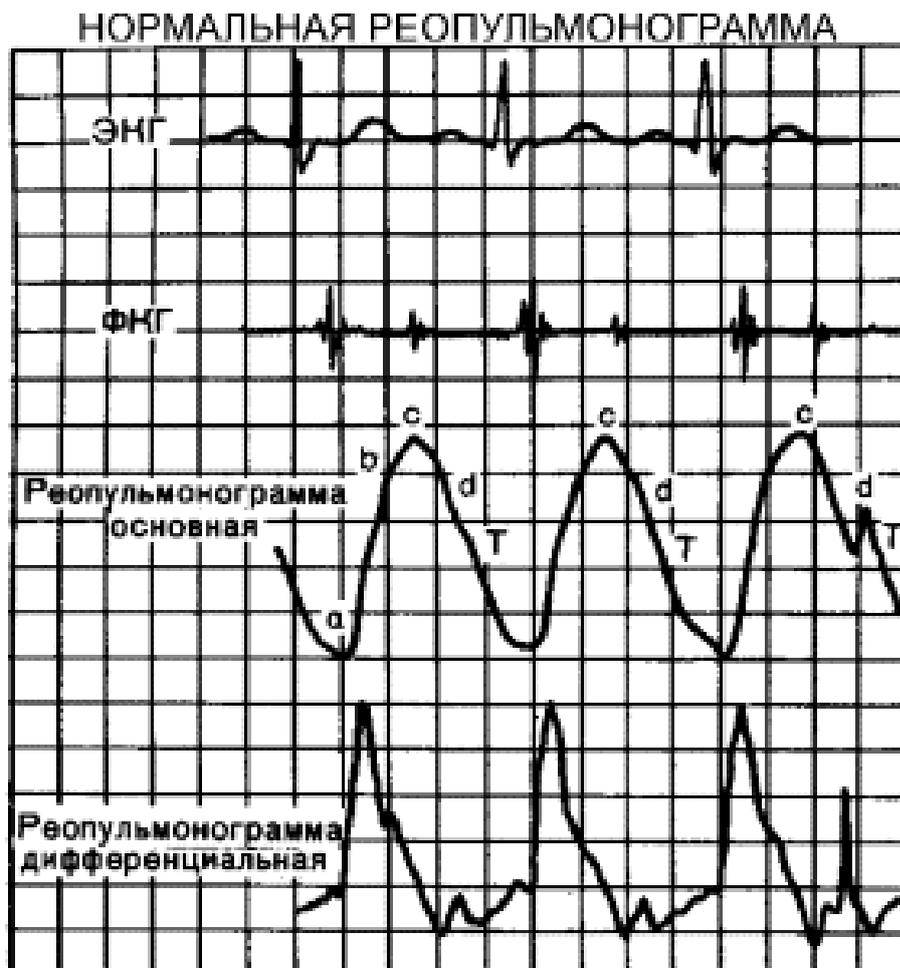


Рис. 7. Нормальная реопульмонограмма.

пресистолическую волну (начинается одновременно с зубцом Q ЭКГ, продолжается 0,10—0,15 с и обусловлена кровенаполнением полых вен и предсердий), систолическую (соответствует началу фазы изгнания в момент открытия полулунных клапанов) — отражающая кровенаполнение легочной артерии, ее ветвей и аорты. На систолической волне различают восходящее колено (рис. 3), «а—Ь», оно более крутое на реограмме аорты), вершину («с»), спуск с инцизурой («d») и дикроту.

За систолической волной следует диастолическая волна («Т»), отражающая время заполнения желудочков сердца кровью, больших полых вен, отток крови из аорты и легочных сосудов. Реограмма аорты и легочной артерии используется для расчета структуры правого и левого желудочков сердца, оценивающей сократительную функцию миокарда. Сопоставление опознавательных точек объемной и дифференциальной реограмм дает возможность выделить отдельные фазы сокращения желудочков.

Расчет указанных фазовых показателей и их условное обозначение следующие (рис.8):

T , с — период напряжения (в секундах)—интервал от начала зубца Q на ЭКГ до начала крутого подъема объемной реограммы в точке «а»;

$ИС$, с — фаза изометрического сокращения (в секундах)— интервал от начала I тона на ФКГ до точки «а» на объемной реограмме;

$АС$, с — фаза асинхронного сокращения (в секундах) — разность $Ac=T-ИС$;

E , с — период изгнания (в секундах)— интервал от точки «а» до точки «с» на объемной реограмме;

$Еб$, с — фаза быстрого изгнания (в секундах)— интервал от точки «а» до точки «Ь» на объемной реограмме;

S_m — механическая систола — сумма $S_m=E+ИС$;

S_o —общая систола—сумма $S_o= =E+T$.

ПОКАЗАТЕЛИ РЕОГРАММЫ ЛЕГОЧНОЙ АРТЕРИИ У ЗДОРОВЫХ ВЗРОСЛЫХ (Мmm)	
Показатель	Величина
C, с	$0,842 \pm 0,029$
AC, с	$0,052 \pm 0,005$
IC, с	$0,044 \pm 0,002$
T, с	$0,096 \pm 0,003$
E, с	$0,262 \pm 0,006$
VH _{быстр.} , с	$0,058 \pm 0,002$
VH _{медл.} , с	$0,111 \pm 0,004$
VH _{макс.} , с	$0,169 \pm 0,006$
S _{м.} , с	$0,313 \pm 0,008$
S _{с.} , с	$0,365 \pm 0,008$
ВСП, %	$85,94 \pm 0,65$
ИНМ, %	$26,30 \pm 0,44$
МК, ед.	$2,80 \pm 0,11$
РИ, ед.	$1,97 \pm 0,12$
V _{макс.} , Ом/с	$2,01 \pm 0,13$
V _{сред. медл.} , Ом/с	$0,56 \pm 0,03$

Рис 8. Показатели реограммы лёгочной артерии у здоровых взрослых.

$V_{\text{макс}}$.- максимальная скорость во время быстрого систолического наполнения характеризует сократительную способность миокарда, ударный объем желудочка (на рис.3) это интервал «2). Расчет производится таким образом вершина основного положительного систолического зубца дифференциальной реограммы проецируется на восходящую часть систолической волны объемной реограммы, из полученной точки проводится касательная к восходящей части систолической волны объемной реограммы — гипотенуза прямоугольного треугольника, а катетами его являются взаимно перпендикулярные отрезки, один параллелен оси абсцисс, а другой — оси ординат. Искомый показатель является тангенсом угла, образованного гипотенузой и катетом, параллельным оси абсцисс (катет вертикальный выражается в Ом, а горизонтальный в секундах, поэтому показатель выражается в Ом/с).

По данным реограммы легких можно сделать ряд диагностических выводов. Легочная гипертензия (прекапиллярная) приводит к изменению ряда показателей нарушается фазовая структура систолы правого желудочка, изменяется величина ударного реографического систолического индекса. Повышение систолического давления отражается тенденцией к уменьшению фазы максимального изгнания. Выраженная легочная гипертензия приводит к абсолютному снижению реографического систолического индекса. Повышение диастолического давления проявляется удлинением периода напряжения и фазы максимального изгнания. При посткапиллярной гипертензии увеличивается межамплитудный индекс за счет амплитуды диастолической волны. Уменьшение амплитуды систолической волны может быть при снижении ударного объема правого желудочка, прекапиллярной гипертензии, склеротических изменениях легочной артерии и ее ветвей.

Увеличение амплитуды систолической волны обусловлено увеличением ударного объема правого желудочка и может иметь место при некоторых врожденных пороках сердца, артериовенозных фистулах. Полное отсутствие

диастолической волны — показатель снижения резервной емкости венозного русла малого круга.

По состоянию гемодинамики малого круга кровообращения различают 3 типа реографических кривых а) гиповолемический — снижена амплитуда, имеются зазубрины на анакроте, нечеткая инцизура на диастолической волне, удлинено время максимального систолического наполнения, б) гиперволемический — высокая амплитуда систолической волны, крутая анакрота и катакрота, плохая выраженность инцизуры и диастолической волны, в) гипертонический — увеличенная амплитуда систолической волны, закругленная вершина, высокое расположение инцизуры.

Реоэнцефалография (РЭГ) — один из вариантов реографического метода исследования, направленный на изучение гемодинамики головного мозга в норме и при патологии [28].

Для записи РЭГ используют упомянутые выше реографы, которые подключают к любому многоканальному самописцу. Электроды представляют собой пластинки круглой формы диаметром 15— 20 мм, с надетыми фланелевыми чехлами, смоченные раствором хлорида натрия. Места наложения электродов обрабатывают спиртом. Условия регистрации и режим работы аппарата такие же, как описаны выше. Обычно используют следующие отведения: фронтомастоидальное, отражающее межполушарную асимметрию и нарушение кровообращения в бассейне внутренней сонной артерии (электроды расположены над бровями и в области сосцевидного отростка за ухом справа и слева); окципитомастоидальное (выявляет локальные изменения в системе позвоночной артерии); лобные и лобно-височные (выявляются нарушения в системе передней мозговой артерии); височно-височные (выявляют нарушения в бассейне средней мозговой артерии) и др. Можно одновременно записывать несколько отведений, но не более трех — четырех.

Рассчитывают и оценивают РЭГ по вышеописанной методике.

При нормальном кровообращении в мозгу межполушарная асимметрия не превышает 10 %, а РИ колеблется в пределах от 1,2 до 1,6. При этом $V_{\text{макс.}}=1,7$ Ом/с и $V_{\text{медл.}}=0,3—0,4$ Ом/с. Если нарушается мозговое кровообращение (стеноз сонной артерии, шок, артериальная гипертония, синдром позвоночной артерии и др.), указанные показатели изменяются в соответствии со степенью дефицита кровотока. Для уточнения характера нарушений кровообращения (органическое или функциональное) проводят лекарственные пробы (нитроглицерин, эуфиллин и др.) и записывают РЭГ в динамике.

Реовазография — метод исследования кровообращения в конечностях [27]. Существуют 2 методики: продольная (электроды накладываются на крайние точки исследуемого участка конечностей — проксимально и дистально) и поперечная (электроды располагаются на одном и том же уровне напротив друг друга). Наиболее признанной и обоснованной является первая методика. При биполярной реовазографии накладывают 2 электрода при тетраполярной — 4. Электроды представляют собой полоски из токопроводящего материала (свинцовые и др.) шириной 5—10 мм и другой формы. При динамическом снятии реовазограмм необходимо фиксировать электроды в одном и том же положении. Условия записи реовазограмм, режим работы аппарата, а также анализ количественных показателей такие же, как и при регистрации реограмм других зон. Реовазограммы, записанные на различных реографах, отличаются величиной амплитудных показателей. Поэтому при динамическом обследовании следует пользоваться одним и тем же аппаратом и на нем предварительно отработать нормативные показатели.

Поданным Ф. Д. Акуловой (1986), амплитуда систолической волны реовазограммы в норме составляет на предплечье 0,07—0,10; на кисти—0,11—0,15; на бедре — 0,05—0,06; на голени—0,08—0,12; на стопе 0,10—0,13 Ом. Соотношение между амплитудными показателями реовазограмм плеча и бедра составляет 1:1,5. При спазме артериальных сосудов амплитуда реовазограммы снижается, а при гипотонии резко возрастает.

Реовазография — высокоинформативный метод диагностики нарушений артериального или венозного кровотока в конечностях (тромбооблитерирующие процессы, атеросклеротическое поражение сосудов, болезнь Рейно и др.).

Интегральная реография [28]. Эта методика основана на изменении базового импеданса. При этом можно измерить базовый импеданс на протяженности всего тела или в каком-нибудь регионе. Существуют несколько зон наложения электродов для изучения регионарного базового импеданса. Наибольшее применение нашли: методика Е. Hoffer (1970), когда размещаются кольцевые электроды в нижней трети правого предплечья и левой голени, и методика М. И. Тищенко (1971— 1973), предусматривающая фиксацию одной пары объединенных электродов в нижней трети обоих предплечий, а другой — в нижней трети обеих голеней. По мнению Hoffer., между величиной интегрального импеданса и общим объемом воды в организме существует линейная зависимость, что выражается следующей формулой:

$$V = 0,4 \frac{H^2}{Z_s} + 14,$$

где V — объем воды, л; H — рост исследуемого, см; Z_s — импеданс тела, Ом. Коэффициент корреляции при этом равен +0,92. Интегральная реография может быть использована для определения ударного объема крови, ибо, как установлено, существует зависимость между базовым импедансом, пульсовым его изменением и объемом крови. На основании такой зависимости Кедров предложил формулу для расчета ударного объема крови по интегральной реограмме туловища:

$$\Delta V = \frac{\Delta R}{R} \cdot P,$$

где ΔV —ударный выброс крови, мл; ΔR — изменение электрического сопротивления во время систолы, Ом; R — исходное базовое сопротивление тела току высокой частоты, создаваемому генератором реографа, Ом; P — масса тела, г.

М. И. Тищенко предлагает такую формулу для вычисления ударного объема крови:

$$V_s = \frac{h/K \cdot L^2}{R} \cdot \frac{T}{D},$$

где V_s — ударный объем; L — рост обследуемого, см, h — максимальная амплитуда кривой, мм; K — амплитуда калибровочного сигнала 0,1 Ом, D — продолжительность диастолы; T — сердечный цикл, с.

Полученные данные умножают на коэффициент 0,275 для мужчин и 0,247 для женщин.

Проведенные Б. И. Мажбичем и Т. П. Шевченко (1984) исследования по уточнению природы интегральной реограммы тела указывают на то, что интегральная реограмма является не чем иным, как реограммой конечностей, а определение величины сердечного выброса левого желудочка по методике Тищенко носит эмпирический характер.

Метод W. Kubicek (1966), предусматривающий расположение токовых электродов на шее и грудной клетке, при сравнении с прямыми методами определения ударного объема (Фика, термодилуции и др.) дает почти 100 % совпадение данных и получил наибольшее признание. Его следует применять в лечебно-профилактических учреждениях. Тетраполярная реограмма регистрируется с применением импедансного кардиографа или реоплетизмографа. Одновременно регистрируют дифференциальную реограмму и ФКГ (для контроля). Ударный объем рассчитывают по формуле:

$$УО = \rho \frac{L^2}{Z^2} T \frac{dZ}{dt},$$

где ρ — удельное сопротивление крови при 100 кГц (150 Ом-см); L — расстояние между электродами; Z — базисное сопротивление в Ом; dZ/Dt — максимальная скорость изменения сопротивления во время сердечного цикла, определяемая по амплитуде дифференциальной кривой; T — длительность периода изгнания (также определяется по дифференциальной реограмме: спустя 0,015 с от начала подъема кривой до самой глубокой точки инцизуры).

Реокардиография [26] широко используется во всем мире как методика оценки параметров центральной гемодинамики. Метод позволяет достаточно точно определить величину сердечного выброса. Величины ударного объема сердца, получаемые при помощи реокардиографии сопоставимы с таковыми при инвазивном его определении путем разведения по Фику (“золотой стандарт” определения величины ударного объема сердца) и превышает по точности эхокардиографию.

Методом реокардиографии можно определить:

Величину сердечного выброса

Тип центральной гемодинамики

Инотропную функцию миокарда (сократимость)

Определение типа центральной гемодинамики с высокой степенью достоверности можно определить только методом реокардиографии. Это имеет большое значение у больных артериальной гипертензией.

Существует три типа центральной гемодинамики:

Эукинетический (нормальные величины сердечного выброса и общего периферического сопротивления сосудов);

Гипокинетический (низкий сердечный выброс при повышенном общем периферическом сопротивлении сосудов);

Гиперкинетический (высокий сердечный выброс при нормальном или пониженном общем периферическом сопротивлении сосудов).

В зависимости от типа системной (центральной) гемодинамики определяется антигипертензивный препарат выбора:

Эукинетический тип – показаны диуретики (арифон), либо ингибиторы АПФ, особенно когда на кардиоритмограмме определяется преобладание гуморально-метаболических влияний на регуляцию сердечного ритма;

Гипокинетический тип – препаратом выбора являются антагинисты кальция (ретардные дигидропиридины, амлодипины), особенно когда преобладают парасимпатические влияния на регуляцию сердечного ритма. Если преобладают гуморально-метаболические влияния, то необходимо применять ингибиторы АПФ;

Гиперкинетический тип – препарат выбора – бета-блокаторы (чаще – атенолол, конкор).

Таким образом, при подборе антигипертензивной терапии необходимо учитывать в комплексе данные реокардиографии и кардиоритмографии. В этом случае мы получим максимально эффективное и безопасное для больного лечение.

Итак, при реокардиографии определяется ударный объем сердца методом тетраполярной грудной реографии по Кубичеку. Однако, определение ударного объема (УО) не является конечной целью реокардиографии. Его можно рассматривать лишь как в заветную дверь системной гемодинамики. Минимальным целесообразным объемом, необходимым для ее оценки используют следующие количественные показатели: минутный объем кровообращения (МОК), его нормализованный эквивалент – сердечный индекс (СИ), показатели периферического сопротивления сосудов – общее периферическое сопротивление (ОПС) и удельное периферическое сопротивление (УПС), а также объемная скорость выброса (ОСВ) и мощность сокращения сердечной мышцы (МСС). Расчет этих показателей осуществляется по специальным формулам.

Оценка сократимости миокарда методом реокардиографии. В свое время эту задачу пытались решить методом поликардиографии. Для этого

синхронно записывались одно отведение ЭКГ, один канал ФКГ и сфигмограмма общей сонной артерии. Выполнялся хронометрический анализ по В.Л. Карпману. На этот метод возлагались определенные надежды объективной диагностики состояния сократительной способности миокарда. Усилия сотен исследователей были направлены на поиск путей клинической адаптации метода. Однако метод целевого назначения – оценки сократимости миокарда давал слишком большую ошибку.

Л.Б. Иванов предложил оценивать сократимость миокарда (инотропную функцию миокарда) методом реокардиографии. Для этого необходимо определить так называемый показатель сократимости миокарда. (ПСМ) [26].

В норме показатель ПСМ находится в пределах от 98 до 102%.

Увеличение значения ПСМ свидетельствует о снижении сократимости миокарда, а снижение ПСМ – на увеличение сократимости.

Имея значение ПСМ можно определить не только инотропную функцию миокарда, но и стадии миокардиальной недостаточности, причем по показателю ПСМ можно также проводить дифференцированный подбор терапии.

В первую стадию (компенсации) под влиянием активации симпато-адреналовой системы происходит компенсаторное усиление инотропной стимуляции миокарда (ПСМ достигает 80%). МОК находится в пределах нормы, пульс учащен.

Дальнейшее компенсаторное усиление сократительной способности миокарда приводит к “пересокращению” миокардиальных волокон, затруднению их расслабления и снижению пропульсивной способности сердца. Развивается вторая стадия миокардиальной формы сердечной недостаточности (субкомпенсации). В эту стадию наблюдается выраженная тахикардия, МОК имеет тенденцию к снижению. ПСМ достигает 60%. Такое состояние миокардиальных волокон не позволяет реализовать преднагрузку в адекватном объеме. Именно на этой стадии, когда наблюдается чрезмерное сокращение сердца при ухудшающемся его расслаблении, целесообразно

использование МАЛЫХ доз бета-блокаторов. Причем оказалось, что эффективные дозы были в 3-5 раз ниже рекомендованных в фармакологических справочниках. Коррекция сократимости должна осуществляться форсированно. Следует снижать уровень инотропной стимуляции с 60% до 80-85%. Передозировка (даже дозой официально рекомендуемой) может способствовать переходу сердечной недостаточности из стадии субкомпенсации в стадию декомпенсации.

Третья стадия сердечной недостаточности – декомпенсации, имеет три фазы развития:

ранняя – характеризуется постепенным снижением хроно-и инотропной стимуляции с переходом ее через короткий период псевдонормализации (ПСМ около 100%), СВ снижается, повышается ОПС.

развитая – тахикардия сменяется брадикардией, МОК резко снижен, ПСМ превышает 100%.

поздняя стадия – указанные проявления прогрессируют.

Таким образом, совокупное применение показателей СВ и ПСМ может быть с большой пользой использовано при подборе эффективной дозы сердечных гликозидов и других кардиотропных препаратов в кардиологической практике методом индивидуального титрования.

Выводы к главе 1

Приведён обзор реографической методики и видов реографии. Сделан вывод о перспективности реографии в исследовании центральной и местной гемодинамики, как о точном методе.

ГЛАВА 2

МАТЕМАТИЧЕСКАЯ ОБРАБОТКА ОБРАБОТКИ КАРДИОГРАФИЧЕСКИХ И РЕОГРАФИЧЕСКИХ ДАННЫХ

2.1. Функции нахождения глобального максимума и минимума сигналов

Глобальные максимумы понадобятся для пересчёта координат графиков в файле в координаты компонента Изображение (TImage).

Алгоритм работает по принципу последовательного сравнения каждого значения из массива с некоторым максимальным (минимальным) значением. Если текущее значение больше(меньше) максимального(минимального), то это значение становится максимальным(минимальным) [17, 18, 23].

2.2. Функция пересчёта координат

И и J – функция пересчёта координат (абсциссу и ординату, соответственно) к координатам на экране. Формула для преобразования координат получается с помощью свойства подобия [17]. Необходимо также учитывать, что на экране ось ординат направлена вниз. На нижерасположенных рисунках показано, как необходимо делать преобразование координат для построения графиков зоне изображения Image1 (рис.9-11).

$I1:=0;$ $I2:=Image1.Width$ (ширина) – размеры изображения по горизонтальной оси

$J1:=0;$ $J2:=Image1.Height$ (высота) – размеры изображения по вертикальной оси.

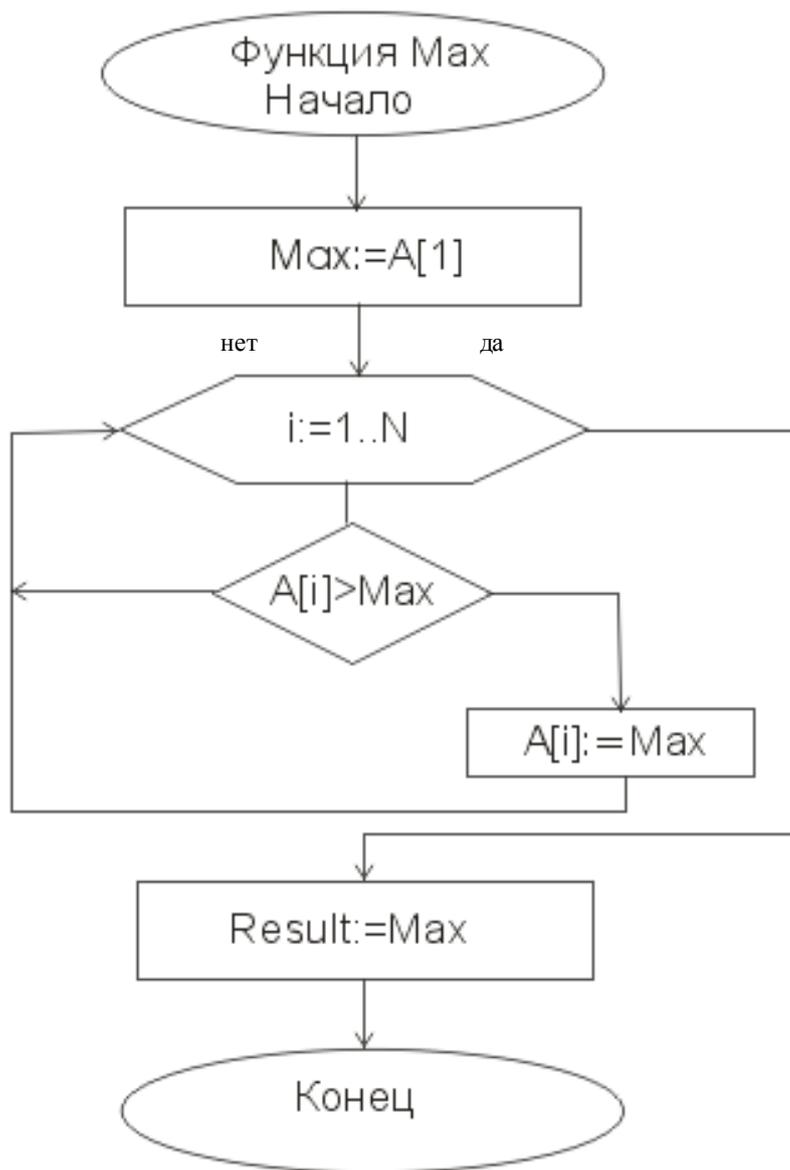


Рис.9. Блок-схема (для нахождения максимума):

$A[i]$ – массив, в котором находится максимальное значение.

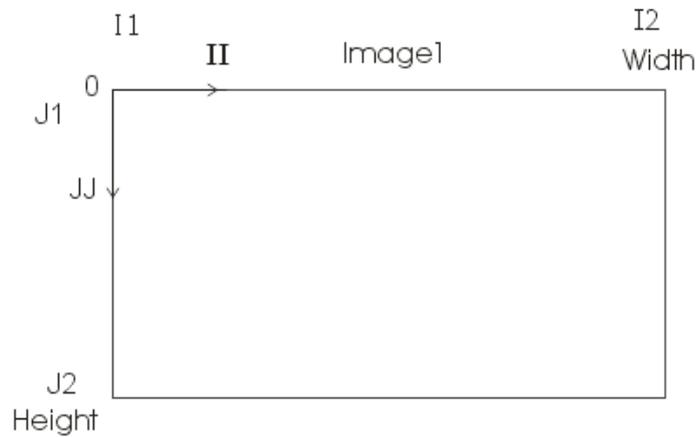


Рис.10. Координатные оси компонента TImage.

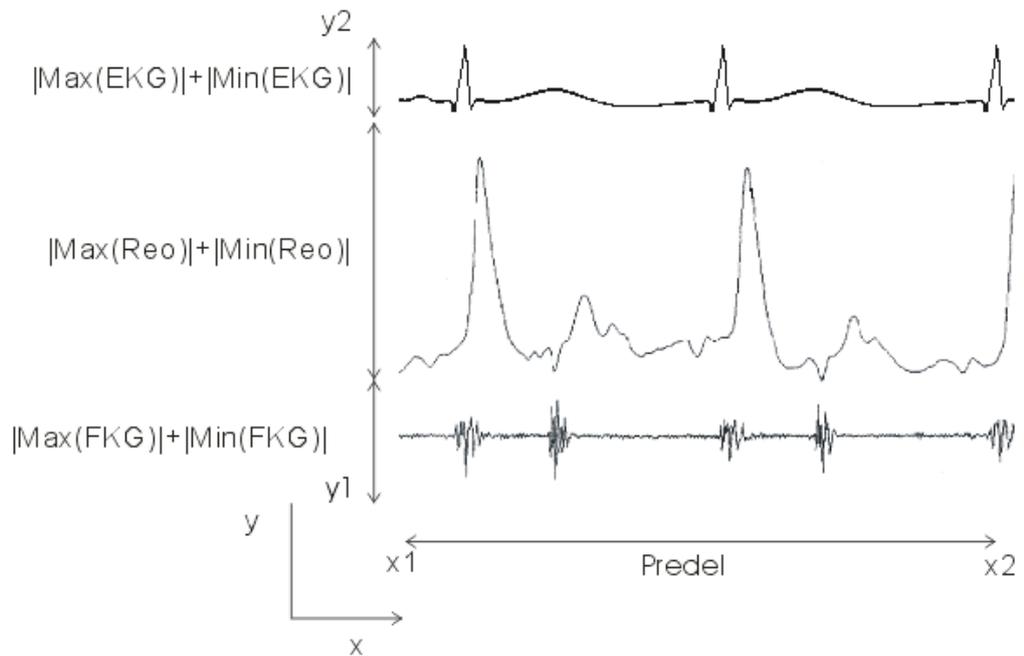


Рис.11. Координаты графиков в массиве.

$x1:=0$; $x2:=Predel$ – количество элементов массива

$y1:=0$; $y2:=|Max(Reo)|+|Min(Reo)|+|Max(FKG)|+|Min(FKG)|+|Max(EKG)|+|Min(EKG)|$ - сумма модулей глобальных максимумов и минимумов трёх графиков.

Из формулы подобия $\frac{x-x_1}{x_2-x_1} = const$ получаются следующие

преобразования координат:

$$\frac{II - I_1}{I_2 - I_1} = \frac{x - x_1}{x_2 - x_1}; \quad II = I_1 + \frac{(I_2 - I_1)(x - x_1)}{x_2 - x_1}$$
$$\frac{JJ - J_1}{-(J_2 - J_1)} = \frac{y - y_1}{y_2 - y_1}; \quad JJ = J_1 + \frac{(J_1 - J_2)(y - y_1)}{y_2 - y_1}$$

2.3. Построение графиков по данным из массива

Если функция для построения графика задана в массиве (т.е. поточечно), то необходимо соединять последовательно пары точек i и $i+1$. Каждая точка имеет координаты: по оси абсцисс – порядковый номер точки в массиве, а по оси ординат – значение в массиве этой точки. В программе это можно сделать с помощью методов `MoveTo(X,Y:integer)` и `LineTo(X,Y:integer)`. Метод `MoveTo(X,Y:integer)` устанавливает курсор на канву компонента `TImage` в точку с координатами X, Y . Метод `LineTo(X,Y:integer)` проводит линию от исходной точки к точке с координатами, указанными в `LineTo` [24, 25].

2.4. Нахождение R-зубцов на ЭКГ

Определить R-зубцы на ЭКГ или соответствующие им зубцы на дифференцированной ЭКГ можно с помощью следующего алгоритма. Вычисляется некий порог, и среди значений, которые его превышают, ищется локальный максимум [11, 12].

Порог может вычисляться, например, как сумма среднего значения ЭКГ и половины максимального значения ЭКГ. Далее ищется первое значение, которое больше порога. От этого значения до первого значения меньшего,

чем порог, ищется максимальное значение, которое и определяется как R-зубец (рис. 12-16).

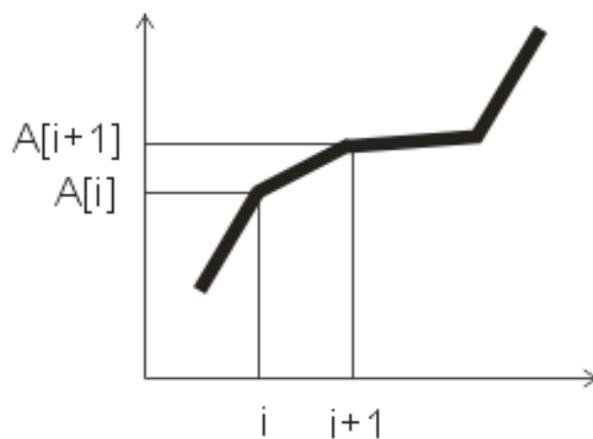


Рис.12. Принцип построения графика по точкам

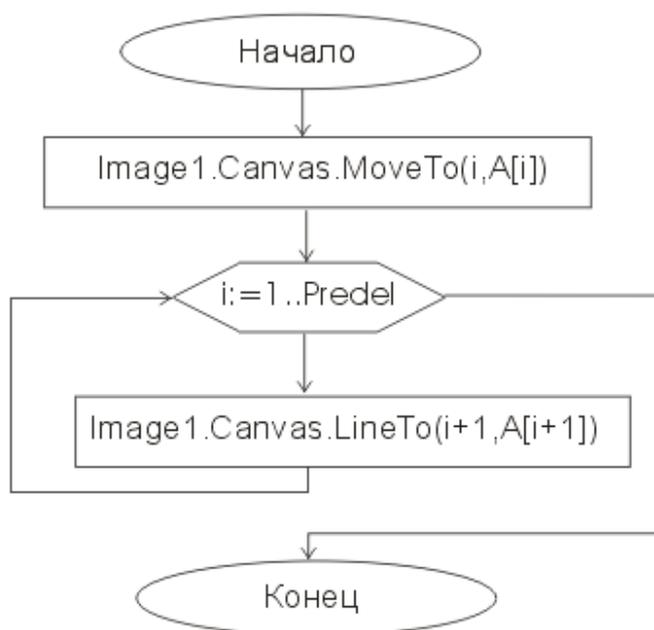


Рис.13. Блок-схема для алгоритма построения графика.

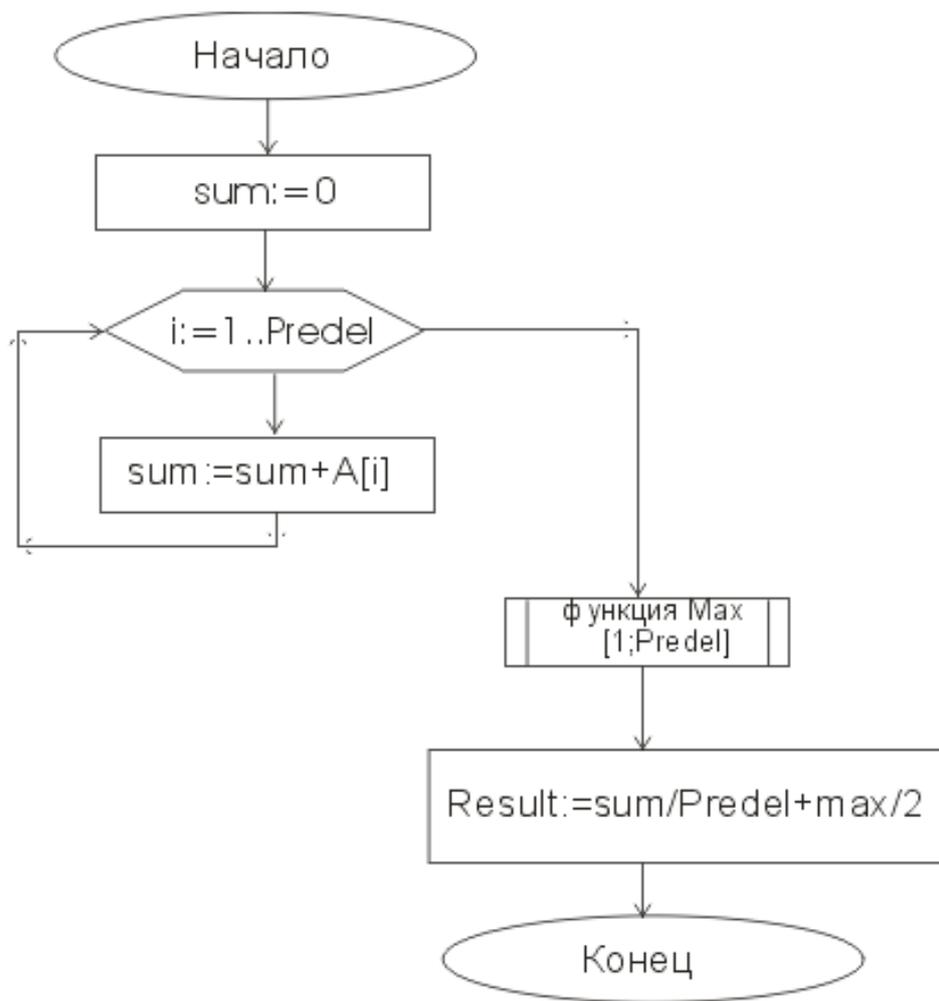


Рис.14. Блок-схема для алгоритма нахождения порога.

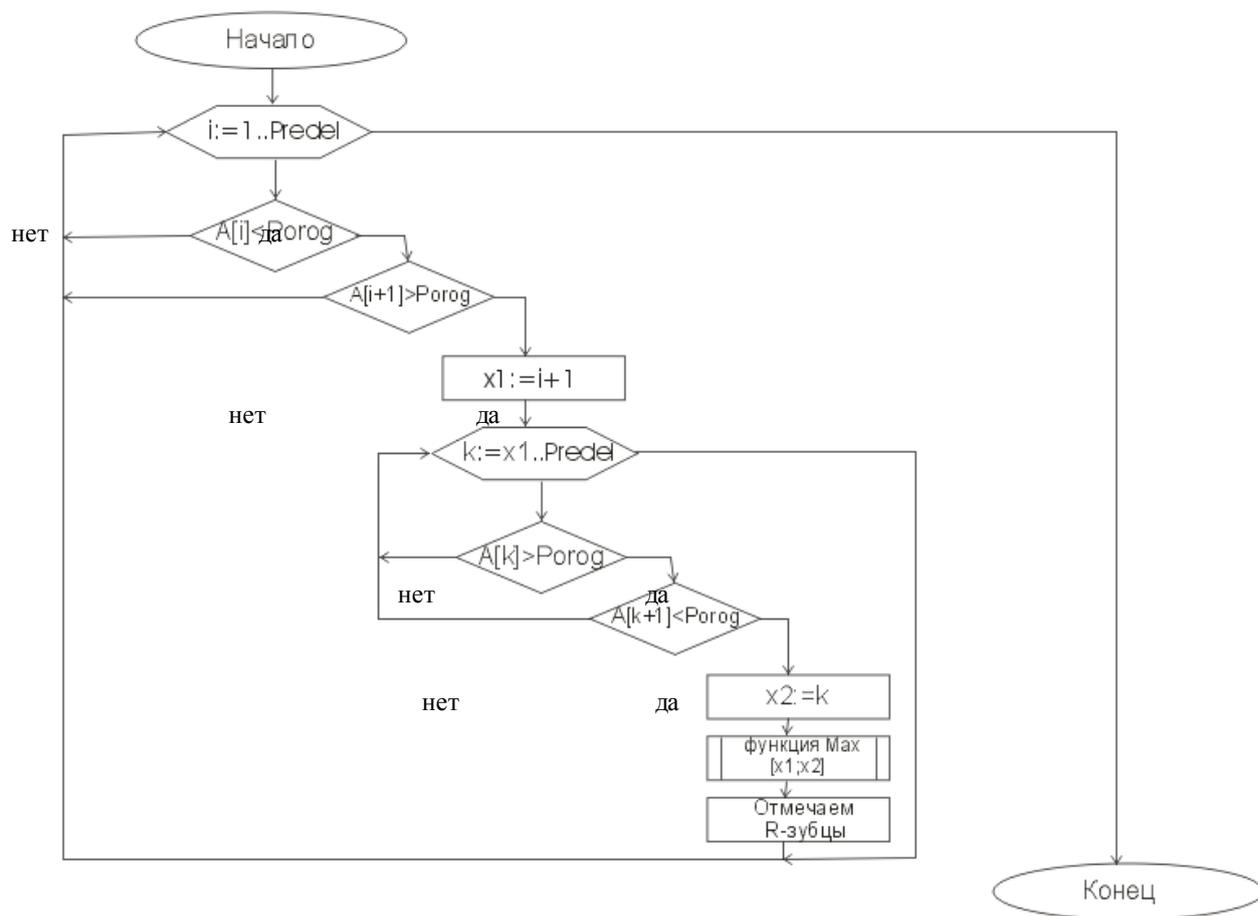


Рис. 15. Блок-схема для алгоритма нахождения R-зубцов.

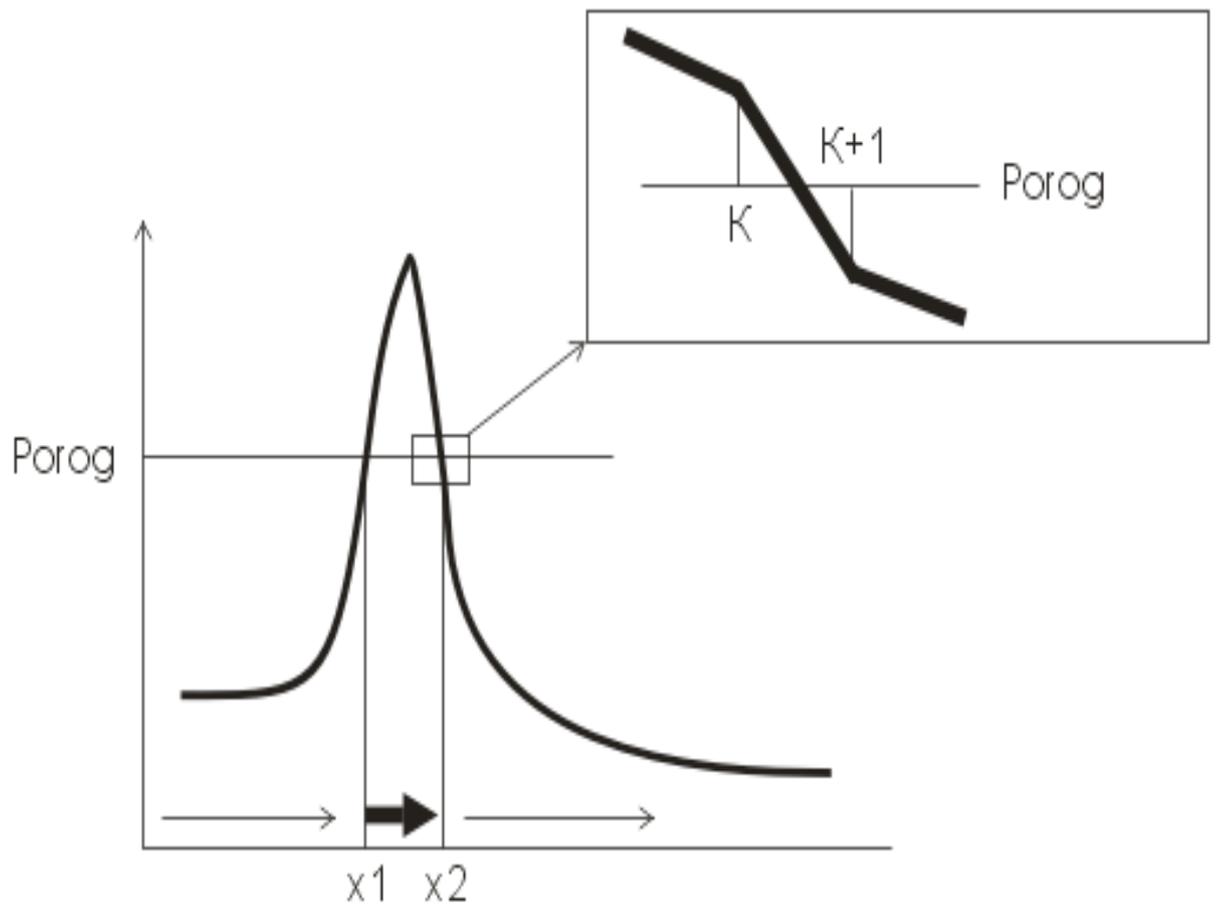


Рис.16 . Иллюстрация к алгоритму нахождения R-зубца.

Нахождение максимумов дифференцированной реограммы происходит аналогичным образом.

2.5. Определение точек начала изгнания крови из левого желудочка сердца

К кривой дифференцированной реограммы в точке максимума второй производной реограммы проводится касательная до пересечения ее с нулевым уровнем [18-22]. Эта точка является абсциссой точки начала изгнания крови из левого желудочка сердца.

Для нахождения этих точек на графике необходимо рассматривать каждый период в отдельности. Поэтому предварительно необходимо разбить график на периодические части. Точками разделения этих периодов могут служить максимумы дифференцированной реограммы. Их координаты абсцисс заносятся в статический массив на стадии нахождения максимумов дифференцированной реограммы. После нахождения координат происходит анализ реограммы на каждом отдельном периоде (рис. 17).

Так как график функции строился по точкам и представляет собой ломанную, то провести касательную к одной точке не представляется возможным. Однако, по определению касательной можно воспользоваться двумя соседними точками и провести через них секущую (касательная является предельным случаем секущей, см. рис.18).

Геометрический смысл производной - коэффициент наклона прямой, т.е. тангенс угла наклона. Тангенс – это отношение противолежащего катета прямоугольного треугольника к прилежащему. Следовательно,

$$Y' = k = \operatorname{tg} \alpha = \frac{A[i+1] - A[i-1]}{(i+1) - (i-1)} = \frac{A[i+1] - A[i-1]}{2}$$

Далее ищется максимальная производная на всем периоде с помощью функции Max. Абсцисса точки начала выброса определяется через тангенс угла.

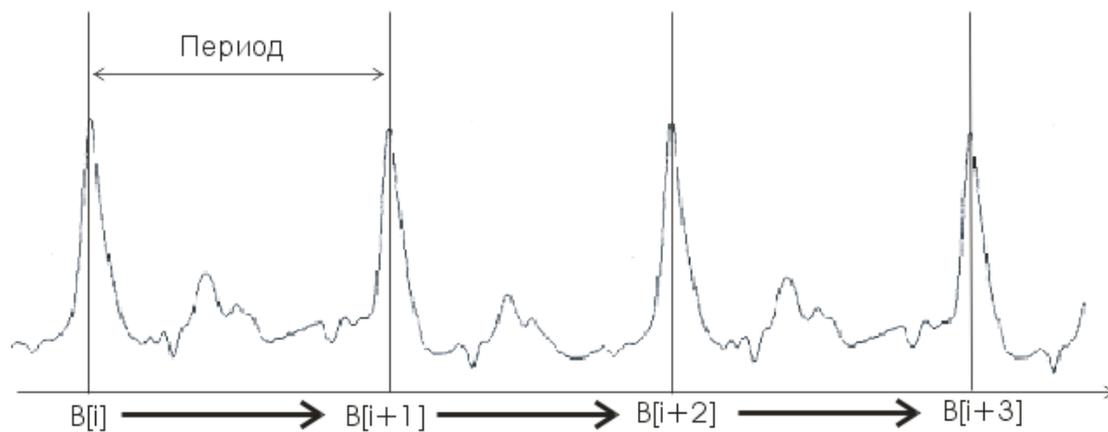


Рис. 17. Периодичность реограммы.

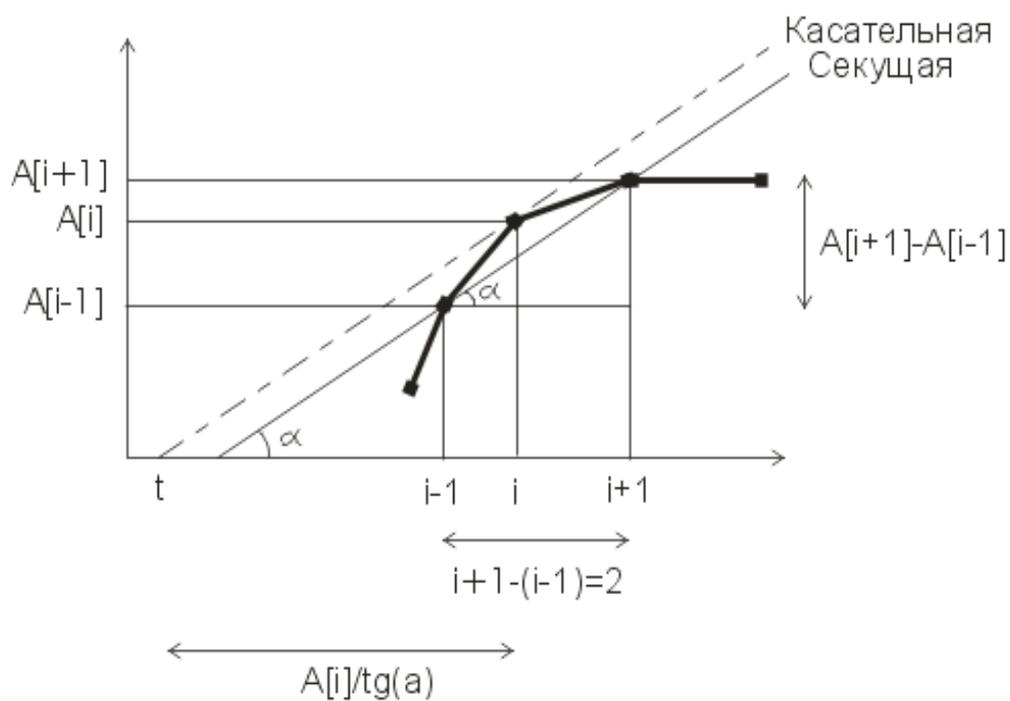


Рис. 18. Нахождение производной и её геометрический смысл.

Чтобы компенсировать сдвиг между секущей и касательной (они параллельны), в формулу будет подставляться координата точки i .

Пусть t – абсцисса точки начала выброса крови. Она равна разности абсциссы точки максимума производной i и основания касательной к точке i .

$$t = i - \frac{A[i]}{\operatorname{tg} \alpha}, \text{ где } i - \text{ точка максимума второй производной основной}$$

реограммы (рис.19).

Выводы к главе 2

Приведены алгоритмы совместной обработки физиологических данных реограммы и кардиограммы различными математическими методами.

Это сравнение нужно, т.к. в массиве с абсциссами максимумов диф. реограммы неизвестное число значащих элементов. Поэтому нужно проверять, пока не появятся нули в массиве.

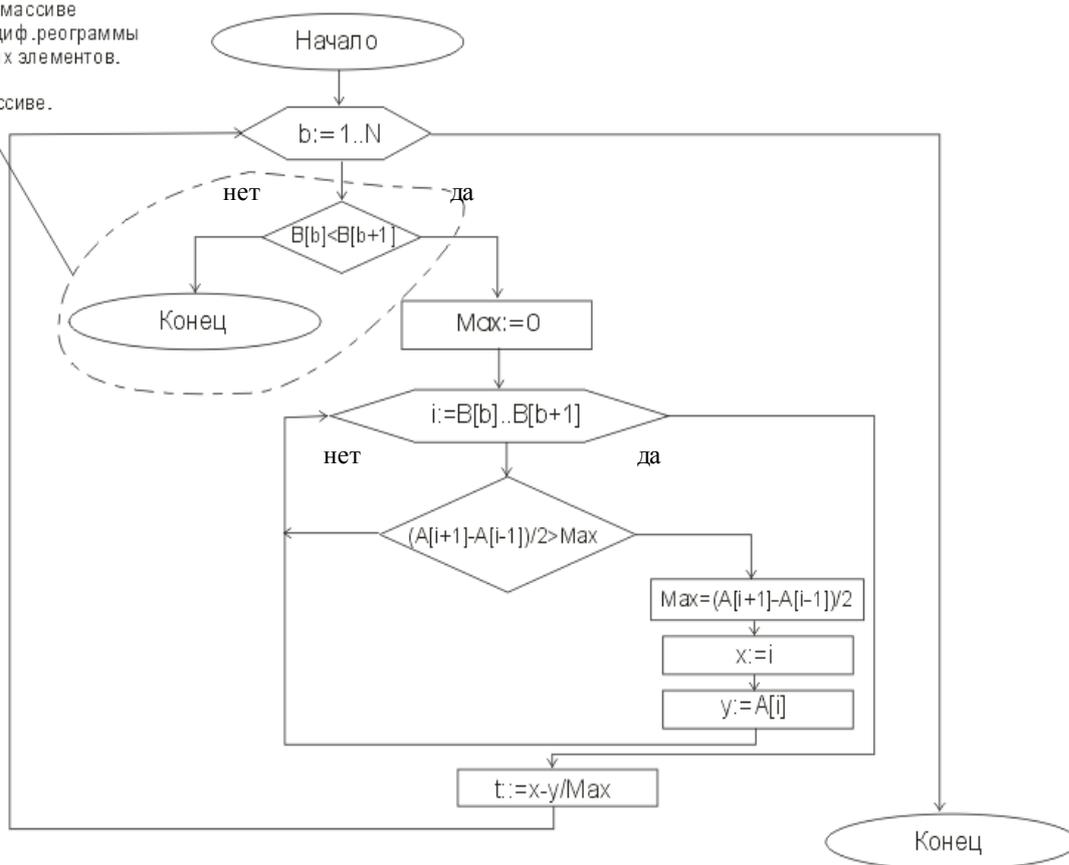


Рис. 19. Блок-схема для алгоритма нахождения точек начала изгнания крови из левого желудочка сердца.

ГЛАВА 3

3.1. Формальное описание программы

Данные сигналов и информация о пациенте располагается в типизированных *.dat файлах (тип запись) [29].

Поля записи

RegNumber: smallint	- регистрационный номер;
Da_ta, Name: shortstring -	дата записи, ФИО;
Distance, Perimetr: smallint –	расстояние между электродами, обхват грудной клетки;
Leng_s, Weight: smallint –	рост, вес;
Sist_BP, Diast_BP: smallint–	систолическое давление, диастолическое давление;
B: array[1..4,1..Predel] of smallint–	массивы сигналов ЭКГ, реограммы, ФКГ, опорный сигнал;
Kalibr_Z0, Kalibr_DZ: Real48 –	значения калибровки;
	Константа
Predel=7000	(количество точек в массиве).

Процедуры и функции:

procedure Action1Execute(Sender: TObject); -	процедура при нажатии на кнопке Открыть
procedure Action2Execute(Sender: TObject); -	процедура при нажатии на кнопке Данные -ввод информации в поля TEdit и построение графика (п.1.8)
procedure Action3Execute(Sender: TObject); -	процедура при нажатии на кнопке Разместить

		(п.1.9, 1.10)
procedure TObject);	Action4Execute(Sender: TObject);	- процедура при нажатии на кнопке Печать (п.1.11)
procedure TObject);	Action4Execute(Sender: TObject);	- процедура при нажатии на кнопке Печать п.1.11)
procedure TObject); -	Action1Execute(Sender: TObject); -	процедура при нажатии на кнопке Открыть (п.1.4)
procedure TObject);	Action2Execute(Sender: TObject);	- процедура при нажатии на кнопке Данные - ввод информации в поля TEdit и построение графика (п.1.8)
procedure TObject); -	Action3Execute(Sender: TObject); -	процедура при нажатии на кнопке Разместить (п.1.9, 1.10)
procedure TObject); -	Action4Execute(Sender: TObject); -	процедура при нажатии на кнопке Печать (п.1.11)
procedure MaxR; -		процедура поиска R-зубцов (п.1.9)
procedure MaxM; -		процедура поиска максимумов дифференцированной реограммы (п.1.9)
procedure GelZ;		- процедура поиска точек выброса крови из левого желудочка сердца (п.1.10)
procedure PrintGrafik; -		печать графиков (ФКГ, Рео, ЭКГ) (п.1.11)
procedure PrintZ; -)		обозначение точек выброса крови из левого желудочка сердца при печати (п.1.10, 1.11)
procedure PrintR; -		обозначение R-зубцов при печати (п.1.9, 1.11)
procedure PrintM; -		обозначение максимумов

	дифференцированной реограммы при печати (п.1.9, 1.11)
function WidthX: Integer;	- получение ширины страницы в пикселях для перевода координат при печати (п.1.7, 1.11)
function HeightY: Integer;	- получение высоты страницы в пикселях для перевода координат при печати (п.1.7, 1.11)
function П(x: smallint):integer; -	функция пересчёта координат по оси X (п.1.7)
function JJ(y,y1,y2: smallint):integer;	- функция пересчёта координат по оси Y (п.1.7)
function MinEKG: smallint; -	нахождение минимального значения ЭКГ (п.1.6)
function MaxEKG: smallint; -	нахождение максимального значения ЭКГ (п.1.6)
function MinReo: smallint; (п.1.6)	- нахождение минимального значения дифференцированной реограммы
function MaxReo: smallint; -	нахождение максимального значения дифференцированной реограммы (п.1.6)
function MinFKG: smallint; -	нахождение минимального значения ФКГ (п.1.6)
function MaxFKG: smallint; -	нахождение максимального значения ФКГ (п.1.6)
function PorogR: smallint; -) (п.1.9)	нахождение порогового значения (для поиска R-зубцов
function PorogM: smallint; -	нахождение порогового значения (для поиска максимумов

дифференцированной реограммы)
(п.1.9)

Тип:

TReo= **record**

RegNumber: smallint -	регистрационный номер;
Da_ta, Name: shortstring -	дата записи, ФИО;
Distance, Perimetr: smallint -	расстояние между электродами, обхват грудной клетки;
Length_s, Weight: smallint -	рост, вес;
Syst_BP, Diast_BP: smallint -	систолическое давление, диастолическое давление;
B: array[1..4,1..Predel] of smallint -	массивы сигналов ЭКГ, реограммы, ФКГ, опорный сигнал;
Kalibr_Z0, Kalibr_DZ: Real48 -	значения калибровки;

Глобальные переменные:

Reo1: TReo; -	переменная, в которую заносятся данные из файла
F: file of TReo; -	файловая переменная (п.1.4.)
I1, I2: integer -	размеры изображения по горизонтальной оси. Эти параметры необходимы для пересчёта координат. $ I2-I1 $ - ширина изображения. (п.1.6.)
J1, J2: integer -	размеры изображения по вертикальной оси. Эти параметры необходимы для пересчёта координат. $ J2-J1 $ - высота изображения. (п.1.6.)

ARR: array[1..80] of smallint-		- массив, в который вносятся абсциссы максимумов дифференцированной реограммы (п.1.10.)
h1: smallint; -	} Необходимы для правильного расположения графиков по вертикали (п.1.7.)	сумма модулей глобального максимума и минимума графиков ЭКГ, РЕО и ФКГ
h2: smallint; -		сумма модулей глобальных минимумов графиков ЭКГ, РЕО и глобального максимума ЭКГ
h2: smallint; -		сумма модулей глобальных минимумов графиков ЭКГ, РЕО, ФКГ и глобальных максимумов РЕО и ЭКГ
Константа:		
Predef=7000 -		количество точек в массиве

3.2. Работа с файлами

В типизированных файлах данные хранятся в виде структуры, представляющие какой-либо тип. В данном случае - запись. Для начала работы с файлом необходимо инициализировать файловую переменную этого типа (в данной программе она обозначается как F). Файловая переменная задаётся с помощью структуры file of.

F:=file of TReo.

Связывание файловой переменной с именем файла происходит при помощи процедуры AssignFile(<файловая переменная>,<название файла>). Файл открывается с помощью процедуры Reset(<файловая переменная>).

Считывание данных из файла переменной в переменную для сохранения прочитанных данных происходит с помощью переменной Read(<файловая переменная>,<переменная для сохранения непрочитанных данных>).

После считывания данных из файла в независимую переменную файл нужно закрыть с помощью метода CloseFile (<файловая переменная>).

3.3. Обработка исключительных ситуаций при открытии файла

Исключительные ситуации в среде программирования Delphi управляются с помощью нескольких операторов. В данной программе использовался оператор try...except.

Так как при открытии файла могут возникать различные исключительные ситуации, то был применён базовый класс исключений Exception.

```
try
  AssignFile(F,OpenDialog1.FileName);      ///Открытие файла
  Reset(F);
  Read(F,Reo1);                            ///Чтение данных из файла в переменную
  CloseFile(F);                             ///Заккрытие файла
except
  on E:Exception do ShowMessage(E.Message); /// вызов сообщения об
ошибке
end;
end;
```

Сообщение об ошибке изображено на иконке



При нажатии на кнопку ОК, окошко ошибки закрывается и основная программа снова становится доступна. Программа имеет вид, представленный на рис. 16-18.

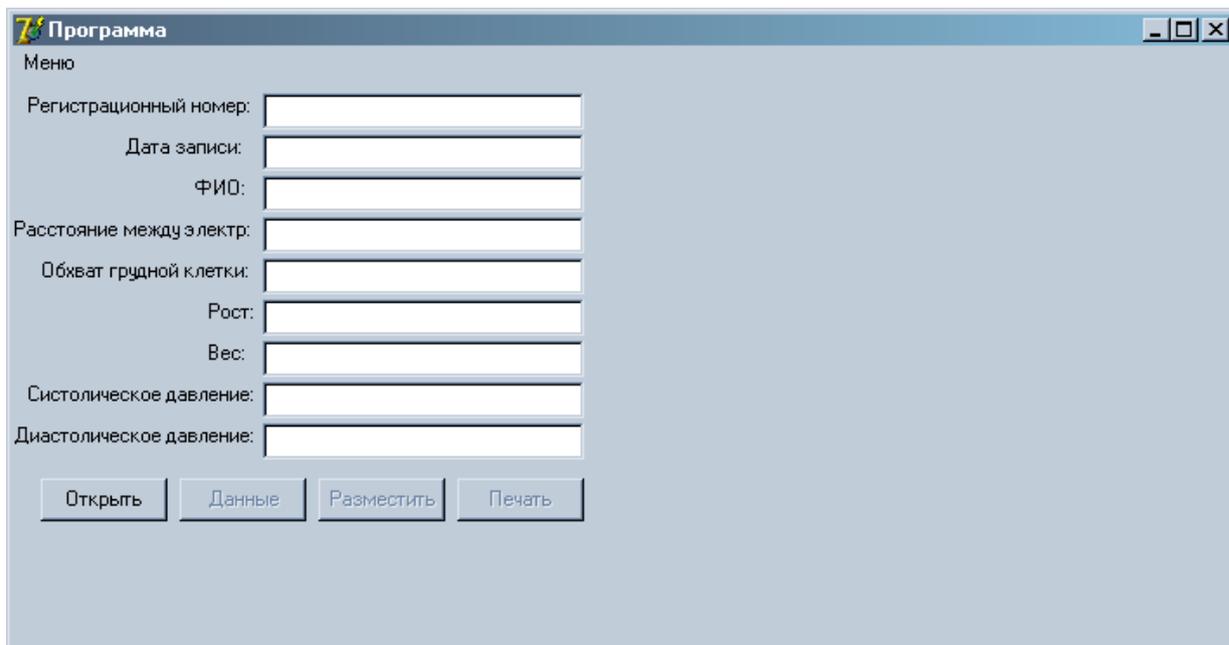


Рис.20. Окно программы до открытия файла.

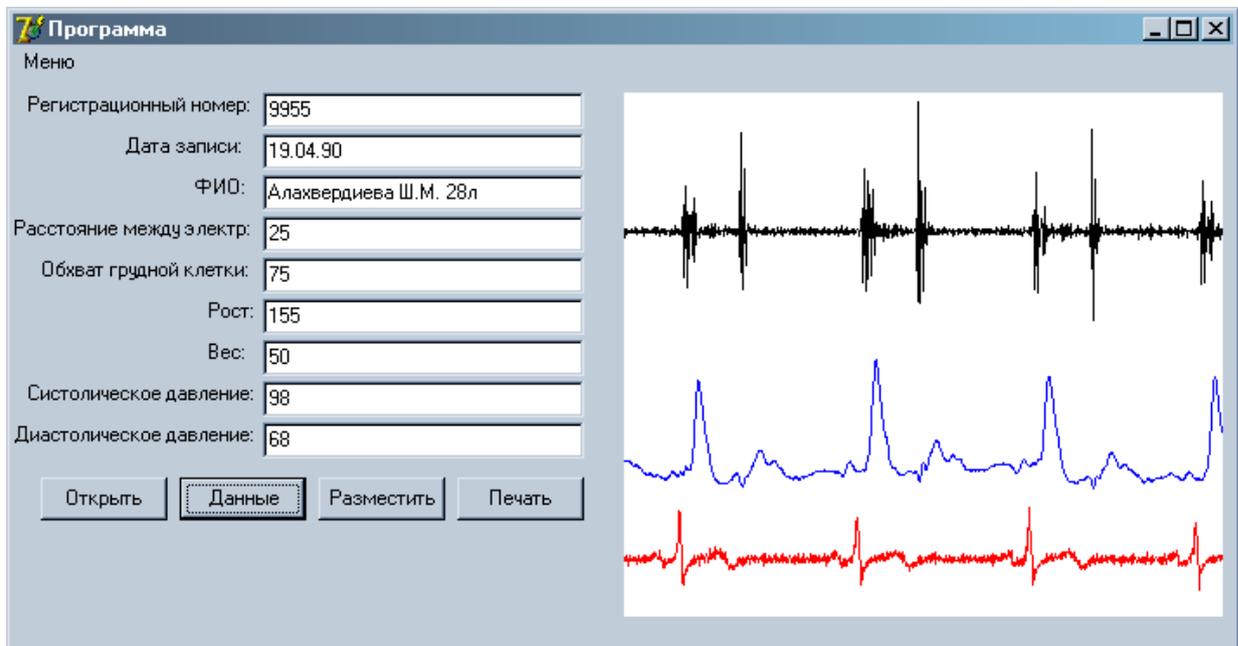


Рис.21. Окно программы после нажатия на кнопку Данные

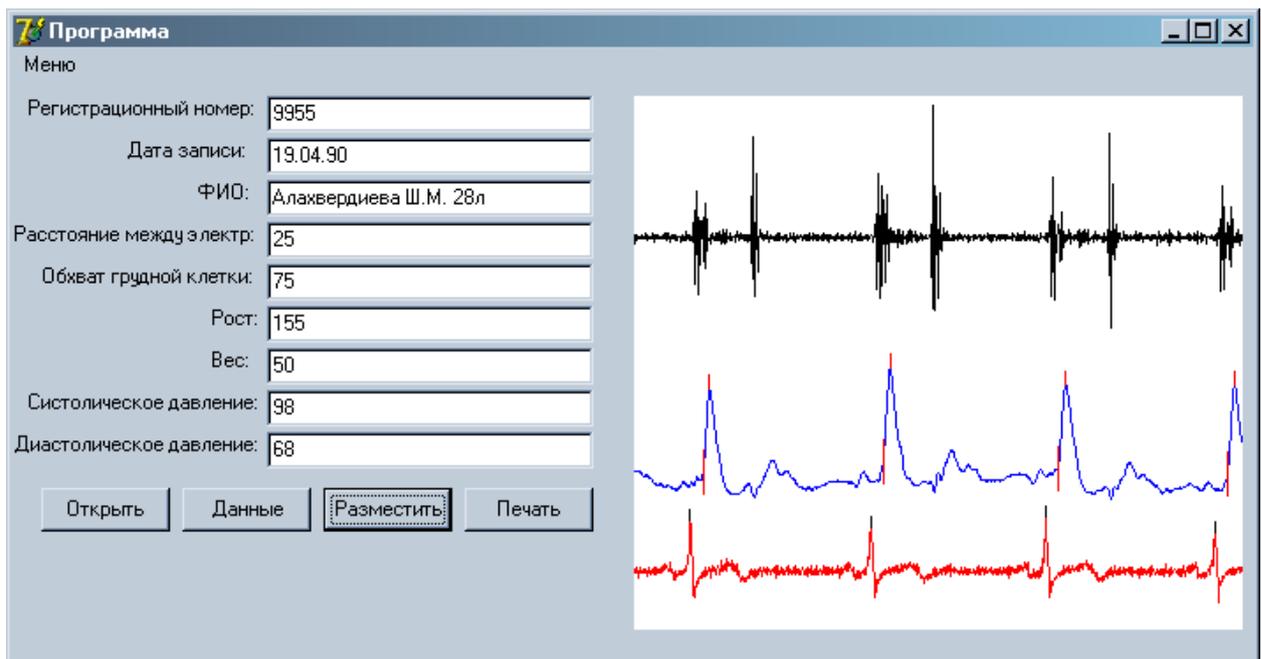


Рис. 22. Окно программы после нажатия на кнопку Разместить

3.4. Работа с канвой принтера

Для работы с принтером необходимо подключить модуль `Printers`. Начало и окончание печати сигнализируются операторами `BeginDoc` и `EndDoc`.

Для вывода текста применяются метод канвы принтера `TextOut(X,Y:integer; Текст:string)`. Для печати графиков применяются методы `MoveTo` и `LineTo`. Их действие описано в п.2.3.

Пересчёт координат с экрана на лист осуществляется с помощью специальных функций.

`Result:=GetDeviceCaps(Printer.Canvas.Handle, VertRes)` – получение высоты страницы в пикселях, заданной в свойствах принтера. Метод `Handle` берёт эту информацию из `API Windows`. Аналогично действует получение ширины страницы, но вместо параметра `VertRes` используется `HorzRes`.

`Result:=round(GetDeviceCaps(printer.Handle,LogPixelsX)/25.4*x)` – преобразование координат по оси X с экрана на бумагу по формуле: $X_{бум} = PPI \cdot X_{экр}$, где PPI – количество пикселей на дюйм. Т.к. в русской системе предпочтительнее использовать миллиметры, то значение PPI надо разделить на количество миллиметров в дюйме – 25.4 мм. `LogPixelsX` возвращает PPI по оси X . Аналогично происходит пересчёт координат заменой X на Y .

Выводы к главе 3

Приведены сведения по работе с программой по оцифровыванию географического аналогового сигнала и иллюстрации интерфейса программного продукта.

Заключение

Составлен обзор реографической методики и видов реографии. Сделан вывод о перспективности реографии в исследовании центральной и местной гемодинамики, как о точном методе.

Разработаны алгоритмы совместной обработки физиологических параметров реограммы и кардиограммы различными математическими методами.

Приведены сведения по работе с программой по оцифровыванию реографического аналогового сигнала и иллюстрации интерфейса программного продукта.

Список литературы

1. Инструментальные методы исследования сердечно-сосудистой системы /Под ред. Т.С. Виноградовой. – М.: Медицина, 1996.
2. Полищук В.И., Терехова Л.Г. Техника и методика реографии и реоплетизмографии. – М.: Медицина, 1993.
3. Яруллин Х.Х. Клиническая реоэнцефалография. М., Медицина, 1993, 270 С.
4. Ремизов А.Н. Медицинская и биологическая физика.- М.: Высш. шк., 1987.
5. Сапин М.Р., Билич Г.Л. Анатомия человека. – М.: Высш. шк., 1989
6. Попечителев Е.П. Инженерные аспекты медико-биологических исследований. – Л.: ЛЭТИ, 1985.
7. Г.П. Матвейков, С.С. Пшоник. Клиническая реография. Минск, “Беларусь”, 1986.-175 С.
8. Информационно-измерительная техника и технологии: Учеб. для вузов/В.И. Калашников, С.В. Нефедов, А.Б. Путилин и др./Под ред. Г.Г. Раннепа. — М.: Высшая школа, 2002. — 454 с.
9. Медицинские приборы. Разработки и проектирование. — М.: Медицинская книга, 2004. — 720 с.
10. Эман А.А. Биофизические основы измерения артериального давления.—Л.: Медицина, 1983. — 126 с.
- 11.Сергиенко А.Б. Цифровая обработка сигналов. - СШ.: Пинтер, 2002.-608 с.
12. Яковлев П.В., Яковлев В.М. Клинико-патофизиологическая интерпретация электрокардиограмм. - Н.Новгород: Изд-во НГМА, Москва: Медицинская книга, 1999. - 100 с.
13. Колпаков Ф.Ф. Ультразвуковые измерительные преобразователи медицинских диагностических систем: Консп. лекций по курсу

"Биомедицинские измерительные преобразователи". - Х.: Гос. аэрокосм. ун-т «Харьк. авиац. ин-т», 2000.

14. Мустецов Н.П. Инструментальные методы медико-биологических исследований: Учеб. пособие. – Х.: ХТУРЭ, 1999.

15. Олейник В.П., Кулиш С.Н., Овчаренко В.Е. Методы медико-биологических исследований: Учеб. пособие. – Х.: Нац. аэрокосм. ун-т «Харьк. авиац. ин-т», 2003.

16. Олейник В.П., Олейник В.Н., Кулиш С.Н. Методы медико-биологических исследований: Учеб. пособие по лаб. практикуму. – Х.: Гос. аэрокосм. ун-т «Харьк. авиац. ин-т», 1999.

17. Цегельникова А.Л. Методические указания по выполнению лабораторных работ. Реография. Обработка сигнала и расчёт гемодинамических показателей в системе Delphi. – М.:МИРЭА, 2004. – 20 с.

18. Полуавтоматическая и автоматическая расшифровка реограмм. Метод. рекомен. МЗ Составители: Молоканов Н.Я., Милягин В.А., Стельмах В.М. Смоленск, 1988. 21 С.

19. Комплексная оценка функционального состояния систем кровообращения и дыхания методом интегральной реографии тела. Метод. рекомен. МЗ Российской Федерации. Составители: Волков Ю.Н., Большов В.М., Сингаевский С.Б., Земцовский Э.В., Гуссейнов Б.А. М., 1989. 21С.

20. Комплексная оценка функционального состояния гемодинамики методом импедансной электроплетизмографии и импедансометрии. Метод. рекомен. МЗ Российской Федерации. Составители: Е.А, Лужников, А.И. Ишмухаметов, Л.Г. Костомарова, А.С. Савина, К.К. Ильяшенко, А.А. Цветков, М., 1985, 33 С.

21. Методические рекомендации по определению основных параметров гемодинамики и тонуса сосудов методом тетраполярной, трансторакальной, импедансной реоплетизмографии. Метод. рекомен. МЗ Украины. Составители: Б.М, Щепотин, С.В. Волосюк, М.И. Гуревич, А.Г. Карцева,

Л.П. Литовченко, А.И. Соловьев, Л.Д. Фесенко, Л.Б. Долман. Киев, Наукова думка, 1980, 19 С.

22. Применение импедансной плетизмографии для оценки состояния центральной гемодинамики. Метод. рекомен. МЗ Белоруссии. Составители: Л.З. Полонецкий, В.М. Альхимович. Минск. 1985. 23 С.

23. Мальцев А. И. Основы линейной алгебры. М.: Наука, 1970.

24. Кетков Ю., Кетков А., Шульц М. MATLAB 6.x: программирование численных методов. СПб.: БХВ_Петербург, 2004.

25. Стечкин С. Б., Субботин Ю. Н. Сплайны в вычислительной математике. М. Наука. 1986. -248 с.

26. Неинвазивная диагностика нарушений периферического и церебрального кровообращения. Метод. рекомен. ЛОО при СМ. Составители: К.И. Овчаренко, В.П. Седов. Москва, 1990. - 47 С.

27. В.В. Карпенко, Е.А. Евдокимов Интегральная реография в оценке функции системы кровообращения. Лекция ЦОЛИУВ. Москва, 1985. -24 С.

28. <http://lainslav.narod.ru/> - Медицина – Реография.

29. Язык программирования Delphi

ПРИЛОЖЕНИЕ

Листинг основной программы.

```
unit Unit1;

interface

uses
  Windows, Classes, ActnList, Dialogs, Menus, StdCtrls, Controls, ExtCtrls,
  Printers, Messages, SysUtils, Variants, Graphics, Forms,
  Buttons, Grids;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    PopupMenu1: TPopupMenu;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Edit8: TEdit;
    Edit9: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    OpenFileDialog1: TOpenDialog;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    ActionList1: TActionList;
    Action1: TAction;
    Image1: TImage;
    Action2: TAction;
    Action3: TAction;
```

```

Action4: TAction;
procedure Action1Execute(Sender: TObject);
procedure Action2Execute(Sender: TObject);
procedure Action3Execute(Sender: TObject);
procedure Action4Execute(Sender: TObject);
procedure MaxR;
procedure MaxM;
procedure GeZ;
procedure PrintGrafik;
procedure PrintZ;
procedure PrintR;
procedure PrintM;
private
  { Private declarations }
public
  { Public declarations }
end;

function WidthX: Integer;
function HeightY: Integer;
function II(x:smallint):integer;
function JJ(y,y1,y2:smallint):integer;

const
  Predel=7000;

type TReo=record
  RegNumber:smallint;
  Da_ta,Name:shortstring;
  Distance,Perimetr:smallint;
  Leng_s,Weight:smallint;
  Sist_BP,Diast_BP:smallint;
  B:array[1..4,1..Predel] of smallint;
  Kalibr_Z0,Kalibr_DZ:Real48;
end;

var
  Form1: TForm1;
  Reo1: TReo;
  F:file of TReo;
  I1,I2,J1,J2:integer;
  ARR:array[1..80] of smallint;
  h1,h2,h3:smallint;

implementation

{$R *.dfm}

////процедура при нажатии на кнопке Открыть
procedure TForm1.Action1Execute(Sender: TObject);
begin

```

```

OpenDialog1.Filter:='Файлы данных (*.dat)|*.dat|Все файлы|*.*';
OpenDialog1.FilterIndex:=1;
OpenDialog1.Title:='Выбор файла';
if OpenDialog1.Execute then
begin
try
AssignFile(F,OpenDialog1.FileName);      ////Открытие файла
Reset(F);
Read(F,Reo1);                            ////Чтение данных из файла в переменную
CloseFile(F);                             ////Закрытие файла
Button2.Enabled:=True;
Button3.Enabled:=True;
Button4.Enabled:=True;
N3.Enabled:=True;
N4.Enabled:=True;
N5.Enabled:=True;
N7.Enabled:=True;
N8.Enabled:=True;
N9.Enabled:=True;
except                                     ////Обработка исключительных ситуаций
on E:EinOutError do ShowMessage(E.Message);
end;
end;

////Нахождение минимального значения ЭКГ
function MinEKG:smallint;
var
j:integer;
begin
Result:=Reo1.B[1,1];
for j:=1 to Predel do
if Reo1.B[1,j]<Result then Result:=Reo1.B[1,j];
end;

////Нахождение максимального значения ЭКГ
function MaxEKG:smallint;
var
j:integer;
begin
Result:=Reo1.B[1,1];
for j:=1 to Predel do
if Reo1.B[1,j]>Result then Result:=Reo1.B[1,j]
end;

////Нахождение минимального значения дифференцированной реограммы
function MinReo:smallint;
var
j:integer;
begin
Result:=Reo1.B[2,1];
for j:=1 to Predel do

```

```
    if Reo1.B[2,j]<Result then Result:=Reo1.B[2,j];
end;
```

```
////Нахождение максимального значения дифференцированной реограммы
```

```
function MaxReo:smallint;
var
    j:integer;
begin
    Result:=Reo1.B[2,1];
    for j:=1 to Predel do
        if Reo1.B[2,j]>Result then Result:=Reo1.B[2,j]
    end;
```

```
////Нахождение минимального значения ФКГ
```

```
function MinFKG:smallint;
var
    j:integer;
begin
    Result:=Reo1.B[3,1];
    for j:=1 to Predel do
        if Reo1.B[3,j]<Result then Result:=Reo1.B[3,j];
    end;
```

```
////Нахождение максимального значения ФКГ
```

```
function MaxFKG:smallint;
var
    j:integer;
begin
    Result:=Reo1.B[3,1];
    for j:=1 to Predel do
        if Reo1.B[3,j]>Result then Result:=Reo1.B[3,j]
    end;
```

```
////функция пересчёта координат по оси X
```

```
function П(x:smallint):integer;
var
    x1,x2:integer;
begin
    x1:=1;
    x2:=1000;
    П:=П1+Trunc((x-x1)*(I2-I1)/(x2-x1));
end;
```

```
////функция пересчёта координат по оси Y
```

```
function JJ(y,y1,y2:smallint):integer;
begin
    JJ:=J2+Trunc((y-y1)*(J1-J2)/(y2-y1));
end;
```

```
////процедура при нажатии на кнопке Данные - ввод информации в поля TEdit и построение графика
procedure TForm1.Action2Execute(Sender: TObject);
```

```

var
  j:integer;
  A:array[0..30] of char;
begin
  Image1.Picture:=nil;          ///Очистка TImage1
  Edit1.Text:=IntToStr(Reo1.RegNumber);    ///Ввод данных в TEdit
  Edit2.Text:=Reo1.Da_ta;
  OemToChar(StrPCopy(A,Reo1.Name),A);      ///Перекодировка строки из DOS в Win-1251
  Edit3.Text:=A;
  Edit4.Text:=IntToStr(Reo1.Distance);
  Edit5.Text:=IntToStr(Reo1.Perimetr);
  Edit6.Text:=IntToStr(Reo1.Leng_s);
  Edit7.Text:=IntToStr(Reo1.Weight);
  Edit8.Text:=IntToStr(Reo1.Sist_BP);
  Edit9.Text:=IntToStr(Reo1.Diast_BP);
  I1:=0;          ///Ввод координат, в пределах которых будет происходить
  построение графиков
  I2:=Image1.Width;
  J1:=0;
  J2:=Round(Image1.Height)-30;

  h1:=Abs(MaxEKG)+Abs(MinEKG)+Abs(MaxFKG)+Abs(MinFKG)+Abs(MaxReo)+Abs(MinReo);
  h2:=Abs(MaxEKG)+Abs(MinEKG)+Abs(MinReo);
  h3:=Abs(MaxEKG)+Abs(MinEKG)+Abs(MaxReo)+Abs(MinReo)+Abs(MinFKG);
  Image1.Canvas.Pen.Color:=clRed;
  Image1.Canvas.MoveTo(I1,JJ(Reo1.B[1,1],0,h1));
  for j:=1 to 1000 do          ///Построение графиков
  begin
    Image1.Canvas.LineTo(I1(j+1),JJ(Reo1.B[1,j+1],0,h1));
  end;
  Image1.Canvas.Pen.Color:=clBlue;
  Image1.Canvas.MoveTo(I1,JJ(Reo1.B[2,1]+h2,0,h1)+5);
  for j:=1 to 1000 do
  begin
    Image1.Canvas.LineTo(I1(j+1),JJ(Reo1.B[2,j+1]+h2,0,h1)+5);
  end;
  Image1.Canvas.Pen.Color:=clBlack;
  Image1.Canvas.MoveTo(I1,JJ(Reo1.B[3,1]+h3,0,h1)+5);
  for j:=1 to 1000 do
  begin
    Image1.Canvas.LineTo(I1(j+1),JJ(Reo1.B[3,j+1]+h3,0,h1)+5);
  end;
end;

///нахождение порогового значения (для поиска R-зубцов)
function PorogR:smallint;
var
  j:integer;
  sum,max:smallint;
begin
  sum:=0;

```

```

for j:=1 to Predel do
  sum:=sum+Reo1.B[1,j];
max:=Reo1.B[1,1];
for j:=1 to Predel do
  if Reo1.B[1,j]>max then max:=Reo1.B[1,j];
Result:=Round(sum/Predel+max/2);
end;

///нахождение порогового значения (для поиска M-зубцов)
function PorogM:smallint;
var
  j:integer;
  sum,max:smallint;
begin
  sum:=0;
  for j:=1 to Predel do
    sum:=sum+Reo1.B[2,j];
    max:=Reo1.B[2,1];
  for j:=1 to Predel do
    if Reo1.B[2,j]>max then max:=Reo1.B[2,j];
  Result:=Round(sum/Predel+max/2);
end;

///нахождение R-зубцов
procedure TForm1.MaxR;
var
  j,Rx,Ry,p,p1,p2,k:integer;
begin
  Image1.Canvas.Brush.Style:=bsClear;
  Image1.Canvas.Font.Color:=clBlack;
  Image1.Canvas.Pen.Color:=clBlack;
  for j:=1 to 1000 do
    if Reo1.B[1,j]<PorogR
    then
      begin
        if PorogR<Reo1.B[1,j+1]
        then
          begin
            p1:=j+1;
            for k:=p1 to 1000 do
              if Reo1.B[1,k]>PorogR
              then
                if Reo1.B[1,k+1]<PorogR
                then
                  begin
                    p2:=k;
                    Rx:=p1;
                    Ry:=Reo1.B[1,p1];
                    for p:=p1 to p2 do
                      if Reo1.B[1,p]>Ry
                      then begin Ry:=Reo1.B[1,p]; Rx:=p; end;
                    Image1.Canvas.MoveTo(II(Rx),JJ(Ry,0,h1)-7);

```

```

        Image1.Canvas.LineTo(П(Rx),JJ(Ry,0,h1));
    end;
end;
end;
end;

////нахождение максимумов дифференцированной реограммы
procedure TForm1.MaxM;
var
    a,i,Mx,My,r,r1,r2,m: integer;
begin
    Image1.Canvas.Brush.Style:=bsClear;
    Image1.Canvas.Pen.Color:=clRed;
    ARR[1]:=1;
    a:=2;
    for i:=1 to 1000 do
        if Reo1.B[2,i]<PorogM
        then
            begin
                if PorogM<=Reo1.B[2,i+1]
                then
                    begin
                        r1:=i+1;
                        for m:=r1 to 1000 do
                            if Reo1.B[2,m]>=PorogM
                            then
                                if Reo1.B[2,m+1]<PorogM
                                then
                                    begin
                                        r2:=m;
                                        Mx:=r1;
                                        My:=Reo1.B[2,r1];
                                        for r:=r1 to r2 do
                                            if Reo1.B[2,r]>My
                                            then begin My:=Reo1.B[2,r]; Mx:=r; end;
                                        ARR[a]:=Mx; a:=a+1;          ////заполнение массива координатами точек
                                        максимума дифференцированной реограммы по оси X (это нужно для нахождения точек
                                        выброса крови из левого желудочка)
                                        Image1.Canvas.MoveTo(П(Mx),JJ(My+h2,0,h1)+5);
                                        Image1.Canvas.LineTo(П(Mx),JJ(My+h2,0,h1)-4);
                                        Break;
                                    end;
                                end;
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;

////Нахождение точек выбрасывания крови из левого желудочка сердца
procedure TForm1.GetZ;
var
    b,c,Px,Py,tinteger;
    Max:real;
begin

```

```

Image1.Canvas.Brush.Style:=bsClear;
Image1.Canvas.Pen.Color:=clRed;
Px:=1;
Py:=1;
for b:=1 to 80 do
if ARR[b]<ARR[b+1]
then
begin
Max:=0.00001;
for c:=ARR[b] to ARR[b+1] do
begin
if (Reo1.B[2,c+1]-Reo1.B[2,c-1])/2>Max
then
begin
Max:=(Reo1.B[2,c+1]-Reo1.B[2,c-1])/2;
Px:=c;
Py:=Reo1.B[2,c];
end;
end;
t:=Round(Px-Py/Max);
Image1.Canvas.MoveTo(П(t),JJ(Reo1.B[2,t]+h2,0,h1)+20);
Image1.Canvas.LineTo(П(t),JJ(Reo1.B[2,t]+h2,0,h1)-6);
end
else Break;
end;

```

```

////процедура при нажатии на кнопку Разместить
procedure TForm1.Action3Execute(Sender: TObject);

```

```

begin
MaxR;
MaxM;
GeZ;
end;

```

```

////функция получения высоты страницы в пикселях
function HeightY: Integer;
begin
Result:=GetDeviceCaps(Printer.Canvas.Handle, VertRes)
end;

```

```

////функция получения ширины страницы в пикселях
function WidthX: Integer;
begin
Result:=GetDeviceCaps(Printer.Canvas.Handle, HorzRes)
end;

```

```

////функция преобразования координат с бумаги по оси X
function PrinterCoordX(x:integer):integer;
begin
Result:=round(GetDeviceCaps(printer.Handle,LogPixelsX)/25.4*x);
end;

```

```

///функция преобразования координат с бумаги по оси Y
function PrinterCoordY(Y:integer):integer;
begin
  Result:=round(GetDeviceCaps(printer.Handle,LogPixelsY)/25.4*Y);
end;

///печать графиков
procedure TForm1.PrintGrafik;
var
  j:integer;
begin
  I1:=0;
  I2:=Round(WidthX);
  J1:=0;
  J2:=Round(HeightY/2);
  Printer.Canvas.MoveTo(I1,PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[1,1],0,h1));
  for j:=1 to 1000 do
    begin
      Printer.Canvas.LineTo(I1(j+1),PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[1,j+1],0,h1));
    end;
  Printer.Canvas.MoveTo(I1,PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[2,1]+h2,0,h1)+5);
  for j:=1 to 1000 do
    begin

Printer.Canvas.LineTo(I1(j+1),PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[2,j+1]+h2,0,h1)+5);
    end;
  Printer.Canvas.MoveTo(I1,PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[3,1]+h3,0,h1)+5);
  for j:=1 to 1000 do
    begin

Printer.Canvas.LineTo(I1(j+1),PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[3,j+1]+h3,0,h1)+5);
    end;
  end;

///печать R-зубцов
procedure TForm1.PrintR;
var
  j,Rx,Ry,p,p1,p2,k:integer;
begin
  Printer.Canvas.Brush.Style:=bsClear;
  I1:=0;
  I2:=Round(WidthX);
  J1:=0;
  J2:=Round(HeightY/2);
  for j:=1 to 1000 do
    if Reo1.B[1,j]<PorogR
    then
      begin
        if PorogR<Reo1.B[1,j+1]
        then
          begin
            p1:=j+1;

```

```

for k:=p1 to 1000 do
  if Reo1.B[1,k]>PorogR
  then
    if Reo1.B[1,k+1]<PorogR
    then
      begin
        p2:=k;
        Rx:=p1;
        Ry:=Reo1.B[1,p1];
        for p:=p1 to p2 do
          if Reo1.B[1,p]>Ry
          then begin Ry:=Reo1.B[1,p]; Rx:=p; end;
        Printer.Canvas.MoveTo(II(Rx),PrinterCoordY(10+9*Font.Size)+JJ(Ry,0,h1)-20);
        Printer.Canvas.LineTo(II(Rx),PrinterCoordY(10+9*Font.Size)+JJ(Ry,0,h1)+30);
        end;
      end;
    end;
  end;
end;

```

////печать точек выброса крови из левого желудочка

```

procedure TForm1.PrintZ;
var
  b,c,Px,Py,t:integer;
  Max:real;
begin
  Printer.Canvas.Brush.Style:=bsClear;
  I1:=0;
  I2:=Round(WidthX);
  J1:=0;
  J2:=Round(HeightY/2);
  Px:=1;
  Py:=1;
  for b:=1 to 80 do
    if ARR[b]<ARR[b+1]
    then
      begin
        Max:=1;
        for c:=ARR[b] to ARR[b+1]do
          begin
            if (Reo1.B[2,c+1]-Reo1.B[2,c-1])/2>Max
            then
              begin
                Max:=(Reo1.B[2,c+1]-Reo1.B[2,c-1])/2;
                Px:=c;
                Py:=Reo1.B[2,c];
              end;
            end;
          t:=Round(Px-Py/Max);
          Printer.Canvas.MoveTo(II(t),PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[2,t]+h2,0,h1)-40);
          Printer.Canvas.LineTo(II(t),PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[2,t]+h2,0,h1)+40);
          end
        else Break;
      end;

```

end;

///
печать максимумов дифференцированной реограммы

procedure TForm1.PrintM;

var

a,i,Mx,My,r,r1,r2,m:integer;

begin

Printer.Canvas.Brush.Style:=bsClear;

I1:=0;

I2:=Round(WidthX);

J1:=0;

J2:=Round(HeightY/2);

ARR[1]:=1;

a:=2;

for i:=1 to 1000 do

if Reo1.B[2,i]<PorogM

then

begin

if PorogM<=Reo1.B[2,i+1]

then

begin

r1:=i+1;

for m:=r1 to 1000 do

if Reo1.B[2,m]>=PorogM

then

if Reo1.B[2,m+1]<PorogM

then

begin

r2:=m;

Mx:=r1;

My:=Reo1.B[2,r1];

for r:=r1 to r2 do

if Reo1.B[2,r]>My

then begin My:=Reo1.B[2,r]; Mx:=r; end;

ARR[a]:=Mx; a:=a+1;

Printer.Canvas.MoveTo(I(Mx),PrinterCoordY(10+9*Font.Size)+JJ(My+h2,0,h1)-20);

Printer.Canvas.LineTo(I(Mx),PrinterCoordY(10+9*Font.Size)+JJ(My+h2,0,h1)+30);

Break;

end;

end;

end;

end;

///
процедура при нажатии на кнопку Печать

procedure TForm1.Action4Execute(Sender: TObject);

begin

Printer.BeginDoc;

Printer.Canvas.TextOut(PrinterCoordX(10), PrinterCoordY(10),Label1.Caption+' '+Edit1.Text);

Printer.Canvas.TextOut(PrinterCoordX(10), PrinterCoordY(10+Round(Font.Size/2)),Label2.Caption+' '+Edit2.Text);

Printer.Canvas.TextOut(PrinterCoordX(10),

```

PrinterCoordY(10+Round(2*Font.Size/2)),Label3.Caption+' '+Edit3.Text);
  Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(3*Font.Size/2)),Label4.Caption+' '+Edit4.Text);
  Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(4*Font.Size/2)),Label5.Caption+' '+Edit5.Text);
  Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(5*Font.Size/2)),Label6.Caption+' '+Edit6.Text);
  Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(6*Font.Size/2)),Label7.Caption+' '+Edit7.Text);
  Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(7*Font.Size/2)),Label8.Caption+' '+Edit8.Text);
  Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(8*Font.Size/2)),Label9.Caption+' '+Edit9.Text);
  PrintGrafik;
  PrintR;
  PrintM;
  PrintZ;
  Printer.EndDoc;
end;

end.

```

Приложение А. Листинг основной программы.

```

unit Unit1;

interface

uses
  Windows, Classes, ActnList, Dialogs, Menus, StdCtrls, Controls, ExtCtrls,
  Printers, Messages, SysUtils, Variants, Graphics, Forms,
  Buttons, Grids;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    PopupMenu1: TPopupMenu;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Edit8: TEdit;
    Edit9: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;

```

```

Label8: TLabel;
Label9: TLabel;
OpenDialog1: TOpenDialog;
Button1: TButton;
Button2: TButton;
Button3: TButton;
Button4: TButton;
N1: TMenuItem;
N2: TMenuItem;
N3: TMenuItem;
N4: TMenuItem;
N5: TMenuItem;
N6: TMenuItem;
N7: TMenuItem;
N8: TMenuItem;
N9: TMenuItem;
ActionList1: TActionList;
Action1: TAction;
Image1: TImage;
Action2: TAction;
Action3: TAction;
Action4: TAction;
procedure Action1Execute(Sender: TObject);
procedure Action2Execute(Sender: TObject);
procedure Action3Execute(Sender: TObject);
procedure Action4Execute(Sender: TObject);
procedure MaxR;
procedure MaxM;
procedure GeIZ;
procedure PrintGrafik;
procedure PrintZ;
procedure PrintR;
procedure PrintM;
private
  { Private declarations }
public
  { Public declarations }
end;

function WidthX: Integer;
function HeightY: Integer;
function II(x:smallint):integer;
function JJ(y,y1,y2:smallint):integer;

const
  PredeI=7000;

type TReo=record
  RegNumber:smallint;
  Da_ta,Name:shortstring;
  Distance,Perimetr:smallint;
  Leng_s,Weight:smallint;

```

```

Sist_BP,Diast_BP:smallint;
B:array[1..4,1..Prede1] of smallint;
Kalibr_Z0,Kalibr_DZ:Real48;
end;

```

```

var
  Form1: TForm1;
  Reo1: TReo;
  F:file of TReo;
  I1,I2,J1,J2:integer;
  ARR:array[1..80] of smallint;
  h1,h2,h3:smallint;

```

implementation

```
{$R *.dfm}
```

```

////процедура при нажатии на кнопке Открыть
procedure TForm1.Action1Execute(Sender: TObject);
begin
  OpenFileDialog.Filter:='Файлы данных (*.dat)*.dat|Все файлы|*.*';
  OpenFileDialog.FilterIndex:=1;
  OpenFileDialog.Title:='Выбор файла';
  if OpenFileDialog.Execute then
    begin
      try
        AssignFile(F,OpenDialog1.FileName);      ////Открытие файла
        Reset(F);
        Read(F,Reo1);                            ////Чтение данных из файла в переменную
        CloseFile(F);                            ////Закрытие файла
        Button2.Enabled:=True;
        Button3.Enabled:=True;
        Button4.Enabled:=True;
        N3.Enabled:=True;
        N4.Enabled:=True;
        N5.Enabled:=True;
        N7.Enabled:=True;
        N8.Enabled:=True;
        N9.Enabled:=True;
      except                                     ////Обработка исключительных ситуаций
        on E:EinOutError do ShowMessage(E.Message);
      end;
    end;
  end;
end;

////Нахождение минимального значения ЭКГ
function MinEKG:smallint;
var
  j:integer;
begin
  Result:=Reo1.B[1,1];

```

```

for j:=1 to Predel do
if Reo1.B[1,j]<Result then Result:=Reo1.B[1,j];
end;

```

```

///Нахождение максимального значения ЭКГ
function MaxEKG:smallint;

```

```

var
j:integer;
begin
Result:=Reo1.B[1,1];
for j:=1 to Predel do
if Reo1.B[1,j]>Result then Result:=Reo1.B[1,j]
end;

```

```

///Нахождение минимального значения дифференцированной реограммы
function MinReo:smallint;

```

```

var
j:integer;
begin
Result:=Reo1.B[2,1];
for j:=1 to Predel do
if Reo1.B[2,j]<Result then Result:=Reo1.B[2,j];
end;

```

```

///Нахождение максимального значения дифференцированной реограммы
function MaxReo:smallint;

```

```

var
j:integer;
begin
Result:=Reo1.B[2,1];
for j:=1 to Predel do
if Reo1.B[2,j]>Result then Result:=Reo1.B[2,j]
end;

```

```

///Нахождение минимального значения ФКГ
function MinFKG:smallint;

```

```

var
j:integer;
begin
Result:=Reo1.B[3,1];
for j:=1 to Predel do
if Reo1.B[3,j]<Result then Result:=Reo1.B[3,j];
end;

```

```

///Нахождение максимального значения ФКГ
function MaxFKG:smallint;

```

```

var
j:integer;
begin
Result:=Reo1.B[3,1];
for j:=1 to Predel do
if Reo1.B[3,j]>Result then Result:=Reo1.B[3,j]

```

```

end;

////функция пересчёта координат по оси X
function П(x:smallint):integer;
var
  x1,x2:integer;
begin
  x1:=1;
  x2:=1000;
  П:=I1+Trunc((x-x1)*(I2-I1)/(x2-x1));
end;

////функция пересчёта координат по оси Y
function JJ(y,y1,y2:smallint):integer;
begin
  JJ:=J2+Trunc((y-y1)*(J1-J2)/(y2-y1));
end;

////процедура при нажатии на кнопке Данные - ввод информации в поля TEdit и
построение графика
procedure TForm1.Action2Execute(Sender: TObject);
var
  j:integer;
  A:array[0..30] of char;
begin
  Image1.Picture:=nil;          ////Очистка TImage1
  Edit1.Text:=IntToStr(Reo1.RegNumber);  ////Ввод данных в TEdit
  Edit2.Text:=Reo1.Da_ta;
  OemToChar(StrPCopy(A,Reo1.Name),A);    ////Перекодировка строки из DOS в Win-1251
  Edit3.Text:=A;
  Edit4.Text:=IntToStr(Reo1.Distance);
  Edit5.Text:=IntToStr(Reo1.Perimetr);
  Edit6.Text:=IntToStr(Reo1.Leng_s);
  Edit7.Text:=IntToStr(Reo1.Weight);
  Edit8.Text:=IntToStr(Reo1.Sist_BP);
  Edit9.Text:=IntToStr(Reo1.Diast_BP);
  I1:=0;          ////Ввод координат, в пределах которых будет происходить
построение графиков
  I2:=Image1.Width;
  J1:=0;
  J2:=Round(Image1.Height)-30;

h1:=Abs(MaxEKG)+Abs(MinEKG)+Abs(MaxFKG)+Abs(MinFKG)+Abs(MaxReo)+Abs(MinR
eo);
h2:=Abs(MaxEKG)+Abs(MinEKG)+Abs(MinReo);
h3:=Abs(MaxEKG)+Abs(MinEKG)+Abs(MaxReo)+Abs(MinReo)+Abs(MinFKG);
Image1.Canvas.Pen.Color:=clRed;
Image1.Canvas.MoveTo(П(1),JJ(Reo1.B[1,1],0,h1));
for j:=1 to 1000 do          ////Построение графиков
begin
  Image1.Canvas.LineTo(П(j+1),JJ(Reo1.B[1,j+1],0,h1));
end;

```

```

Image1.Canvas.Pen.Color:=clBlue;
Image1.Canvas.MoveTo(II(1),JJ(Reo1.B[2,1]+h2,0,h1)+5);
for j:=1 to 1000 do
begin
Image1.Canvas.LineTo(II(j+1),JJ(Reo1.B[2,j+1]+h2,0,h1)+5);
end;
Image1.Canvas.Pen.Color:=clBlack;
Image1.Canvas.MoveTo(II(1),JJ(Reo1.B[3,1]+h3,0,h1)+5);
for j:=1 to 1000 do
begin
Image1.Canvas.LineTo(II(j+1),JJ(Reo1.B[3,j+1]+h3,0,h1)+5);
end;
end;

```

///нахождение порогового значения (для поиска R-зубцов)

```

function PorogR:smallint;
var
j:integer;
sum,max:smallint;
begin
sum:=0;
for j:=1 to Predel do
sum:=sum+Reo1.B[1,j];
max:=Reo1.B[1,1];
for j:=1 to Predel do
if Reo1.B[1,j]>max then max:=Reo1.B[1,j];
Result:=Round(sum/Predel+max/2);
end;

```

///нахождение порогового значения (для поиска M-зубцов)

```

function PorogM:smallint;
var
j:integer;
sum,max:smallint;
begin
sum:=0;
for j:=1 to Predel do
sum:=sum+Reo1.B[2,j];
max:=Reo1.B[2,1];
for j:=1 to Predel do
if Reo1.B[2,j]>max then max:=Reo1.B[2,j];
Result:=Round(sum/Predel+max/2);
end;

```

///нахождение R-зубцов

```

procedure TForm1.MaxR;
var
j,Rx,Ry,p,p1,p2,k:integer;
begin
Image1.Canvas.Brush.Style:=bsClear;
Image1.Canvas.Font.Color:=clBlack;
Image1.Canvas.Pen.Color:=clBlack;

```

```

for j:=1 to 1000 do
  if Reo1.B[1,j]<PorogR
  then
  begin
  if PorogR<Reo1.B[1,j+1]
  then
  begin
  p1:=j+1;
  for k:=p1 to 1000 do
  if Reo1.B[1,k]>PorogR
  then
  if Reo1.B[1,k+1]<PorogR
  then
  begin
  p2:=k;
  Rx:=p1;
  Ry:=Reo1.B[1,p1];
  for p:=p1 to p2 do
  if Reo1.B[1,p]>Ry
  then begin Ry:=Reo1.B[1,p]; Rx:=p; end;
  Image1.Canvas.MoveTo(II(Rx),JJ(Ry,0,h1)-7);
  Image1.Canvas.LineTo(II(Rx),JJ(Ry,0,h1));
  end;
  end;
  end;
end;

end;

///нахождение максимумов дифференцированной реограммы
procedure TForm1.MaxM;
var
  a,i,Mx,My,r,r1,r2,m:integer;
begin
  Image1.Canvas.Brush.Style:=bsClear;
  Image1.Canvas.Pen.Color:=clRed;
  ARR[1]:=1;
  a:=2;
  for i:=1 to 1000 do
  if Reo1.B[2,i]<PorogM
  then
  begin
  if PorogM<=Reo1.B[2,i+1]
  then
  begin
  r1:=i+1;
  for m:=r1 to 1000 do
  if Reo1.B[2,m]>=PorogM
  then
  if Reo1.B[2,m+1]<PorogM
  then
  begin
  r2:=m;
  Mx:=r1;

```

```

My:=Reo1.B[2,r1];
for r:=r1 to r2 do
  if Reo1.B[2,r]>My
  then begin My:=Reo1.B[2,r]; Mx:=r; end;
  ARR[a]:=Mx; a:=a+1;          ///заполнение массива координатами точек
максимума дифференцированной реограммы по оси X (это нужно для нахождения точек
выброса крови из левого желудочка)
  Image1.Canvas.MoveTo(П(Mx),JJ(My+h2,0,h1)+5);
  Image1.Canvas.LineTo(П(Mx),JJ(My+h2,0,h1)-4);
  Break;
end;
end;
end;
end;
end;

```

///Нахождение точек выбрасывания крови из левого желудочка сердца
procedure TForm1.GelZ;

```

var
  b,c,Px,Py,t:integer;
  Max:real;
begin
  Image1.Canvas.Brush.Style:=bsClear;
  Image1.Canvas.Pen.Color:=clRed;
  Px:=1;
  Py:=1;
  for b:=1 to 80 do
    if ARR[b]<ARR[b+1]
    then
      begin
        Max:=0.00001;
        for c:=ARR[b] to ARR[b+1] do
          begin
            if (Reo1.B[2,c+1]-Reo1.B[2,c-1])/2>Max
            then
              begin
                Max:=(Reo1.B[2,c+1]-Reo1.B[2,c-1])/2;
                Px:=c;
                Py:=Reo1.B[2,c];
              end;
            end;
          end;
          t:=Round(Px-Py/Max);
          Image1.Canvas.MoveTo(П(t),JJ(Reo1.B[2,t]+h2,0,h1)+20);
          Image1.Canvas.LineTo(П(t),JJ(Reo1.B[2,t]+h2,0,h1)-6);
        end
      else Break;
    end;
  end;
end;

```

///процедура при нажатии на кнопку Разместить
procedure TForm1.Action3Execute(Sender: TObject);

```

begin
  MaxR;
  MaxM;

```

```

    GeZ;
end;

///функция получения высоты страницы в пикселях
function HeightY: Integer;
begin
    Result:=GetDeviceCaps(Printer.Canvas.Handle, VertRes)
end;

///функция получения ширины страницы в пикселях
function WidthX: Integer;
begin
    Result:=GetDeviceCaps(Printer.Canvas.Handle, HorzRes)
end;

///функция преобразования координат с бумаги по оси X
function PrinterCoordX(x:integer):integer;
begin
    Result:=round(GetDeviceCaps(printer.Handle,LogPixelsX)/25.4*x);
end;

///функция преобразования координат с бумаги по оси Y
function PrinterCoordY(Y:integer):integer;
begin
    Result:=round(GetDeviceCaps(printer.Handle,LogPixelsY)/25.4*Y);
end;

///печать графиков
procedure TForm1.PrintGrafik;
var
    j:integer;
begin
    I1:=0;
    I2:=Round(WidthX);
    J1:=0;
    J2:=Round(HeightY/2);
    Printer.Canvas.MoveTo(I1(PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[1,1],0,h1)));
    for j:=1 to 1000 do
        begin
            Printer.Canvas.LineTo(I1(PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[1,j+1],0,h1)));
        end;
    Printer.Canvas.MoveTo(I1(PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[2,1]+h2,0,h1)+5));
    for j:=1 to 1000 do
        begin
            Printer.Canvas.LineTo(I1(PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[2,j+1]+h2,0,h1)+5));
        end;
    Printer.Canvas.MoveTo(I1(PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[3,1]+h3,0,h1)+5));
    for j:=1 to 1000 do
        begin
            Printer.Canvas.LineTo(I1(PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[3,j+1]+h3,0,h1)+5));
        end;
    end;
end;

```

```

    end;
end;

////печатать R-зубцов
procedure TForm1.PrintR;
var
    j,Rx,Ry,p,p1,p2,k:integer;
begin
    Printer.Canvas.Brush.Style:=bsClear;
    I1:=0;
    I2:=Round(WidthX);
    J1:=0;
    J2:=Round(HeightY/2);
    for j:=1 to 1000 do
        if Reo1.B[1,j]<PorogR
        then
            begin
                if PorogR<Reo1.B[1,j+1]
                then
                    begin
                        p1:=j+1;
                        for k:=p1 to 1000 do
                            if Reo1.B[1,k]>PorogR
                            then
                                if Reo1.B[1,k+1]<PorogR
                                then
                                    begin
                                        p2:=k;
                                        Rx:=p1;
                                        Ry:=Reo1.B[1,p1];
                                        for p:=p1 to p2 do
                                            if Reo1.B[1,p]>Ry
                                            then begin Ry:=Reo1.B[1,p]; Rx:=p; end;
                                        Printer.Canvas.MoveTo(II(Rx),PrinterCoordY(10+9*Font.Size)+JJ(Ry,0,h1)-20);
                                        Printer.Canvas.LineTo(II(Rx),PrinterCoordY(10+9*Font.Size)+JJ(Ry,0,h1)+30);
                                        end;
                                    end;
                                end;
                            end;
                        end;
                    end;
                end;
            end;
        end;
end;

////печатать точек выброса крови из левого желудочка
procedure TForm1.PrintZ;
var
    b,c,Px,Py,tinteger;
    Max:real;
begin
    Printer.Canvas.Brush.Style:=bsClear;
    I1:=0;
    I2:=Round(WidthX);
    J1:=0;
    J2:=Round(HeightY/2);
    Px:=1;

```

```

Py:=1;
for b:=1 to 80 do
if ARR[b]<ARR[b+1]
then
begin
Max:=1;
for c:=ARR[b] to ARR[b+1] do
begin
if (Reo1.B[2,c+1]-Reo1.B[2,c-1])/2>Max
then
begin
Max:=(Reo1.B[2,c+1]-Reo1.B[2,c-1])/2;
Px:=c;
Py:=Reo1.B[2,c];
end;
end;
t:=Round(Px-Py/Max);
Printer.Canvas.MoveTo(II(t),PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[2,t]+h2,0,h1)-40);
Printer.Canvas.LineTo(II(t),PrinterCoordY(10+9*Font.Size)+JJ(Reo1.B[2,t]+h2,0,h1)+40);
end
else Break;
end;
end;

```

////печать максимумов дифференцированной реограммы

```

procedure TForm1.PrintM;
var
a,i,Mx,My,r,r1,r2,m: integer;
begin
Printer.Canvas.Brush.Style:=bsClear;
I1:=0;
I2:=Round(WidthX);
J1:=0;
J2:=Round(HeightY/2);
ARR[1]:=1;
a:=2;
for i:=1 to 1000 do
if Reo1.B[2,i]<PorogM
then
begin
if PorogM<=Reo1.B[2,i+1]
then
begin
r1:=i+1;
for m:=r1 to 1000 do
if Reo1.B[2,m]>=PorogM
then
if Reo1.B[2,m+1]<PorogM
then
begin
r2:=m;
Mx:=r1;
My:=Reo1.B[2,r1];

```

```

    for r:=r1 to r2 do
    if Reo1.B[2,r]>My
    then begin My:=Reo1.B[2,r]; Mx:=r; end;
    ARR[a]:=Mx; a:=a+1;
    Printer.Canvas.MoveTo(II(Mx),PrinterCoordY(10+9*Font.Size)+JJ(My+h2,0,h1)-20);
    Printer.Canvas.LineTo(II(Mx),PrinterCoordY(10+9*Font.Size)+JJ(My+h2,0,h1)+30);
    Break;
    end;
    end;
end;
end;
end;

```

////процедура при нажатии на кнопку Печать

```

procedure TForm1.Action4Execute(Sender: TObject);
begin
    Printer.BeginDoc;
    Printer.Canvas.TextOut(PrinterCoordX(10), PrinterCoordY(10),Label1.Caption+'
'+Edit1.Text);
    Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(Font.Size/2)),Label2.Caption+' '+Edit2.Text);
    Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(2*Font.Size/2)),Label3.Caption+' '+Edit3.Text);
    Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(3*Font.Size/2)),Label4.Caption+' '+Edit4.Text);
    Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(4*Font.Size/2)),Label5.Caption+' '+Edit5.Text);
    Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(5*Font.Size/2)),Label6.Caption+' '+Edit6.Text);
    Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(6*Font.Size/2)),Label7.Caption+' '+Edit7.Text);
    Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(7*Font.Size/2)),Label8.Caption+' '+Edit8.Text);
    Printer.Canvas.TextOut(PrinterCoordX(10),
PrinterCoordY(10+Round(8*Font.Size/2)),Label9.Caption+' '+Edit9.Text);
    PrintGrafik;
    PrintR;
    PrintM;
    PrintZ;
    Printer.EndDoc;
end;

end.

```