

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ПЕРМСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
БЕРЕЗНИКОВСКИЙ ФИЛИАЛ

П.В. Плехов

ОБРАБОТКА XML ДОКУМЕНТА  
В BORLAND DELPHI 7

Методическое пособие по выполнению курсовой работы  
по дисциплине «Программирование»

Березники, 2013

ББК

УДК 004.3

П

Рецензент:

доктор техн. наук, доцент Затонский А.В

Плехов П.В.

П\*\* Программирование. Обработка xml документа в Borland Delphi 7: Учебно-методическое пособие для выполнения курсовой работы/ Перм. гос. техн. ун-т., Березниковский филиал, 2011.– 43 с.

Рассмотрен пример обработки структурированного файла формата xml. Приведено подробное описание процесса создания программы в среде Borland Delphi 7. Рассмотрены основные приемы при обработке xml-файла.

Предназначено для студентов направления «Информатика и вычислительная техника».

© Пермский национальный  
исследовательский политехнический  
университет, 2013

## **Содержание**

Содержание .....	3
Введение .....	4
Общие положения .....	5
Пример.....	7
Варианты заданий на курсовую работу .....	9
Пример программы .....	13
Создание формы.....	13
Исходный текст программы (основной модуль).....	17
Исходный текст программы (модуль работы со стеком) .....	22
Приложение 1.....	24
Приложение 2.....	44

## **Введение**

В современных прикладных задачах часто приходится встречаться с проблемой обмена данными между различными программами. Нередко самая простая задача импорта-экспорта данных из одной системы в другую приводит к необходимости серьёзных разработок модулей на стыке систем. Задача существенно облегчается, если данные определенного класса будут перемещаться между системами, при условии, что в этих системах заложена возможность воспринимать извне и отдавать наружу данные в стандартном формате импорта-экспорта. Наиболее простым и распространенным в настоящее время способом обмена данными между системами является формат *XML*. Также, этот формат применяется, например, для хранения структурированных данных, обмена информацией между программами. Таким образом, практикующему программисту необходимо уметь обрабатывать *XML* – файлы.

Настоящие указания к выполнению курсовой работы по предмету «Программирование» нацелены на развитие навыков в обработке таких файлов. При выполнении курсовой работы студенты должны разобраться в структуре предлагаемого файла и написать программу для его обработки. Кроме того, необходимо уметь самостоятельно работать с законодательной литературой, где описываются форматы файлов. Для развития этого навыка в приложениях приведены оригинальные нормативные документы.

## **Общие положения**

Настоящее методическое пособие по выполнению курсовой работы по дисциплине «Программирование» (в дальнейшем просто Курсовая) устанавливает единые требования по содержанию, выполнения и защиты курсовой работы по дисциплине «Программирование» направления Информатика и вычислительная техника.

Выполнение Курсовой должно быть начато во втором семестре дисциплины «Программирование» после изучения темы «Обработка XML-файлов».

Курсовая должна быть сдана во время зачётной недели в соответствии с графиком учебного процесса, либо по соответствующему допуску из Деканата.

Если студент показывает готовую рабочую программу, полностью соответствующую заданию до зачётной недели, то такая Курсовая может претендовать на оценку «Отлично». Полностью готовая программа, соответствующая заданию, но впервые показанная на зачётной недели может претендовать только на оценку «Хорошо». Полностью готовая программа, соответствующая заданию впервые показанная после зачётной недели может претендовать только на оценку «Удовлетворительно».

Курсовая состоит из нескольких частей:

- Первая часть Курсовой это реализация программы по обработке XML-файла формата НДСФЛ2 версии, соответствующей версии файла разобранный в данном методическом пособии в качестве примера. Выполнение данной части Курсовой необходимо для получения оценки «Удовлетворительно»;
- Вторая часть Курсовой это полностью самостоятельная разработка программы для обработки файла НДСФЛ2 более новой версии, например, версии 5.02. Наличие этой части в курсовой работе студента необходимо для получения оценки «Хорошо». Необходимо отметить, что программа в курсовой работе должна

быть одна, и она автоматически должна определять версию открываемого файла и, соответственно, обрабатывать его по одному или другому алгоритму;

- Третья часть это дополнение программы какими-либо функциями сбора статистики по обрабатываемому файлу. Наличие таких функций в программе необходимо для получения оценки «Отлично».

Необходимо отметить, что итоговая оценка курсовой работы определяется по ответам студента во время процедуры защиты курсовой работы.

Результатом выполнения Курсовой является программа соответствующая варианту задания выданному студенту и пояснительная записка, содержащая следующие разделы:

1. Титульный лист.
2. Задание на курсовую работу (взято из данного пособия слово-в-слово).
3. Краткое описание предметной области, в которой применяются файлы указанного типа.
4. Описание последовательности действий программы при выполнении заданной обработки файла (алгоритм).
5. Краткая инструкция по работе с программой.
6. Листинг программы. Рекомендуется подробно комментировать ключевые моменты программы, это упрощает процесс защиты работы.
7. Список литературы.

Записка выполняется в соответствии с требованиями к оформлению курсовых и дипломных работ (Приложение 3). На защиту курсовой работы предоставляются записка и исходные тексты программы.

Программа может быть выполнена на языке программирования *Object Pascal* в любой среде программирования (*Borland Delphi, Lazarus*).

Использование иных языков программирования допускается при предварительном согласовании с преподавателем, не менее чем за три недели до зачётной недели.

## Пример

*XML*-файл хранит информацию в древовидной структуре, поэтому для перемещения по его узлам разумно использовать стек.

Стек, как известно, это очередь с дисциплиной *LIFO* (*Last In First Out*, т.е. Последним вошел – первым вышел) при обработке *xml*-файла используется для хранения узлов при перемещении по узлам дерева. Использование стека не является обязательным, но тогда возникнет необходимость в использовании большого количества переменных для хранения узлов дерева, причём количество переменных будет соответствовать высоте дерева.

Программа, реализующую первую часть Курсовой, может содержать примерно такую последовательность действий:

1. Предложить пользователю выбрать файл для обработки.
2. После выбора пользователем файла – открыть выбранный файл.
3. Найти корневой элемент и добавить его в стек.
4. Определить количество элементов для обработки (узлы «Справка»<sup>1</sup>, содержащем информацию по конкретному человеку).
5. По порядку обрабатываем все узлы «Справка»:
  - a. Находим узел «ПолучДох» (Получатель Доходов, т.е. сведения о физическом лице, на которого собрана информация в узле «Справка»).
  - b. Находим узел «ИННФЛ» (ИНН физического лица), если узел существует<sup>2</sup>, считываем данные.

---

<sup>1</sup> Пример файла приведён в Приложение 1 в пункте 6.

<sup>2</sup> В соответствии с форматом, описанном в Приложении 1 данный узел может не присутствовать в узле «Справка»

- c. Находим узел «ФИО», в этом узле находим узлы: «Фамилия», «Имя», «Отчество» считываем данные, если узел «Отчество» не содержит данных, то сохраняем пустую строку.
  - d. Находим узел «ДатаРожд» (Дата рождения), считываем данные если они есть.
6. После окончания считывания данных о человеке необходимо проверить условие (есть ли у человека отчество) и при необходимости выводим данные о человеке на экран (в компонент *TStrinGrid*, вывод реализован в процедуре *AddToSG*).
  7. Для перехода к следующему узлу «Справка» извлекаем данные о корневом элементе из стека.
  8. После окончания обработки всех записей закрыть файл.
  9. Вывести на экран результат работы программы.

## **Варианты заданий на курсовую работу**

1. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) проживающих на территории Пермского края не в г. Березники. Подсчитать и вывести на экран их количество.
2. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, код субъекта федерации) проживающие за пределами Пермского края (на территории РФ). Подсчитать и вывести на экран их количество.
3. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) имеющих адрес места жительства за пределами РФ. Подсчитать и вывести на экран их количество.
4. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) у которых ИНН выдан не в Пермской области (крае). Подсчитать и вывести на экран их количество.
5. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) возрастом более 45 лет. Подсчитать и вывести на экран их количество.
6. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) имеющих статус резидент. Подсчитать и вывести на экран их количество.
7. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) у которых, при начислении налога по ставке 13% была допущена ошибка более чем на 2 рубля. Подсчитать и вывести на экран их количество и сумму ошибок по всему файлу.
8. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) в данных о постоянном месте жительства в РФ которых, не указан почтовый индекс. Подсчитать и вывести на экран их количество.

9. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, общая сумма вычета) одновременно имеющих вычет по кодам 101 и 103. Подсчитать и вывести на экран их количество.
10. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) у которых не указан почтовый индекс. Подсчитать и вывести на экран их количество.
11. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, сумма исчисленная, сумма удержанная) у которых, сумма налога исчисленная, не равна сумме налога удержанной. Подсчитать и вывести на экран их количество подсчитать и вывести сумму ошибок по всему файлу.
12. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) имеющих не нулевой совокупный доход. Подсчитать и вывести на экран их количество и сумму.
13. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, код документа) у которых документ удостоверяющие личность - не паспорт РФ. Подсчитать и вывести на экран их количество.
14. Составить и вывести на экран список работников (фамилия, имя, отчество), не имеющих ИНН. Подсчитать и вывести на экран их количество.
15. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) не являющихся гражданами РФ. Подсчитать и вывести на экран их количество.
16. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) у которых сумма вычета по кодам не соответствует общей сумме вычета. Подсчитать и вывести на экран их количество.
17. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) имеющих вычет по коду 101 более 3000 рублей. Подсчитать и вывести на экран их количество.

18. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) пенсионного возраста. Подсчитать и вывести на экран их количество.
19. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, сумма налога) имеющих не нулевую сумму, излишне удержанную налоговым агентом. Подсчитать и вывести на экран их количество.
20. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, сумма налога) имеющих не нулевую сумму, удержанную при уплате налога по перерасчету с доходов прошлых лет. Подсчитать и вывести на экран их количество.
21. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) имеющих статус не резидент и год рождения до 1981 включительно. Подсчитать и вывести на экран их количество.
22. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, количество месяцев) проработавших в организации полгода и более. Подсчитать и вывести на экран их количество.
23. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, сумма налога) имеющих не нулевую сумму налога, переданную на взыскание в налоговый орган. Подсчитать и вывести на экран их количество.
24. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) с ошибочными ИНН. Подсчитать и вывести на экран их количество.
25. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, сумма налога) имеющих задолженность по уплате налога. Подсчитать и вывести на экран их количество.
26. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, сумма дохода) имеющих доход по коду 2000. Подсчитать и вывести на экран их количество и сумму.

27. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН) работающих в организации весь год. Подсчитать и вывести на экран их количество.
28. Составить и вывести на экран список работников (фамилия, имя, отчество, ИНН, сумма по месяцам, общая сумма) у которых, сумма дохода по месяцам не соответствует общей сумме дохода. Подсчитать и вывести на экран их количество.
29. Составить и вывести на экран список всех кодов дохода, встречающихся в файле, и сумм доходов по ним.
30. Составить и вывести на экран список всех кодов вычетов, встречающихся в файле, и сумм вычетов по ним.

## Пример программы

**Задание:** Сформировать и вывести на экран список всех работников (фамилия, имя, ИНН), не имеющих отчества. Подсчитать и вывести на экран их количество.

### Создание формы

МОЖЕТ КАРТИНКИ КОМПОНЕНТОВ ДОБАВИТЬ В ТЕКСТ

1. После открытия *Delphi* сохраняем проект и все связанные с ним файлы в отдельной папке.
2. Переименовываем форму с *Form1* в *Form\_Main*, изменив свойство *Name*.
3. Меняем заголовок главного окна (*Form\_Main*) на «Обработка файла формата НДСФЛ2», изменяя свойство *Caption*.
4. Из палитры компонентов, с вкладки *Standard* добавляем на форму компонент *MainMenu*.
5. Двойным щелчком по добавленному компоненту на форме открываем его для редактирования, добавляем пункт «Файл» (свойство *Caption*), добавляем подпункты:
  - 5.1. «открыть», в свойстве *ShortCut* можно выбрать сочетание клавиш для быстрого доступа, например [Ctrl]+[O];
  - 5.2. разделитель (один знак минус);
  - 5.3. выход.
6. С вкладки *Standard* добавляем компонент *Memo*, в свойстве *Align* устанавливаем значение *alBottom* (компонент расположен у нижнего края), подстраиваем высоту компонента. Выбрав свойство *Lines*, нажимаем на появившуюся рядом кнопку (☰), стираем все строки и нажимаем кнопку *OK*.
7. С вкладки *Additional* добавляем компонент *Splitter*, в свойстве *Align* выбираем значение *alBottom*, устанавливаем высоту элемента с помощью свойства *Height*, задав значение 5.

**Примечание [u1]:** Не знаю, на этот момент они уже со всеми этими компонентами работали

8. С вкладки *Additional* добавляем компонент *StringGrid*, в свойстве *Align* выбираем значение *alClient* (компонент занимает всё доступное пространство).
9. Настраиваем объект *StringGrid*:
  - 9.1. *ColCount* (Количество столбцов): 5 (зависит от задания);
  - 9.2. *DefaultRowHeight* (высота строки по умолчанию): 16;
  - 9.3. раскрываем свойство *Options*, и устанавливаем:
    - 9.3.1. *goRangeSelect* (запрещаем выбор нескольких ячеек): false;
    - 9.3.2. *goRowSizing* (изменение высоты строки): true;
    - 9.3.3. *goColSizing* (изменение ширины столбца): true;
    - 9.3.4. *goRowSelect* (выделение будет не для одной ячейки, а для всей строки): true;
  - 9.4. *RowCount* (количество строк, потом в процессе вывода информации строки будут добавляться дополнительно): 2;
10. С вкладки *Dialogs* добавляем компонент *OpenDialog*.
11. Настраиваем объект *OpenDialog1*.
  - 11.1. Выбрав свойство *Filter*, нажимаем на появившуюся рядом кнопку (☰). В столбце *FilterName* вводим описание фильтра, например «XML-файлы», в поле *Filter* вводим фильтр, для *xml* файлов это «\*.xml», также добавляем второй пункт «Все файлы»-«\*», нажимаем кнопку «OK».
  - 11.2. Раскрываем свойство *Options* и устанавливаем: *ofFileMustExist*(выбираемый файл должен существовать): true;
12. С вкладки *Internet* добавляем компонент *XMLDocument*, изменим имя объекта с «*XMLDocument1*» на «*xml*». Результат выполнения предыдущих пунктов 1-12 представлен на рис. 1.

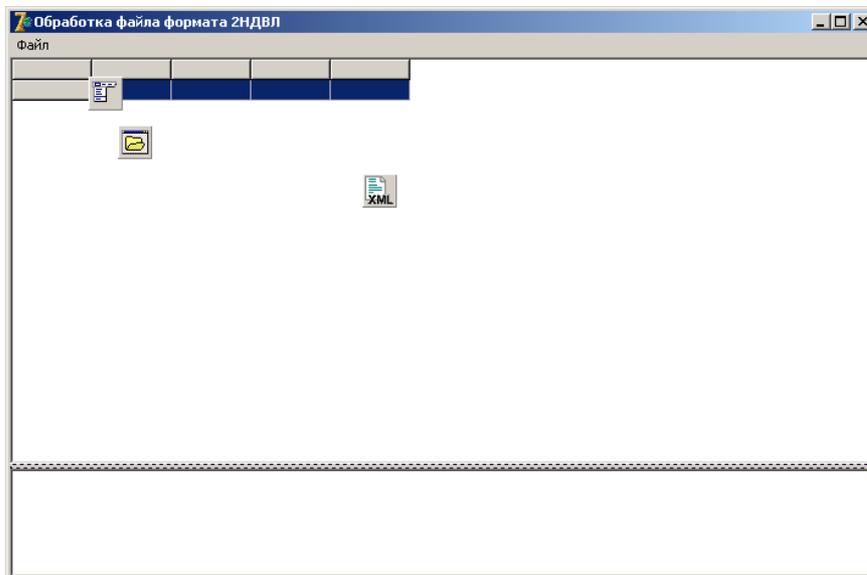


Рис. 1 Вид окна программы в *Delphi 7*

13. Описываем тип данных *man\_info*<sup>1</sup> – это record, которая содержит такие поля, как фамилия, имя, отчество, ИНН, дату рождения и флаг, который указывает есть ли в поле дата рождения данные, или у человека не была указана дата рождения.
14. Добавляем глобальную переменную *FileName*.
15. Подключаем дополнительный модуль *stack.pas*<sup>2</sup> (файл необходимо создать, используя пункт меню *File→New→Unit*, затем поместить в созданный файл исходный код модуля *stack.pas* и сохранить файл в папке с проектом), добавив его в разделе *uses*.
16. Создаём дополнительный процедуры. В меню *View* выбираем пункт *CodeExplorer*, вызываем контекстное меню для пункта *TForm\_Main* (рис 2.) и, выбирая каждый раз пункт *New*, по очереди добавляем три процедуры (введя их название и если необходимо параметры, без знака «;» в конце):
  - 16.1. *procedure process ;*

<sup>1</sup> Объявление типа приведено в исходном тексте программы ниже.

<sup>2</sup> Текст модуля приведён ниже, после исходного текста главной формы.

16.2. *procedure AddToSG(m:man\_info);*

16.3. *procedure FormClearAll.*

17. Для того, что бы было проще работать с узлами *xml*-файла, которые могут содержать *null* (признак отсутствия значения). Создаем функцию *is\_null*, данная функция будет принимать два входящих параметра: переменную, которая может содержать *null*, и строку. Функция будет возвращать значение переменной, если оно не *null*, или строку если в переменной содержится *null*. Текст функции необходимо поместить перед всеми методами (процедурами) в программе, как и приведено в примере ниже.

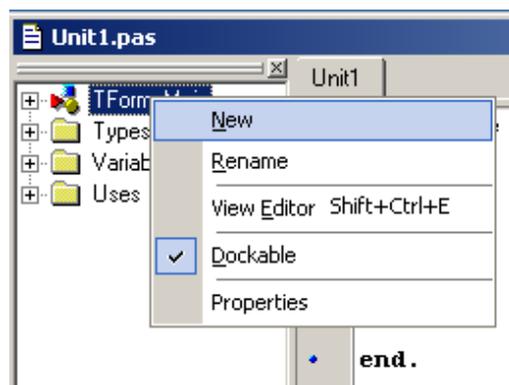


Рис. 2 Добавление нового метода

18. Чтобы настроить ширину столбцов, заголовки столбцов и прочее создадим событие *OnShow* для главной формы, выбрав объект *Form\_Main* в инспекторе объектов, переходим на вкладку *Events* (События). Двойным щелчком напротив поля *OnShow* (при отображении формы) создаем соответствующее событие, пишем его обработчик.
19. Пишем код для метода (процедуры) *process*, который будет вызываться для обработки файла.

20. Пишем код для метода (процедуры) *AddToSG*, который будет использоваться для вывода на экран информации о человеке (передана в качестве параметра в виде типа данных *man\_info*).
21. Пишем код для метода (процедуры) *FormClearAll* который будет отчищать содержимое формы, что бы можно было обработать еще один *xml*-файл.
22. Создаем событие для пункта меню «Открыть», щелкнув по нему мышкой, описываем необходимые действия.
23. Создаем событие для пункта меню «Выход», щелкнув по нему мышкой, описываем необходимые действия.
24. Запускаем программу кнопкой F9. Вид окна программы после обработки файла представлен на рис. 3.

№	Фамилия	Имя	ИНН	Дата рождения
1	Анферов	Дмитрий	591100899480	25.12.1961
2	Кельская	Юлия	591109916027	30.11.1962
3	Соцкий	Егор	591111670677	23.05.1970

Количество записей о людях: 66  
Из них не имеют отчества: 3

Рис. 3 Вид окна программы после обработки файла

### **Исходный текст программы (основной модуль)**

```
unit Unit1;
interface

//{$DEFINE DEBUG}           //для работы в режиме отладки

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
```

```
Dialogs, Menus, Grids, ExtCtrls, ComCtrls, xmldom, XMLIntf,
msxmldom,
XMLDoc, stack, StdCtrls;
```

**type**

```
man_info=record      //запись, данные о человеке
  f,i,o:String[50];  //ФИО
  inn:String[12];    //ИНН
  dr:TDate;          //Дата рождения
  dr_exist:boolean;  //флаг, указана дата рождения или нет
end;
```

```
TForm_Main = class(TForm)
```

```
  Splitter1: TSplitter;
  StringGrid1: TStringGrid;
  MainMenu1: TMainMenu;
  N1: TMenuItem;
  N2: TMenuItem;
  OpenDialog1: TOpenDialog;
  xml: TXMLDocument;
  Memo1: TMemo;
  N3: TMenuItem;
  N4: TMenuItem;
  procedure N2Click(Sender: TObject);
  procedure process;
  procedure N4Click(Sender: TObject);
  procedure AddToSG(m:man_info);
  procedure FormShow(Sender: TObject);
  procedure FormClearAll;
```

```
private
```

```
  { Private declarations }
```

```
public
```

```
  { Public declarations }
```

```
end;
```

**var**

```
  Form_Main: TForm_Main;
  FileName:String;      //Имя обрабатываемого файла
```

**implementation**

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

```
//Функция проверки аргумента v на значение null
```

```
//возвращает str, если v имеет значение null
```

```
function is_null(v:Variant;str:String):String;
```

```
begin
```

```
  if (v=null) then
```

```
    Result:=str
```

```
  else
```

```
    Result:=v;
```

```
end;
```

```
//процедура добавления данных о человеке в список
```

```

//m - структура с данными о человеке
procedure TForm_Main.AddToSG(m:man_info);
var j:integer; //номер строки
begin
j:=StringGrid1.RowCount-1; //вычисляем, куда добавить
новые данные
StringGrid1.Cells[0,j]:=IntToStr(j); //ставим порядковый
номер
StringGrid1.Cells[1,j]:=m.f; //выводим фамилию
StringGrid1.Cells[2,j]:=m.i; //имя
StringGrid1.Cells[3,j]:=m.inn; //ИНН
if (m.dr_exist) then //Если дата рождения есть
StringGrid1.Cells[4,j]:=FormatDateTime('dd.mm.yyyy',m.dr);
//то выводим
j:=j+1; //добавляем
StringGrid1.RowCount:=j+1; // строку
end;

//процедура обработки файла
procedure TForm_Main.process;
var iNode:IXMLNode; //переменная, для перемещения между
элементами
i,cnt,cnt_usl:integer;mn:man_info;
begin
iNode:=xml.DocumentElement; //Корень всего документа
push_node(iNode); //Сохраняем корневой узел
cnt:=iNode.ChildNodes.Count; //получаем кол-во
потомков
Memo1.Lines.Add('Кол-во записей о людях: ' + IntToStr(cnt));
cnt_usl:=0; //кол-во людей без отчества
for i:=0 to cnt-1 do //для обхода всех "Справок"
begin
iNode:=iNode.ChildNodes[i]; //Узел "Справка"
push_node(iNode); //Сохраняем этот узел

iNode:=iNode.ChildNodes.FindNode('ПолучДох'); //Переходим к
узлу
push_node(iNode); //сохраняем этот узел

iNode:=iNode.ChildNodes.FindNode('ИННФЛ'); //Ищем ИНН
if (iNode<>nil) then //Проверяем, есть или нет ИНН
mn.inn:=is_null(iNode.NodeValue,'') //Если ИНН есть, то
сохраняем его
else //Если ИНН нет, то сохраняем
пустую строку
mn.inn:='';
iNode:=read_node(); //переходим к узлу "ПолучДох"
iNode:=iNode.ChildNodes.FindNode('ФИО'); //Ищем узел
"ФИО"
push_node(iNode); //Сохраняем узел "ФИО"

iNode:=iNode.ChildNodes.FindNode('Фамилия'); //ищем фамилию
mn.f:=iNode.NodeValue; // фамилия есть всегда

```

```

iNode:=read_node(); //Переходим К "ФИО"

iNode:=iNode.ChildNodes.FindNode('Имя'); //ищем имя
mn.i:=iNode.NodeValue; // Имя тоже, обязательное поле
iNode:=read_node(); //Переходим К "ФИО"

iNode:=iNode.ChildNodes.FindNode('Отчество'); //ищем
"Отчество"
if (iNode<>nil) then //поле отчество - не
обязательное
mn.o:=is_null(iNode.NodeValue, '') //если оно есть,
записываем его
else //иначе записываем
mn.o:=''; //пустую строку
iNode:=pop_node(); //Переходим К "ФИО"
iNode:=read_node(); //Считываем "ПолучДох"
iNode:=iNode.ChildNodes.FindNode('ДатаРожд'); //ищем дату
рождения
if (iNode<>nil) then //если поле есть
begin
mn.dr:=StrToDate(iNode.NodeValue); //считываем дату
mn.dr_exist:=true; //устанавливаем флаг, что
дата существует
end
else
mn.dr_exist:=false; //если поля нет, то даты рождения
нет
iNode:=pop_node(); //К узлу "ПолучДох"
iNode:=pop_node(); //К узлу "Справка"
iNode:=read_node; //К узлу "НДФЛ2" (корню), но он остается
в стеке
//Получение информации завершено, проверяем условие отбора
if (mn.o='') then
begin
AddToSG(mn); //выводим на экран данные
cnt_usl:=cnt_usl+1; //учет количества людей без отчества
end;
end;
iNode:=pop_node(); //Освобождаем память, занятую корневым
элементом
Memo1.Lines.Add('Из них не имеют отчества: ' +
IntToStr(cnt_usl));
end;

//пункт меню "Открыть"
procedure TForm_Main.N2Click(Sender: TObject);
begin
{$IFDEF DEBUG} //В режиме "Debug"
OpenDialog1.FileName:='C:\ДОХОД_2НДФЛ.xml'; //файл не
запрашивается
{$ELSE} //В обычном режиме
if (OpenDialog1.Execute) then //Запрашиваем файл
{$ENDIF}

```

```

begin
  FormClearAll(); //Отчистка формы, перед выбором нового
  файла
  FileName:=OpenDialog1.FileName; //Сохраняем имя файла
  Form_Main.Caption:='Файл: '+FileName; //выводим пояснение
  в заголовков
  xml.FileName:=FileName; //указываем, какой файл будем
  обрабатывать
  try //пытаемся обработать
  xml.Active:=true; //открываем xml файл
  process(); //выполняем обработку
  except //Если возникла ошибка, выводим сообщение
  on E: EDOMParseError do
  MessageBox(handle,PChar(E.Message),'Файл не обработан',MB_OK
or MB_ICONSTOP);
  end;
  end;
end;

//пункт меню "Выход"
procedure TForm_Main.N4Click(Sender: TObject);
begin
  Close(); //закрытие окна программы
end;

//При отображение формы на экране выполняем подготовительные
  действия
procedure TForm_Main.FormShow(Sender: TObject);
begin
  StringGrid1.RowCount:=2; //количество строк
  StringGrid1.ColCount:=5; //количество столбцов
  StringGrid1.Cells[0,0]:='№'; //
  StringGrid1.Cells[1,0]:='Фамилия'; //Записываем
  StringGrid1.Cells[2,0]:='Имя'; // пояснение к
  StringGrid1.Cells[3,0]:='ИНН'; // столбцам
  StringGrid1.Cells[4,0]:='Дата рождения'; //
  StringGrid1.ColWidths[0]:=20; //Номер по порядку
  StringGrid1.ColWidths[1]:=100; //фамилия
  StringGrid1.ColWidths[2]:=100; //имя
  StringGrid1.ColWidths[3]:=120; //ИНН
  StringGrid1.ColWidths[4]:=100; //Дата рождения
end;

//Отчистка формы
procedure TForm_Main.FormClearAll;
var i,j:integer;
begin
for i:=1 to StringGrid1.RowCount - 1 do //Обходим все строки
  for j:= 0 to StringGrid1.ColCount -1 do // и столбцы
    StringGrid1.Cells[j,i]:=''; //записываем пустые
  значения
  StringGrid1.RowCount:=2; //устанавливаем начальное значение
  кол-ва строк

```

```
Memo1.Lines.Clear();      //Отчистка окна сообщений
end;
```

```
end.
```

### **Исходный текст программы (модуль работы со стеком)**

```
unit stack;

interface

uses XMLIntf, Dialogs, SysUtils;

//Перечисляем функции доступные из других модулей
procedure push_node(_node:IXMLNode);
function pop_node:IXMLNode;
function read_node:IXMLNode;

type                                //объявляем типы
PListElem=^ListElem; //Указатель на элемент списка
ListElem=record          //Элемент списка
  node:IXMLNode;        //узел xml документа
  Next:PListElem;      //Указатель на следующий элемент списка
end;

var                                //объявляем глобальную переменную-корень
списка
  root:PListElem=nil; //инициализируем её

implementation

//Из вкладки Internet помещаем объект XMLDocument
//Не забываем его переименовать

//добавление узла в стек
procedure push_node(_node:IXMLNode);
var tmp:PListElem;
begin
New(tmp); //выделяем память под элемент
tmp^.Next:=root; //переставляем связь
tmp^.node:=_node; //копируем данные
root:=tmp; //изменяем корневой элемент
end;

//получение узла из стека
//проверка на существование элемента не осуществляется!
function pop_node:IXMLNode;
var tmp:PListElem;
begin
Result:=root^.node; //сохраняем данные
tmp:=root; //сохраняем адрес этого элемента
root:=root^.Next; //изменяем корень списка
Dispose(tmp); //освобождаем память занятую элементом
end;
```

```
//функция чтения верхнего элемента в стеке,  
//без удаления этого элемента  
function read_node:IXMLNode;  
begin  
Result:=root^.node; //возвращаем значение  
end;  
  
end.
```

## Приложение 1.

### Формат сведений о доходах по форме № 2-НДФЛ "Справка о доходах физического лица за 200\_ год " в электронном виде (на основе XML) (версия 4.00001)

#### 1. Общие сведения.

Настоящий документ описывает требования к составу и структуре сведений о доходах физических лиц по форме № 2-НДФЛ в электронном виде (далее сведения в электронном виде).

Сведения в электронном виде в налоговые органы ФНС России представляются в виде XML файлов.

#### 2. Описание сведений в электронном виде

##### 2.1. Общие положения

##### Имя файла сведений в электронном виде

Имя файла сведений в электронном виде, должно иметь следующий вид:

R\_T\_P\_O\_ggggmmdd\_N.xml ,

где:

R\_T – префикс имени, принимает значение: "ДОХОД\_2НДФЛ";

P – код получателя информации. Код территориального органа ФНС России по классификатору СОНО;

O - идентификатор отправителя.

Десятизначный идентификатор (ИНН и КПП) для юридического лица. Двенадцатиразрядный идентификатор (ИНН) для физического лица;

gggg – год, mm - месяц, dd – день формирования сведений в электронном виде;

N - идентификационный номер файла – номер передаваемого файла (длина до 36 знаков). В случае невозможности обеспечения уникальности номера файла, рекомендуется использовать 36-символьный глобально уникальный идентификатор (GUID).

##### *Параметры первой строки файла сведений*

Первая строка XML файла должна иметь следующий вид:

```
<?xml version="1.0" encoding = "windows-1251"?>
```

##### 2.2. Логическая модель сведений в электронном виде

Логическая модель сведений в электронном виде, приведена в графическом виде в Разделе 3 на рис.1. Логическая модель сведений в электронном виде включает элементы и атрибуты XML файла. Полный перечень структурных элементов логической модели и их описания приведены в Разделе 4.

Для каждого структурного элемента в таблице приводятся следующие сведения:

- *Наименование элемента.* Приводится полное наименование элемента.
- *Сокращенное наименование элемента.* Приводится сокращенное наименование элемента. Сокращенные наименования могут записываться буквами кириллицы и цифрами.
- *Признак типа элемента.* Может принимать следующие значения: "П" – простой элемент (не имеющий вложенных); "С" – сложный элемент (имеющий вложенные), А - атрибут. После признака типа элемента может указываться признак "Т", если для определения элемента используется пользовательский тип данных. Например: ПТ, СТ. Наименование типа данных (типового элемента) указывается в графе "Дополнительные сведения".
- *Формат элемента.* Формат<sup>1</sup> определяется для простых элементов (т.е. имеющих значение "П" в столбце "Признак типа элемента"). Представляется в условных обозначениях, которым соответствуют следующие значения: Т – символьная строка; N – числовое значение (целое или дробное). Формат символьной строки указывается в виде Т(к) или =Т(к). Где к – максимальное количество знаков в строке для формата Т(к) (значение "к" не указывается для символьной строки неограниченной длины) и к – фиксированное количество знаков в строке для формата =Т(к). Формат целого числа указывается в виде N(m), где m – максимальное количество знаков в числе. Если значением элемента является дробное десятичное число, то формат его представления имеет вид N(m.k), где m – максимальное количество знаков в числе, включая целую и дробную часть числа без разделяющей десятичной точки, а k – максимальное число знаков дробной части числа. Для простых элементов, являющимися базовыми в XML (определенными в <http://www.w3.org/TR/xmlschema-0>) и имеющими строго определенную длину (например, элемент с типом "date"), поле

<sup>1</sup> При описании структуры формата сведений в электронном виде используются следующие металингвистические конструкции:

::= – метасимвол, означающий "есть по определению";

< > – метасимволы, используемые для выделения элементов структуры сообщения (логической модели);

// - метасимволы, содержащие значения элементов структуры сообщения;

[ ] – метасимволы, указывающие на необязательность элемента металингвистической структуры, заключенного в них;

{ } – метасимволы, указывающие на использование элемента металингвистической структуры, заключенного в них, один и более раз;

| – метасимвол, означающий возможность выбора среди нескольких вариантов значений элемента металингвистической структуры.

- “Формат элемента” не заполняется. Формат и тип для таких элементов указываются в поле “Дополнительная информация”.
- *Признак обязательности элемента* определяет обязательность наличия элемента в сведениях в электронном виде. Признак обязательности элемента может принимать следующие значения: “О” – обязательное наличие элемента (наименование элемента и его значение должны присутствовать); “Н” – присутствие элемента необязательно (наименование элемента и его значение могут отсутствовать). Если элемент может принимать ограниченный перечень значений (по классификатору или перечню кодов), то признак обязательности элемента дополняется символом “К”. Например: “ОК”. В случае если количество реализаций элемента может быть более одной, то признак обязательности элемента дополняется символом “М”. Например: “ОМ, ОКМ”.
  - *Дополнительная информация.* Для элементов, кодируемых по классификатору (кодovому словарию), перечню значений, указывается соответствующее сокращенное наименование классификатора (кодovого словаря), перечень его возможных значений или ссылка на таблицу классификатора (кодovого словаря). Для типовых элементов указывается наименование типового элемента. Для сложных элементов указывается ссылка на таблицу, в которой описывается структура данного элемента.

Классификаторы и кодовые словари (справочники), используемые при формировании некоторых элементов представляемых сведений в электронном виде, приведены в Приложении №2 к форме № 2-НДФЛ “Справочники”.

XSD схема сведений, представляемых в электронном виде, размещена на сайте ГНИВЦ ФНС России.

Имя файла, содержащего XSD схему, имеет следующий вид:

R\_T\_N.xsd ,

где:

R\_T - префикс имени файла XSD схемы, принимает значение ДОХОД\_2НДФЛ;

N – двузначный идентификационный номер версии XSD схемы файла .

### 3. Диаграмма сведений в электронном виде

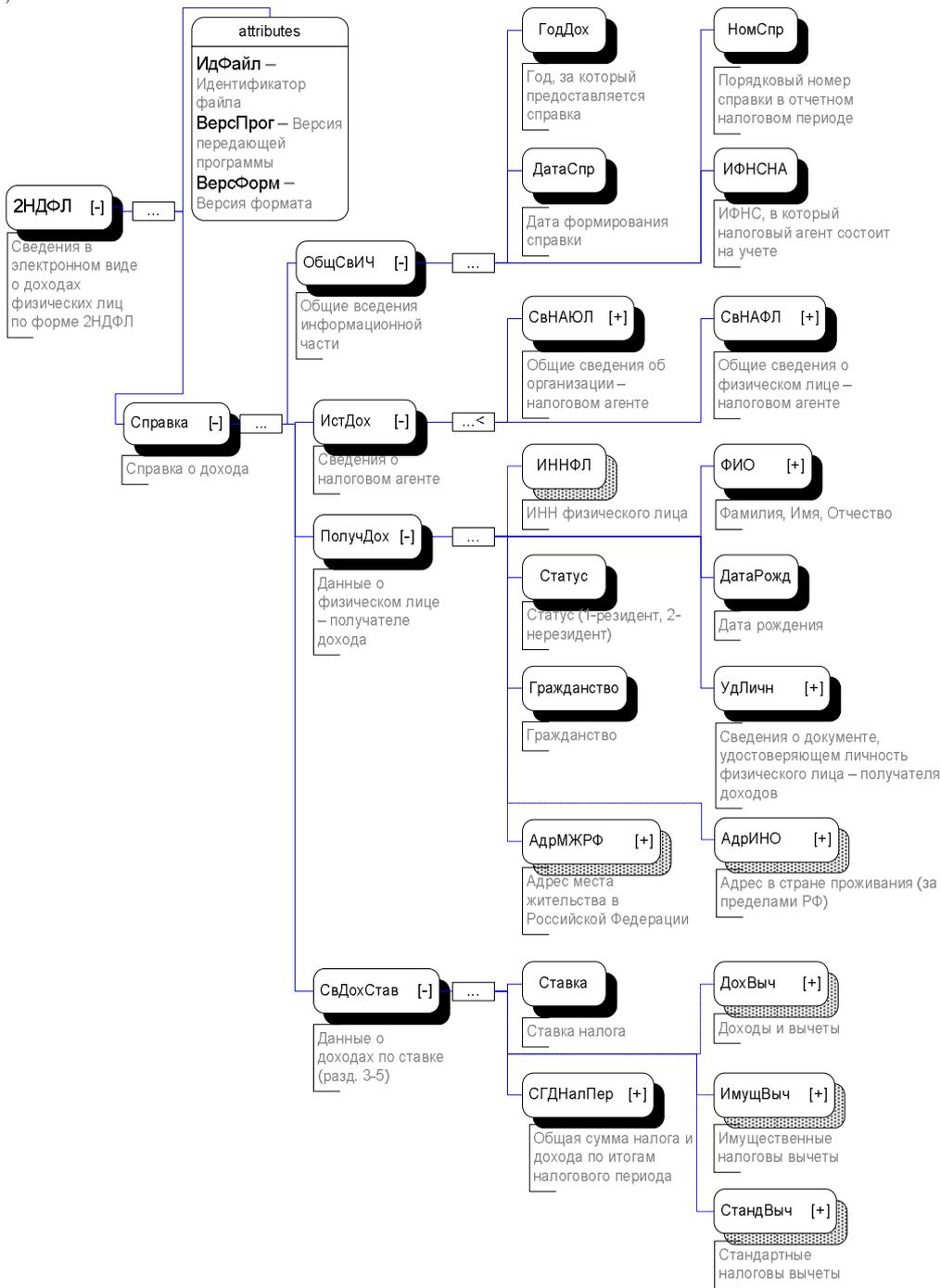


Рис.1. Диаграмма структуры сведений в электронном виде

#### 4. Перечень структурных элементов логической модели сведений в электронном виде

Перечень структурных элементов логической модели сведений в электронном виде приведен в таблицах 4.1 – 4.17

Таблица 4.1

#### Сведения в электронном виде о доходах физических лиц по форме 2-НДФЛ (НДФЛ2)

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Идентификатор файла	ИдФайл	А	T(100)	О	Совпадает с именем файла
Версия передающей программы	ВерсПрог	А	T(40)	О	
Версия формата	ВерсФорм	А	T(7)	О	4.00001
Справка о доходах	Справка	С		ОМ	Структура представлена в табл.4.2

Таблица 4.2

#### Справка о доходах (Справка)

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Общие сведения информационной части	ОбщСвИЧ	С		О	Структура представлена в табл.4.3
Сведения о налоговом агенте	ИстДох	С		О	Структура представлена в табл.4.4
Данные о физическом лице - получателе дохода	ПолучДох	С		О	Структура представлена в табл.4.7
Данные о доходах по ставке (разд. 3-5)	СвДохСтав	С		О	Структура представлена в табл.4.10

Таблица 4.3

## Общие сведения информационной части (ОбщСвИЧ)

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Год, за который представляется справка	ГодДох	П		О	Элемент в формате ГГГГ тип «gYear»
Порядковый номер справки в отчетном налоговом периоде	НомСпр	П	T(8)	О	
Дата формирования справки	ДатаСпр	ПТ	=T(10)	О	Типовой элемент <ДатаТип> в формате ДД.ММ.ГГГГ
ИФНС, в которой налоговый агент состоит на налоговом учете	ИФНСНА	ПТ	=T(4)	ОК	Типовой элемент <СОНОТип> Код по Классификатору «Система обозначений налоговых органов» (СОНО)

Таблица 4.4

## Сведения о налоговом агенте (ИстДох)

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
-----------------------	---	-----------------------	-----------------	---------------------------------	---------------------------

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Общие сведения об организации - налоговом агенте	СвНАЮЛ I	С I		О I	Структура представлена в табл.4.5 I
Общие сведения о физическом лице - налоговом агенте	СвНАФЛ	С		О	Структура представлена в табл.4.6 (в сведениях о налоговом агенте должны присутствовать или СвНАЮЛ или СвНАФЛ)

Таблица 4.5

**Общие сведения об организации - налоговом агенте (СвНАЮЛ)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
ИНН организации	ИННЮЛ	ПТ	=Т(10)	О	Типовой элемент <ИННЮЛТип>
КПП организации	КПП	ПТ	=Т(9)	О	Типовой элемент <КППТип>
Наименование организации	НаимОрг	П	Т(1000)	О	
Код по ОКАТО	ОКАТО	ПТ	=Т(11)	ОК	Типовой элемент <ОКАТОТип> Код по Классификатору объектов административно-территориального деления (ОКАТО)
Телефон (контактный)	Телефон	П	Т(20)	Н	

Таблица 4.6

**Общие сведения о физическом лице - налоговом агенте (СвНАФЛ)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
ИНН физического лица	ИННФЛ	ПТ	=Т(12)	О	Типовой элемент <ИННФЛТип>
Фамилия, имя, отчество физического лица	ФИОФЛ	СТ		О	Типовой элемент <ФИОТип>. Структура представлена в табл.4.17
Код по ОКАТО	ОКАТО	ПТ	=Т(11)	ОК	Типовой элемент <ОКАТОТип> Код по Классификатору объектов административно-территориального деления (ОКАТО)
Телефон (контактный)	Телефон	П	Т(20)	Н	

Таблица 4.7

**Данные о физическом лице - получателе дохода (ПолучДох)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
ИНН физического лица	ИННФЛ	ПТ	=Т(12)	Н	Типовой элемент <ИННФЛТип>
Фамилия, имя, отчество	ФИО	СТ		О	Типовой элемент <ФИОТип>. Структура представлена в табл.4.17

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Статус	Статус	П	=Т(1)	ОК	Принимает значение: 1 – резидент   2 – нерезидент
Дата рождения	ДатаРожд	ПТ	=Т(10)	О	Типовой элемент <ДатаТип> в формате ДД.ММ.ГГГГ
Гражданство	Гражданство	ПТ	=Т(3)	ОК	Типовой элемент <ОКСМТип> Код по Общероссийскому классификатору стран мира (ОКСМ)
Сведения о документе, удостоверяющем личность физического лица - получателя дохода	УдЛичн	С		О	Структура представлена в табл.4.8
Адрес места жительства в Российской Федерации	АдрМЖРФ	С		О   Н	Обязателен: - в случае, если <Статус>=1 и <Гражданство>=643; - в случае, если отсутствует <АдрИНО> Структура представлена в табл.4.16
Адрес в стране проживания	АдрИНО	С		О   Н	Обязателен в случае, если отсутствует <АдрМЖРФ> Структура представлена в табл.4.9

Таблица 4.8

**Сведения о документе, удостоверяющем личность физического лица - получателя дохода (УдЛичн)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Код вида документа, удостоверяющего личность	КодУдЛичн	ПТ	=Т(2)	ОК	Типовой элемент <СПДУЛТип> Значение выбирается из Справочника «Коды документов» (Приложение № 2 к форме № 2-НДФЛ)
Серия и номер документа, удостоверяющего личность	СерНомДок	П	Т(25)	О	Значение в соответствии с шаблоном из Справочника шаблонов, приведенных в разделе 5

Таблица 4.9

**Адрес в стране проживания (за пределами Российской Федерации) (АдрИно)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Код страны	КодСтр	ПТ	=Т(3)	ОК	Типовой элемент <ОКСМТип> Код по Общероссийскому классификатору стран мира (ОКСМ)
Адрес	АдрТекст	П	Т(255)	О	

Таблица 4.10

## Данные о доходах по ставке (разделы 3-5 Справки) (СвДохСтав)

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Ставка налога	Ставка	П	N(2)	О	
Доходы и вычеты	ДохВыч	С		НМ	Структура представлена в табл.4.11
Стандартные налоговые вычеты	СтандВыч	С		Н	Структура представлена в табл.4.12
Имущественные налоговые вычеты	ИмуцВыч	С		Н	Структура представлена в табл.4.13
Общая сумма дохода и налога на доходы по итогам налогового периода	СГДНалПер	С		О	Структура представлена в табл.4.14

Таблица 4.11

**Доходы и вычеты (ДохВыч)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Месяц	НомМес	П	=Т(2)	О	Принимает значение: от 01 до 12
Код дохода	КодДоход	П	=Т(4)	ОК	Значение выбирается из Справочника «Коды доходов» (Приложение № 2 к форме № 2-НДФЛ)
Сумма дохода	СумДоход	П	N(15.2)	О	
Код вычета	КодВычет	П	=Т(3)	ОК   НК	Значение выбирается из Справочника «Коды вычетов» (Приложение № 2 к форме № 2-НДФЛ) Обязательно, если присутствует <СумВычет>
Сумма вычета	СумВычет	П	N(15.2)	О   Н	Обязательно, если присутствует <КодВычет>

Таблица 4.12

**Стандартные налоговые вычеты (СтандВыч)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Суммы налоговых вычетов, право на получение которых имеется у налогоплательщика	ПравоВыч	СТ		ОМ	Типовой элемент <ПравоВыч>. Структура представлена в табл.4.15
Общая сумма предоставленных стандартных налоговых вычетов	СумСтНалВыч	П	N(15.2)	О	

Таблица 4.13

**Имущественные налоговые вычеты (ИмущВыч)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Суммы налоговых вычетов, право на получение которых имеется у налогоплательщика	ПравоВыч	СТ		ОМ	Типовой элемент <ПравоВыч>. Структура представлена в табл.4.15
Номер уведомления, подтверждающего право на имущественный налоговый вычет	НомерУвед	П	T(20)	О	
Дата выдачи уведомления	ДатаУвед	ПТ	=T(10)	О	Типовой элемент <ДатаТип> в формате ДД.ММ.ГГГГ

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Код налогового органа, выдавшего уведомление	ИФНСУвед	ПТ	=Т(4)	ОК	Типовой элемент <СОНОТип> Код по Классификатору «Система обозначений налоговых органов» (СОНО)
Общая сумма предоставленных имущественных налоговых вычетов	СумИмуВыч	П	N(15.2)	О	

Таблица 4.14

**Общая сумма дохода и налога на доходы по итогам налогового периода (СГДНалПер)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Общая сумма дохода	СГДСумм	П	N(15.2)	О	
Облагаемая сумма дохода	ОблСумм	П	N(15.2)	О	
Сумма налога исчисленная	НИОблСумм	П	N(12.0)	О	
Сумма налога удержанная	НУОблСумм	П	N(12.0)	О	
Сумма возврата налога по перерасчету с доходов прошлых лет	ВозврСуммПЛ	П	N(12.0)	Н	

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Сумма, зачтенная при уплате налога по перерасчету с доходов прошлых лет	ЗачСуммПЛ	П	N(12.0)	Н	
Сумма, удержанная при уплате налога по перерасчету с доходов прошлых лет	УдСуммПЛ	П	N(12.0)	Н	
Задолженность по налогу за налогоплательщиком	ДолгНП	П	N(12.0)	Н	
Сумма налога, излишне удержанная налоговым агентом	ДолгНА	П	N(12.0)	Н	
Сумма налога, переданная на взыскание в налоговый орган	ВзыскИФНС	П	N(12.0)	Н	

Таблица 4.15

**Суммы налоговых вычетов, право на получение которых имеется у налогоплательщика (ПравоВыч)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Код вычета	КодВычет	П	=T(3)	ОК	Значение выбирается из Справочника «Коды вычетов» (Приложение № 2 к форме № 2-НДФЛ)

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Сумма вычета	СумВычет	П	N(15.2)	О	

Таблица 4.16

**Адрес в Российской Федерации (АдрРФТип)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Почтовый индекс	Индекс	П	=Т(6)	Н	
Код региона	КодРегион	ПТ	=Т(2)	О	Значение в соответствии с КЛАДР
Район	Район	П	Т(50)	О   Н	Значение в соответствии с КЛАДР Обязательно для городов и населенных пунктов районного подчинения
Город	Город	П	Т(50)	О   Н	Значение в соответствии с КЛАДР Обязательно при отсутствии населенного пункта (кроме субъектов Российской Федерации – Москва и Санкт-Петербург)

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Населенный пункт	НасПункт	П	T(50)	О   Н	Значение в соответствии с КЛАДР Обязательно при отсутствии города (кроме субъектов Российской Федерации – Москва и Санкт-Петербург)
Улица	Улица	П	T(50)	О   Н	Значение в соответствии с КЛАДР
Дом (владение)	Дом	П	T(8)	Н	
Корпус	Корпус	П	T(8)	Н	
Квартира	Кварт	П	T(8)	Н	

Таблица 4.17

**Фамилия, имя, отчество физического лица (ФИОТип)**

Наименование элемента	Сокращённое наименование (код) элемента	Признак типа элемента	Формат элемента	Признак обязательности элемента	Дополнительная информация
Фамилия	Фамилия	П	T(60)	О	
Имя	Имя	П	T(60)	О	
Отчество	Отчество	П	T(60)	Н	

**5. Шаблоны для ввода реквизита "серия и номер документа, удостоверяющего личность"**

<b>Код</b>	<b>Наименование документа</b>	<b>Шаблон серии, номера</b>
01	Паспорт гражданина СССР	R-ББ 999999
03	Свидетельство о рождении	SSSSSSSSSSSSSSSSSSSSSSSSSS
07	Военный билет солдата (матроса, сержанта, старшины)	ББ 0999999
10	Паспорт иностранного гражданина	SSSSSSSSSSSSSSSSSSSSSSSSSS
11	Свидетельство о рассмотрении ходатайства о признании беженцем на территории Российской Федерации по существу	SSSSSSSSSSSSSSSSSSSSSSSSSS
12	Вид на жительство	SSSSSSSSSSSSSSSSSSSSSSSSSS
13	Удостоверение беженца в Российской Федерации	SSSSSSSSSSSSSSSSSSSSSSSSSS
14	Временное удостоверение личности гражданина Российской Федерации	SSSSSSSSSSSSSSSSSSSSSSSSSS
15	Разрешение на временное проживание в Российской Федерации	SSSSSSSSSSSSSSSSSSSSSSSSSS
21	Паспорт гражданина Российской Федерации	99 99 9999990
23	Свидетельство о рождении, выданное уполномоченным органом иностранного государства	SSSSSSSSSSSSSSSSSSSSSSSSSS
91	Иные документы	SSSSSSSSSSSSSSSSSSSSSSSSSS

Примечания к таблице.

В графе "Шаблон серии, номера" приведены данные для контроля значения серии, номера документа.

Шаблон состоит из символов "R", "Б", "9", "0", "S", "-" (тире) и " " (пробел).

Используются следующие обозначения:

**R** - на месте одного символа **R** располагается целиком римское число, заданное символами "I", "V", "X", "L", "C", набранными на верхнем регистре латинской клавиатуры;

**9** - любая десятичная цифра (обязательная);

**0** - любая десятичная цифра (необязательная, может отсутствовать);

**Б** - любая русская заглавная буква;

**S** - символ не контролируется (может содержать любую букву, цифру или вообще отсутствовать);

“–“ (тире) - указывает на обязательное присутствие данного символа в контролируемом значении.

Пробелы используются для разделения групп символов, а также вместо знаков “№” или “№” для разделения серии и номера документа.

## 6. Пример сведений в электронном виде

```
<?xml version="1.0" encoding="windows-1251"?>
<НДФЛ2 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
ВерсФорм="4.00001" ВерсПрог="4.214"
ИдФайл="ДОХОД_2НДФЛ_1111_11111111111111111111_20070115_1">
  <Справка>
    <ОбщСвИЧ>
      <ГодДох>2006</ГодДох>
      <НомСпр>1</НомСпр>
      <ДатаСпр>15.01.2007</ДатаСпр>
      <ИФНСНА>1111</ИФНСНА>
    </ОбщСвИЧ>
    <ИстДох>
      <СвНАЮЛ>
        <ИННЮЛ>1111111111</ИННЮЛ>
        <КПП>1111111111</КПП>
        <НаимОрг>ЗАО "Рассвет"</НаимОрг>
        <ОКАТО>1111111111</ОКАТО>
        <Телефон>(111)111-11-11</Телефон>
      </СвНАЮЛ>
    </ИстДох>
    <ПолучДох>
      <ИННФЛ>1111111111</ИННФЛ>
      <ФИО>
        <Фамилия>Иванов</Фамилия>
        <Имя>Иван</Имя>
        <Отчество>Иванович</Отчество>
      </ФИО>
      <Статус>1</Статус>
      <ДатаРожд>11.11.1955</ДатаРожд>
      <Гражданство>643</Гражданство>
      <УдЛичн>
        <КодУдЛичн>22</КодУдЛичн>
        <СерНомДок>11 11 111111</СерНомДок>
      </УдЛичн>
      <АдрМЖРФ>
        <Индекс>111111</Индекс>
        <КодРегион>77</КодРегион>
        <Улица>Ленинский пр-кт</Улица>
        <Дом>11</Дом>
        <Корпус>1</Корпус>
        <Кварт>111</Кварт>
      </АдрМЖРФ>
    </ПолучДох>
    <СвДохСтав>
      <Ставка>13</Ставка>
```

```
<ДоxВыч>
  <НомМес>01</НомМес>
  <КодДоход>2000</КодДоход>
  <СумДоход>10000</СумДоход>
</ДоxВыч>
<ДоxВыч>
  <НомМес>12</НомМес>
  <КодДоход>2000</КодДоход>
  <СумДоход>10000</СумДоход>
</ДоxВыч>
<СтандВыч>
  <ПравоВыч>
    <КодВычет>103</КодВычет>
    <СумВычет>4800</СумВычет>
  </ПравоВыч>
  <СумСтНалВыч>4800</СумСтНалВыч>
</СтандВыч>
<ИмушВыч>
  <ПравоВыч>
    <КодВычет>311</КодВычет>
    <СумВычет>5000</СумВычет>
  </ПравоВыч>
  <НомерУвед>111111</НомерУвед>
  <ДатаУвед>11.01.2006</ДатаУвед>
  <ИФНСУвед>1111</ИФНСУвед>
  <СумИмуВыч>5000</СумИмуВыч>
</ИмушВыч>
<СГДНалПер>
  <СГДСумм>20000</СГДСумм>
  <ОблСумм>10200</ОблСумм>
  <НИОблСумм>1326</НИОблСумм>
  <НУОблСумм>1326</НУОблСумм>
</СГДНалПер>
</СвДоxСтав>
</Справка>
</НДФЛ2>
```

## Приложение 2.

### Разъяснения о формировании ИНН в соответствии с ПРИКАЗОМ ГОСУДАРСТВЕННОЙ НАЛОГОВОЙ СЛУЖБЫ РФ «ОБ УТВЕРЖДЕНИИ ИНСТРУКЦИИ О ПОРЯДКЕ УЧЕТА НАЛОГОПЛАТЕЛЬЩИКОВ» от 13.06.96 № ВА-3-12/49

Идентификационный номер налогоплательщика (далее - ИНН) присваивается налогоплательщикам (организациям и физическим лицам) государственными налоговыми инспекциями по субъектам Российской Федерации, по районам, городам, районам в городах (далее - налоговая инспекция). Под налогоплательщиками - организациями понимаются имеющие обязанности уплачивать и/или перечислять хотя бы один вид налога или иного обязательного платежа в федеральный бюджет, бюджеты субъектов Российской Федерации и местные бюджеты:

- представительства, филиалы, а равно и другие обособленные подразделения юридического лица, созданные в соответствии с законодательством Российской Федерации;
- организации, созданные в соответствии с законодательством иностранных государств, и международные организации, осуществляющие деятельность и/или имеющие имущество в Российской Федерации;
- физические лица, осуществляющие предпринимательскую деятельность без образования юридического лица (далее - предприниматели);
- нотариусы, занимающиеся в установленном законодательством порядке частной практикой.

ИНН организации, созданной в соответствии с законодательством Российской Федерации, представляет собой десятизначный цифровой код: INNININIXIXIXIXIC!, где:

- NNNN – код налоговой инспекции, осуществившей постановку на учет налогоплательщика;
- XXXXX – порядковый номер налогоплательщика;
- C – контрольное число, рассчитанное по утвержденному Федеральной налоговой службой России алгоритму.

ИНН организации, созданной в соответствии с законодательством иностранного государства, и международной организации, осуществляющей деятельность и/или имеющей имущество в Российской Федерации, представляет собой цифровой код, аналогичный описанному в п. 2 настоящего документа, но с учетом того, что перед указанным десятизначным цифровым кодом дополнительно проставляется буквенный символ "F": FNNINININIXIXIXIXIC!

ИНН физического лица, имеющего статус предпринимателя или имеющего право заниматься в установленном законодательством Российской Федерации порядке частной практикой, представляет собой двенадцатизначный цифровой код: INNINININIXIXIXIXIXIXIC!, где:

- NNNN – код налоговой инспекции, осуществившей постановку на учет налогоплательщика;
- XXXXXX – порядковый номер налогоплательщика;
- СС – контрольное число, рассчитанное по утвержденному Федеральной налоговой службой России алгоритму.

№ цифры в ИНН по порядку	Множитель для 10-зн.	Множитель для 12-зн. (1-ая контр. цифра)	Множитель для 12-зн (2-я контр. цифра)
1	2	7	3
2	4	2	7
3	10	4	2
4	3	10	4
5	5	3	10
6	9	5	3
7	4	9	5
8	6	4	9
9	8	6	4
10		8	6
11			8

Для расчета десятого контрольного разряда в 10-значном ИНН каждая цифра ИНН (кроме десятой) умножается на соответствующий множитель в соответствии с табл.1 , затем все значения суммируются , сумма берется по модулю 11, затем полученное число берется по модулю 10 это и есть десятый разряд.

Для расчета 11-го контрольного разряда (1-й контрольной цифры) в 12-значном ИНН каждая цифра ИНН (кроме 11-й и 12-й) умножается на соответствующий множитель в соответствии с табл.1 , затем все значения суммируются , сумма берется по модулю 11, затем полученное число берется по модулю 10, это и есть 11-й разряд.

Для расчета 12-го контрольного разряда (2-й контрольной цифры) в 12-значном ИНН каждая цифра ИНН (кроме 12-й) , 11-я вычисляется в соответствии с предыдущим пунктом, умножается на соответствующий множитель в соответствии с табл.1 , затем все значения суммируются, сумма берется по модулю 11, затем полученное число берется по модулю 10, это и есть 12-й разряд.

Приложение 3.

# **Требования оформлению.**

**к**