**Задание «Автоматизация Excel».** В среде объектно-ориентированного программирования *Visual Studio* разработать программу в режиме *Windows Forms Application* на языке *Visual C#*, представляющую собой средство взаимодействия с электронными таблицами *Microsoft Office Excel.*

**Цель работы**: освоить навыки подключения внешних модулей и их использования в разрабатываемых программах; освоить навыки программируемой настройки параметров электронных таблиц *Microsoft Office Excel*.

**Рабочее задание:**

1. Создать графический пользовательский интерфейс с кнопкой «Выгрузка в *Excel*».

2. Нажатие на кнопку инициирует запуск на исполнение метода составления и заполнения электронной таблицы заданной по варианту структуры.

**В качестве результата выполненной работы сдаётся**:

1. Архив *\*.rar* или *\*.zip*, содержащий разработанное решение (*solution*). Для отправки архива по электронной почте рекомендуется использовать сервисы файлового обмена, генерирующие ссылки на файлы (например, *Google Drive*, *Яндекс.Диск*, *Облако Mail.Ru* и тому подобные), поскольку протоколы электронной почты часто автоматически отклоняют архивы, содержащие исполняемые (*exe*) файлы.

2. Электронная таблица в формате *Microsoft Office Excel*, выгруженная приложением.

3. Отчёт в формате *Microsoft Office Word* по выполненной работе.

**Памятка программисту №1. Обнаружение и перенос библиотеки *Microsoft Office Excel* в проект (методика аналогична переносу библиотеки *Microsoft Office Word*):**

1. В меню «Проект» перейти к диалоговому окну «Добавить ссылку…» (Рисунок 1).

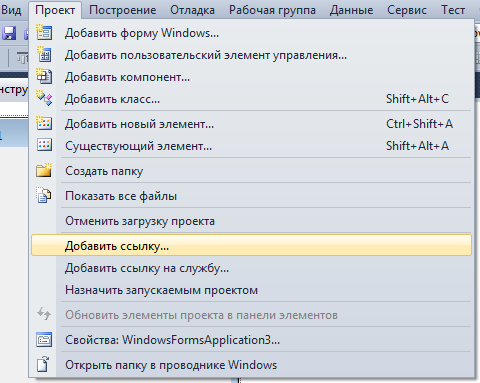


Рисунок 1 – Содержимое пункта меню «Проект» (*Visual Studio 2010*)

2. В открывшемся окне переключиться на вкладку «*COM*» (Рисунок 2).

3. Найти в списке компонент, содержащий слова «*Microsoft Excel*» (Рисунок 2).

4. Определить путь к соответствующей библиотеке по всплывающему *tool-tip*’у или в одном из столбцов диалогового окна (Рисунок 2).

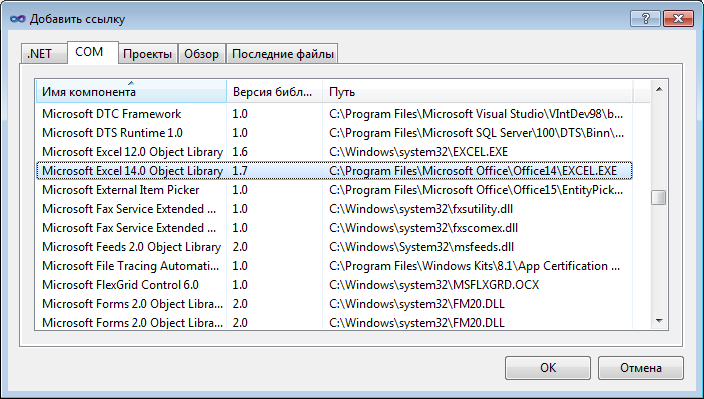


Рисунок 2 – Путь к библиотеке *Microsoft Office Excel* (*Visual Studio 2010*)

Нужным файлом может оказаться «*EXCEL.EXE*», который находиться, например, в *«\Program Files\Microsoft Office\Office14\*» или *«\Program Files\Microsoft Office\Office15\*».

Важно, чтобы число, указанное после слова «*Office*» (это кодовая версия *Microsoft Office*), было не старше того, которое будет указано на компьютере, где планируется последующая эксплуатация приложения.

В том случае, если известно, что на компьютере пользователя установлена более ранняя версия *Office* по сравнению с установленной на компьютере разработчика, то не лишним шагом будет предварительное копирование этого файла с компьютера пользователя.

Полезная информация относительно соответствия рабочих версий пакетов *Microsoft Office* следующая:

***Office 11 – Microsoft Office 2003;***

***Office 12 – Microsoft Office 2007;***

***Office 14 – Microsoft Office 2010;***

***Office 15 – Microsoft Office 2013.***

5. Найти по определённому пути запускаемый файл, содержащий библиотеку *EXCEL* «*EXCEL.EXE*» (Рисунок 3).

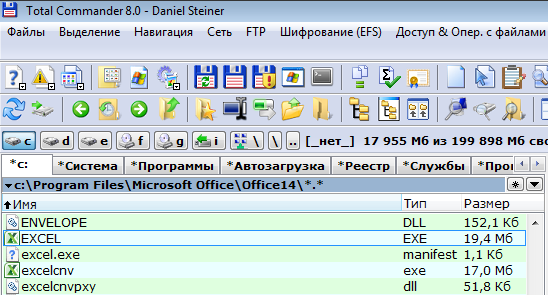


Рисунок 3 – Обнаружение библиотеки *Microsoft Office Excel* в директории *Microsoft Office*

6. Скопировать запускаемый файл в директорию с разрабатываемой оболочкой (Рисунок 4).

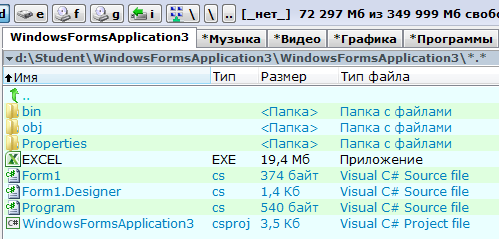


Рисунок 4 – Перемещённая к проекту библиотека *Microsoft Office Excel*

**Памятка программисту №2. Подключение библиотеки *Microsoft Office Excel* к решению**

1. В меню «Проект» перейти к диалоговому окну «Добавить ссылку…» (Рисунок 1).

2. В открывшемся окне переключиться на вкладку «Обзор» (Рисунок 5).

3. Выбрать библиотеку *Microsoft Office Excel* и подтвердить выбор нажатием на кнопку «ОК» (Рисунок 5).

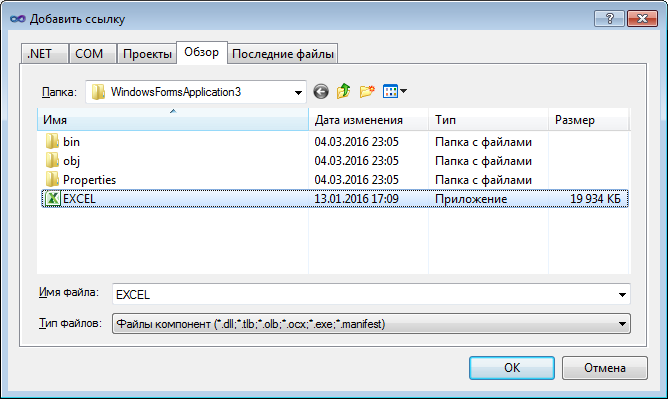


Рисунок 5 – Переопределение места размещения библиотеки *Microsoft Office Excel* в среде программирования *Visual Studio* для текущего проекта

4. В обозревателе решений (Рисунок 6) развернуть папку «Ссылки» и убедиться в подключении указанной библиотеки (*Microsoft.Office.Core,* *Microsoft.Office.Interop.Excel, VBIDE*).

|  |  |
| --- | --- |
| до | после |

Рисунок 6 – Контроль изменения ссылок при подключении модуля *Microsoft Office Excel* в обозревателе решений

5. Прописать в разделе подключаемых к решению модулей ***using*** следующий:

***Microsoft.Office.Interop.Excel.***

Для удобства обращения к модулю присвоить его переменной, например, *Excel* или *E* (Рисунок 7).

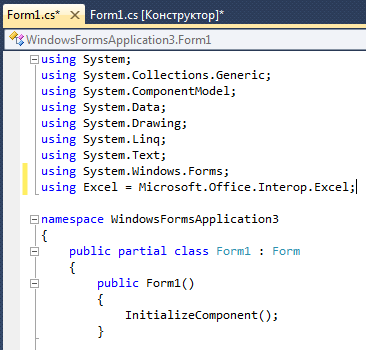


Рисунок 7 – Добавление используемого модуля *Microsoft Office Excel* в раздел *using* главной формы проекта

**Памятка программисту №3. Работа с *Microsoft Excel* программным способом**

1. Обобщённая конструкция для работы с *Microsoft Office Excel* может быть записана следующим образом:

а) В разделе объявления используемых модулей ***using*** прописать

***System.Reflection****.*

он необходим для обращения к *Missing* (отсутствующему параметру).

б) Создать объект под отсутствующие параметры *Microsoft Office Excel*

***object ObjMissing = Missing.Value.***

в) Поместить блок отлова ошибок ***try … catch … finally … .***

г) Создать переменную под приложение *Microsoft Office Excel* (подложка)

***Excel.\_Application ObjExcel = new Excel.Application().***

д) Создать метод, в котором в качестве параметров необходимо передавать *Excel*-приложение и отсутствующее значение.

е) Закрыть *Excel*-приложение методом **Quit()**.

ж) В разделе ***catch*** вызвать побуждающее сообщение на случай возникновения непредвиденной ошибки в процессе работы с *Excel*.

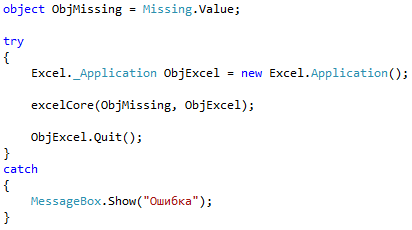


Рисунок 8 – Обобщённый кодовый фрагмент, позволяющий обеспечить нормальную работу с *Microsoft Office Excel*

Пониманию программной структуры *Microsoft Office Excel* может способствовать следующая схема (Рисунок 9):



Рисунок 9 – Краткая объектно-ориентированная модель *Microsoft Office Excel*

Процедура, в рамках которой книга *Microsoft Office Excel* наполняется содержимым, может, в общем виде, быть представлена следующей конструкцией:

а) Для экономии ресурсов *Excel*-приложение делается невидимым

**ObjExcel.Visible = false.**

б) В *Excel*-приложении создаётся новая книга

**Excel.Workbook ObjWorkBook =**

**ObjExcel.Workbooks.Add(Missing.Value).**

в) В книге *Microsoft Office Excel* выбирает лист (таблица), с которой будет вестись работа. Первый лист присутствует всегда. В версиях до 2013 года гарантированно присутствовали 3 листа

**Excel.Worksheet ObjWorkSheet = ObjWorkBook.Sheets[1].**

г) При необходимости добавить дополнительный лист в начало

**ObjWorkBook.Worksheets.Add().**

д) При необходимости добавить дополнительный лист в конец

**ObjWorkBook.Worksheets.Add(Missing.Value,**

**ObjWorkBook.Sheets[ObjWorkBook.Sheets.Count]).**

е) Если новый лист был добавлен, необходимо запустить метод активации установленного рабочего листа

**ObjWorkSheet.Activate().**

ж) Обращение к диапазону ячеек

**Excel.Range cells = ObjWorkSheet.get\_Range("G3", "O6").**

з) Выделение диапазона ячеек

**cells.Select().**

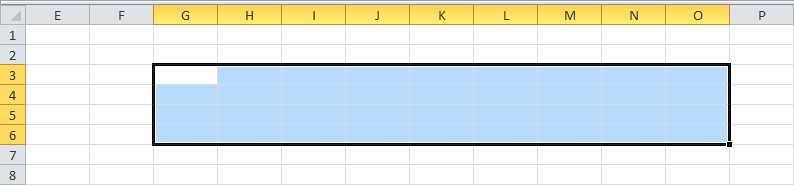


Рисунок 10 – Реакция на программное выделение диапазона с *G3* по *O6* методом *Select()*

и) Запись в ячейку диапазона конкретного значения

**cells.Cells[2, 3] = "Мой текст".**

Объединяя пункты ж) и и) получается, что надпись «Мой текст» будет записана в ячейку *I4* (смещение от G3 на 2 позиции по строкам и на 3

позиции по столбцам).

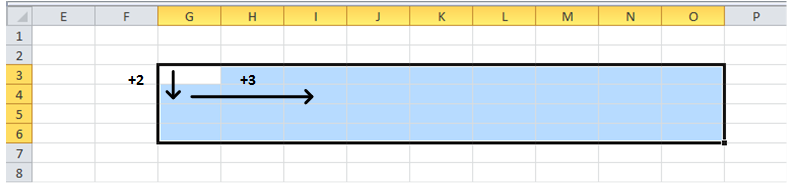


Рисунок 11 – Иллюстрация поведения индексации в выделенном диапазоне

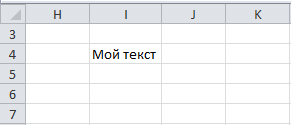


Рисунок 12 – Реакция на программную запись текста в ячейку

к) Выполнить слияние ячеек (объединить можно только выделенный диапазон ячеек)

**cells.Merge().**

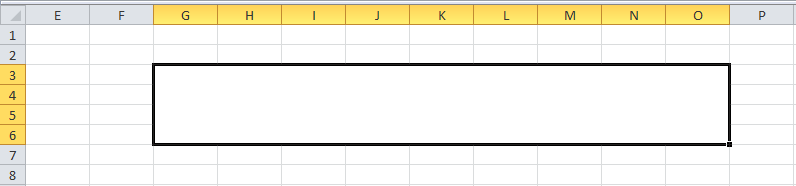


Рисунок 13 – Реакция на программное слияние ячеек диапазона с *G3* по *O6* методом *Merge()*

л) Для удобства отладки перевести книгу *Excel* в ручное управление перед сохранением

**ObjExcel.UserControl = true.**

м) Выполняется сохранение файла (например, в той же директории, где размещён исполняемый (*exe*) файл)

**ObjWorkBook.SaveCopyAs**

**(Application.StartupPath + @"\" + Name + ".xlsx");\*.**

н) Закрыть книгу *Microsoft Office Excel*

**ObjWorkBook.Close(false, "", Missing.Value);\*\*.**

\* В переменную *Name* в любой удобный момент времени, предшествующий сохранению, необходимо записать название файла, который будет записан.

\*\* 1-й параметр – не сохранять изменения, так как сохранение уже выполнено;

2-й параметр – пустое имя файла, опять же, так как сохранение уже выполнено;

3-й параметр – маршрутизация книги (по умолчанию).

**Памятка программисту №4. Полезные процедуры управления диапазоном**

Для удобства работы со ссылочной структурой электронной таблицы *Microsoft Office Excel* удобно составить дополнительный типизированный метод (функцию), интерпретирующий численные значения в эквивалент *Microsoft Office Excel*, содержащий буквы латинского алфавита (Рисунок 14).

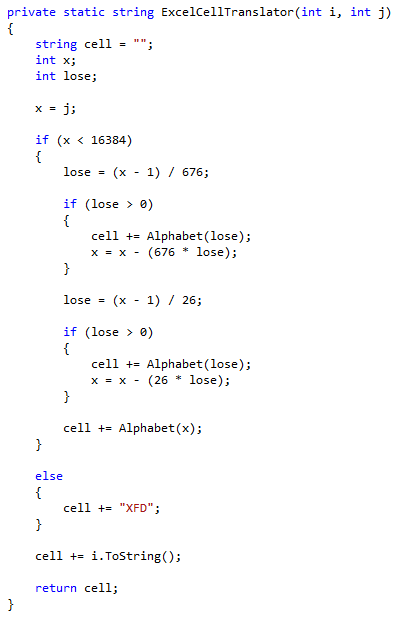


Рисунок 14 – Метод-переводчик ссылок на ячейки *Microsoft Office Excel*

Функция возвращает строку формата\*\*\*:

**Буква(-ы)Цифра**

Например, *ExcelCellTranslator(1, 1)* вернёт

«A1»;

Например, *ExcelCellTranslator(5, 40)* вернёт

«AN5»;

Использование:

**Excel.Range cells = ObjWorkSheet.get\_Range(**

**ExcelCellTranslator(3, 7), ExcelCellTranslator(6, 15) );**

\*\*\* Функция содержит вложенный типизированный метод (функцию) *Alphabet* для определения буквы латинского алфавита по указанному численному значению (Рисунок 15).

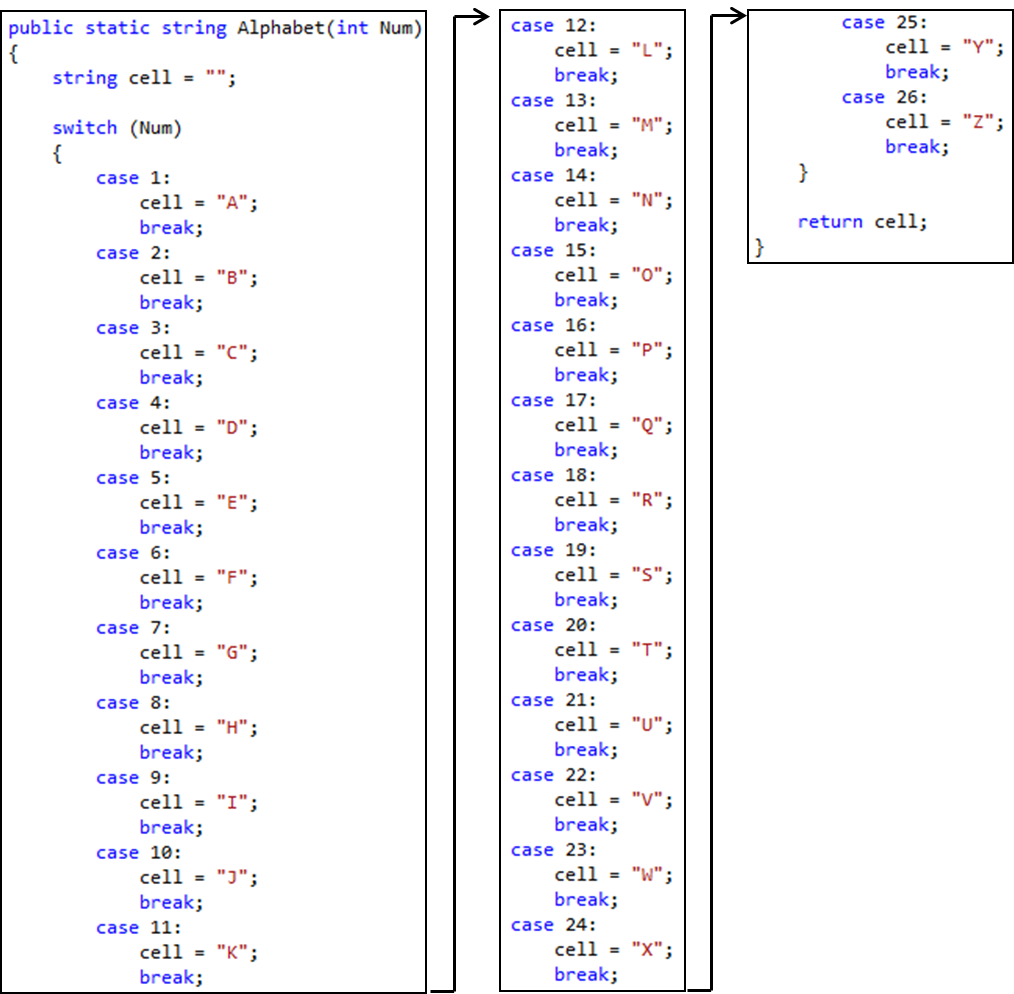


Рисунок 15 – Код метода *Alphabet*

**Памятка программисту №5. Пример создания многоуровневой шапки таблицы**

Рассматривается пример составления электронной таблицы, по структуре аналогичной третьей (если вести отсчёт сверху) из приведённых на Рисунке 16.

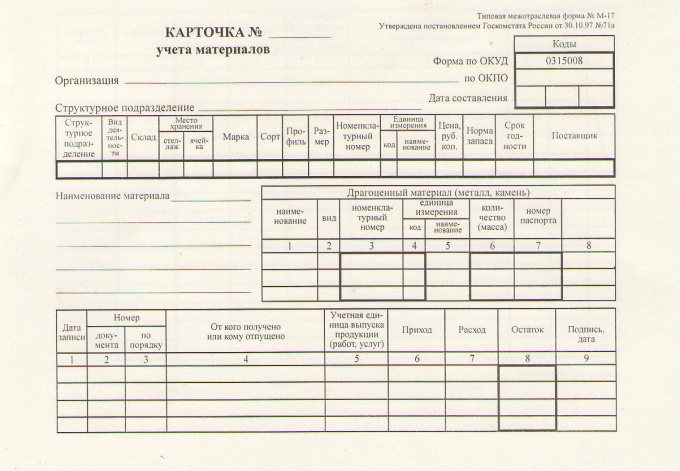


Рисунок 16 – Пример некоторой печатной формы

Перед составлением необходимо тщательно продумать алгоритм формирования таблицы заданной структуры. Очень важно составить блок-схему такого алгоритма, чтобы заведомо устранить возможную избыточность планируемых действий (Рисунок 17).

**Обобщённый алгоритм**



Рисунок 17 – Блок-схема для формирования электронной таблицы из рассматриваемого примера

**Формирование первой строки**

Разберём перечень действий, относящихся к формированию первой строки вида, представленного на Рисунке 18.

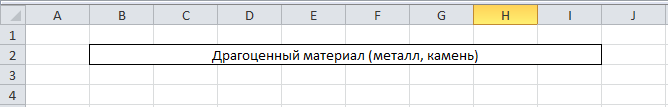


Рисунок 18 – Первая строка таблицы из примера

Перечень действий может быть описан кодом, аналогичным представленному во фрагменте на Рисунке 19.

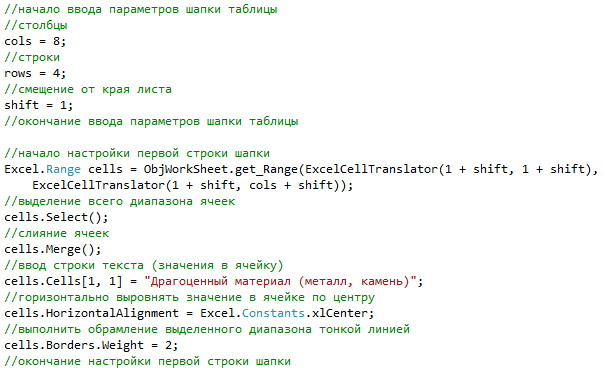


Рисунок 19 – Фрагмент кода, соответствующий составлению первой строки таблицы из примера

**Формирование второй и третьей строк**

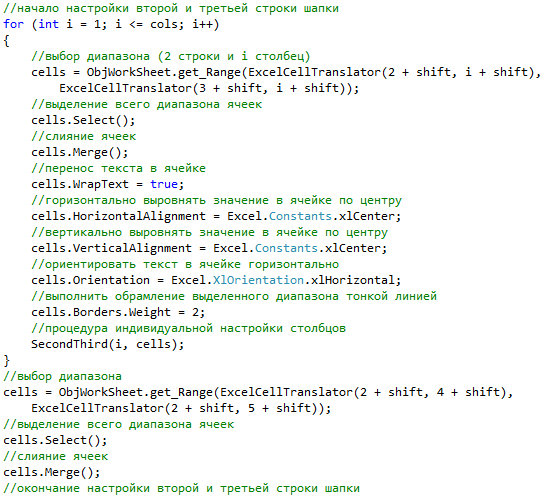


Рисунок 20 – Фрагмент кода, соответствующий составлению второй и третьей строк таблицы из примера

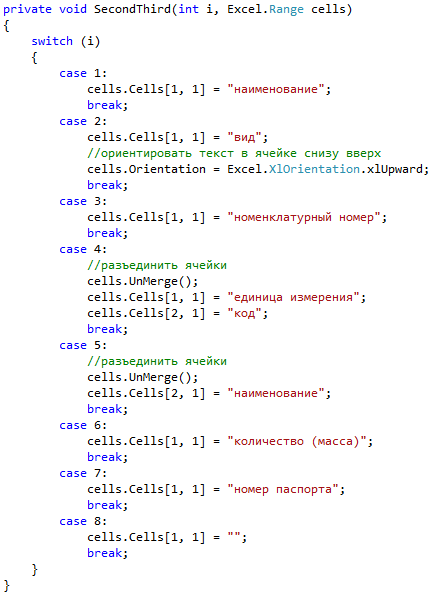


Рисунок 21 – Фрагмент кода вложенного метода, входящего в состав ранее представленного кода для составления второй и третьей строк таблицы из примера

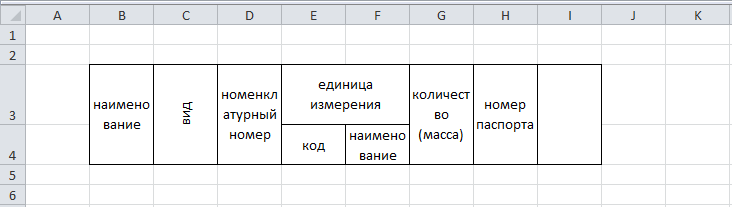


Рисунок 22 – Вторая и третья строки таблицы из примера. Для наглядности использовано иное управляющее воздействие в столбце «вид»

**Формирование четвёртой строки**

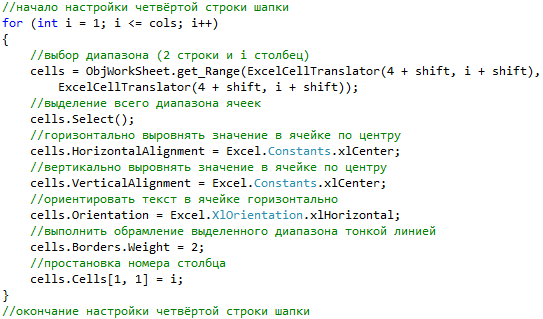


Рисунок 23 – Фрагмент кода, соответствующий составлению четвёртой строки таблицы из примера

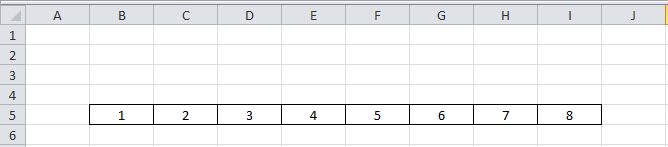


Рисунок 24 – Четвёртая строка таблицы из примера.

**Итоговый вид шапки таблицы из примера**

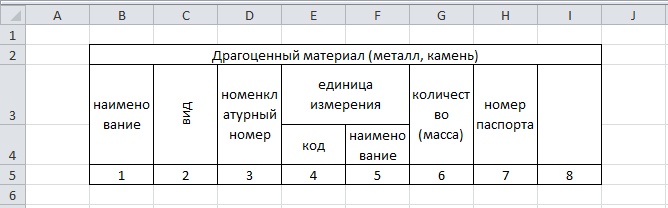


Рисунок 25 – Результат автоматизированного составления структуры электронной таблицы

**Определение рациональных размеров столбцов\*\*\*\***

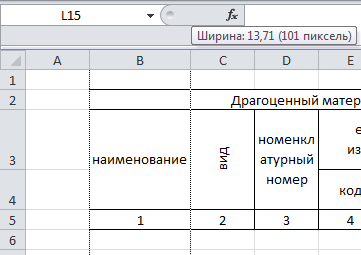


Рисунок 26 – Изменение ширины столбца и фиксация значения для последующего использования в настройке кодом

\*\*\*\* Начать растяжение столбца и зафиксировать значение ширины (13,71).

**Рационализированная шапка таблицы**



Рисунок 27 – Дополненный значениями ширины столбцов фрагмент кода для работы по составлению второй и третьей строк таблицы из примера. Фрагмент содержит и некоторые другие настройки по изменению ориентации текста, способствующие пониманию процессов автоматизации

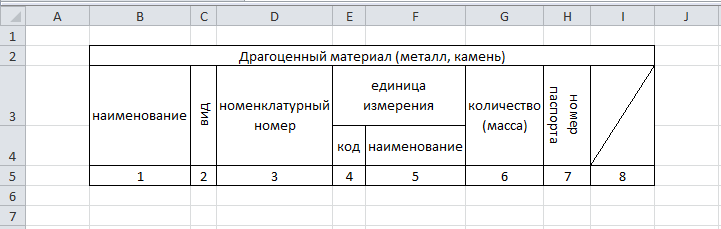


Рисунок 28 – Модифицированный вид таблицы из примера

Представленный ниже перечень иллюстраций на Рисунках 29-32 призван помочь разобраться с элементами интерфейса *Microsoft Office Excel* при сопоставлении их объектно-ориентированной модели.

**Соответствие границ**

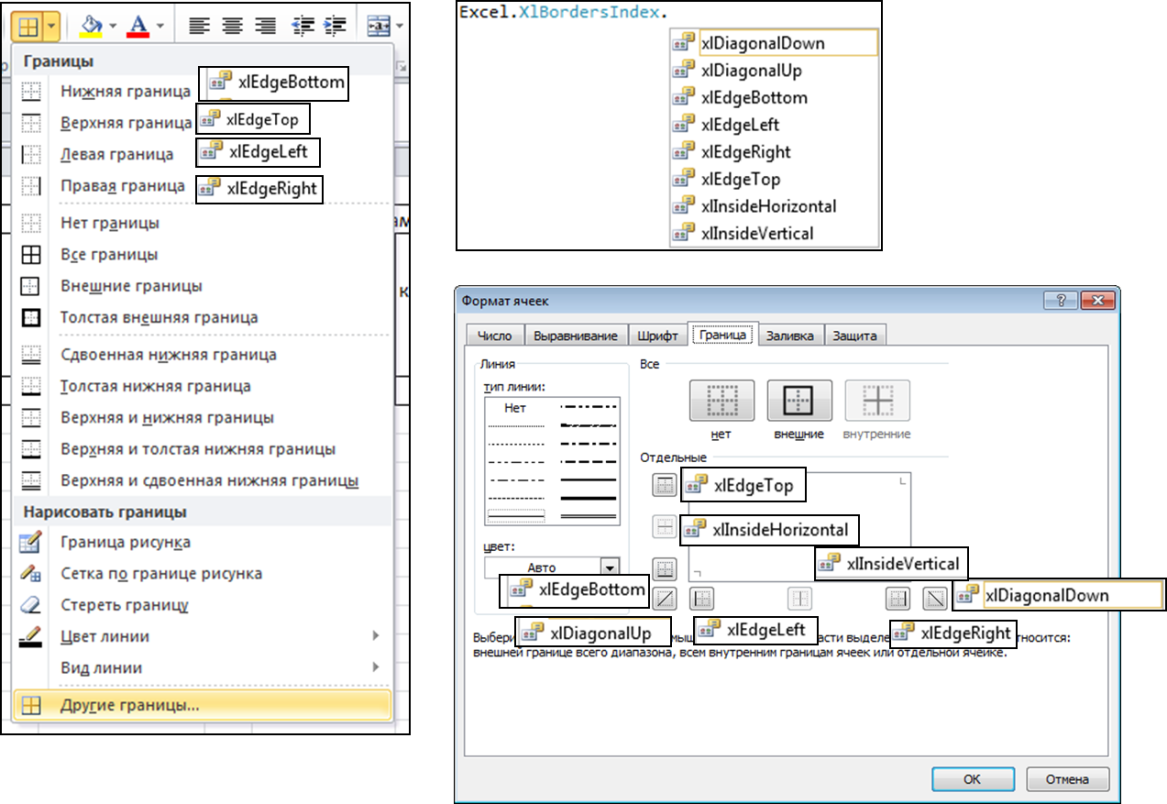


Рисунок 29 – Имена констант, соответствующих границам таблиц и ячеек в *Microsoft Office Excel*

**Соответствие типов линий**

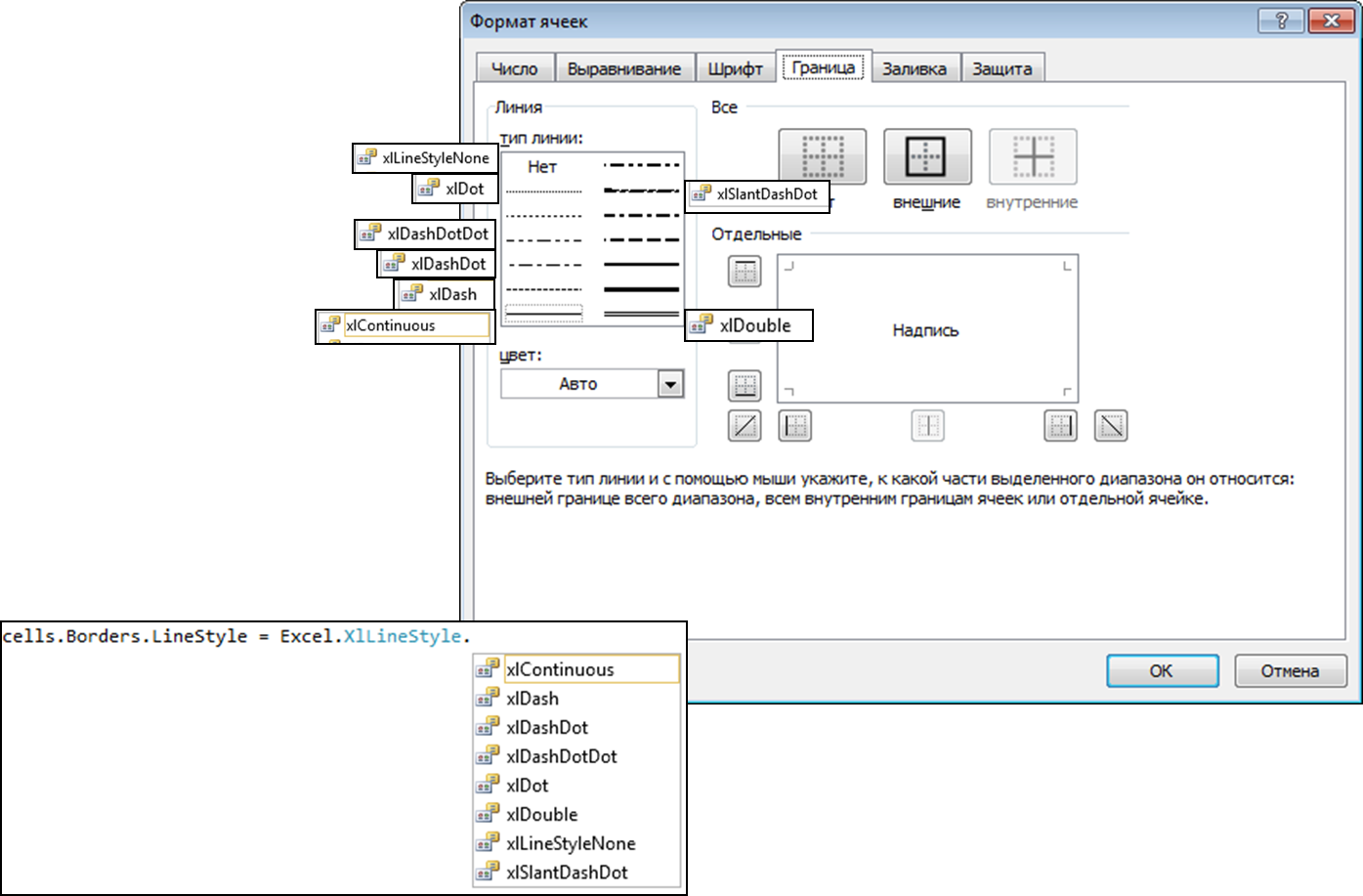


Рисунок 30 – Имена констант, соответствующих типам линий для границ таблиц и ячеек в *Microsoft Office Excel*

**Соответствие направлений текста**

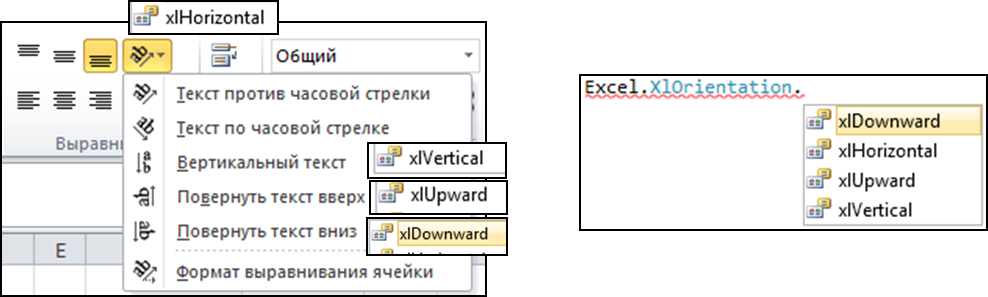


Рисунок 31 – Имена констант, соответствующих ориентации текста в ячейках *Microsoft Office Excel*

**Соответствие выравнивания текста**

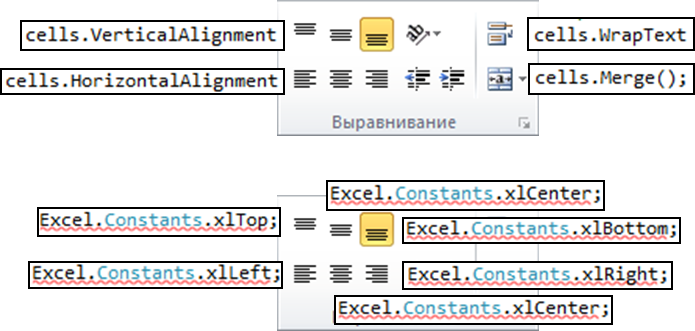


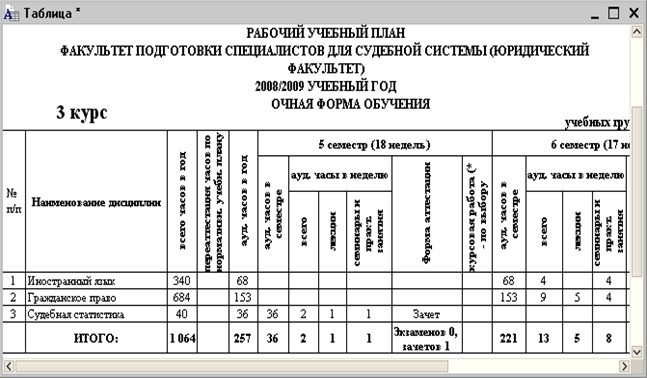
Рисунок 32 – Имена констант, соответствующих выравниванию текста в ячейках *Microsoft Office Excel*

**Варианты индивидуального задания**

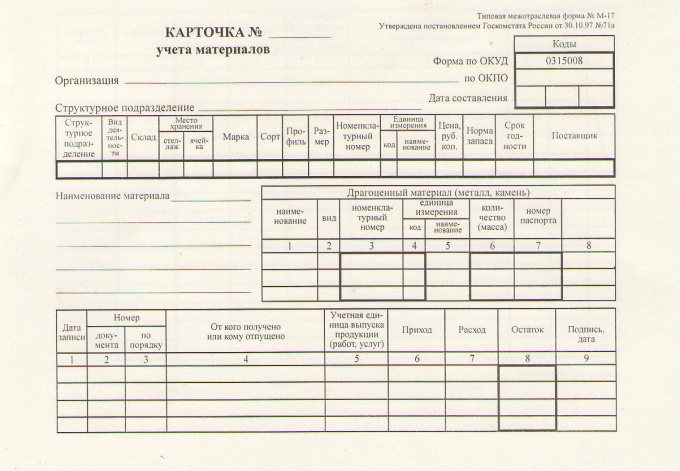
Вариант 1



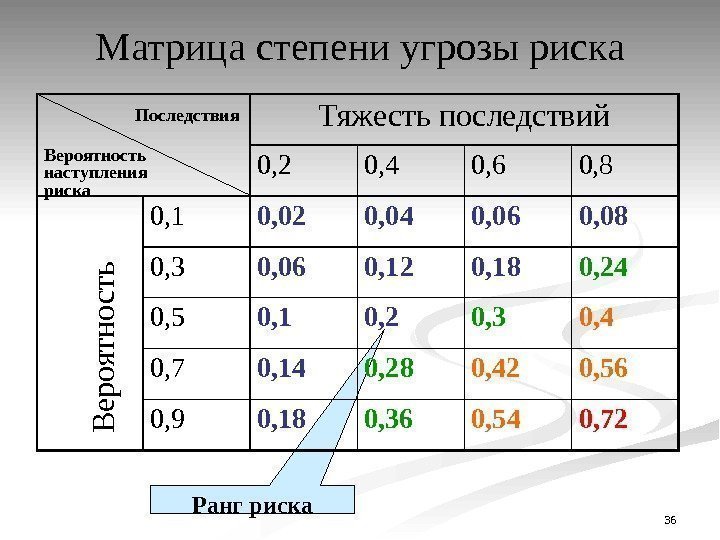
Вариант 2



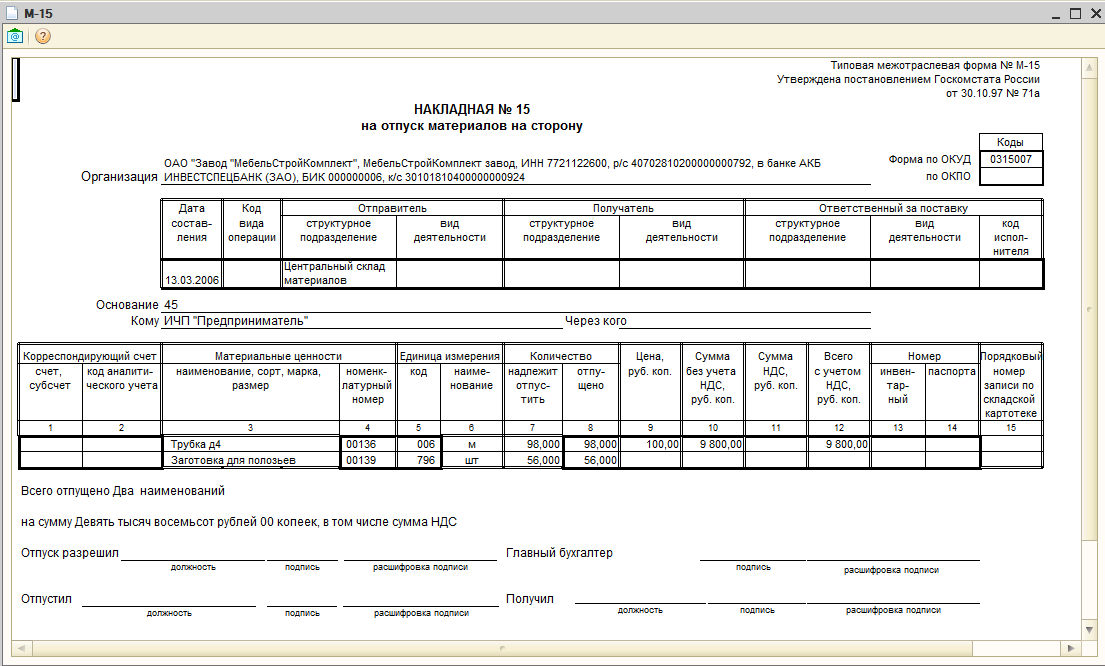
Вариант 3



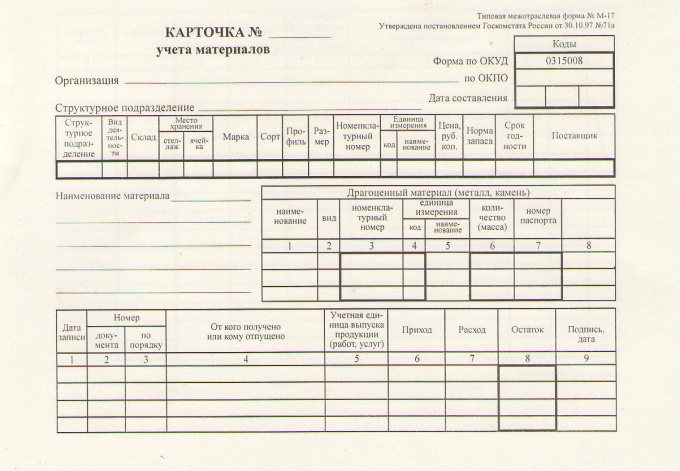
Вариант 4



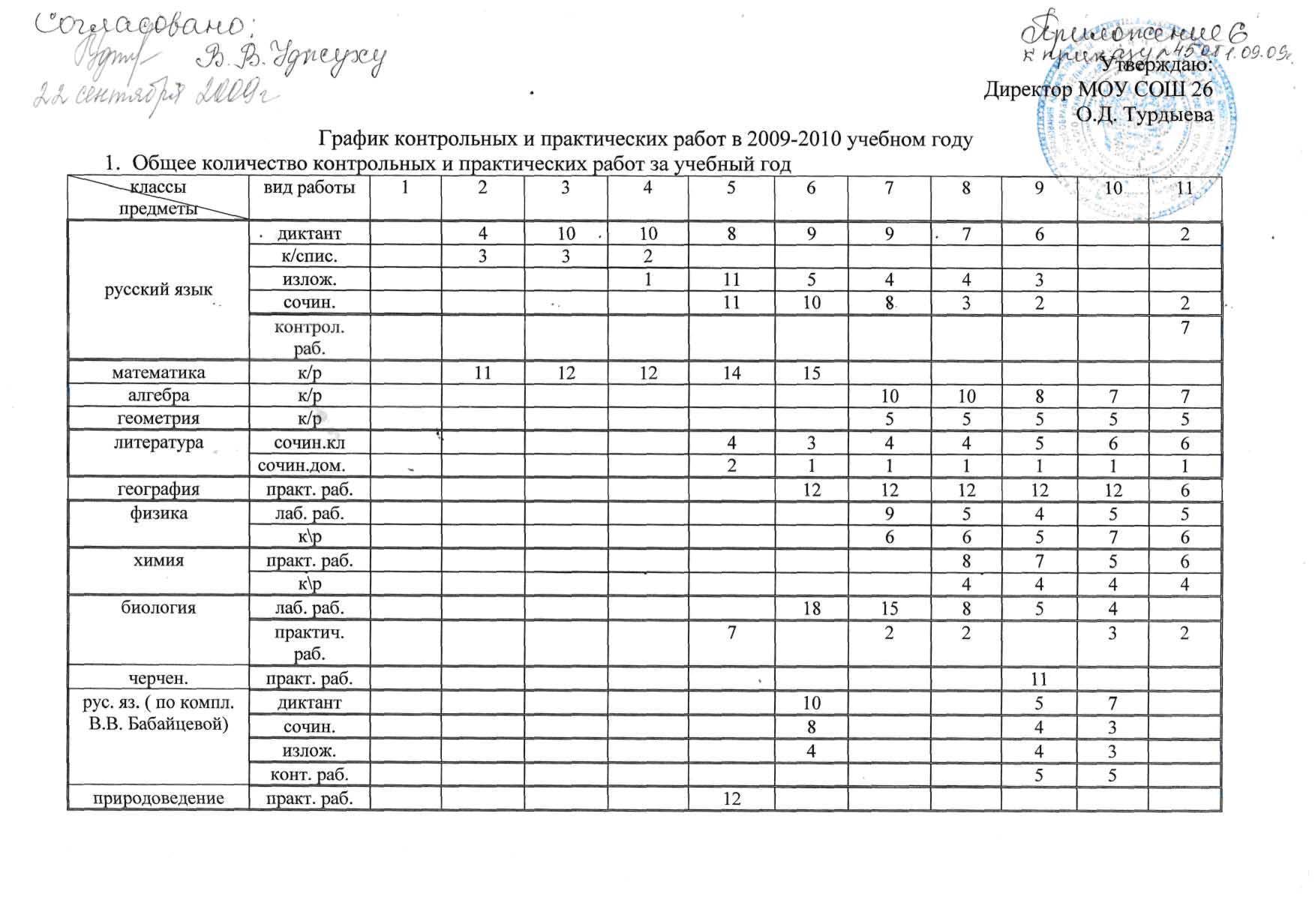
Вариант 5



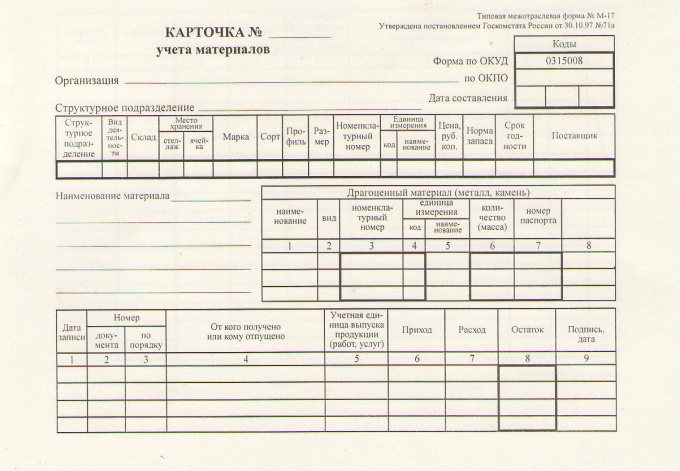
Вариант 6



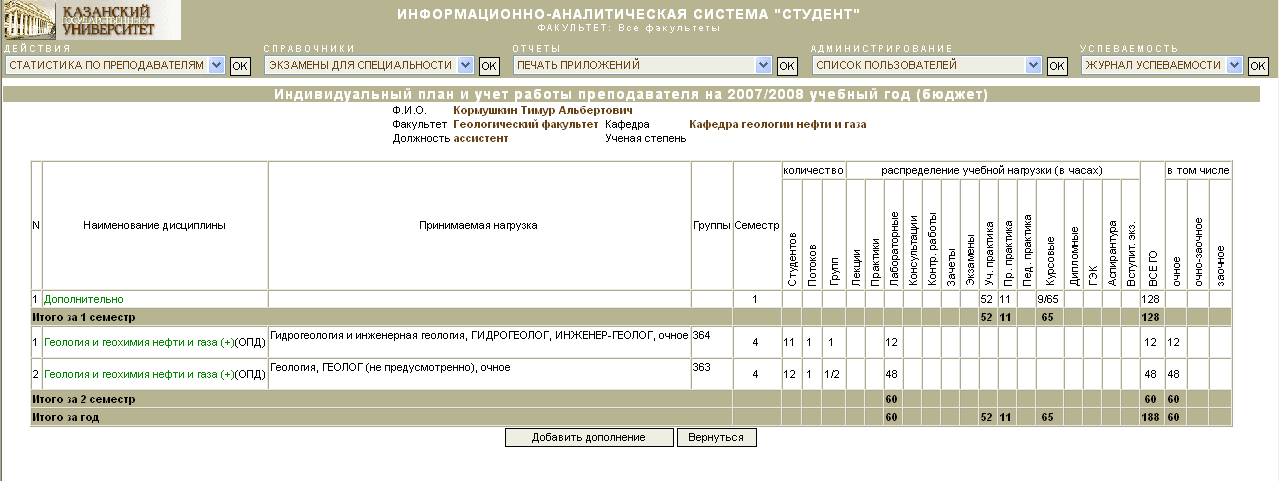
Вариант 7



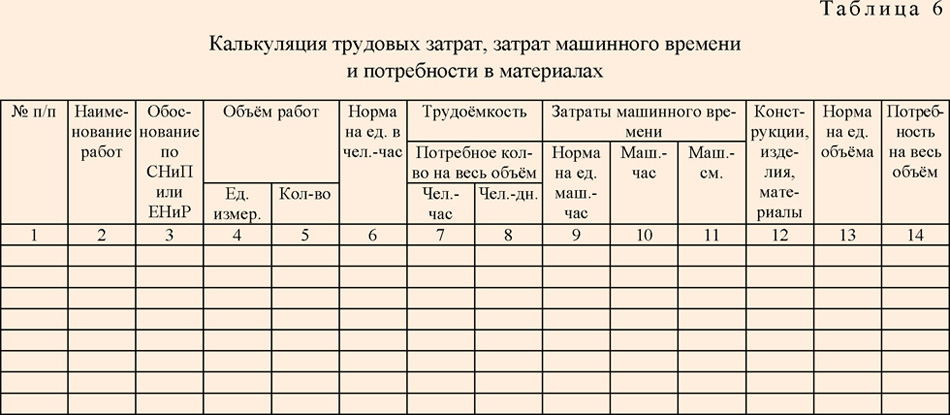
Вариант 8



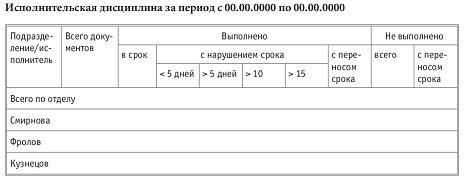
Вариант 9



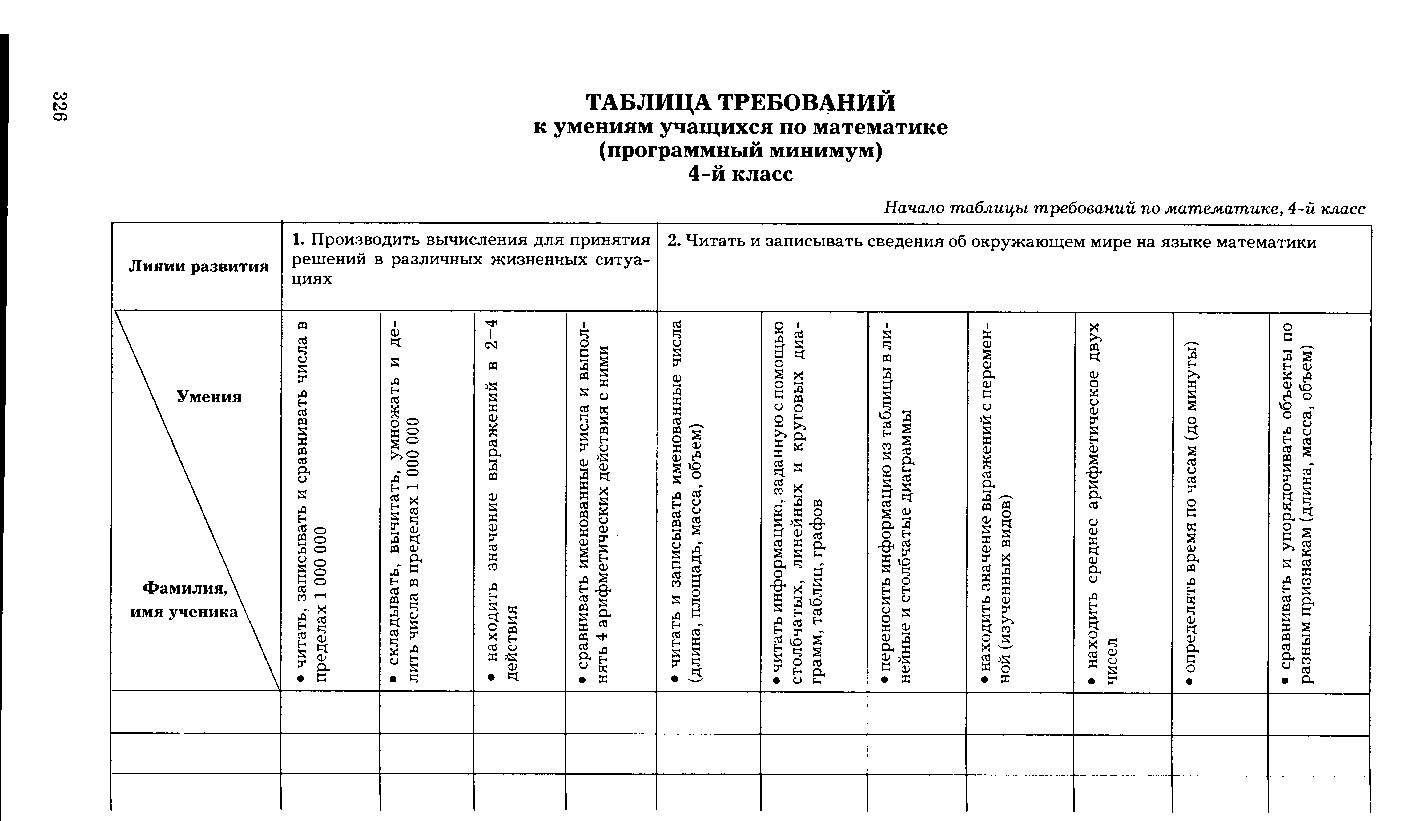
Вариант 10



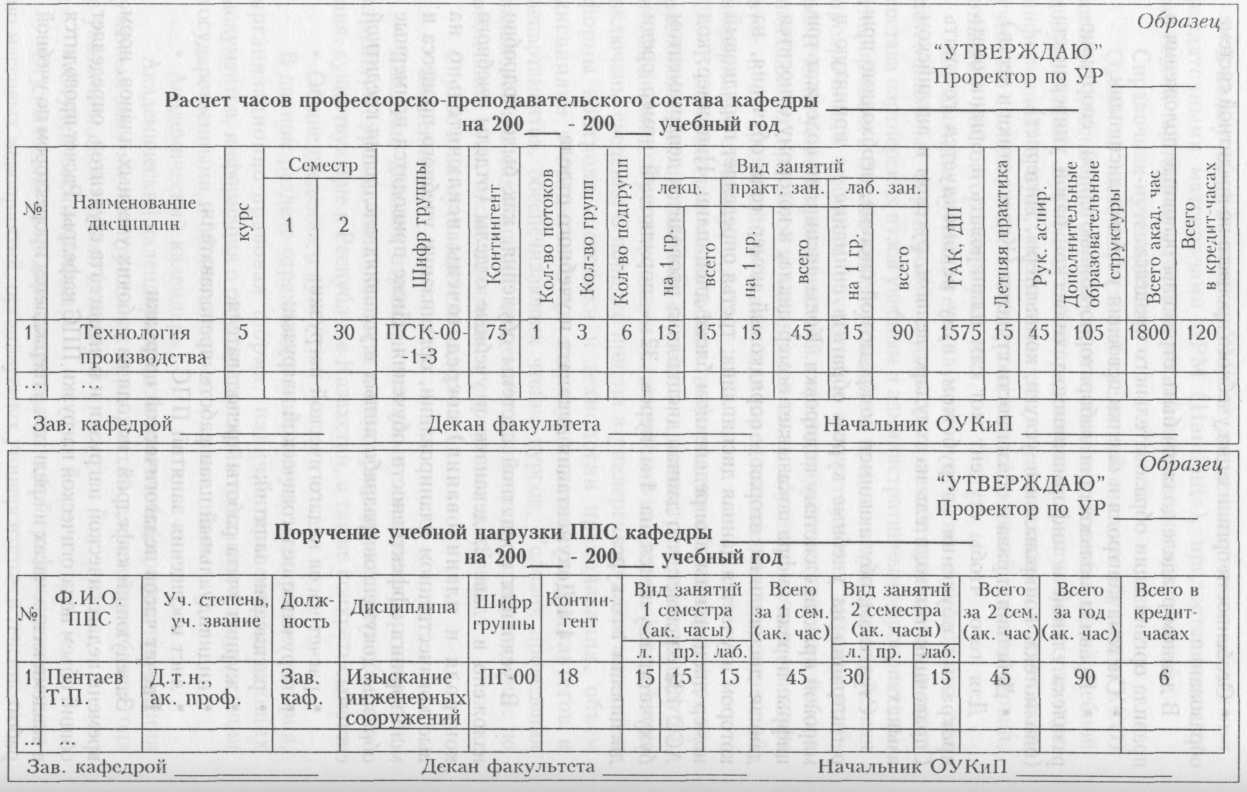
Вариант 11



Вариант 12



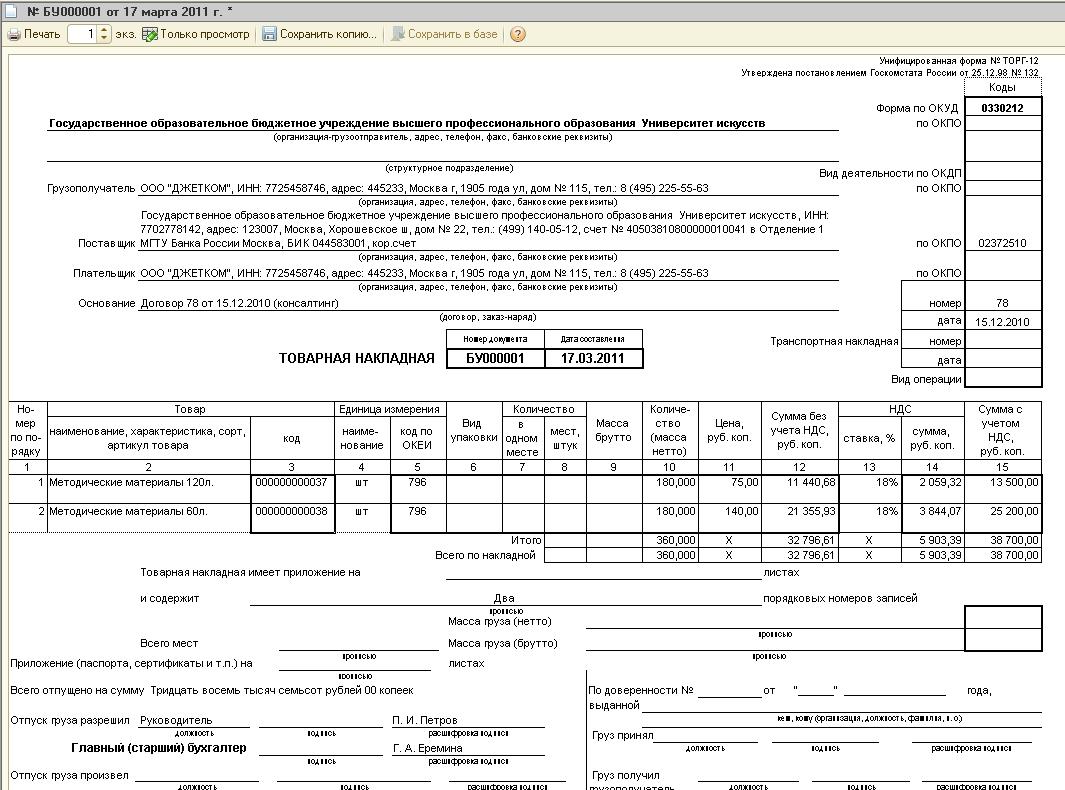
Вариант 13



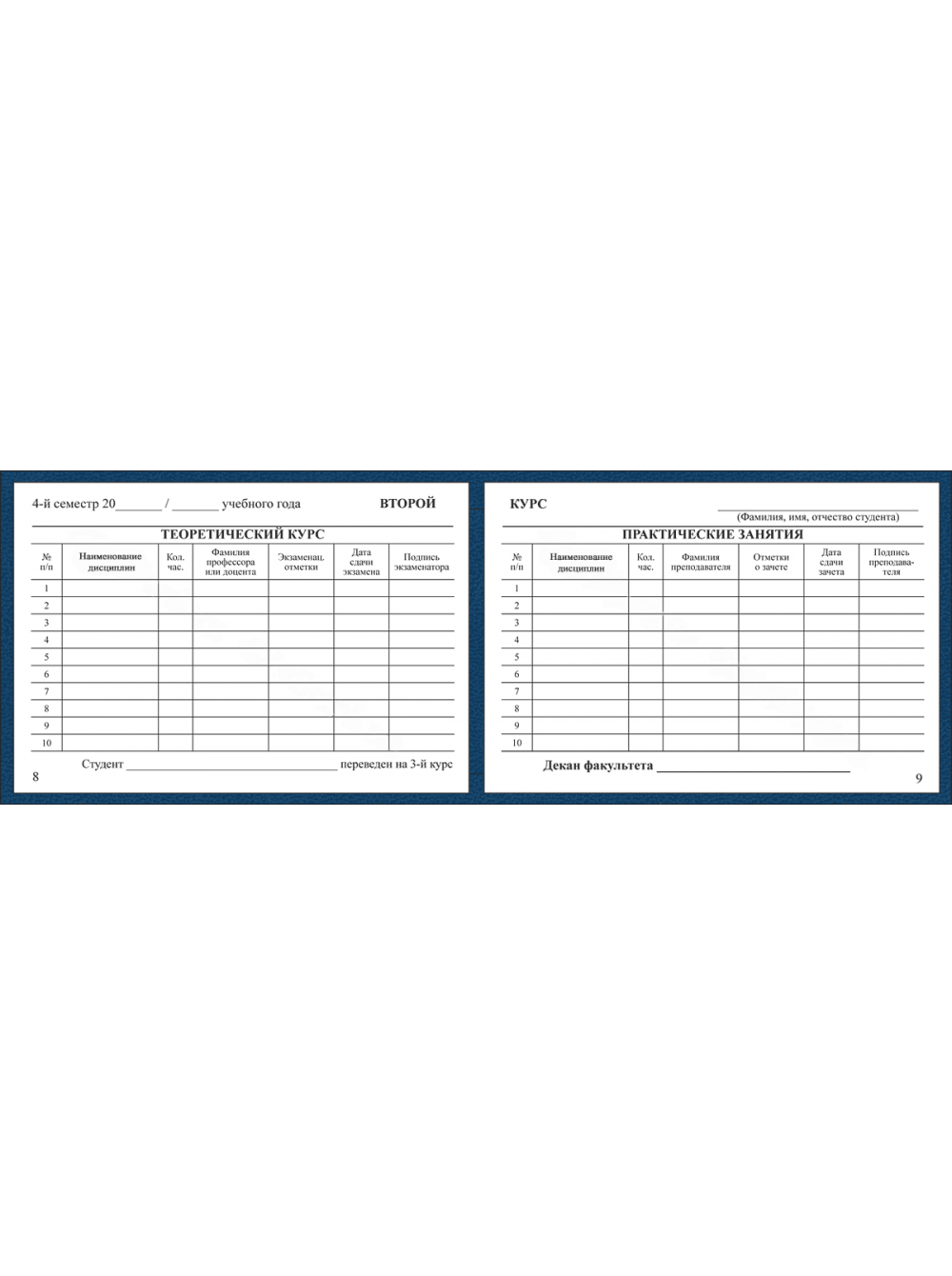
Вариант 14



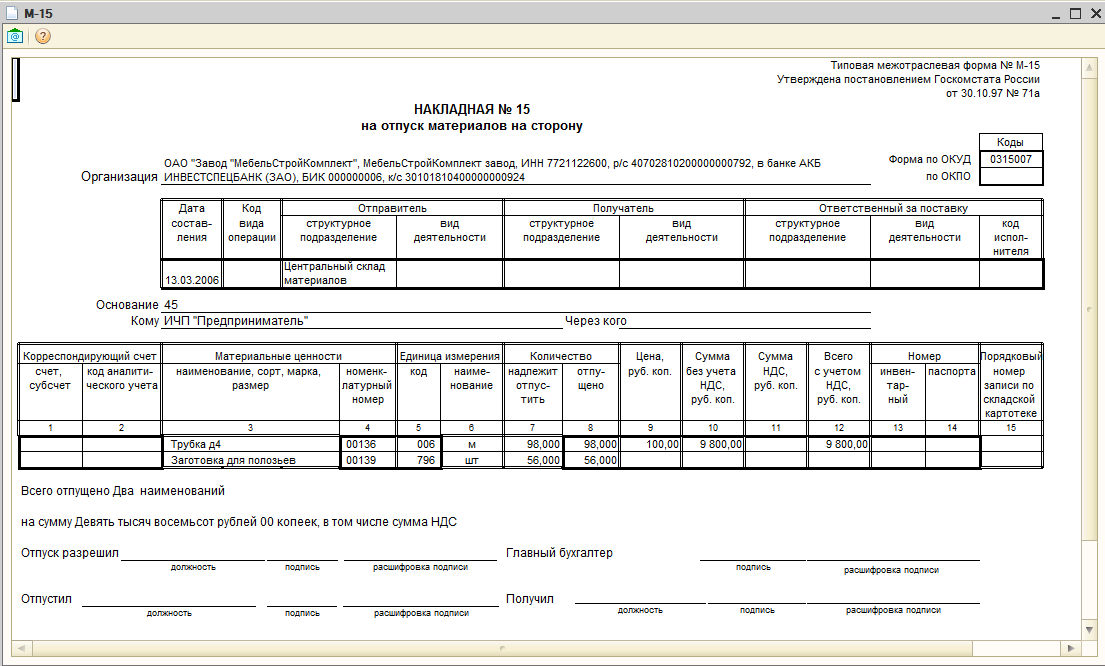
Вариант 15



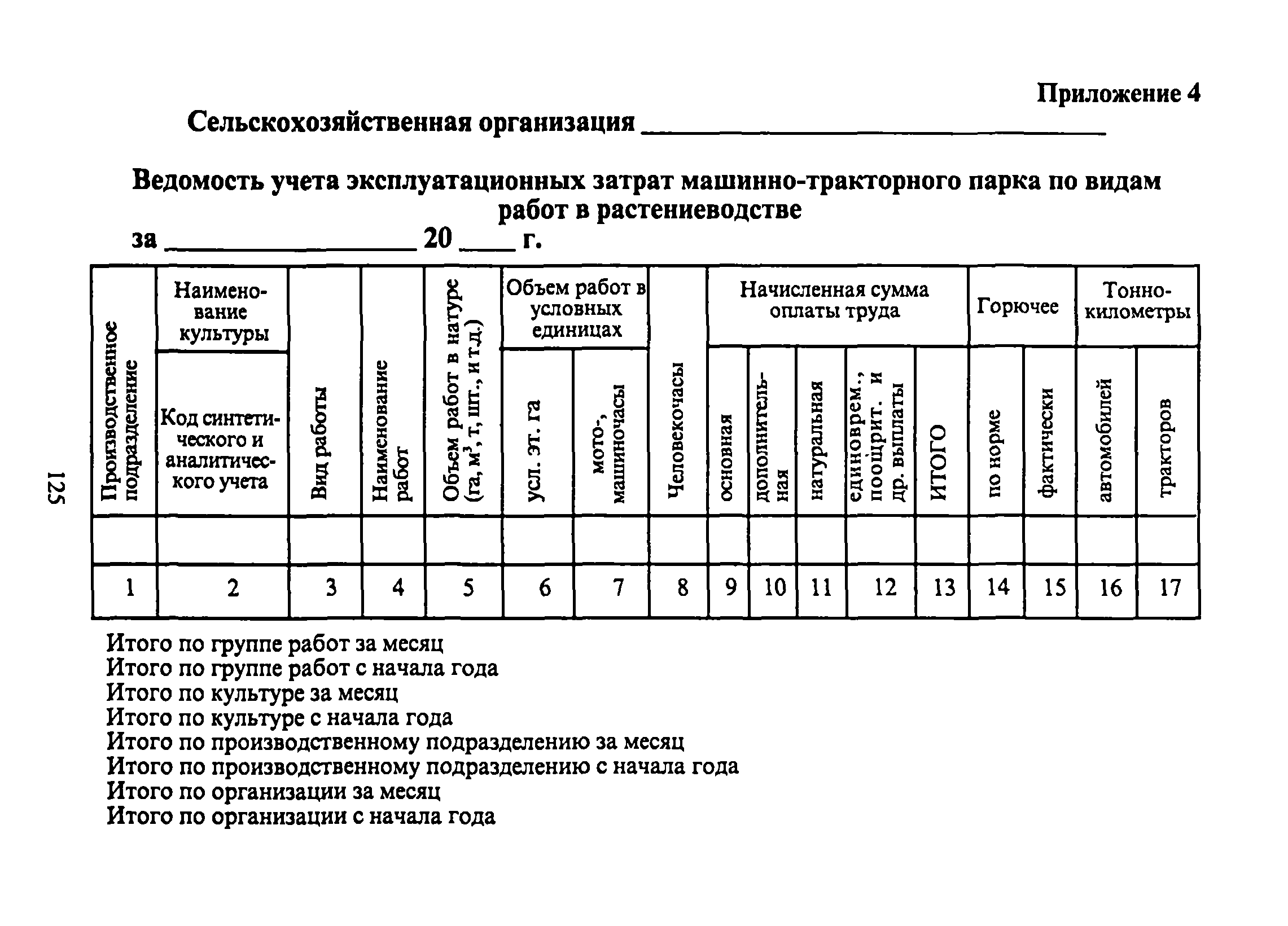
Вариант 16



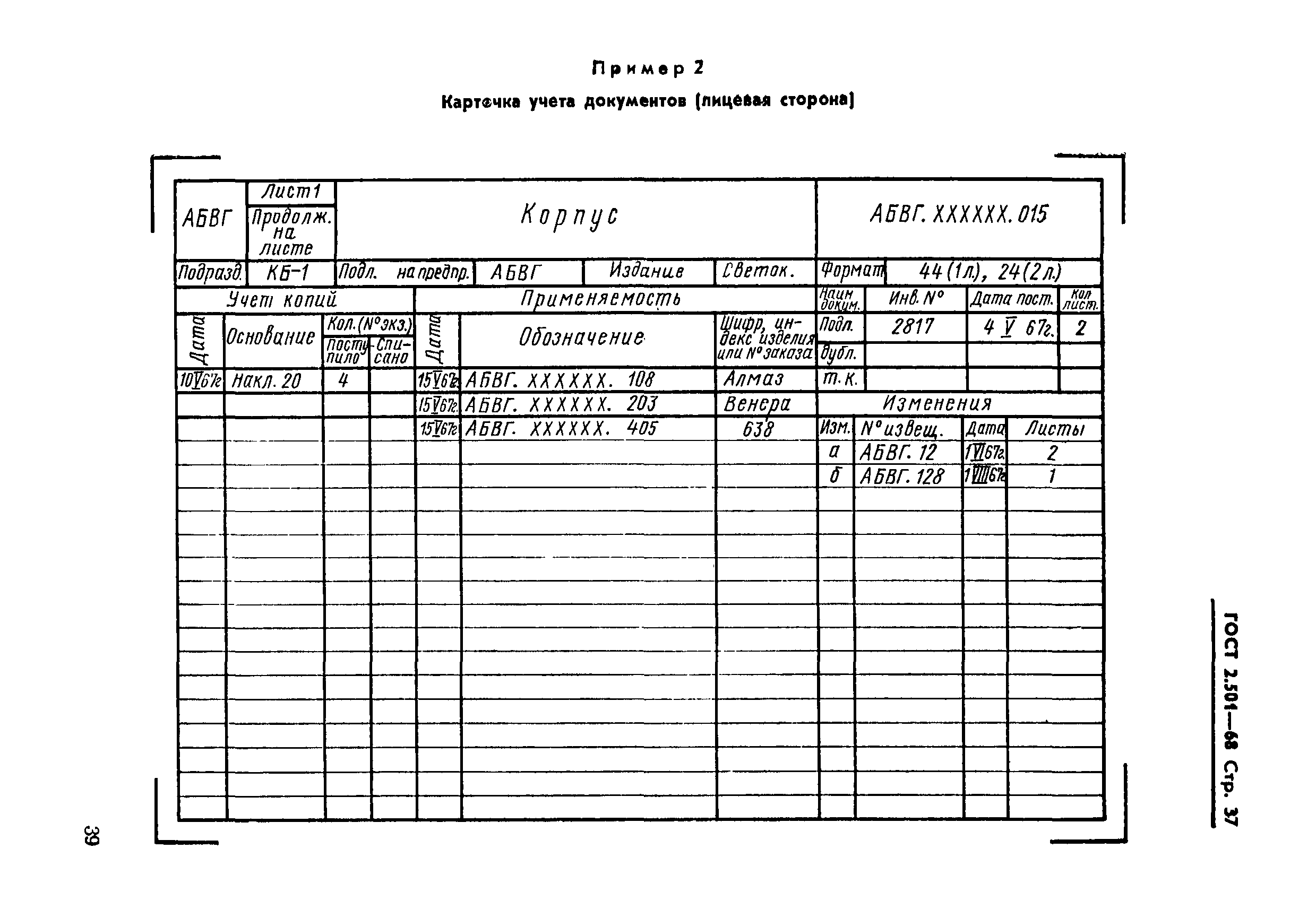
Вариант 17



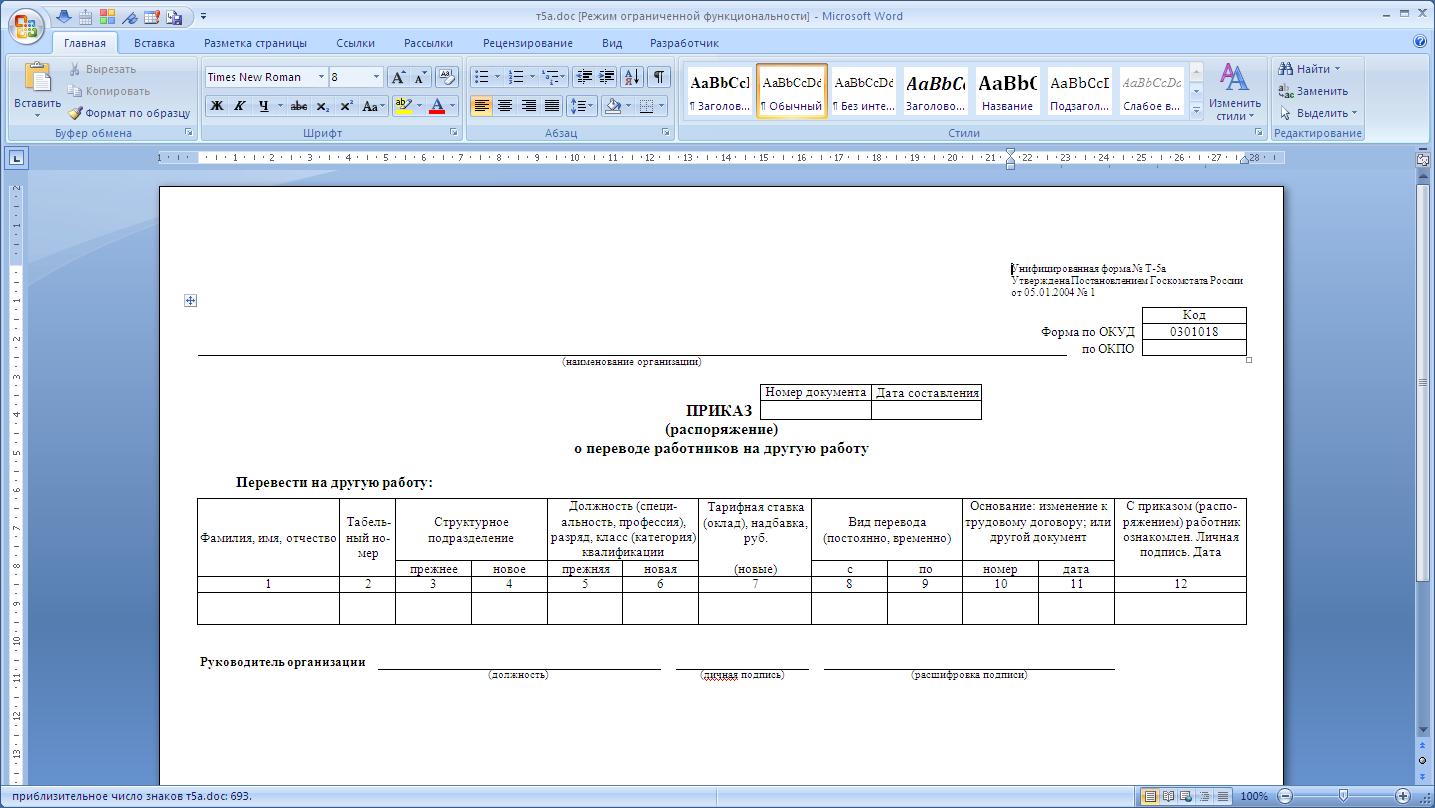
Вариант 18



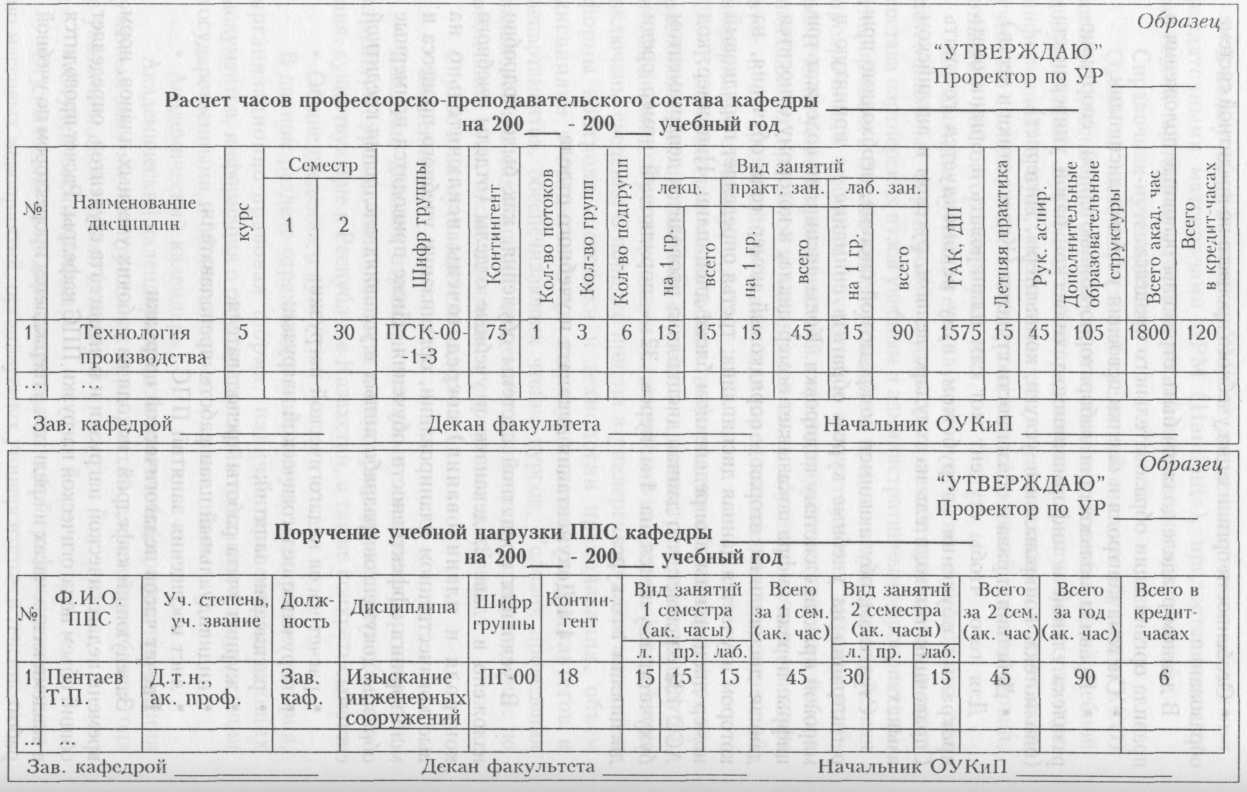
Вариант 19



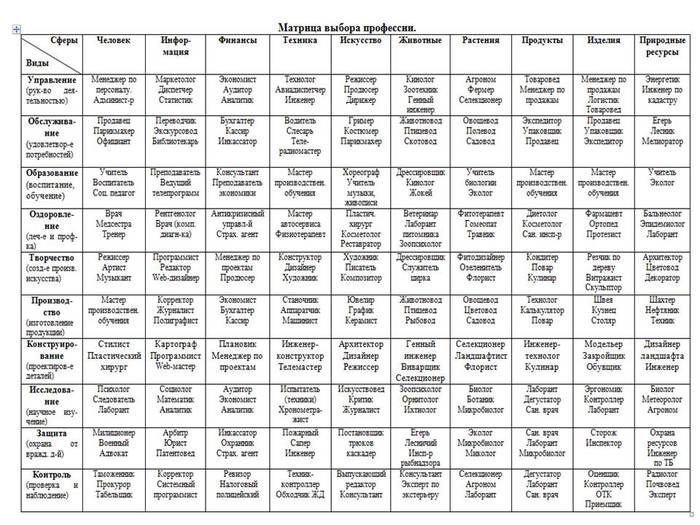
Вариант 20



Вариант 21



Вариант 22



Вариант 23



Вариант 24

