

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт **математики, физики и информационных технологий**  
Кафедра «**Прикладная математика и информатика**»

01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

### **БАКАЛАВРСКАЯ РАБОТА**

на тему Оптимизация информационной структуры сайта ТГУ в соответствии с требованиями нормативной базы

Студентка \_\_\_\_\_ Т.С. Фефелова \_\_\_\_\_

Руководитель \_\_\_\_\_ Г.А. Тырыгина \_\_\_\_\_

**Допустить к защите**  
Заведующий кафедрой к.т.н., доцент, А.В. Очеповский \_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.

Тольятти 2016

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»  
Институт математики, физики и информационных технологий  
Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ  
Зав.кафедрой «Прикладная  
математика и информатика»  
\_\_\_\_\_ А.В.Очеповский

« \_\_\_\_\_ » \_\_\_\_\_ 2016 г.

**ЗАДАНИЕ**

**на выполнение бакалаврской работы**

Студент Фефелова Татьяна Сергеевна

1. Тема Оптимизация информационной структуры сайта ТГУ в соответствии с требованиями нормативной базы

2. Срок сдачи студентом законченной бакалаврской работы  
24.06.2016

3. Исходные данные к бакалаврской работе Методические рекомендации представления информации об образовательной организации в открытых источниках с учетом соблюдения требований законодательства в сфере образования

(для образовательных организаций высшего образования), структура сайта ТГУ

4. Содержание бакалаврской работы (перечень подлежащих разработке вопросов, разделов)

Введение

- 1) Анализ структуры сайта ТГУ на соответствие требованиям нормативной базы
- 2) Оптимизация и проектирование информационной структуры сайта ТГУ
- 3) Разработка новой информационной структуры сайта ТГУ

Заключение

Список используемой литературы

5. Ориентировочный перечень графического и иллюстративного материала  
Презентация, графики, таблицы

6. Дата выдачи задания «11» января 2016 г.

Руководитель бакалаврской  
работы \_\_\_\_\_

\_\_\_\_\_ Г.А. Тырыгина

Задание принял к исполнению \_\_\_\_\_

\_\_\_\_\_ Т.С. Фефелова

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ  
Зав.кафедрой «Прикладная  
математика и информатика»  
\_\_\_\_\_ А.В.Очеповский

« \_\_\_\_ » \_\_\_\_\_ 2016 г.

**КАЛЕНДАРНЫЙ ПЛАН  
выполнения бакалаврской работы**

Студента Фефеловой Татьяны Сергеевны  
по теме Оптимизация информационной структуры сайта ТГУ в соответствии с  
требованиями нормативной базы

Наименование раздела работы	Плановый срок выполнения раздела	Фактический срок выполнения раздела	Отметка о выполнении	Подпись руководителя
Изучение теоретических основ	08.02.2016	08.02.2016	Выполнено	
Написание введения бакалаврской работы	11.02.2016	11.02.2016	Выполнено	
Изучение математических основ	13.02.2016	13.02.2016	Выполнено	
Написание 1 главы бакалаврской работы	15.02.2016	15.02.2016	Выполнено	
Создание ориентированного графа структуры сайта ТГУ	17.02.2016	17.02.2016	Выполнено	
Изменение структуры сайта	20.02.2016	20.02.2016	Выполнено	
Написание 2 главы	08.03.2016	08.03.2016	Выполнено	

бакалаврской работы				
Написание 3 главы бакалаврской работы	01.04.2016	01.04.2016	Выполнено	
Написание заключения бакалаврской работы	21.04.2016	21.04.2016	Выполнено	
Подведение итогов, редактирование бакалаврской работы	24.04.2016	24.04.2016	Выполнено	
Представление бакалаврской работы	01.05.2016	01.05.2016	Выполнено	
Создание презентационного материала	11.05.2016	11.05.2016	Выполнено	
Проверка на наличие заимствований (плагиата) в системе antiplagiat.ru	17.06.2016	17.06.2016	Выполнено	
Предварительная защита	30.05.2016 - 11.06.2016	30.05.2016 - 11.06.2016	Выполнено	
Сдача на кафедру отзыва научного руководителя и ознакомление с ним	20.06.2016	20.06.2016	Выполнено	
Сдача на кафедру комплекта документов для защиты	24.06.2016	24.06.2016	Выполнено	
Защита бакалаврской работы	27.06.2016 - 30.06.2016	27.06.2016 - 30.06.2016	Выполнено	

Руководитель бакалаврской работы \_\_\_\_\_

Г.А. Тырыгина \_\_\_\_\_

Задание принял к исполнению \_\_\_\_\_

Т.С. Фефелова \_\_\_\_\_

## **Аннотация**

Тема: Оптимизация информационной структуры сайта ТГУ в соответствии с требованиями нормативной базы.

Работа выполнена студенткой Тольяттинского государственного университета, института математики, физики и информационных технологий, группы ПМИБ-1201, Фефеловой Татьяной Сергеевной.

**Объект исследования** - сайт ТГУ.

**Предмет** – информационная структура сайта.

**Цель:** оптимизировать информационную структуру сайта ТГУ.

Для этого необходимо решить следующие **задачи**:

- проанализировать информационную структуру сайта ТГУ;
- проанализировать статью 29 Федерального закона об образовании и методические рекомендации Рособнадзора;
- сравнить существующую структуру сайта ТГУ и структуру из статьи 29 ФЗ об образовании, методических рекомендаций Рособнадзора;
- спроектировать информационную структуру сайта ТГУ;
- реализовать новую информационную структуру сайта.

Работа состоит из введения, трёх глав и заключения.

Во введении описываются актуальность, цели и задачи данной работы.

Первая глава работы посвящена анализу структуры сайта ТГУ на соответствие требованиям нормативной базы, а именно статье 29 Федерального закона «Об образовании» и методическим рекомендациям представления информации об образовательной организации в открытых источниках с учетом соблюдения требований законодательства в сфере образования.

Вторая глава описывает процесс оптимизации и проектирования информационной структуры сайта Тольяттинского государственного университета.

Третья глава раскрывает процесс реализации новой информационной структуры сайта.

Бакалаврская работа представлена на 46 страницах, включает 30 иллюстраций, 13 формул, список используемой литературы, состоящий из 21 источника.

## Оглавление

Введение.....	3
Глава 1 Анализ структуры сайта Тольяттинского государственного университета на соответствие требованиям нормативной базы.....	4
Глава 2 Оптимизация и проектирование информационной структуры сайта Тольяттинского государственного университета .....	8
2.1 Построение ориентированного графа структуры сайта Тольяттинского государственного университета .....	8
2.2 Оптимизация информационной структуры сайта .....	10
2.3 Закон Фиттса.....	11
2.5 Закон Хика .....	24
2.6 Проектирование информационной структуры сайта ТГУ.....	26
Глава 3 Разработка новой информационной структуры сайта Тольяттинского государственного университета.....	28
3.1 CMS «1С Битрикс: Управление сайтом».....	28
3.2 Инфоблоки в CMS «1С Битрикс: Управление сайтом» .....	30
3.3 Комплексный компонент «Новости» .....	31
3.4 Основной шаблон страниц .....	36
3.5 Разработка шаблона страницы «Структура и органы управления образовательной организацией».....	37
3.6 Разработка шаблона отображения детальной информации о подразделениях.....	41
Заключение .....	43
Список используемой литературы .....	45
Приложение А. Листинг кода файла template.php .....	47

## Введение

В нашей стране практически каждое высшее учебное заведение имеет собственный веб-сайт и Тольяттинский государственный университет не исключение. На сайтах университетов располагается такая информация, как структура ВУЗа, нормативные документы, информация о сотрудниках, информация для студентов и абитуриентов. В соответствии с Федеральным законом об образовании РФ [1], проектом постановления Правительства России об утверждении Порядка размещения на официальном сайте образовательной организации в информационно-телекоммуникационной сети «Интернет» и обновления информации об образовательной организации [3], методическими рекомендациями [2], на сайтах образовательных учреждений должны находиться определенные документы и все эти документы должны располагаться в специальном разделе «Сведения об образовательной организации». Сайт ТГУ должен соответствовать перечисленным выше нормативным актам. Это определяет актуальность данной работы.

Объект исследования: сайт ТГУ.

Предмет – информационная структура сайта.

Цель: оптимизировать информационную структуру сайта ТГУ.

Для этого необходимо решить следующие задачи:

- проанализировать информационную структуру сайта ТГУ;
- проанализировать статью 29 Федерального закона об образовании и методические рекомендации. Сравнить существующую структуру сайта ТГУ и структуру из статьи 29 ФЗ об образовании, методических рекомендаций;
- спроектировать информационную структуру сайта ТГУ;
- реализовать новую информационную структуру сайта.

## **Глава 1 Анализ структуры сайта Тольяттинского государственного университета на соответствие требованиям нормативной базы**

В наше время для каждого высшего учебного заведения сайт – это средство привлечения абитуриентов, средство информирования студентов и сотрудников. Можно сказать, что для ВУЗов веб-сайт становится необходимостью [16].

Согласно Федеральному закону от 29.12.2012 N 273-ФЗ "Об образовании в Российской Федерации" на официальном сайте образовательной организации в сети интернет должна публиковаться информация:

- о дате создания образовательной организации, об учредителе, учредителях образовательной организации, о месте нахождения образовательной организации и её филиалов (при наличии), режиме, графике работы, контактных телефонах и адресах электронной почты;
- о структуре и об органах управления образовательной организации;
- об уровне образования;
- о формах обучения;
- о нормативном сроке обучения;
- о сроке действия государственной аккредитации образовательной программы (при наличии государственной аккредитации);
- об описании образовательной программы с приложением её копии;
- об учебном плане с приложением его копии;
- об аннотации к рабочим программам дисциплин (по каждой дисциплине в составе образовательной программы) с приложением их копий (при наличии);
- о календарном учебном графике с приложением его копии;
- о методических и об иных документах, разработанных образовательной организацией для обеспечения образовательного процесса;

- о реализуемых образовательных программах с указанием учебных предметов, курсов, дисциплин (модулей), практики, предусмотренных соответствующей образовательной программой;
- о численности обучающихся по реализуемым образовательным программам за счёт бюджетных ассигнований федерального бюджета, бюджетов субъектов Российской Федерации, местных бюджетов и по договорам об образовании за счёт средств физических и (или) юридических лиц;
- о языках, на которых осуществляется образование (обучение);
- о федеральных государственных образовательных стандартах и об образовательных стандартах с приложением их копий (при наличии);
- о руководителе образовательной организации, его заместителях, руководителях филиалов образовательной организации (при их наличии);
- о персональном составе педагогических работников с указанием уровня образования, квалификации и опыта работы;
- о материально-техническом обеспечении образовательной деятельности;
- о числе вакантных мест для приёма (перевода) по каждой образовательной программе, профессии, специальности и направлению подготовки (на места, финансируемые за счёт бюджетных ассигнований федерального бюджета, бюджетов субъектов Российской Федерации, местных бюджетов, по договорам об образовании за счёт средств физических и (или) юридических лиц);
- о наличии и условиях предоставления обучающимся стипендий и мер социальной поддержки;
- о наличии общежития, интерната, количестве жилых помещений в них для иногородних обучающихся, о формировании платы за проживание в общежитии;

- об объёме образовательной деятельности, финансовое обеспечение которой осуществляется за счёт бюджетных ассигнований федерального бюджета, бюджетов субъектов Российской Федерации, местных бюджетов, по договорам об образовании за счёт средств физических и (или) юридических лиц;
- о поступлении финансовых и материальных средств и об их расходовании по итогам финансового года;
- о трудоустройстве выпускников.

Согласно методическим рекомендациям на сайте образовательной организации должен быть специальный раздел «Сведения об образовательной организации». В данном разделе должно быть 11 подразделов:

1. Основные сведения.
2. Структура и органы управления образовательной организацией.
3. Документы.
4. Образование.
5. Образовательные стандарты.
6. Руководство. Педагогический (научно-педагогический) состав.
7. Материально-техническое обеспечение и оснащённость образовательного процесса.
8. Стипендии и иные виды материальной поддержки.
9. Платные образовательные услуги.
10. Финансово-хозяйственная деятельность.
11. Вакантные места для приема (перевода).

Сейчас этот раздел располагается в разделе «Документы». На рисунке 1.1 можно увидеть главное меню сайта. На главную страницу сайта приходится около 55% просмотров, а на страницу «Сведения об общеобразовательной

организации» менее 2%.

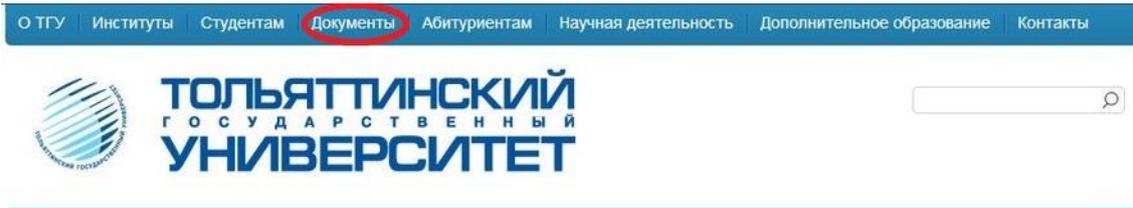


Рисунок 1.1 – Главное меню сайта ТГУ

Информация размещена в виде гиперссылок (Гиперссылка – часть гипертекстового документа, ссылающаяся на другой элемент в самом документе или на другой документ в сети Интернет) на документы или раздел на сайте ТГУ. На рисунке 1.2 можно увидеть страницу «Информационная открытость».

#### ТГУ: Информационная открытость

Федеральный закон от 29.12.2012 N 273-ФЗ "Об образовании в Российской Федерации"  
 Статья 29. Информационная открытость образовательной организации  
 1. Образовательные организации формируют открытые и общедоступные информационные ресурсы, содержащие информацию об их деятельности, и обеспечивают доступ к таким ресурсам посредством размещения их в информационно-телекоммуникационных сетях, в том числе на официальном сайте образовательной организации в сети "Интернет".  
 2. Образовательные организации обеспечивают открытость и доступность:  
 1) информации:  
 а) о дате создания образовательной организации, об учредителе, учредителях образовательной организации, о месте нахождения образовательной организации и ее филиалов (при наличии), режиме, графике работы, контактных телефонах и об адресах электронной почты;  
 б) о структуре и об органах управления образовательной организацией;  
 в) о реализуемых образовательных программах с указанием учебных предметов, курсов, дисциплин (модулей), практик, предусмотренных соответствующей образовательной программой;  
 г) о численности обучающихся по реализуемым образовательным программам за счет бюджетных ассигнований федерального бюджета, бюджетов субъектов Российской Федерации, местных бюджетов и по договорам об образовании за счет средств физических и (или) юридических лиц;  
 д) о языках образования;  
 е) о федеральных государственных образовательных стандартах, об образовательных стандартах (при их наличии);  
 ж) о руководителе образовательной организации, его заместителях, руководителях филиалов образовательной организации (при их наличии).

Сведения об образовательной организации
Лицензия
Свидетельство об аккредитации
Устав
ЕГРЮЛ
ИНН
Сертификат ARES 2014
Нормативные документы
Образцы бланков
Реклама в ТГУ

Рисунок 1.2 – Страница «Информационная открытость»

## Глава 2 Оптимизация и проектирование информационной структуры сайта Тольяттинского государственного университета

### 2.1 Построение ориентированного графа структуры сайта Тольяттинского государственного университета

Начнем с рассмотрения рисунка 2.1, который показывает часть дорожной карты и части электрической сети.

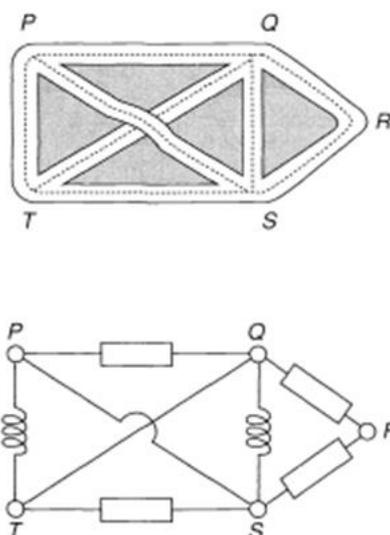


Рисунок 2.1 – Часть дорожной карты и часть электрической сети

Любая из этих ситуаций может быть представлена схематически посредством пунктов и линий, как на рисунке 2.2. Точки P, G, R, S и T называются вершинами, линии называются ребра, а вся схема называется графом. Обратите внимание на то, что пересечение PS линий и QT не является вершиной, так как оно не соответствует перекрестку или пересечению двух проводов. Степенью вершины называется число ребер, выходящих с этой вершины в качестве конечной точки, на рисунке 2.1 она соответствует количеству дорог на перекрестке. Например, степень вершины Q = 4.

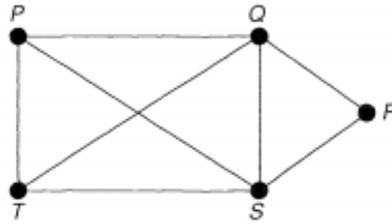


Рисунок 2.2 – Пример графа

Изучение ориентированных графов (или орграфов, как их сокращенно называют) происходит от создания дороги на улицах с односторонним движением. Пример орграфа приведен на рисунке 2.3, направления на улицах с односторонним движением обозначены стрелками.

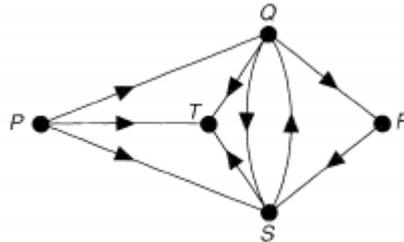


Рисунок 2.3 – Пример ориентированного графа

Построим граф структуры сайта, вершинами которого будут страницы меню, ребрами обозначим связи между страницами.

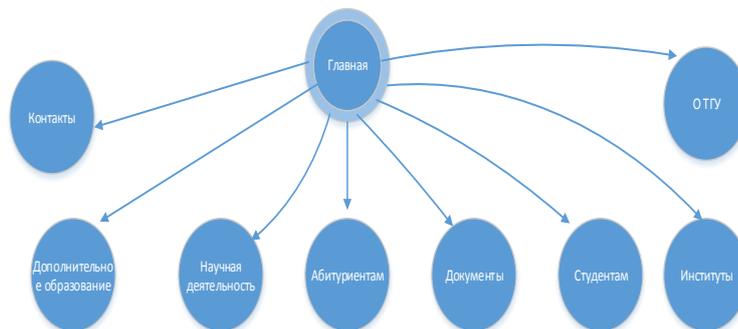


Рисунок 2.4 – Граф структуры сайта ТГУ

Как видно из рисунка 2.4, все разделы связаны с главной и не имеют связи между собой. Чтобы найти связанные элементы, пользователю придется

искать их по сайту. Нет полностью структурированной и связанной информации.

## 2.2 Оптимизация информационной структуры сайта

Оптимизация — модификация системы для улучшения её эффективности. Система может быть одиночной компьютерной программой, цифровым устройством, набором компьютеров или даже целой сетью, такой как Интернет. В нашем случае целью оптимизации будет соответствие структуры сайта в соответствии с требованиями нормативной базы и увеличение числа просмотров раздела «Сведения об образовательной организации».

Пусть  $W = \{w_1, w_2, w_3 \dots w_n\}$  — набор страниц исследуемого веб-сайта,  $w_1$  — стартовая страница этого веб-сайта. Вектор начальных состояний определяет вероятность того, что пользователь выберет страницу из множества  $W$  при начале просмотра сайта.

Множество ссылок между страницами сайта обозначим через  $L = \{L(i, j)\}^{n,n}$ , где  $L(i, j)$  — количество ссылок на странице  $i$ , переходящих на страницу  $j$ . Если ссылки нет, то  $L(i, j) = 0$ , если ссылка одна, то  $L(i, j) = 1$ , две —  $L(i, j) = 2$  и т.д.

Каждая ссылка  $L(i, j)$  имеет вероятность  $P(i, j, a)$ , что пользователь по ней перейдет. Сумма  $\sum_1^m P_{n_m} + g + z = 1$  для каждого  $n$ . Здесь  $g$  — вероятность того, что пользователь покинет сайт,  $z$  — фактор перезагрузки сессии. Вероятность, что пользователь покинет сайт (соответствующим образом будут пересчитываться вероятности) и фактор перезагрузки сессии будут постоянными.

Будем считать, что  $P_{n_m}$  зависит только от количества ссылок на странице  $w_n$ . Вероятность перехода с  $i$ -й страницы на  $j$ -ю определяется по формуле 2.1:

$$P(i, j) = \frac{L(i, j)(1-g-z)}{\sum_{k=1}^n L(i, k)}, \quad (2.1)$$

где  $P(i, j)$  — вероятность перехода с  $i$ -й страницы на  $j$ -ю,

$L(i, j)$  — количество ссылок на странице  $i$ ,

$g$  – вероятность того, что пользователь покинет сайт,

$z$  – фактор перезагрузки сессии.

Из общего числа просмотров страниц сайта ТГУ около 55% - просмотры главной страницы, и 37% - просмотры страниц главного меню. Следовательно, чтобы увеличить количество просмотров раздела на сайте, нужно поместить ссылку на этот раздел в главное меню [10].

### **2.3 Закон Фиттса**

Дословный перевод английского слова “usability” – это удобство и простота использования. Под понятием «юзабилити сайта» подразумевается простое и удобное использование сайта. Очевидно, что чем больше уровень удобства сайта, тем более расположены к нему пользователи. Это напрямую воздействует на лояльность, число посещений и даже конверсию на вашем сайте [21].

Все эти характеристики можно улучшить практически на всех ступенях разработки сайта: при его визуальном оформлении, структурировании навигации, размещении контента. Итак, цель улучшения юзабилити сайта — сделать так, чтобы уже при самом первом взгляде пользователь мог понять, о чём сайт и как им пользоваться.

Существует немного универсально принятых законов юзабилити. Закон Фиттса - один из этих немногих. В законе говорится, что пользователи могут достигнуть цели быстрее, если цель больше или ближе к курсору мыши пользователя (в операционной системе). Цель может быть элементом на экране, на который пользователи пытаются щелкнуть мышью или кнопкой, на которую они пытаются нажать на сенсорном экране.

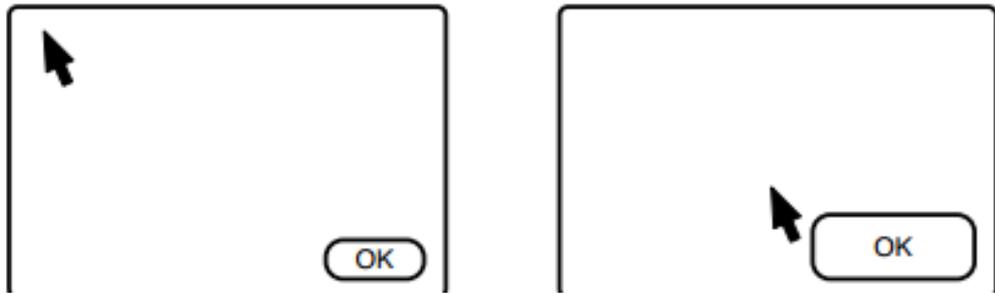


Рисунок 2.5 - Нажатие на кнопку в первом случае займет больше времени, чем во втором

Направление движения к объекту. Размер целевого объекта должен соответствовать направлению движения.

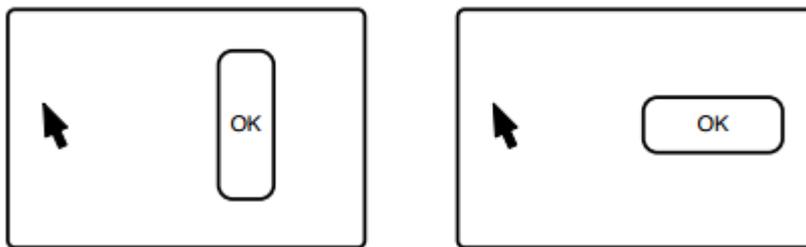


Рисунок 2.6 - Нажатие на кнопку в первом случае займет больше времени, чем во втором

Закон Фиттса назван в честь его первооткрывателя, Пола Фиттса, психолога из Университета штата Огайо, а затем Университета штата Мичиган.

Эта идея кажется довольно простой, и в самом деле, ряд исследований показал, что она верна. Соответствующая часть формулы, используемой для вычисления времени, которое требуется пользователю, чтобы достигнуть цели, называется индексом трудности (ID), который показывает, насколько трудно достичь цели. Эта часть формулы выглядит следующим образом:

$$ID = \log_2 \left( \frac{D}{W} + 1 \right), \quad (2.2)$$

где ID – индекс трудности,

D (расстояние) – измерение дистанции от начальной до конечной точки (до целевого объекта),

W (ширина) – ширина целевого объекта.

Так как формула для ID логарифмическая, даже небольшие изменения в размере могут привести к сравнительно большим изменениям в трудности, если

цель маленькая сначала. Для больших целей небольшие изменения имеют небольшой эффект.

Хотя сам закон кажется достаточно очевидным, некоторые его эффекты могут не быть очевидными. Давайте посмотрим на некоторых.

Элемент интерфейса в самом краю экрана чрезвычайно бесконечен в размере, при попытке навести на него мышью. Независимо от того, как далеко в направлении края экрана вы перемещаете свою мышь, он никогда не будет оставлять экран и всегда оставаться в поле видимости. Таким образом, имеет смысл помещать важные элементы в самый край экрана. Достичь таких элементов намного легче, потому что люди могут просто переместить мышь к краю, и курсор остановится в цели самостоятельно.

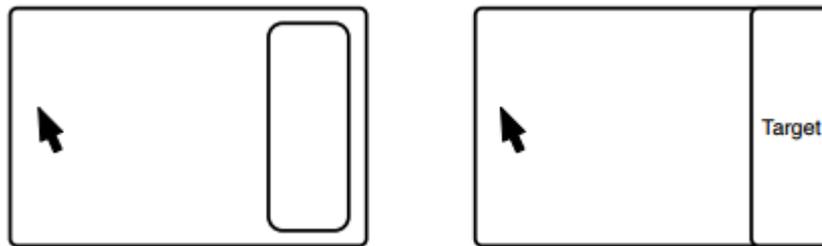


Рисунок 2.7 - Левая цель сложнее достигается, чем правая

Углы особенно ценны, потому что у них есть два бесконечных края.

К сожалению, в большинстве случаев этот же эффект не относится к сенсорным экранам. Достичь элементы ближе к краю сенсорного экрана не легче, чем элементы в другом месте. Области, которых может наиболее легко достигнуть пользователь, зависят от того, как он держит устройство.



Рисунок 2.8 - Если Вы держите свое устройство как это, только части вашего экрана могут легко быть достигнуты

Конечно, когда пользователь перетягивает элементы, край сенсорного экрана становится более легкой целью снова, потому что невозможно тянуть элементы за пределами экрана.

Как уже упоминалось, в углы особенно легко попасть, потому что у них есть две бесконечные края. Существует, однако, одна точка на экране, в которую даже легче попасть: точка под курсором. Вы не должны перемещать мышь вообще, чтобы достичь её. Контекстные меню используют это.

Если есть всплывающее меню под курсором мыши, то имеет смысл организовать отдельные пункты меню вокруг курсора мыши, чтобы уменьшить среднее расстояние до каждой записи. Один из способов добиться этого с помощью радиального контекстного меню.

Многие игры используют радиальное контекстное меню. Вот пример от LucasArts под названием Full Throttle:



Рисунок 2.9 – Пример радиального контекстного меню

Расширение Firefox easyGestures также использует радиальное контекстное меню:



Рисунок 2.10 – Радиальное контекстное меню расширения Firefox easyGestures

Таким образом, если они работают так хорошо, почему радиальные контекстные меню не используются чаще? Одна причина может состоять в том, что трудно вместить много записей меню в маленький круг. Язык майя обходит

это при помощи ярлыков обычного меню, но располагая их в кругу вокруг мыши.



Рисунок 2.11 – Пример контекстного меню

Одну из причин, по которой радиальные контекстное меню не могут быть более популярными, можно найти в VisionScience [Pal99] Стивен Э. Палмер. Объясняя одну модель того, как люди чувствуют визуальную информацию, он отмечает, что «клетки в первой области зрительной зоны коры головного мозга удлинили восприимчивые области, которые отвечают наиболее энергично, если стимулируется краем или линией при конкретной ориентации и положении». Другими словами, кажется, что у людей есть специальные механизмы для наблюдения линий. Более того, мы можем лучше видеть горизонтальные и вертикальные линии, чем линий, идущие в других направлениях. В Sensation and Perception [Cor99], Стэнли Корен полагает, что, так как люди проводят большую часть своего времени в прямолинейных условиях и постоянно подвергаются воздействию вертикальных и горизонтальных линий, наш мозг направлен в сторону линий в этих направлениях.

Действительно, в газете, названной «Восприятие ориентации контура в центральной первой части: Короткие линии» Д.П. Эндрюсом отмечают, что во многих экспериментах, «острота для ориентации была лучше всего около горизонтальных и вертикальных направлений». Другие исследования с этим соглашаются. Например, в газете, названной «Человеческая дискриминация ориентации, проверенная с длинными стимулами», Гай А. Орбан приходит к выводу, что «чувствительность ориентации лучше для узкого круга ориентаций вокруг основных меридианов».

Чтобы сделать действительно длинный список коротким, исследование позволяет предположить, что люди воспринимают

горизонтальные и вертикальные линии более быстро и правильно, чем наклонные линии или круги.

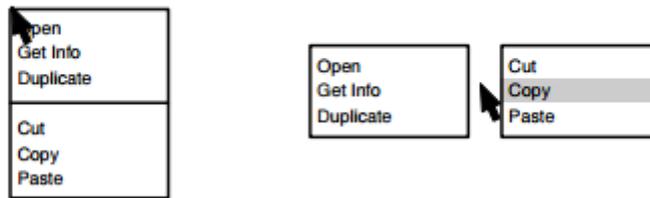


Рисунок 2.12 – Левое меню проще и понятнее пользователю

Еще одно преимущество этого вида меню является то, что лучшее использование горизонтального пространства позволяет отображать больше пунктов меню одновременно. В результате вы не должны скрывать пункты меню в тяжело используемых подменю.

Наконец, помещение горизонтальных элементов меню слева и справа от курсора только уменьшает расстояние, это помогает легкому попаданию на элементы, так как курсор перемещается влево или вправо, и таким образом направление движения курсора соответствует более длинному измерению целевой области.

На большие элементы легче попасть, иногда имеет смысл сделать деструктивные элементы меньше, чтобы уменьшить вероятность того, что пользователь нажмет на них непреднамеренно.

Например, в меню Windows 7 Start, кнопка “Shut down” меньше, чем интерактивные области, которые запускают приложения.

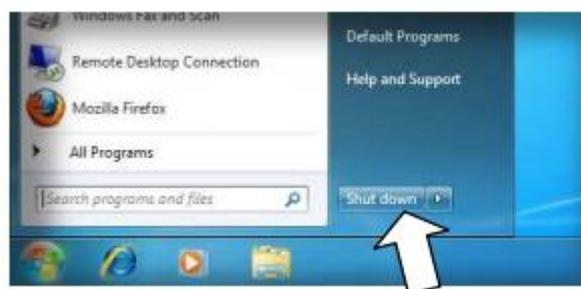


Рисунок 2.13 – Кнопка “Shut down” в Windows 7 Start

Закон Фиттса является такой моделью, которая помогает дизайнерам отыскать научно аргументированные решения для интерфейсов и макетов

интернет-страниц. Он может быть использован вместе с такими теориями, как теория визуального веса, при помощи которой определяется иерархия и место элементов в интерфейсе пользователя.

Закон Фиттса сфокусирован на математическом уравнении, которое вычисляет время, потраченное на достижение целевого объекта. Целевой объект для интерфейса пользователя – это любой интерактивный элемент, такой как гиперссылка, область ввода любой веб-формы. Идея такова: чем быстрее вы достигаете целевого объекта, тем удобнее и проще его использование.

Рассмотрим уравнение 2.3.

$$T = a + b * \log_2 \left( 2 * \frac{D}{W} \right), \quad (2.3)$$

где T (время)– количество времени, требуемое для совершения действия, а и b – определённые эмпирическим путём коэффициенты регрессии, которые указывают на иерархию, обычно a=50, b=150,

D (расстояние) – измерение дистанции от начальной до конечной точки (до целевого объекта),

W (ширина) – ширина целевого объекта.

Сначала вычислим количество времени, необходимое, чтобы выбрать в главном меню раздел «Документы».

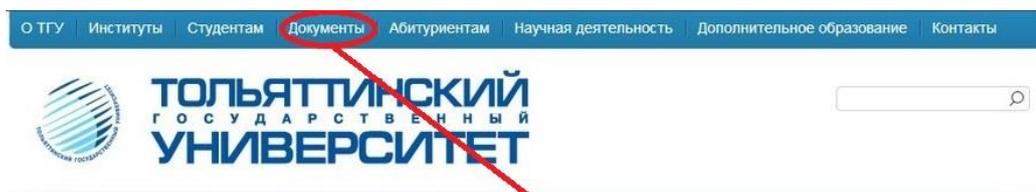


Рисунок 2.14 – Расстояние от начальной точки до пункта «Документы» в главном меню

Ширина пункта меню – 85 px. В качестве начальной точки будем использовать точку под логотипом университета, в середине линии. Расстояние от начальной точки до целевого объекта (пункта «Документы» в главном меню) – 170 px.

$$T_1 = 50 + 150 * \log_2 \left( \frac{170}{85} + 1 \right) = 201,585(\text{мс}). \quad (2.4)$$

Далее вычислим время, которое понадобится, чтобы достичь пункта «Сведения об образовательной организации» в разделе «Документы».

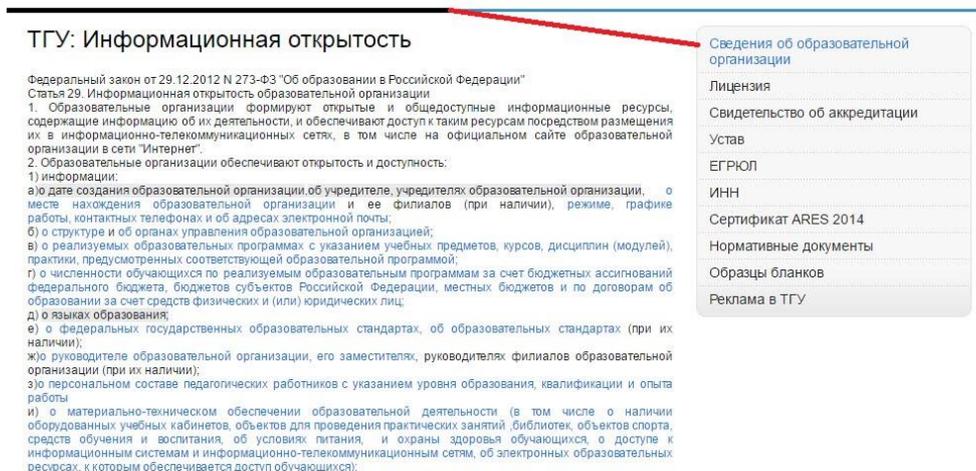


Рисунок 2.15 – Расстояние от начальной точки до целевого объекта (пункта меню)

Ширина пункта в меню – 210 px, расстояние от начальной точки до целевого объекта – 250 px.

$$T_2 = 50 + 150 * \log_2\left(\frac{250}{210} + 1\right) = 219,65(\text{мс}). \quad (2.5)$$

По формуле 2.6 вычислим общее время, которое нам понадобится, чтобы достигнуть страницы «Сведения об образовательной организации» - 2.7.

$$T = T_1 + T_2, \quad (2.6)$$

где  $T$  - общее время, которое нам понадобится, чтобы достигнуть страницы «Сведения об образовательной организации»,

$T_1$  - время, которое понадобится, чтобы достичь пункта «Документы»,

$T_2$  - время, которое понадобится, чтобы достичь пункта «Сведения об образовательной организации» в разделе «Документы».

$$T = 201,585 + 219,65 = 421,235(\text{мс}) \quad (2.7)$$

Заметим, что во внимание берётся только ширина целевого объекта, при двухмерном (плоском) средстве передачи, каким являются экраны мониторов, высота и глубина не принимаются во внимание. Поэтому, к примеру, объект 500 пикселей в ширину и всего 2 пикселя в высоту не будет казаться непригодным к применению целевым объектом, даже если по нему будет

сложно щёлкнуть, при размещении кнопок и продумывании их ширины и высоты, следует обращать внимание на этот факт.

Закон Фиттса действует одновременно только для одной координаты. Однако, при двухмерном (плоском) интерфейсе пользователя необходимо помнить, что и высота, и ширина могут облегчать использование объекта.

Одним из наиболее важных уроков, который можно извлечь из закона Фиттса является, что несмотря на то, что легче всего щёлкнуть по самой большой кнопке, это не обязательно будет самая важная кнопка. Именно из-за этого заблуждения во множестве интерфейсов пользователей кнопки подтверждения просто огромные.

Согласно модели закона Фиттса, большие элементы не всегда более подходящие, чем элементы поменьше

Закон Фиттса – это двоичный логарифм. Это значит, что для получения ожидаемого результата использования объекта нужно выбрать не прямой, а извилистый путь. В веб-дизайне это означает, что по очень маленькому объекту будет значительно легче щёлкать, если его увеличиться на 20%, но если увеличить на 20% очень большой объект, то от этого по нему не станет удобнее щёлкать.

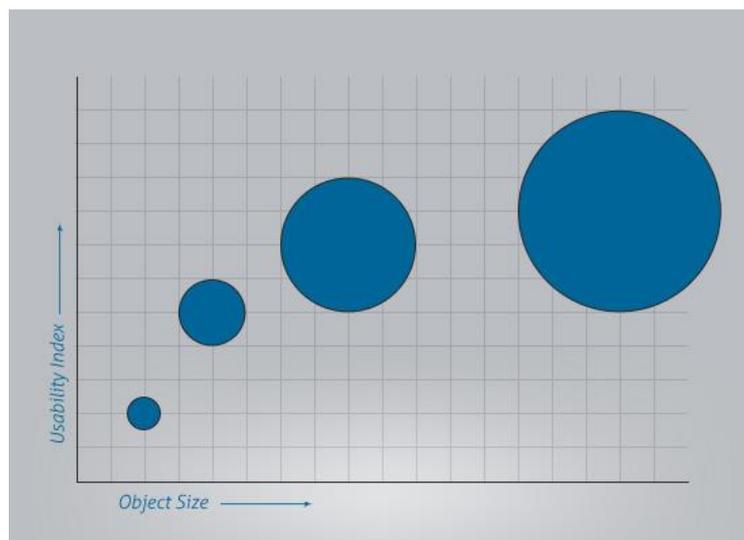


Рисунок 2.16 – Зависимость индекса удобства от размера объекта

Это значительное открытие для дизайнеров всех платформ. Эти нелинейные отношения берегут наши драгоценные пиксели. Наше умение

увеличить использование основано на нелинейном увеличении размера, что позволяет дизайнерам создавать эффективные и профессионально разработанные интерфейсы.

Помимо размера целевого объекта, другим основным фактором закона Фиттса является расстояние между тем, где курсор мыши находится в данный момент и тем, где ему нужно оказаться.

При расположении ключевых компонентов сайта далеко друг от друга, увеличивается время, требуемое на совершение последовательных задач в интерфейсе.

Известно, что расположение общих элементов интерфейса вместе – это стандартная практика в дизайне. Это облегчает поиск однородных элементов.

Кроме того, одинаково короткое расстояние между кнопками упростит их использование.

Совмещая и согласовывая модель закона Фиттса с другими концепциями, такими как Гештальт подходы, теории структуры визуальной иерархии, симметрии, поведения человека (Gestalt principles, power structure for visual hierarchy, symmetry, human behavior theories) и т.д. можно создать интерфейс пользователя, согласно разумным и научно доказанным дизайнерским решениям.

Ключом к оптимизации задач пользователя является знание того, где будет пользователь перед тем как направится куда-либо. К сожалению, для веб-дизайнеров, тут мы находимся в невыгодном положении.

Существует пиксель, который является самым важным из всего, что представлено на экране. Этот супер-пиксель правильно назван исходным пикселем (primerixel), являясь таковым, он обладает основной силой.

Исходный пиксель – это указатель, который используется одной точкой пространства, где на данный момент находится курсор вашего входящего устройства. Для каждого интерфейса компьютера расстояние до целевого объекта всегда будет измеряться исходя из того как далеко этот объект расположен от исходного пикселя.

Программные средства для настольных систем часто пользуются этой идеей. Например, в операционной системе Windows, когда вы щёлкаете правой кнопкой мыши, на месте исходного пикселя обычно появляется контекстное меню (как показано ниже).

Однако, у веб-сайтов нет возможности точно определять где находится исходный пиксель, поэтому веб-дизайнеры не могут воспользоваться этой идеей. Другими словами, расположение исходного пикселя изменяется, т.к. начальный пункт может быть где угодно. Даже если бы пользователь при помощи языка программирования JavaScript проследил за курсором мыши, он не захотел бы чтобы элементы интерфейса постоянно двигались в зависимости от месторасположения курсора мыши, т.к. это бы только запутало его.

Однако, обязательной является настройка активных элементов (таких как кнопки подтверждения и гиперссылки). При помощи языка программирования JavaScript мы можем услышать, что произошло, например, щелчок мышью или наведение курсора мыши на активный элемент. Так как интеракция (взаимодействие) в сети обусловлена вводом пользователем данных, только после щелчка мышью можно достаточно точно предсказать, где будет находиться курсор мыши при загрузке новой страницы.

Например, если пользователь перемещается по странице сайта посредством расположенной сверху горизонтальной панели управления, то мы знаем, что исходный пиксель для следующей страницы будет расположен около элементов панели управления. Принимая во внимание, что после щелчка по ссылке на панели управления, мышь не будет двигаться, мы можем предположить, что последний щелчок или последнее наведение курсора мыши будет исходным пикселем на следующем экране. Это во многом помогает в определении последовательности действий пользователя.

Такая логика может быть наиболее эффективной при многошаговых онлайн процессах, таких как регистрация или заполнение веб-форм. Веб-дизайнер всегда должен принимать во внимание расстояние кнопки на следующем этапе процесса заполнения от полей ввода предыдущего этапа, а

также то, где находится курсор пользователя во время загрузки следующего экрана.

Хотя исходный пиксель является самым важным местом на экране компьютера, он всего лишь один из пяти «волшебных пикселей», которые принимаются во внимание при создании интерфейса.

Четыре других волшебных пикселя применяются нашим экраном в каждом из четырёх его углов.

Веб-дизайнеры и тут находятся в невыгодном положении, так как у нас такое местонахождение волшебных пикселей меняет их функции на противоположные.

Согласно закону Фиттса, стороны и углы экрана играют удивительную роль, так как они создают границы, которых не существует в реальном мире.

В сущности, в схеме с большим количеством величин этим обеспечивается безразмерная форма. Например, когда пользователю не нужно останавливаться в правом верхнем углу, закрытие окна посредством клика по кнопке становится очень простым заданием.

Однако, если бы опция закрыть окно была бы расположена в правом верхнем углу экрана, но не включала бы ближайšie к краю пиксели, использование стало бы намного неудобнее.

У веб-сайтов нет возможности в полном объёме использовать углы и поэтому те углы, которые нам доступны из ценного приобретения превращаются в ненужные.

Так как края экрана на веб-сайте не создают неограниченный размер, его углы являются самыми дальними точками для исходного пикселя.

Итак, как же веб-дизайнеру в своей работе преодолеть отсутствие содействия волшебных пикселей?

Углы веб-сайта неожиданно становятся неподходящим местом для таких важных кнопок как, например, кнопки входа в систему. Как веб-дизайнеру бороться с этим?

Превосходным методом на сегодняшний день является метод перемещения от краёв и углов в центр экрана.

Если посчитать расстояние от углов веб-сайта до исходного пикселя и сравнить его со средним расстоянием до середины экрана, можно обнаружить, что в центре есть явное преимущество.

Выходит, веб-дизайн естественным образом использует ориентацию краёв и углов на центр.

На практике видна целесообразность данного подхода: большинство людей, которые используют широкоэкранные мониторы, могут рассказать, что они предпочитают веб-сайты, у которых активные панели находятся посередине браузера, тем сайтам, где всё сосредоточено справа или слева экрана.

Когда активная панель сайта расположена в центре экрана, пользователь может совмещать действия, производимые на вашем сайте, с действиями, которые предлагает его браузер и операционная система, что является наиболее эффективным.

Важно располагать похожие элементы интерфейса близко друг от друга, что, в соответствии с законом Фиттса, что облегчит их использование. Гештальт закон близости, который утверждает, что ближайšie друг к другу элементы будут восприниматься сгруппированными вместе, позже подтвердил эту идею.

Например, панель управления должна создавать единый рисунок, где одна кнопка неразрывно граничит с другой. На рисунке 2.17 элементы Home и Blog слева вместе занимают столько же пространства, как и такие же элементы справа, но образовавшееся в первом случае расстояние усложняет, согласно закону Фиттса, использование меню, так как увеличивается расстояние от начальной точки до целевого объекта.

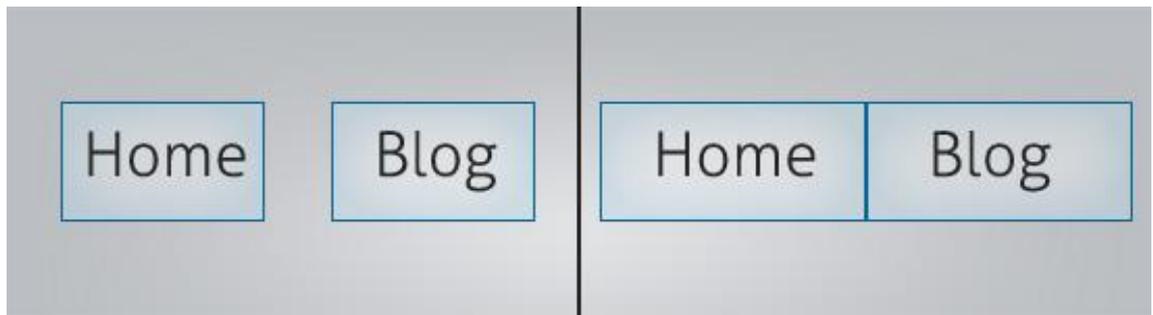


Рисунок 2.17 – Пример панели управления

Закон Фиттса – это модель, которая должна использоваться каждым дизайнером, независимо от того работает он в виртуальном или в реальном пространстве. Основная логика закона Фиттса, состоящая в том, что целенаправленные действия лучше осуществлять в специально созданном для этого пространстве, не может быть оставлена без внимания.

Удивительной стороной теории дизайна является то, что дизайнер может исследовать эту науку столько сколько ему угодно и каждый раз он будет находить что-нибудь новое для себя. Дизайнер, использующий себе во благо закон Фиттса, вскоре найдёт много ценной информации, которая может послужить хорошим толчком для успешного развития его веб-сайта.

## 2.5 Закон Хика

Закон Хика – закон психологии, применимый в проектировании интерфейсов.

Хик (Nick) Вильям Эдуард (1912 – 1975) - английский психотерапевт и специалист по психологии, врач магистр гуманитарных наук (1954), доктор медицины (Дарем, 1949), член Британской академии психологических наук и Совета по медицинским исследованиям. Работал в Кембриджском университете по проблемам экспериментальной психологии. Позднее перешел к частной практике. В 1952 г. вместе с Р. Хайманом вывел закон времени реакции («закон Хика»): время реакции выбора из альтернативных действий на равновероятные раздражители логарифмически связано с количеством альтернатив. В данном исследовании были применены результаты исследования немецкого психолога И. Меркеля, полученные еще в 1885 г.

Закон Хика показывает, что если нужно сделать выбор из некоторого числа( $n$ ) вариантов, то время на выбор одного из них будет пропорционально логарифму по основанию 2 от числа вариантов плюс 1, если все варианты равновероятны [21].

$$T = a + b * \log_2(n + 1) , \quad (2.8)$$

где  $T$  – среднее время выбора одного из вариантов,

$a, b$  – коэффициенты пропорциональности,

$n$  – число равновероятных вариантов.

Вычислим время, которое понадобится, чтобы достичь страницы «Сведения об образовательной организации». Для этого сначала вычислим время, требуемое для выбора раздела «Документы» в главном меню, затем время выбора страницы «Сведения об образовательной организации».

$$T_1 = a + 3,17 * b, \quad (2.9)$$

где  $T_1$ – время, требуемое для выбора раздела «Документы» в главном меню,

$a, b$  – коэффициенты пропорциональности.

$$T_2 = a + 3,459 * b, \quad (2.10)$$

где  $T_2$ – время, которое потребуется для выбора страницы «Сведения об образовательной организации» в меню раздела «Сведения об образовательной организации»,

$a, b$  – коэффициенты пропорциональности.

$$T = a + 6,629 * b, \quad (2.11)$$

где  $T$  – общее время, которое понадобится, чтобы достичь страницы «Сведения об образовательной организации»,

$a, b$  – коэффициенты пропорциональности.

На основании закона Хика, можно понять, что чем меньше элементов в меню, тем проще сделать выбор, а также, что одно меню лучше, чем два. Используя данный закон, несложно понять, что расположение расписания на сайте ТГУ требует изменения.

## 2.6 Проектирование информационной структуры сайта ТГУ

Следуя законам Фиттса и Хика, можно сделать выводы, что чем меньше элементов в меню и чем ближе схожие элементы друг к другу, тем удобнее меню для пользователя. Используя данные законы, был составлен прототип меню, изображенный на рисунке 2.18. Также в главном меню сайта есть раздел «О ТГУ», целесообразно будет его добавить в раздел «Сведения об образовательной организации», так как они схожи по наполнению [4].



Рисунок 2.18 – Прототип интерфейса оптимизированного раздела

Используя законы Фиттса и Хика, вычислим время, которое нам потребуется, чтобы достигнуть этого пункта в меню.

$$T = 50 + 150 * \log_2\left(\frac{185}{310} + 1\right) = 151,25(\text{мс}), \quad (2.12)$$

$$T = a + b * \log_2(7) = a + 2,807 * b. \quad (2.13)$$

Вычисленное время значительно меньше времени, затраченного для достижения данной страницы до оптимизации.

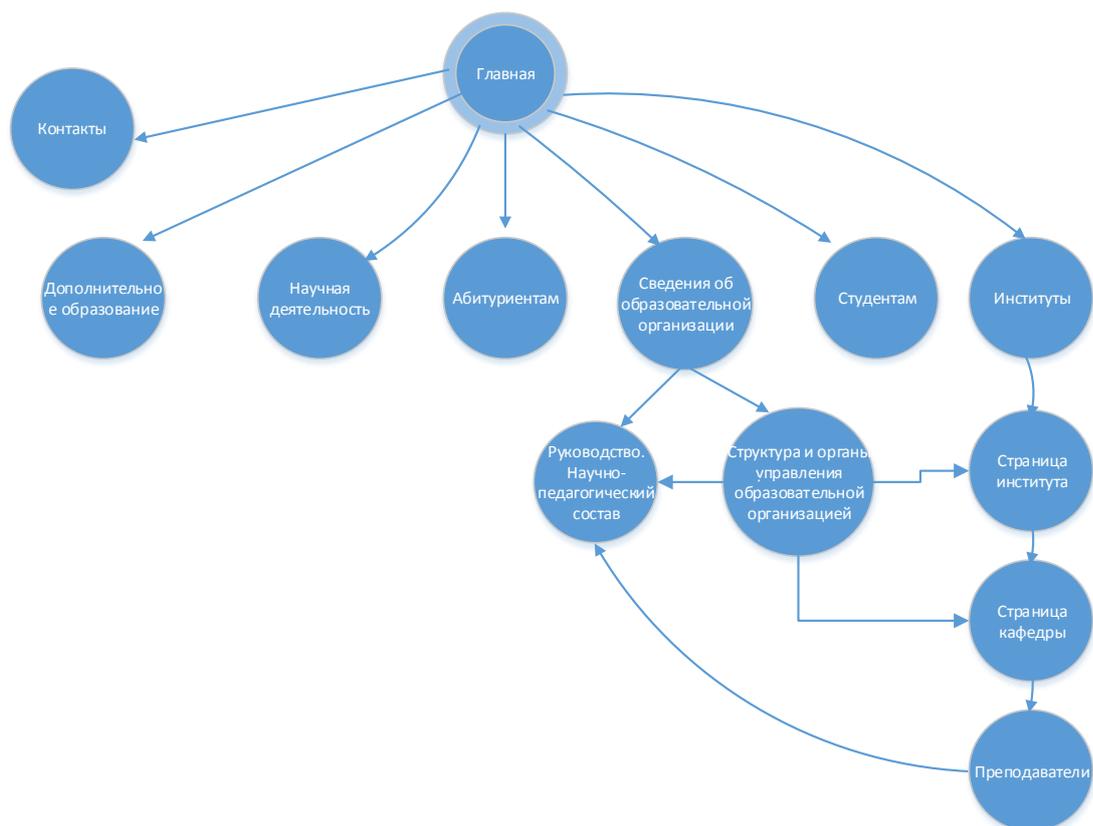


Рисунок 2.19 – Граф оптимизированной структуры сайта

Данная схема демонстрирует новую структуру сайта ТГУ. На ней можно увидеть, что новые страницы связаны между собой. Есть страница каждого преподавателя с каждой кафедры и при просмотре данных преподавателя можно посмотреть страницу кафедры или института и наоборот. При просмотре структуры Тольяттинского государственного университета, можно посмотреть страницы подразделений и руководства этих подразделений, также ознакомиться с положением о создании данного структурного подразделения.

Такие изменения делают навигацию по сайту более удобной для пользователя и сам сайт более информативным [5].

## **Глава 3 Разработка новой информационной структуры сайта Тольяттинского государственного университета**

### **3.1 CMS «1С Битрикс: Управление сайтом»**

Сайт Тольяттинского государственного университета разработан с помощью CMS «1С Битрикс: Управление сайтом». CMS (Content management system) или система управления сайтом – система, используемая для управления содержимым сайта. Основная функция данной системы состоит в автоматизации управления содержимым сайта [13].

CMS «1С Битрикс: Управление сайтом» – одна из самых популярных платных тиражных систем управления сайтом в России. Разработка в данной системе ведется с помощью таких языков программирования как php, javascript, html, css [15].

PHP - это скриптовый язык, а также мощный инструмент для создания динамических и интерактивных веб-страниц. PHP является широко используемым, открытым языком программирования и эффективной альтернативой для конкурентов, таких как ASP от Microsoft [11].

JavaScript является языком программирования, позволяющим сделать веб-страницу интерактивной. Его применяют, чтобы спрограммировать поведение веб-страниц [12].

HTML является стандартным языком разметки, используемым для создания веб-страниц, а его элементы образуют составляющие блоки всех сайтов [14].

CSS (каскадные таблицы стилей) это язык стилей, который описывает представление документа в формате HTML (или XML) [20]. CSS описывает, как элементы должны визуализироваться на экране пользователя [19]. Преимущественно css применяют для оформления внешнего вида веб-страниц, написанных с помощью HTMLи XHTML [15], но также может применяться и к XML-документам [22].

Однако, работать на данной платформе достаточно просто даже без знания языков гипертекстовой разметки и других языков программирования.

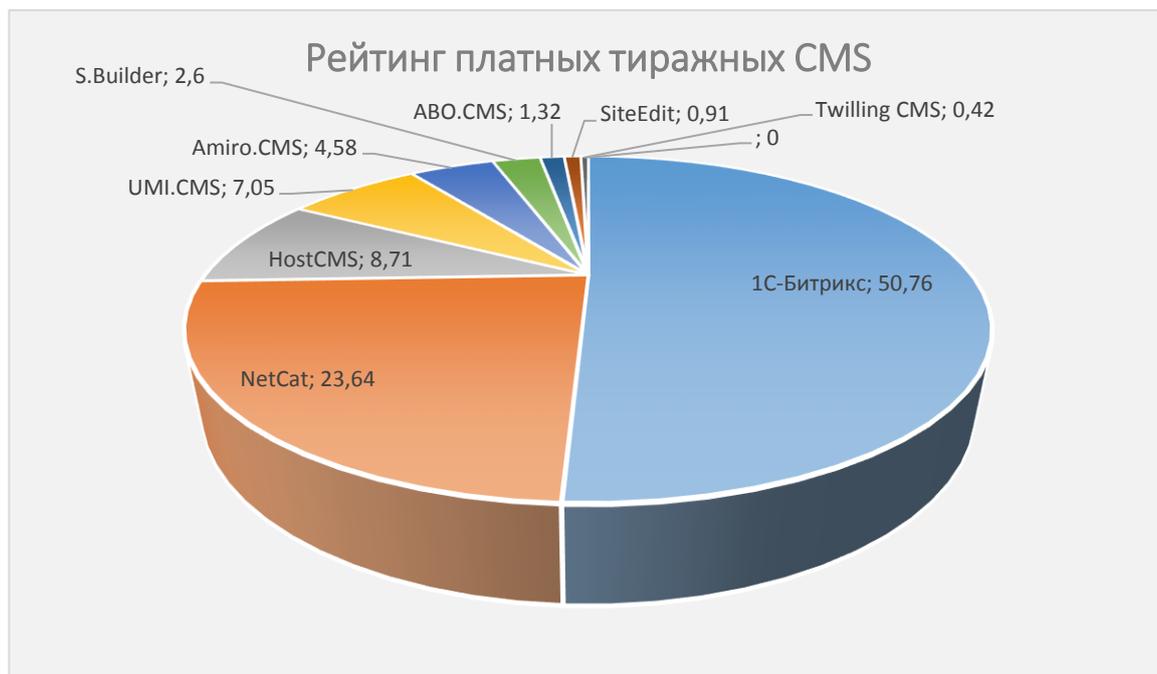


Рисунок 3.1 – Рейтинг платных тиражных CMS

Главные достоинства системы:

- встроенный полноценный визуальный редактор, позволяющий изменять содержимое страниц, вставлять объекты и текст из MSWord и Excel. Данный редактор поддерживает 3 режима работы – режим визуального редактирования, режим редактирования исходного кода и совмещенный режим редактирования;
- инструменты для разработки индивидуальных решений. Платформа «1С Битрикс: Управление сайтом» позволяет полностью изменять оболочку сайта, в результате чего можно получить абсолютно уникальный сайт;
- магазин, в котором располагаются всевозможные платные и бесплатные модули.

Однако данная платформа имеет ряд недостатков:

- стандартные компоненты не предназначены для изменений (их изменение требует дополнительных знаний);
- система платная;

- в сравнении с другими CMS, Битрикс довольно медленная и требовательная к ресурсам система.

В CMS «1С Битрикс: Управление сайтом» используются PHP, HTML и JavaScript.

### **3.2 Инфоблоки в CMS «1С Битрикс: Управление сайтом»**

Так как необходимая информация однотипна, то вывод информации на экран целесообразно сделать с помощью инфоблоков. Информационными блоками (инфоблоками) называют модуль, который позволяет каталогизировать и управлять разнообразными типами (блоками) однотипной информации. Инфоблоки предоставляют возможность реализовать публикацию различных типов динамической информации: каталоги товаров, блоки новостей, справочники и т.д.

Информационные блоки – главный элемент Bitrix Framework. Практически всё, что делается в системе в той или иной мере завязано на этот модуль, даже если это и не отображается явно.

Информационные блоки представляют собой очередной уровень абстракции над обычными таблицами СУБД, своеобразная "база данных в базе данных". Поэтому к ним отчасти применимы все те правила, которых придерживаются при проектировании БД.

Инфоблоки – сущность, которая в физической структуре БД создает 4 таблицы, не меняющиеся при изменении структуры данных: типы объектов, экземпляры объектов, свойства объектов и значения свойств объектов.

Плюсы такого подхода:

- удобный контроль над данными такой структуры из своего приложения;
- универсальность методов;
- общая структура данных для любого проекта;
- возможность многократно менять типы данных для полей без удаления самих данных.

Минусы такого подхода:

- повышенные требования к производительности;
- непрозрачность при прямом доступе к данным.

Упорядочивание элементов инфоблоков по разделам может быть очень удобным для навигации по инфоблоку в административном интерфейсе. Фасетное упорядочивание делает навигацию ещё более удобной.

Также инфоблок с разделами уже сам в себе содержит описание того, каким образом следует его представлять пользователю. То есть достаточно обойти его самым обычным алгоритмом обхода дерева, чтобы вывести его в максимально удобном виде.

Особенности работы заключаются в некотором неудобстве работать отдельно именно с разделами. Так, например, если помимо инфоблока Статьи существует инфоблок Книги, то велика вероятность, что его элементы также будут нуждаться в классификации по дате публикации и по тематикам. В этом случае придётся ещё раз создавать такую же структуру разделов. Также весьма непросто будет вывести, например, список всех материалов (статей и книг) по одной тематике, упорядочив их по дате публикации. Также сложно будет вывести общий рубрикатор тематик в меню сайта.

В таком случае следует завести отдельный инфоблок Тематики и добавить в инфоблоки Книги и Статьи свойство ссылочного типа Тематика, а также свойство Дата Публикации типа Дата. Навигацию в административном интерфейсе тогда будет удобнее осуществлять при помощи установки фильтров по этим свойствам.

### **3.3 Комплексный компонент «Новости»**

В «1С Битрикс: Управление сайтом» компоненты разделяются на простые и комплексные. Простые (обычные, одностраничные) компоненты создают какую-либо область на одной странице. Их удобно применять, когда на одной странице требуется разместить контент из различных модулей (например, блоги и инфоблоки) или данные из разных инфоблоков (новости и каталог

товаров). Для создания полного раздела новостей либо каталога товаров пользоваться ими достаточно неудобно: приходится создавать огромное количество статических страниц и следить за тем, чтобы они были корректно связаны друг с другом.

Комплексные (сложные, многостраничные) компоненты создают разделы веб-сайта. К примеру, компонент каталога создает на сайте весь раздел каталога: список каталогов, список групп и страницы товаров. То есть, комплексный компонент состоит из набора динамических страниц при просмотре сайта, но из одной статической страницы на физическом уровне. Сложные компоненты строятся на базе простых компонентов, применяя их логику. Комплексные компоненты разрешают следующие проблемы:

- отпадает необходимость создания большого количества статических страниц для размещения всех необходимых подкомпонентов. Отпадает необходимость отдельно настраивать для каждого из подкомпонентов общие (пересекающиеся) свойства (к примеру, тип инфоблока и инфоблок);
- происходит установление сложных связей между подкомпонентами. Например, нет необходимости для страницы с перечнем сообщений темы форума настраивать, как данная страница может указать на страницу списка тем форума, а как на страницу профиля посетителя;
- в компонент (даже с настроенными шаблонами вывода) можно добавить новую страницу. Например, если на форуме появится страница (подкомпонент) по выводу 10 посетителей с самым высоким рейтингом, то эта страница станет доступной и в публичной части;
- можно сменить шаблон вывода всего комплексного компонента одним действием, а не настраивать вывод каждого из подкомпонентов.

Алгоритм работы комплексного компонента:

- на основании действий посетителя интернет-сайта (например, переход по пунктам меню) комплексный компонент определяет, какая страница должна быть показана пользователю, и подключает свой шаблон компонента для данной динамической страницы;

- шаблон страницы подключает простые компоненты, автоматически настраивая необходимым образом их свойства;
- простые компоненты выполняют свою работу: запрашивают данные у ядра системы, форматируют их и выводят посетителю, также предоставляя пользователю различные элементы управления (гиперссылки, формы, кнопки и другие);
- пользователь с помощью каких-либо элементов управления, отправляет новый запрос комплексному компоненту.

Рассмотрим упрощённый пример работы комплексного компонента новостей. Пусть у нас имеются обычные компоненты: список новостей и детальной новости (последний принимает во входных параметрах код новости, которую нужно показать).

Раздел новостей можно организовать, к примеру, разместив на главной странице компонент списка новостей, а на странице новостей – компонент детальной новости. При этом у компонента списка новостей необходимо настроить входные параметры так, чтобы он мог формировать гиперссылки на страницу детальной новости (с кодом новости), а у компонента детальной новости нужно настроить входные параметры так, чтобы он имел возможность сформировать ссылку на страницу списка новостей.

Чтобы задать ссылку на страницу детальной новости, нужно задать путь к этой странице, а также название параметра, в котором будет передаваться код новости для показа. То же название параметра нужно задать и во входных параметрах компонента детальной новости, чтобы он знал, где брать код новости для показа. Даже в данном максимально упрощённом случае настройки не элементарны. А если это набор из десятков компонентов форума?

Наиболее удобной альтернативой для сборщика сайта будет использование комплексного компонента новостей. Данный компонент, например, можно просто установить на главную страницу сайт. Согласованием ссылок и параметров будет заниматься сам комплексный компонент. От разработчика сайта никаких дополнительных действий не потребуется.

Страницы шаблона комплексного компонента будут содержать подключение соответствующих обычных компонентов с правильной настройкой их входных параметров. Обычные компоненты будут выполнять свои обычные функции: им все равно, кто их вызвал и зачем. Для обычных компонентов важна только правильная настройка их входных параметров.

Таким образом реализуется паттерн MVC (модель-вид-контроллер – схема применения нескольких шаблонов проектирования, с помощью которых модель приложения (модель), взаимодействие с пользователем(контроллер) и пользовательский интерфейс (вид) разделены на 3 отдельных составляющих таким образом, что модификация одного из компонентов оказывает минимальное воздействие на другие):

- на комплексный компонент новостей (контроллер) приходит HTTP запрос (действия пользователя);
- комплексный компонент новостей (контроллер) проверяет, установлен ли через HTTP запрос код новости и подключает из своего шаблона страницу списка новостей или страницу детальной новости (вид или представление);
- подключенная страница, в свою очередь, подключает соответствующий простой компонент, устанавливая при этом его входные параметры соответствующим образом;
- простой компонент выполняет свою работу: запрашивает данные у ядра (модель), форматирует их и показывает посетителю, а также отображает элементы управления (гиперссылки, формы, кнопки и прочие);
- пользователь с помощью элементов управления посылает новый HTTP запрос на комплексный компонент новостей (контроллер);
- процедура повторяется при необходимости.

Таким образом можно сделать вывод, что комплексные компоненты более удобны в использовании. Комплексный компонент новости содержит 3 шаблона: default, web20 и flat. Компонент стандартный и входит в дистрибутив модуля.

На рисунке 3.2 приведен пример стандартного вызова компонента «Новости».

```

<?APPLICATION->IncludeComponent("bitrix:news","",Array(
    "DISPLAY_DATE" => "Y",
    "DISPLAY_PICTURE" => "Y",
    "DISPLAY_PREVIEW_TEXT" => "Y",
    "SEF_MODE" => "Y",
    "AJAX_MODE" => "Y",
    "IBLOCK_TYPE" => "news",
    "IBLOCK_ID" => "1",
    "NEWS_COUNT" => "20",
    "USE_SEARCH" => "Y",
    "USE_RSS" => "Y",
    "USE_RATING" => "Y",
    "USE_CATEGORIES" => "Y",
    "USE_REVIEW" => "Y",
    "USE_FILTER" => "Y",
    "SORT_BY1" => "ACTIVE_FROM",
    "SORT_ORDER1" => "DESC",
    "SORT_BY2" => "SORT",
    "SORT_ORDER2" => "ASC",
    "CHECK_DATES" => "Y",
    "PREVIEW_TRUNCATE_LEN" => "",
    "LIST_ACTIVE_DATE_FORMAT" => "d.m.Y",
    "LIST_FIELD_CODE" => Array(),
    "LIST_PROPERTY_CODE" => Array(),
    "HIDE_LINK_WHEN_NO_DETAIL" => "Y",
    "DISPLAY_NAME" => "Y",
    "META_KEYWORDS" => "-",
    "META_DESCRIPTION" => "-",
    "BROWSER_TITLE" => "-",
    "DETAIL_SET_CANONICAL_URL" => "Y",
    "DETAIL_ACTIVE_DATE_FORMAT" => "d.m.Y",
    "DETAIL_FIELD_CODE" => Array(),
    "DETAIL_PROPERTY_CODE" => Array(),
    "DETAIL_DISPLAY_TOP_PAGER" => "Y",
    "DETAIL_DISPLAY_BOTTOM_PAGER" => "Y",
    "DETAIL_PAGER_TITLE" => "Страница",
    "DETAIL_PAGER_TEMPLATE" => "",
    "DETAIL_PAGER_SHOW_ALL" => "Y",
    "SET_TITLE" => "Y",
    "ADD_SECTIONS_CHAIN" => "Y",
    "ADD_ELEMENT_CHAIN" => "N",
    "SET_LAST_MODIFIED" => "Y",
    "PAGER_BASE_LINK_ENABLE" => "Y",
    "SET_STATUS_404" => "Y",
    "SHOW_404" => "Y",
    "MESSAGE_404" => "",
    "PAGER_BASE_LINK" => "",
    "PAGER_PARAMS_NAME" => "arrPager",
    "INCLUDE_IBLOCK_INTO_CHAIN" => "Y",
    "USE_PERMISSIONS" => "Y",
    "GROUP_PERMISSIONS" => Array("1"),
    "CACHE_TYPE" => "A",
    "CACHE_TIME" => "3600",
    "CACHE_FILTER" => "Y",
    "CACHE_GROUPS" => "Y",
    "DISPLAY_TOP_PAGER" => "Y",
    "DISPLAY_BOTTOM_PAGER" => "Y",
    "PAGER_TITLE" => "Новости",
    "PAGER_SHOW_ALWAYS" => "Y",
    "PAGER_TEMPLATE" => "",
    "PAGER_DESC_NUMBERING" => "N",
    "PAGER_DESC_NUMBERING_CACHE_TIME" => "36000",
    "PAGER_SHOW_ALL" => "Y",
    "FILTER_NAME" => "",
    "FILTER_FIELD_CODE" => Array(),
    "FILTER_PROPERTY_CODE" => Array(),
    "NUM_NEWS" => "20",
    "NUM_DAYS" => "30",
    "YANDEX" => "Y",
    "MAX_VOTE" => "5",
    "VOTE_NAMES" => Array("0", "1", "2", "3", "4"),
    "CATEGORY_IBLOCK" => Array(),
    "CATEGORY_CODE" => "CATEGORY",
    "CATEGORY_ITEMS_COUNT" => "5",
    "MESSAGES_PER_PAGE" => "10",
    "USE_CAPTCHA" => "Y",
    "REVIEW_AJAX_POST" => "Y",
    "PATH_TO_SMILE" => "/bitrix/images/forum/smile/",
    "FORUM_ID" => "1",
    "URL_TEMPLATES_READ" => "",
    "SHOW_LINK_TO_FORUM" => "Y",
    "POST_FIRST_MESSAGE" => "Y",
    "SEF_FOLDER" => "/",
    "SEF_URL_TEMPLATES" => Array(
        "detail" => "#ELEMENT_ID#/",
        "news" => "search/",
        "section" => "rss/"
    ),
    "AJAX_OPTION_JUMP" => "N",
    "AJAX_OPTION_STYLE" => "Y",
    "AJAX_OPTION_HISTORY" => "N",
    "VARIABLE_ALIASES" => Array(
        "detail" => Array(),
        "news" => Array(),
        "section" => Array(),
    ),
    "USE_SHARE" => "Y",
    "SHARE_HIDE" => "Y",
    "SHARE_TEMPLATE" => "",
    "SHARE_HANDLERS" => array("delicious", "facebook", "lj", "twitter"),
    "SHARE_SHORTEN_URL_LOGIN" => "",
    "SHARE_SHORTEN_URL_KEY" => ""
)
):?>

```

Рисунок 3.2 – Стандартный вызов компонента «Новости»

### 3.4 Основной шаблон страниц

Стандартный шаблон позволяет вывести на экран анонс новости, её подробное описание и свойства в формате: название свойства – значение свойства.

Стандартный вызов компонента открывает пользователю детальную информацию по клику на название инфоблока. Так как у нас нет детальной информации, а есть лишь свойства, которые в списке новостей и при детальном просмотре идентичны, целесообразно будет скрывать ссылку. Для этого при вызове компонента поставим "HIDE\_LINK\_WHEN\_NO\_DETAIL" => "Y".

Ниже приведен код шаблона для вывода общей информации, в нем мы выводим на экран только название инфоблока и значения свойств. Так же этот шаблон используется для страниц «Документы», «Образование», «Образовательные стандарты», «Материально-техническое обеспечение и оснащенность образовательного процесса», «Стипендии и иные виды материальной поддержки», «Платные образовательные услуги» и «Финансово-хозяйственная деятельность». Для каждой страницы создадим инфоблок, зададим названия и значения свойств его элементов, затем добавим необходимую информацию. Для шаблона были заданы индивидуальные css-свойства [17] [18]. Шаблон универсален и применим ко всем комплексным компонентам новости [6].

Часть основного шаблона страниц приведена на рисунке 3.3.

```

<? if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED !== true) die(); ?>
<div class="news-list">
  <? if ($arParams["DISPLAY_TOP_PAGER"]): ?>
    <?= $arResult["NAV_STRING"] ?><br/>
  <? endif; ?>
  <? foreach ($arResult["ITEMS"] as $arResultItem): ?>
    <?
    $this->AddEditAction($arResultItem['ID'], $arResultItem['EDIT_LINK'], $arResultItem['IBLOCK_ID'], "ELEMENT_EDIT");
    $this->AddDeleteAction($arResultItem['ID'], $arResultItem['DELETE_LINK'],
    $arResultItem['IBLOCK_ID'], "ELEMENT_DELETE",
    array("CONFIRM" => GetMessage('CT_BNL_ELEMENT_DELETE_CONFIRM')));
    ?>
    <p class="news-item" id="<?= $this->GetEditAreaId($arResultItem['ID']); ?>">
      <? if ($arParams["DISPLAY_PICTURE"] != "N" && is_array($arResultItem["PREVIEW_PICTURE"]): ?>
        <? if (!($arParams["HIDE_LINK_WHEN_NO_DETAIL"] || ($arResultItem["DETAIL_TEXT"] && $arResult["USER_HAVE_ACCESS"])): ?>
          <a href="<?= $arResultItem["DETAIL_PAGE_URL"] ?>">"
            width="<?= $arResultItem["PREVIEW_PICTURE"]["WIDTH"] ?>"
            height="<?= $arResultItem["PREVIEW_PICTURE"]["HEIGHT"] ?>"
            alt="<?= $arResultItem["NAME"] ?>"
            title="<?= $arResultItem["NAME"] ?>" style="..."/></a>
          <? else; ?>
            "
            width="<?= $arResultItem["PREVIEW_PICTURE"]["WIDTH"] ?>"
            height="<?= $arResultItem["PREVIEW_PICTURE"]["HEIGHT"] ?>" alt="<?= $arResultItem["NAME"] ?>"
            title="<?= $arResultItem["NAME"] ?>" style="..."/>
          <? endif; ?>
        <? endif; ?>
        <? if ($arParams["DISPLAY_DATE"] != "N" && $arResultItem["DISPLAY_ACTIVE_FROM"]): ?>
          <span class="news-date-time"><? echo $arResultItem["DISPLAY_ACTIVE_FROM"] ?></span>
        <? endif; ?>
        <? if ($arParams["DISPLAY_NAME"] != "N" && $arResultItem["NAME"]): ?>
          <? if (!($arParams["HIDE_LINK_WHEN_NO_DETAIL"] || ($arResultItem["DETAIL_TEXT"] && $arResult["USER_HAVE_ACCESS"])): ?>
            <a href="<?= $arResultItem["DETAIL_PAGE_URL"] ?>"><b><? echo $arResultItem["NAME"] ?></b></a><br/>
          <? else; ?>
            <b><? echo $arResultItem["NAME"] ?></b><br/>
          <? endif; ?>
        <? endif; ?>
        <? if ($arParams["DISPLAY_PREVIEW_TEXT"] != "N" && $arResultItem["PREVIEW_TEXT"]): ?>
          <? echo $arResultItem["PREVIEW_TEXT"]; ?>
        <? endif; ?>
        <? if ($arParams["DISPLAY_PICTURE"] != "N" && is_array($arResultItem["PREVIEW_PICTURE"]): ?>
          <div style="clear:both"></div>
        <? endif; ?>
        <? foreach ($arResultItem["FIELDS"] as $code => $value): ?>
          <small>
            <?=> GetMessage("IBLOCK_FIELD_" . $code) ?>:&nbsp;&nbsp;<?= $value; ?>
          </small><br/>
        <? endforeach; ?>
        <? foreach ($arResultItem["DISPLAY_PROPERTIES"] as $pid => $arResultProperty): ?>
          <small>
            <?=> $arResultProperty["NAME"] ?>:&nbsp;&nbsp;&nbsp;
            <? if (is_array($arResultProperty["DISPLAY_VALUE"]): ?>
              <?=> implode("&nbsp;&nbsp;&nbsp;/&nbsp;&nbsp;&nbsp;", $arResultProperty["DISPLAY_VALUE"]); ?>
            <? else; ?>
              <?=> $arResultProperty["DISPLAY_VALUE"]; ?>
            <? endif; ?>
          </small>
        <? endforeach; ?>
      </p>
    <? foreach ($arResultItem["DISPLAY_PROPERTIES"] as $pid => $arResultProperty): ?>
        <small>
          <?=> $arResultProperty["NAME"] ?>:&nbsp;&nbsp;&nbsp;
          <? if (is_array($arResultProperty["DISPLAY_VALUE"]): ?>
            <?=> implode("&nbsp;&nbsp;&nbsp;/&nbsp;&nbsp;&nbsp;", $arResultProperty["DISPLAY_VALUE"]); ?>
          <? else; ?>
            <?=> $arResultProperty["DISPLAY_VALUE"]; ?>
          <? endif; ?>
        </small>
      <? endforeach; ?>
    </div>
  <? endforeach; ?>

```

Рисунок 3.3 – Основной шаблон страниц

### 3.5 Разработка шаблона страницы «Структура и органы управления образовательной организацией»

Для организации структуры данного раздела создадим тип инфоблока «Структура учебного заведения», и в нем создадим два инфоблока – «Учебные подразделения» и «Структурные подразделения». В каждом инфоблоке создадим разделы структурных подразделения и в них, в качестве элементов, добавим подразделения, находящиеся в их подчинении. Например, раздел – институт, элементы – кафедры. В свойства этих элементов добавим всю необходимую информацию: отделы в подчинении или у кого в подчинении находится отдел, сайт, положение о подразделении, информацию о

руководителе, контакты. Этот инфоблок будет связан с инфоблоком научно-педагогического состава по свойству «руководитель».

Далее в папке «structure» создадим файлы, именами которых зададим коды соответствующих разделов.

Для отображения списка структурных подразделений будем использовать компонент «Структура разделов». Компонент выводит список разделов инфоблока. Компонент стандартный, входит в дистрибутив модуля и содержит 4 шаблона: .default, old\_version (шаблон .default до версии 14.0), tree и visual.

В визуальном редакторе компонент расположен по пути Контент > Каталог > Структура разделов.

Стандартный вызов компонента приведен на рисунке 3.4.

```
<?APPLICATION->IncludeComponent("bitrix:catalog.section.list", "",
Array(
    "VIEW_MODE" => "TEXT",
    "SHOW_PARENT_NAME" => "Y",
    "IBLOCK_TYPE" => "",
    "IBLOCK_ID" => "",
    "SECTION_ID" => $_REQUEST["SECTION_ID"],
    "SECTION_CODE" => "",
    "SECTION_URL" => "",
    "COUNT_ELEMENTS" => "Y",
    "TOP_DEPTH" => "2",
    "SECTION_FIELDS" => "",
    "SECTION_USER_FIELDS" => "",
    "ADD_SECTIONS_CHAIN" => "Y",
    "CACHE_TYPE" => "A",
    "CACHE_TIME" => "36000000",
    "CACHE_NOTES" => "",
    "CACHE_GROUPS" => "Y"
)
):?>
```

Рисунок 3.4 – Стандартный вызов компонента «Структура разделов»

При настройке компонента в разделе «Шаблоны ссылок» введем #SECTION\_CODE#.php как url, ведущий на страницу раздела, настроенный вызов компонента изображен на рисунке 3.5. Соответственно код раздела и названия файлов должны быть идентичными.

```

<p style="line-height: 15.9091px;"><?APPLICATION->IncludeComponent(
    "bitrix:catalog.section.list",
    "old_version",
    Array(
        "IBLOCK_TYPE" => "structure",
        "IBLOCK_ID" => "118",
        "SECTION_ID" => $_REQUEST["SECTION_ID"],
        "SECTION_CODE" => "",
        "SECTION_URL" => "#SECTION_CODE#.php",
        "COUNT_ELEMENTS" => "N",
        "TOP_DEPTH" => "1",
        "SECTION_FIELDS" => array(),
        "SECTION_USER_FIELDS" => array(),
        "ADD_SECTIONS_CHAIN" => "Y",
        "CACHE_TYPE" => "A",
        "CACHE_TIME" => "36000000",
        "CACHE_GROUPS" => "Y"
    )
);?></p>

```

Рисунок 3.5 – Настроенный вызов компонента «Структура разделов»

Для вывода детальной информации о каждом подразделении будем использовать компонент «Элементы раздела». Компонент выводит список элементов раздела с указанным набором свойств. Компонент содержит шесть шаблонов: .default, board, links, list, old\_version (шаблон .default до версии 14.0) и visual. Компонент стандартный и входит в дистрибутив модуля.

В визуальном редакторе компонент расположен по пути Контент> Каталог> Элементы раздела.

На рисунке 3.6 приведен пример стандартного вызова компонента.

```

<?APPLICATION->IncludeComponent(
    "bitrix:catalog.section",
    "",
    Array(
        "TEMPLATE_THEME" => "blue",
        "PRODUCT_DISPLAY_MODE" => "N",
        "ADD_FICT_PROP" => "MORE_PHOTO",
        "LABEL_PROP" => "NEW_BOOK",
        "OFFER_ADD_FICT_PROP" => "FILE",
        "OFFER_TREE_PROPS" => array("-",),
        "PRODUCT_SUBSCRIPTION" => "N",
        "SHOW_DISCOUNT_PERCENT" => "N",
        "SHOW_OLD_PRICE" => "N",
        "SHOW_CLOSE_POPUP" => "Y",
        "MESS_BTN_BUY" => "Купить",
        "MESS_BTN_ADD_TO_BASKET" => "В корзину",
        "MESS_BTN_SUBSCRIBE" => "Подписаться",
        "MESS_BTN_DETAIL" => "Подробнее",
        "MESS_NOT_AVAILABLE" => "Нет в наличии",
        "AJAX_MODE" => "Y",
        "SEF_MODE" => "N",
        "IBLOCK_TYPE" => "books",
        "IBLOCK_ID" => "6",
        "SECTION_ID" => $REQUEST["SECTION_ID"],
        "SECTION_CODE" => "",
        "SECTION_USER_FIELDS" => array(),
        "ELEMENT_SORT_FIELD" => "sort",
        "ELEMENT_SORT_ORDER" => "asc",
        "ELEMENT_SORT_FIELD2" => "name",
        "ELEMENT_SORT_ORDER2" => "asc",
        "FILTER_NAME" => "arrFilter",
        "INCLUDE_SUBSECTIONS" => "Y",
        "SHOW_ALL_NO_SECTION" => "Y",
        "SECTION_URI" => "",
        "DETAIL_URL" => "",
        "BASKET_URL" => "/personal/basket.php",
        "ACTION_VARIABLE" => "action",
        "PRODUCT_ID_VARIABLE" => "id",
        "PRODUCT_QUANTITY_VARIABLE" => "quantity",
        "ADD_PROPERTIES_TO_BASKET" => "Y",
        "PRODUCT_PROPS_VARIABLE" => "prop",
        "PARTIAL_PRODUCT_PROPERTIES" => "N",
        "SECTION_ID_VARIABLE" => "SECTION_ID",
        "ADD_SECTIONS_CHAIN" => "Y",
        "DISPLAY_COMPARE" => "N",
        "SET_TITLE" => "Y",
        "SET_BROWSER_TITLE" => "Y",
        "BROWSER_TITLE" => "-",
        "SET_META_KEYWORDS" => "Y",
        "META_KEYWORDS" => "",
        "SET_META_DESCRIPTION" => "Y",
        "META_DESCRIPTION" => "",
        "SET_LAST_MODIFIED" => "Y",
        "USE_MAIN_ELEMENT_SECTION" => "Y",
        "SET_STATUS_404" => "N",
        "PAGE_ELEMENT_COUNT" => "30",
        "LINE_ELEMENT_COUNT" => "3",
        "PROPERTY_CODE" => array(),
        "OFFERS_FIELD_CODE" => array(),
        "OFFERS_PROPERTY_CODE" => array(),
        "OFFERS_SORT_FIELD" => "sort",
        "OFFERS_SORT_ORDER" => "asc",
        "OFFERS_SORT_FIELD2" => "active_from",
        "OFFERS_SORT_ORDER2" => "desc",
        "OFFERS_LIMIT" => "5",
        "BACKGROUND_IMAGE" => "-",
        "PRICE_CODE" => array(),
        "USE_PRICE_COUNT" => "Y",
        "SHOW_PRICE_COUNT" => "1",
        "PRICE_VAT_INCLUDE" => "Y",
        "PRODUCT_PROPERTIES" => array(),
        "USE_PRODUCT_QUANTITY" => "Y",
        "CACHE_TYPE" => "A",
        "CACHE_TIME" => "3600000",
        "CACHE_FILTER" => "Y",
        "CACHE_GROUPS" => "Y",
        "DISPLAY_TOP_PAGER" => "N",
        "DISPLAY_BOTTOM_PAGER" => "Y",
        "PAGER_TITLE" => "Товары",
        "PAGER_SHOW_ALWAYS" => "Y",
        "PAGER_TEMPLATE" => "",
        "PAGER_DESC_NUMBERING" => "Y",
        "PAGER_DESC_NUMBERING_CACHE_TIME" => "36000",
        "PAGER_SHOW_ALL" => "Y",
        "HIDE_NOT_AVAILABLE" => "Y",
        "OFFERS_CART_PROPERTIES" => array(),
        "AJAX_OPTION_JUMP" => "N",
        "AJAX_OPTION_STYLE" => "Y",
        "AJAX_OPTION_HISTORY" => "N",
        "CONVERT_CURRENCY" => "Y",
        "CURRENCY_ID" => "RUB",
        "ADD_TO_BASKET_ACTION" => "ADD",
        "PAGER_BASE_LINK_ENABLE" => "Y",
        "SET_STATUS_404" => "Y",
        "SHOW_404" => "Y",
        "MESSAGE_404" => "",
        "PAGER_BASE_LINK" => "",
        "PAGER_PARAMS_NAME" => "arrPager"
    )
);?>

```

Рисунок 3.6 – Стандартный вызов компонента «Элементы раздела»

Показывать все свойства нам не нужно, некоторые оставляем скрытыми.

Это показано на рисунке 3.7.

```

"PROPERTY_CODE" => array("EMAIL", "ADDRESS", "SUB", "PERSONS", "HEAD", "POLOZHENIE",
    "WORKTIME", "DIRECTOR", "RUKOVODITEL", "SITE", "PHONE", "NAME", "EMAIL",
    "NAME", "ADDRESS", "SUB", "PERSONS", "HEAD", "WORKTIME", "SITE", "PHONE"),

```

Рисунок 3.7 – Настройка отображения свойств элементов компонента

«Элемент раздела»

### 3.6 Разработка шаблона отображения детальной информации о подразделениях

Для вывода детальной информации о подразделениях стандартные шаблоны не подходят, необходимо разработать индивидуальный.

Располагать информацию о каждом подразделении наиболее целесообразно в виде таблицы [7]. Наполнение каждой ячейки таблицы можно увидеть на рисунке 3.8 [9]. При выводе свойств нужно обязательно проверять, существуют ли они для данного элемента, используя функцию `isset` [8]. Функция `isset` определяет, была ли установлена переменная со значением, отличным от `NULL` [12].

```
<td valign="top"><?=$arElement["NAME"]?><br />
<?foreach($arElement["DISPLAY_PROPERTIES"] as $pid=>$arProperty):?>
<?=$arProperty["NAME"]?>:&nbsp;&nbsp;&nbsp;<?
    if ($arProperty["CODE"] != 'SITE'){
        if ($arProperty["CODE"] != 'DIRECTOR'){
            if ($arProperty["CODE"] != 'ZAV'){
                if ($arProperty["CODE"] != 'POLOZHENIE'){
                    $pattern = "/<a[^>]+(.*?)</a>/";
                    if (is_array($arProperty["DISPLAY_VALUE"])){
                        foreach ($arProperty["DISPLAY_VALUE"] as $url){
                            preg_match_all($pattern, $url, $matches);
                            echo "&nbsp;&nbsp;&nbsp;";
                        }
                    } else {
                        if (isset($arProperty["SITE"])){
                            echo $arProperty["SITE"]["DISPLAY_VALUE"];?><br />
                        }
                        if (isset($arProperty["DIRECTOR"])){
                            echo $arProperty["DIRECTOR"]["DISPLAY_VALUE"];?><br />
                        }
                        echo $arProperty["DISPLAY_VALUE"];
                    }
                } else {
                    if (is_array($arProperty["DISPLAY_VALUE"]))
                        echo implode("&nbsp;&nbsp;&nbsp;/", $arProperty["DISPLAY_VALUE"]);
                    else
                        echo $arProperty["DISPLAY_VALUE"];
                }
            } else {
                if (is_array($arProperty["DISPLAY_VALUE"]))
                    echo implode("&nbsp;&nbsp;&nbsp;/", $arProperty["DISPLAY_VALUE"]);
                else
                    echo $arProperty["DISPLAY_VALUE"];
            }
        }
    }
    echo "<br/>";
?>
<?endforeach?>
<br />
<?=$arElement["PREVIEW_TEXT"]?>
```

Рисунок 3.8 – Шаблон для отображения детальной информации

Так как инфоблок структуры связан с инфоблоком научно-педагогического состава, поэтому имя руководителя подразделения будет в виде ссылки. Внешний вид такой структуры представлен на рисунке 3.9.

## Институт математики, физики и информационных технологий

Институт математики, физики и информационных технологий  
E-mail: [svtalalov@ttsu.ru](mailto:svtalalov@ttsu.ru)  
Директор: [Талалов Сергей Владимирович](#)  
Кабинет: (УЛК-410б)  
Положение: Положение о кафедре, Положение об институте  
Сайт: [Институт математики, физики и информационных технологий](#)  
Телефон: 53-93-09

Алгебра и геометрия  
Заведующий: [Утеева Роза Азербаяевна](#)  
Институт: [Институт математики, физики и информационных технологий](#)

Кабинет: УЛК-404  
Сайт: [Кафедра «Алгебра и геометрия»](#)  
Телефон: 53-91-13

Высшая математика и математическое моделирование  
Заведующий: [Зибров Петр Федорович](#)  
Институт: [Институт математики, физики и информационных технологий](#)

Кабинет: УЛК-414  
Сайт: [Кафедра «Высшая математика и математическое моделирование»](#)  
Телефон: 53-92-17, 53-91-17

Общая и теоретическая физика  
Заведующий: [Павлова Анджела Петровна](#)  
Институт: [Институт математики, физики и информационных технологий](#)

Кабинет: Г-316а  
Сайт: [Кафедра «Общая и теоретическая физика»](#)  
Телефон: 53-91-23

Прикладная математика и информатика  
Заведующий: [Очеповский Андрей Викторович](#)  
Институт: [Институт математики, физики и информационных технологий](#)

Кабинет: УЛК-403  
Положение: [Прикладная математика и информатика](#)  
Сайт: [Кафедра «Прикладная математика и информатика»](#)  
Секции: [Общая информатика](#)

Телефон: 53-95-14

Общая информатика  
Институт: [Институт математики, физики и информационных технологий](#)

Кафедра: [Прикладная математика и информатика](#)

[Вернуться к списку](#)

Рисунок 3.9 – Визуальное отображение детальной информации

## Заключение

В ходе работы был проведен анализ сайта Тольяттинского государственного университета на соответствие требованиям нормативной базы, изучены методические рекомендации Рособнадзора и Федеральный закон об образовании РФ. Анализ информационной структуры сайта был проведен с помощью построения графа структуры сайта, в котором отражены все связи между разделами.

В результате анализа информационной структуры сайта ТГУ был выявлен ряд несоответствий методическим рекомендациям и Федеральному закону «Об образовании в Российской Федерации»:

1. Отсутствие одиннадцати необходимых разделов, указанных в методических рекомендациях представления информации об образовательной организации в открытых источниках с учетом соблюдения требований законодательства в сфере образования (для образовательных организаций высшего образования) Рособнадзора.
2. Несоответствие названий документов названиям, приведенным в методических рекомендациях Рособнадзора.
3. Несоответствие формата документов формату, указанному в методических рекомендациях Рособнадзора.

Анализ сайта показал, что структуру сайта нужно оптимизировать. Мною был предложен вариант оптимизации информационной структуры, построен граф оптимизированной структуры сайта образовательного учреждения, которая будет намного удобнее и понятнее пользователю и в которой отражены все методические рекомендации, так же составлен прототип интерфейса для раздела «Сведения об образовательной организации».

На основе законов Фиттса и Хика было вычислено время для достижения страницы «Сведения об образовательной организации» до и после оптимизации. В результате оптимизации время, необходимое для достижения данного раздела, оказалось намного меньше.

В результате работы было создано несколько шаблонов страниц и шаблоны отображения детальной информации. Созданы сами страницы, полностью соответствующие оптимизированной структуре, приведенной выше. Количество просмотров страницы «Сведения об образовательной организации» заметно увеличилось с 306 в месяц до 4827.

## Список используемой литературы

1. Закон Российской Федерации «Об образовании» (от 29 декабря 2012 года №273-ФЗ). [Электронный ресурс]: сайт Министерства образования и науки Российской Федерации: <http://минобрнауки.рф/документы/2974>.
2. Методические рекомендации представления информации об образовательной организации в открытых источниках с учетом соблюдения требований законодательства в сфере образования (для образовательных организаций высшего образования) [Электронный ресурс]: сайт Рособнадзора: [http://www.obrnadzor.gov.ru/common/upload/doc\\_list/Methodicheskie\\_rekomendatsii.pdf](http://www.obrnadzor.gov.ru/common/upload/doc_list/Methodicheskie_rekomendatsii.pdf).
3. Проект постановления Правительства России об утверждении Порядка размещения на официальном сайте образовательной организации в информационно-телекоммуникационной сети «Интернет» и обновления информации об образовательной организации [Электронный ресурс]: официальный сайт Министерства образования и науки Российской Федерации: <http://www.mon.gov.ru>.
4. Горячкин, Б.С., Яроц, Е.В. Метод для определения эргономичности текстовых редакторов // Инженерный вестник, №12, 2014. – С. 33-35.
5. Баканов, А.С. Эргономика пользовательского интерфейса. От проектирования к моделированию человеко-компьютерного интерфейса / А.С. Баканов, А.А. Обознов. – М.: «Институт психологии РАН», 2011. - 176 с.
6. Нидерст-Роббинс, Д. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Д. Нидерст-Роббинсон - М.: Эксмо, 2014. – 528 с.
7. Роббинс, Д. HTML5. Карманный справочник / Д. Роббинс - М.: Вильямс, 2015. – 192 с.
8. Зандстра, М. PHP. Объекты, шаблоны и методики программирования / М. Зандстра - М.: Вильямс, 2011. - 560 с.
9. Сухов, К. HTML 5. Путеводитель по технологии / К. Сухов - М.: ДМК Пресс, 2013. – 312 с.

10. Прохоренок, Н. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера/ Н. Прохоренок, В. Дронов. - СПб.: БХВ-Петербург, 2015. – 768 с.
11. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон - СПб.: Питер, 2012. – 688 с.
12. Клименко, Р. Веб-мастеринг на 100% / Р. Клименко - СПб.: Питер, 2013. – 455 с.
13. Хоган Б., HTML5 и CSS3. Веб-разработка по стандартам нового поколения / Б. Хоган - СПб.: Питер, 2014. – 272 с.
14. Титтел, Э. HTML, XHTML и CSS для чайников. / Э. Титтел, Д. Ноубл. - М.: «Диалектика», 2011. – 400 с.
15. Якушина, Е. В. Характеристика образовательного пространства Интернет. [Электронный ресурс]: Лаборатория ТСО и медиаобразования ИСМО РАО: <http://www.mediaeducation.ru/publ/stat2.shtml>.
16. Asadi, A., HTML 5 & CSS3 Genius Guide. - Imagine Publishing Ltd, 2015.
17. Goldstein, A., HTML5 & CSS3 For The Real World. - SitePoint, 2015.
18. Cederholm D., CSS3 for Web Designers. - A Book Apart, 2014.
19. Sawyer McFarland. D., CSS: The Missing Manual. - O'Reilly Media, 2015.
20. Mathis, L., Designed for Use. - Pragmatic Bookshelf, 2011.
21. Hampton-Smith, S., Pro CSS3 Layout Techniques. - Apress, 2016.

## Приложение А

### Листинг кода файла template.php

```
<? if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED !== true)
die(); ?>

<div class="catalog-section">

    <? if ($arParams["DISPLAY_TOP_PAGER"]): ?>

        <?= $arResult["NAV_STRING"] ?><br/>

    <? endif; ?>

    <table cellpadding="0" cellspacing="0" border="0">

        <? foreach ($arResult["ITEMS"] as $cell => $arResult): ?>

            <? $this->AddEditAction($arResult['ID'], $arResult['EDIT_LINK'],
CIBlock::GetArrayByID($arParams["IBLOCK_ID"], "ELEMENT_EDIT"));

            $this->AddDeleteAction($arResult['ID'], $arResult['DELETE_LINK'],
CIBlock::GetArrayByID($arParams["IBLOCK_ID"], "ELEMENT_DELETE"),
array("CONFIRM" =>
 GetMessage('CT_BCS_ELEMENT_DELETE_CONFIRM')));?>

            <? if ($cell % $arParams["LINE_ELEMENT_COUNT"] == 0): ?>

                <tr>

            <? endif; ?>

            <td valign="top" width="<?= round(100 /
$arParams["LINE_ELEMENT_COUNT"]) ?>%"
            id="<?= $this->GetEditAreaId($arResult['ID']); ?>">

                <table cellpadding="0" cellspacing="2" border="0">

                    <tr>
```

```

        <? if (is_array($arResult["PREVIEW_PICTURE"])): ?>

            <td valign="top">

                "
                    width="<?=$arResult["PREVIEW_PICTURE"]["WIDTH"]?>"
                    height="<?=$arResult["PREVIEW_PICTURE"]["HEIGHT"] ?>"
                    alt="<?=$arResult["PREVIEW_PICTURE"]["ALT"] ?>"
                    title="<?=$arResult["PREVIEW_PICTURE"]["TITLE"] ?>" />

                <br />

            </td>

        <? elseif (is_array($arResult["DETAIL_PICTURE"])): ?>

            <td
                valign="top">"
                    width="<?=$arResult["DETAIL_PICTURE"]["WIDTH"] ?>"
                    height="<?=$arResult["DETAIL_PICTURE"]["HEIGHT"] ?>"
                    alt="<?=$arResult["DETAIL_PICTURE"]["ALT"] ?>"
                    title="<?=$arResult["DETAIL_PICTURE"]["TITLE"] ?>" /><br />

            </td>

        <? endif ?>

        <td valign="top"><?=$arResult["NAME"] ?><br />

        <? foreach ($arResult["DISPLAY_PROPERTIES"] as $pid =>
$arResultProperty): ?>

            <?=$arResultProperty["NAME"] ?>:&nbsp;&nbsp;&nbsp;<?

            if ($arResultProperty["CODE"] != 'SITE') {

```

```

if ($arParams["CODE"] != 'DIRECTOR') {
    if ($arParams["CODE"] != 'ZAV') {
        if ($arParams["CODE"] != 'POLOZHENIE') {
            $pattern = "/<a[^>]+>(.*?)</a>"/;
            if (is_array($arParams["DISPLAY_VALUE"])) {
                foreach ($arParams["DISPLAY_VALUE"] as
$url){
                    preg_match_all($pattern, $url, $matches);
                    echo "&nbsp;/&nbsp;"; . $matches[1][0];
                }
            } else {
                if (isset($arParams["SITE"])) {
                    echo $arParams["SITE"]["DISPLAY_VALUE"];
                }
                if (isset($arParams["DIRECTOR"])) {
                    echo
$arParams["DIRECTOR"]["DISPLAY_VALUE"];
                }
                if (isset($arParams["ZAV"])) {
                    echo $arParams["ZAV"]["DISPLAY_VALUE"];
                }
            }
        }
    }
}

```

```

        if (isset($arParams["POLOZHENIE"])) {
            echo
$arParams["POLOZHENIE"]["DISPLAY_VALUE"]; ?><br/>
            <?>
            preg_match_all($pattern,
$arParams["DISPLAY_VALUE"], $matches);
            if (!empty($matches[1])) {
                echo $matches[1][0] . "<br />";
            } else {
                echo $arParams["DISPLAY_VALUE"];
            }
        }
    } else {
        if (is_array($arParams["DISPLAY_VALUE"]))
            echo          implode("&nbsp;/&nbsp;",
$arParams["DISPLAY_VALUE"]);
        else
            echo $arParams["DISPLAY_VALUE"];
    } else {
        if (is_array($arParams["DISPLAY_VALUE"]))
            echo          implode("&nbsp;/&nbsp;",
$arParams["DISPLAY_VALUE"]);
        else

```

```

        echo $arProperty["DISPLAY_VALUE"];}
    } else {
        if (is_array($arProperty["DISPLAY_VALUE"]))
            echo          implode("&nbsp;/&nbsp;",
$arProperty["DISPLAY_VALUE"]);
        else
            echo $arProperty["DISPLAY_VALUE"];
    }
} else {
    if (is_array($arProperty["DISPLAY_VALUE"]))
        echo          implode("&nbsp;/&nbsp;",
$arProperty["DISPLAY_VALUE"]);
    else
        echo $arProperty["DISPLAY_VALUE"];

    echo "<br/>";?>
<? endforeach ?>
<br/>
<?= $arElement["PREVIEW_TEXT"] ?>
</td>
</tr>
</table>
<?      if      (is_array($arElement["OFFERS"])      &&
!empty($arElement["OFFERS"])): ?>

```



```

                class="catalog-price"><?=
$arParams["PRINT_DISCOUNT_VALUE"] ?></span>

                <? else: ?>

                <span                                class="catalog-price"><?=
$arParams["PRINT_VALUE"] ?></span>

                <? endif ?>

            </p>

        <? endif; ?>

    <? endforeach; ?>

<p>

<? if ($arParams["DISPLAY_COMPARE"]): ?>

    <noindex></noindex>

<? endif ?>

<? if ($arParams["CAN_BUY"]): ?>

    <? if ($arParams["USE_PRODUCT_QUANTITY"]): ?>

        <form    action="<?=    POST_FORM_ACTION_URI    ?>"
method="post" enctype="multipart/form-data">

            <table border="0" cellspacing="0" cellpadding="2">

                <tr valign="top">

                    <td><?    echo    GetMessage("CT_BCS_QUANTITY")
?></td>

                    <td>

                        <input                                type="text"
name="<? echo $arParams["PRODUCT_QUANTITY_VARIABLE"] ?>"

```

```

                value="1" size="5">
            </td>
        </tr>
    </table>

    <input        type="hidden"        name="<?        echo
$arParams["ACTION_VARIABLE"] ?>" value="BUY">

    <input        type="hidden"        name="<?        echo
$arParams["PRODUCT_ID_VARIABLE"] ?>"

                value="<? echo $arResult["ID"] ?>">

    <input        type="submit"        name="<?        echo
$arParams["ACTION_VARIABLE"] . "BUY" ?>"

                value="<? echo GetMessage("CATALOG_BUY") ?>">

    <input        type="submit"        name="<?        echo
$arParams["ACTION_VARIABLE"] . "ADD2BASKET" ?>" value="<? echo
GetMessage("CATALOG_ADD") ?>">

    </form>

    <? else: ?>

        <noindex></noindex>

    <? endif; ?>

    <? elseif (count($arResult["PRICES"]) > 0): ?>

        <?= GetMessage("CATALOG_NOT_AVAILABLE") ?>

    <?

                                                $APPLICATION-
>IncludeComponent("bitrix:sale.notice.product", ".default", array(
                "NOTIFY_ID" => $arResult['ID'],

```



```

<thead>

<tr>

    <? if (count($arElement["PRICE_MATRIX"]["ROWS"]) >= 1
    && ($arElement["PRICE_MATRIX"]["ROWS"][0]["QUANTITY_FROM"] > 0 ||
    $arElement["PRICE_MATRIX"]["ROWS"][0]["QUANTITY_TO"] > 0)): ?>

        <td                valign="top"                nowrap><?=
    GetMessage("CATALOG_QUANTITY") ?></td>

        <? endif ?>

        <? foreach ($arElement["PRICE_MATRIX"]["COLS"] as
    $TypeID => $arType): ?>

            <td valign="top" nowrap><?= $arType["NAME_LANG"]
    ?></td>

        <? endforeach ?>

    </tr>

</thead>

<? foreach ($arElement["PRICE_MATRIX"]["ROWS"] as $ind =>
    $arQuantity): ?>

    <tr>

        <? if (count($arElement["PRICE_MATRIX"]["ROWS"]) > 1 ||
    count($arElement["PRICE_MATRIX"]["ROWS"]) == 1 &&
    ($arElement["PRICE_MATRIX"]["ROWS"][0]["QUANTITY_FROM"] > 0 ||
    $arElement["PRICE_MATRIX"]["ROWS"][0]["QUANTITY_TO"] > 0)): ?>

            <th nowrap><?

                if (IntVal($arQuantity["QUANTITY_FROM"]) > 0 &&
    IntVal($arQuantity["QUANTITY_TO"]) > 0)

```

```

                                echo                                str_replace("#FROM#",
$arResult["QUANTITY_FROM"],                                str_replace("#TO#",
$arResult["QUANTITY_TO"],
getMessage("CATALOG_QUANTITY_FROM_TO"));

                                elseif (intval($arResult["QUANTITY_FROM"]) > 0)

                                echo                                str_replace("#FROM#",
$arResult["QUANTITY_FROM"],
getMessage("CATALOG_QUANTITY_FROM"));

                                elseif (intval($arResult["QUANTITY_TO"]) > 0)

                                echo                                str_replace("#TO#",
$arResult["QUANTITY_TO"], getMessage("CATALOG_QUANTITY_TO"));

                                ?></th>

                                <? endif ?>

                                <? foreach ($arResult["PRICE_MATRIX"]["COLS"] as
$TypeID => $arType): ?>

                                <td>

                                <?if
($arResult["PRICE_MATRIX"]["MATRIX"][$TypeID][$ind]["DISCOUNT_PRIC
E"] < $arResult["PRICE_MATRIX"]["MATRIX"][$TypeID][$ind]["PRICE"]):?>

                                <s><? =
FormatCurrency($arResult["PRICE_MATRIX"]["MATRIX"][$TypeID][$ind]["PR
ICE"],
$arResult["PRICE_MATRIX"]["MATRIX"][$TypeID][$ind]["CURRENCY"])
?></s>

                                <span                                class="catalog-price"><? =
FormatCurrency($arResult["PRICE_MATRIX"]["MATRIX"][$TypeID][$ind]["DI

```

```

SCOUNT_PRICE"],
$arResult["PRICE_MATRIX"]["MATRIX"][$TypeID][$ind]["CURRENCY"]);
?></span>

        <? else: ?>

                <span                                class="catalog-price"><?=
FormatCurrency($arResult["PRICE_MATRIX"]["MATRIX"][$TypeID][$ind]["PR
ICE"],
$arResult["PRICE_MATRIX"]["MATRIX"][$TypeID][$ind]["CURRENCY"]);
?></span>

                <? endif ?>&nbsp;

        </td>

        <? endforeach ?>

</tr>

<? endforeach ?>

</table><br/>

<? endif ?>

<? if ($arParams["DISPLAY_COMPARE"]): ?>

        <noindex> </noindex>

<? endif ?>

<? if ($arResult["CAN_BUY"]): ?>

        <?if                ($arParams["USE_PRODUCT_QUANTITY"]                ||
count($arResult["PRODUCT_PROPERTIES"]): ?>

                <form                action="<?=                POST_FORM_ACTION_URI                ?>"
method="post" enctype="multipart/form-data">

```

```

<table border="0" cellspacing="0" cellpadding="2">
  <? if ($arParams["USE_PRODUCT_QUANTITY"]): ?>
    <tr valign="top">
      <td><? echo GetMessage("CT_BCS_QUANTITY")
?></td>
      <td>
        <input type="text"
name="<? echo $arParams["PRODUCT_QUANTITY_VARIABLE"] ?>" value="1"
size="5">
      </td>
    </tr>
  <? endif; ?>
  <? foreach ($arResult["PRODUCT_PROPERTIES"] as $pid
=> $product_property): ?>
    <tr valign="top">
      <td>
        <? echo $arResult["PROPERTIES"][$pid]["NAME"] ?></td>
      <td>
        <?if
($arResult["PROPERTIES"][$pid]["PROPERTY_TYPE"] == "L"
&&
$arResult["PROPERTIES"][$pid]["LIST_TYPE"] == "C"): ?>
          <? foreach ($product_property["VALUES"] as $k =>
$v): ?>

```

```

                <label><input type="radio" name="<? echo
$arParams["PRODUCT_PROPS_VARIABLE"] ?>[<? echo $pid ?>]" value="<?
echo $k ?>" <? if ($k == $product_property["SELECTED"]) echo "checked" ?>><?
echo $v ?>

                </label><br>

                <? endforeach; ?>

                <? else: ?>

                <select name="<? echo
$arParams["PRODUCT_PROPS_VARIABLE"] ?>[<? echo $pid ?>]">

                <? foreach ($product_property["VALUES"] as $k
=> $v): ?>

                <option value="<? echo $k ?>" <? if ($k ==
$product_property["SELECTED"]) echo "selected" ?>><? echo $v ?></option>

                <? endforeach; ?>

                </select>

                <? endif; ?>

                </td>

                </tr>

                <? endforeach; ?>

                </table>

                <input type="hidden" name="<? echo
$arParams["ACTION_VARIABLE"] ?>" value="BUY">

                <input type="hidden" name="<? echo
$arParams["PRODUCT_ID_VARIABLE"] ?>" value="<? echo $arResult["ID"]
?>">

```

```
                <input                type="submit"                name="<?                echo
$arParams["ACTION_VARIABLE"]        .        "BUY"        ?>"        value="<?        echo
    GetMessage("CATALOG_BUY") ?>">
```

```
                <input                type="submit"                name="<?                echo
$arParams["ACTION_VARIABLE"]        .        "ADD2BASKET"        ?>"        value="<?        echo
    GetMessage("CATALOG_ADD") ?>">
```

```
        </form>
```

```
        <? else: ?>
```

```
        <? endif; ?>
```

```
        <?        elseif        ((count($arResult["PRICES"])        >        0)        ||
is_array($arResult["PRICE_MATRIX"]): ?>
```

```
        <?= GetMessage("CATALOG_NOT_AVAILABLE") ?>
```

```
        <? $APPLICATION->IncludeComponent("bitrix:sale.notice.product",
".default", array(
```

```
                "NOTIFY_ID" => $arResult['ID'],
```

```
                "NOTIFY_URL"                                ==>
```

```
                htmlspecialchars($arResult["SUBSCRIBE_URL"]),
```

```
                "NOTIFY_USE_CAPTCHA" => "N" ),
```

```
                $component?>
```

```
        <? endif ?>
```

```
        <? endif ?>
```

```
    </td>
```

```
    <? $cell++;
```

```
    if ($cell % $arParams["LINE_ELEMENT_COUNT"] == 0):?>
```

```
</tr>

<? endif ?>

<? endforeach; // foreach($arResult["ITEMS"] as $arResultElement):?>

<? if ($cell % $arParams["LINE_ELEMENT_COUNT"] != 0): ?>

    <? while (($cell++) % $arParams["LINE_ELEMENT_COUNT"] != 0): ?>

        <td>&nbsp;</td>

    <? endwhile; ?>

</tr>

<? endif ?>

</table>

<a href=".">Вернуться к списку</a>

<? if ($arParams["DISPLAY_BOTTOM_PAGER"]): ?>

    <br/><?=$arResult["NAV_STRING"] ?>

<? endif; ?>

</div>
```