

## Содержание

Введение.....	4
1 Анализ задачи.....	7
1.1 Постановка задачи.....	7
1.2 Инструменты разработки.....	10
1.3 Требования к приложению.....	13
2 Проектирование задачи.....	16
2.1 Организация данных.....	16
2.2 Процессы.....	16
2.3 Описание внешнего пользовательского интерфейса.....	18
3 Реализация.....	20
3.1 Структура программы.....	20
3.1.1 Структура и описание процедур и функций пользователя.....	20
3.1.2 Описание использованных компонентов.....	22
3.2 Спецификация программы.....	23
4 Тестирование.....	26
5 Применение.....	28
5.1 Общие сведения о программном продукте.....	28
5.2 Инсталляция.....	28
5.3 Выполнения программы.....	29
5.3.1 Запуск программы.....	29
5.3.2 Инструкции по работе с программой.....	29
5.4 Использование системы справочной информации.....	35
Заключение.....	36
Список использованных источников.....	37
Приложение А Листинг программы.....	38

						КП 2-40 01 01 01.35.32.06.21 ПЗ			
Изм.	Кол	Лист	Издок	Подпись	Дата	Разработка программного обеспечения по рациональному распределению доходов и расходов «Студенческий кошелек»	Стадия	Лист	Листов
Разрабо-		Муханюк						3	57
Проверил		Заяц							
Н. контр.							УО ГГПК		
Утв.									

## Введение

На курсовое проектирование была поставлена задача, разработать программного обеспечения по рациональному распределению доходов и расходов «Студенческий кошелек»..

Цель курсового проекта заключается в ознакомлении студентов с понятием «бюджет» и его значение в его жизни, закреплении знаний и автоматизации следующих действий: понятие «бюджет», как рационально распределить доходы и расходы.

Создаваемая программа будет рассчитана на любого рода пользователей. Любой студент сможет ознакомиться с понятием «бюджет», создать свой бюджет, узнать куда большую часть его денег уходит, рационально распределить доходы и расходы, начать правильно вести учет за своим бюджетом.

Важным аспектом при решении данной задачи, является то, что программа будет предоставлять необходимые знания в данной области. Это может пригодиться, если студент захочет начать жить самостоятельно и перестать зависеть от родителей, перестать бессознательно тратить свои деньги и вести учет за их расходом.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название “Анализ задачи”. В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно-экономической сущности. Также в этом разделе вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе. В подразделе “Инструменты разработки” будет рассмотрена среда, в которой создается данный курсовой проект. Здесь также будут установлены минимальные и оптимальные требования

к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе “Проектирование задачи” будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет четко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

“Реализация задачи” – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе будут четко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов используемых в данном проекте.

Четвертый раздел – “Тестирование”. В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Будут смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе “Применение” будет описано назначение, область применения, среда функционирования курсовой программы. Также в нем будет описано использование справочной системы.

“Заключение” будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В “Литературе” будет приведен список используемых при разработке источников.

В приложениях к пояснительной записке будет приведен листинг программы с необходимыми комментариями.

Схема работы системы будет представлена в графической части.

						КП 2-40 01 01 01.35.32.06.21 ПЗ	Лист
Изм.	Коп.	Лист	№ док	Подпись	Дата		6

## 1 Анализ задачи

### 1.1 Постановка задачи

Темой данного курсового проекта является разработка программного обеспечения по рациональному распределению доходов и расходов «Студенческий кошелек».

За время учебы в вузе через руки студентов проходят такие суммы денег, посчитав которые можно ужаснуться. Пополняется студенческий кошелек не так часто, как хотелось бы, а опустошается он еще быстрее. Почему так происходит не сможет объяснить не один студент, даже самый щепетильный, записывающий все свои доходы и расходы.

Поступив в вуз и попав в непривычную обстановку, первое время студенты стараются экономить на всем. Но все это обманчиво. Вокруг столько много соблазнов, что удержаться просто невозможно. На что же уходит львиная доля бюджета? Основные расходы студентов состоят в следующем:

- Счет за жилье. В зависимости от места проживания, тратятся различные суммы денег. Общежитие обходится значительно дешевле съемной комнаты. А студенты, желающие проживать с комфортом, снимают квартиры и оплачивают дополнительно счета за коммунальные услуги.

- Питание и проезд. Многие студенты стараются готовить самостоятельно, не всегда это удается, но экономия существенная. Продукты нужно закупать в супермаркетах или на рынках, где можно и поторговаться. Питаться в кафе дорого, да и еда там не очень полезная. Для проезда на общественном транспорте лучше всего купить проездной. Но студенты — народ непоседливый, возвращаясь с вечеринки расходов на такси не избежать. Сэкономить и в этом случае можно — протопать весь маршрут пешком.

– Учебный процесс. Расходы студентов, помимо оплаты за учебу (если они учатся на коммерческой основе), заключаются в покупке канцелярских товаров, частенько требуются копии документов. Со временем можно потратиться на принтер и сканер. Здесь уже можно подзаработать, распечатывая документы сокурсникам.

– Телефон и интернет. Сотовая связь и пользование интернетом также съедает часть бюджета. Экономия на бесплатном доступе в интернет-кафе оборачивается дополнительными расходами на еду и напитки.

– Отдых и развлечения. Самая огромная дыра в студенческом кошельке. Походы в клубы, развлекательные центры, отдых на природе — съедают деньги со скоростью света. Вроде бы вчера была приличная сумма, а наутро денег едва находится на обед в студенческой столовой. Здесь не помогает никакая экономия.

А ведь еще хочется хорошо выглядеть, иметь стильный телефон, девушкам нужны деньги на косметику. Могут возникнуть непредсказуемые обстоятельства, требующие внесения энной суммы. Расходов много, а доходы не стремятся к увеличению. Хотя и это поправимо, заработать некоторую сумму денег для студентов не проблема, главное не лениться.

Кропотливо следить за доходами и расходами сможет не каждый студент, но стремиться к этому необходимо. В итоге незаметно для себя в кошельке обнаружится достаточное количество денег, которые можно потратить на давно приглянувшуюся вещь.

Изучение программного обеспечения по рациональному распределению доходов и расходов «Студенческий кошелек» может понадобиться многим обычным пользователям. В частности, с изучением и определением методом распределения доходов и расходов.

Аналогов для такого приложения можно привести не так уж и много. В основном в сети интернет находятся различного рода статьи по данной те-

матике, сочинения других людей. В более редких случаях – ознакомительный материал с примерами и проверками на внимательность читателя в виде теста и вывода результата прохождения на экран. Собранный же воедино материал с интерактивным и обучающим элементом встречается единично на просторах интернет.

Периодичность использования данного программного продукта неограниченна. Пользователь может в любое время установить, а также удалить приложение после полного изучения материала по данной теме и прохождения тестов.

Цель данного курсового проекта – разработать программный продукт, который будет сочетать в себе элементы обучения, проверки усвоенного материала и развлечения для людей, интересующихся своими расходами и доходами, а также разработкой игр «Копейка в копейку», «Студенческий кошелёк».

За основу программы следует взять из соответствующих источников (сайтов, статей, энциклопедий) наиболее полную, достоверную и точную информацию. При реализации проекта разработчик должен стараться сделать его простым и максимально доступным для пользователей.

Данный проект должен стать общедоступным для всех пользователей. В поставленной задаче необходимо реализовать максимально простой пользовательский интерфейс, позволяющий использовать проект пользователю, не обладающему дополнительными знаниями электронно-вычислительной машины.

Разрабатываемый программный продукт должен позволять выполнять следующие действия:

- просмотр информации о рациональном распределении доходов и расходов;
- прохождение игры «Студенческий кошелёк» для закрепления информации;

- просмотр справки;
- просмотр пункта «о разработчике»;
- просмотр пункта «о программе»;
- возможность сыграть в игру «Копейка в копейку».

В разрабатываемой программе будут использоваться три вида данных. К входной информации можно отнести вводимые пользователем значения, информацию о пользователе. К выходной – статьи на тему рациональное распределение доходов и расходов, игры «Копейка в копейку» и «Студенческий кошелек».

Постоянной информацией в проекте будут являться текстовые файлы, картинки, медиа файлы и др.

Программный продукт предоставляет функционал для следующего ряда пользователей: зарегистрированный пользователь.

## 1.2 Инструменты разработки

Для разработки данного проекта будет выбрана среда Delphi Lite Edition v. 7.3.4.3, так как это самая удобная и доступная среда разработки на данный момент. Delphi - язык программирования, относящийся к классу RAD- (Rapid Application Development – «Средство быстрой разработки приложений») средств CASE – технологии. Delphi сделал разработку приложений для windows быстрым и приятным процессом. Теперь разрабатывать сложные и интересные проекты можно только одним человеком, использующим Delphi.

Интерфейс Windows обеспечивает полное перенесение CASE-технологии в интегральную систему поддержки работ по созданию прикладной системы на всех фазах жизненного цикла работы и проектирования системы.

Delphi обладает широким набором возможностей, начиная от проекти-



ровщика форм и заканчивая поддержкой всех форматов популярных баз данных. Среда устраняет необходимость программировать такие компоненты Windows общего назначения, как метки, программы и даже диалоговые панели. Работая в Windows, можно видеть одинаковые «объекты» во многих разнообразных приложениях. Диалоговые панели (например, Choose File и Save File) являются примерами многократно-используемых компонентов, встроенных непосредственно в Delphi, который позволяет приспособить эти компоненты к имеющейся задаче, чтобы они работали именно так, как требуется создаваемому приложению. Также здесь имеются предварительно-определенные визуальные и не визуальные объекты.

Три основные части разработки интерфейса следующие: проектирование панели, проектирование диалога и представление окон. Для общего пользовательского доступа также должны учитываться условия применения архитектуры прикладных систем.

Сегодня появилась реальная возможность с помощью моделирования на современных многофункциональных средствах обработки и отображения информации таких как Delphi конкретизировать тип и характеристики используемых информационных моделей, выявить основные особенности будущей деятельности операторов, сформулировать требования к параметрам аппаратно-программных средств интерфейса взаимодействия и т.д.

Delphi позволяет создать различные виды программ: консольные приложения, оконные приложения, приложения для работы с Интернетом и базами данных. То есть, Delphi является не только средствами для работы с языком программирования Паскаль, но дополнительные инструменты, призванные для максимального упрощения и ускорения создание приложений. К дополнительным инструментам можно отнести визуальный редактор форм, благодаря которому можно с легкостью создать полноценную программу, и другие визуальные составляющие разработки программного обеспечения. С Delphi вам не нужно вручную просчитывать расположение

каждого элемента интерфейса пользователя, поэтому при разработке программы значительно экономится время.

Выгоды от проектирования в среде Windows с помощью Delphi:

1. устраняется необходимость в повторном вводе данных;
2. обеспечивается согласованность проекта и его реализации;
3. увеличивается производительность разработки и переносимость программ.

Ни одно серьезное программное обеспечение не обходится без модуля справочной информации и руководства пользователя. Это придает программе законченный вид и показывает заботу о пользователе.

**Help Manual** – легкий в использовании и функциональный инструмент, упрощающий создание справочных файлов Windows, печать справочных руководств и документации в целом. Программа имеет интуитивно понятный интерфейс. Все созданные проекты можно сохранить в различных форматах: HTML Help, Winhelp и MS Help 2.0 / Visual Studio Help, Browser-based Help, PDF и Word RTF, а также печатной документации при использовании одного и того же проекта. В основном окне программы содержатся оглавление (в виде древовидного списка) и текстовый редактор. Это дает возможность легко ориентироваться в оглавлении, редактировать или перемещать разделы справки без каких-либо проблем. Также утилита позволяет конвертировать help-файлы из одного формата в другой. Кроме приложений для работы с текстом в данном продукте содержатся утилиты для создания скриншотов и редактирования графических файлов.

Для создания инсталлятора будет использоваться мощное и удобное средство - **Smart Install Maker**. Программа обладает удобным и интуитивно понятным интерфейсом, а также полным набором необходимых функций для создания профессиональных инсталляторов с минимальным размером, высокой степенью сжатия файлов и приятным интерфейсом. Помимо стандартного минимума, Smart Install Maker позволяет редактировать системный реестр

и INI-файлы, создавать программные ярлыки, запускать ассоциируемые и исполняемые файлы, регистрировать новые шрифты и ActiveX компоненты, отображать тексты информации и лицензионного соглашения. Также, с помощью этой утилиты, можно создать мультязыковые инсталляторы с поддержкой более 20-ти популярных языков мира.

**Microsoft Word 2019** – редактор текста для написания документации.

**Microsoft Power Point 2019** – программа для создания презентации.

**Блокнот** – для создания html страниц.

Для нормального функционирования разрабатываемого проектанеобходимо наличие у компьютера следующих параметров:

- процессор не ниже Pentium 500 Mhz;
- объем ОЗУ не меньше 128 Mb;
- объем места на HDD не менее 200 Mb;
- видеоподсистема не менее 1024x768 точек с глубиной цвета 16 Bit;
- ОС – Windows 7,8,10.

Как видно разрабатываемое приложение не очень требовательно к аппаратным ресурсам, что, является большим плюсом.

### 1.3 Требования к приложению

На этапе исследования предметной области был установлен целый ряд требований, которые предъявляются к разрабатываемой программе.

При моделировании форм следует учесть такой момент: основная проблема состоит в том, что нынешняя молодежь достаточно поверхностно ознакомлена с тем как нужно правильно распределять доходы и расходы. По этой причине особенно актуальной становится разработка программного электронного продукта, который способствовал бы и давал возможность вспомнить, изучить как рационально распределить свой бюджет. Чтобы эта

информация лучше усваивалась необходимо подобрать правильный интерфейс.

**Требования к интерфейсу:** в связи с частым использованием программы она должна быть с приятной цветовой гаммой и понятной для пользователя. Следовательно, каждое окно должно иметь ясную визуальную иерархию своих элементов. Фрагменты текста должны располагаться на экране так, чтобы пользователя было просто и понятно принимать информацию.

Пользователь не должен испытывать какого-либо дискомфорта в плане восприятия информация, отображённой на экране. Объекты (рисунки и символы) не должны быть слишком мелкие. Все окна приложения по возможности должны помещаться на экран полностью, так как использование в процессе работы полос прокруток достаточно неудобно.

На одной форме нельзя допускать избытка и нагромождения данных. Формы должны быть эффектно оформлены согласно тематике разрабатываемого проекта.

#### **Требования к надежности:**

Специальных требований к надежности не предъявляется.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств. Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой через графический интерфейс не должны влиять на конечный результат. ПП должен иметь средства фиксации всех

действий в процессе изготовления программного продукта. Это необходимо для восстановления состояния процесса разработки на любом его этапе (при комплексной отладке).

## 2 Проектирование задачи

### 2.1 Организация данных

Основным средством хранения информации в приложении будут html-странички с информацией о данной сфере, БД для хранения всех пользователей. В проекте будут присутствовать материалы по расходу и доходу, игры на данную тематику.

В базе данных будет храниться информация о пользователях в следующем виде:

- Номер пользователя;
- Никнейм пользователя;
- Пароль пользователя;
- Имя пользователя;
- Фамилия пользователя;
- Отчество пользователя;
- Пол пользователя;
- Дата рождения пользователя;

Система справочной информации будет представлена файлом справки, который будет содержать информацию о приложении и правила его использования.

### 2.2 Процессы

Основными функциями приложения будут являться:

- изучение теоретического материала ;
- прохождение игр;
- просмотр видео в стили «Лайфхаки для студента».

Воспользовавшись пунктом меню «Справка», можно будет узнать информацию о разработчике.

Согласно всем перечисленным требованиям и указаниям, которые были рассмотрены в разделе «Анализ задачи», было определено, чем конкретно должна заниматься разрабатываемая программа. Главной её задачей будет являться ознакомления пользователя с понятием «бюджет» и как можно рационально распределить свои доходы и расходы.

Для реализации задач будут использоваться процедуры. С помощью процедуры, например, будет осуществляться загрузка html-страницы на форму.

Заходя на главную форму, пользователь сможет перейти на вкладку «О программе», «О разработчике», Справка и выйти.

Затем, после главной формы, пользователь может авторизоваться, но если не зарегистрирован ему будет предоставлено возможность зарегистрироваться и, затем, снова попытаться авторизоваться.

После авторизацию пользователю будет видна информация о нем и доступ к материалам по данной теме курсового проекта, также он сможет поиграть в игру «Копейка в копейку» и «Студенческий кошелёк».

В игре «Копейка в копейку» пользователь сможет проверить свои знания по классификации продуктов. Пользователю необходимо за определённую сумму нужно выбрать хотя бы 1 товар из каждой группы. Выбор товара происходит по нажатию на CheckBox. Подсчет суммы, на которую закупился пользователь и подсчет количества товара в корзине происходит путем нажатия на Button «Подсчитать сумму и кол-во товара в корзине», а проверка условия происходит после нажатия на Button «Проверить условия игры».

В игре «Студенческий кошелёк» пользователь может проконтролировать свой бюджет за месяц. В ней будут содержаться доход, полученный за счёт стипендии и денег, данные родителями или кем-то из близких родственников, и расходы, в них будут включаться оплата за общежитие, если он в нем проживает, продуктовый магазин, где пользователь может расписать трату за каждую неделю по отдельности, транспорт, канцелярские расходы, мобильная связь, отдых в клубах или ресторанах, отдых на природе, и промтоварный магазин, где пользователь может расписать растраты по каждой из 4 недель. Данные о рас-

тратах пользователь вводит через Edit, а вид расходов выбирает по нажатию на CheckBox. Доход пользователь получает после введения своего среднего балла и также он может ввести сумму, полученную от родителей или близких родственников.

Материал будет генерироваться на основе выбора пункта в объекте TButton. В WebBrowser будет загружаться данная страничка с помощью процедуры showHtml(). Далее пользователь сможет ознакомиться с информацией и дальше использовать ее в своей жизни.

### 2.3 Описание внешнего пользовательского интерфейса

Важным при выполнении курсового проекта является организация диалога между пользователем и самой программой. Во многом это зависит от того, как программист разработает данную программу, какие компоненты будут использованы и какие методы будут автоматизированы.

Особое внимание следует уделить интерфейсу. Разработчик должен так организовать внешний вид своей программы, что бы пользователь понял, что от него требуется. При решении задачи будут использованы компоненты для ввода информации и её вывода, для вставки видео, текста, и др.

Для организации эффективной работы пользователя нужно создать целостное приложение данной предметной области, в которой все компоненты приложения будут сгруппированы по функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс пользователя.

Таким образом, для успешной работы всего проекта в целом следует обеспечить интуитивно понятный интерфейс с приятной гаммой цветов и шрифтами.

Структура навигации по проекту представлена на рисунке 1:



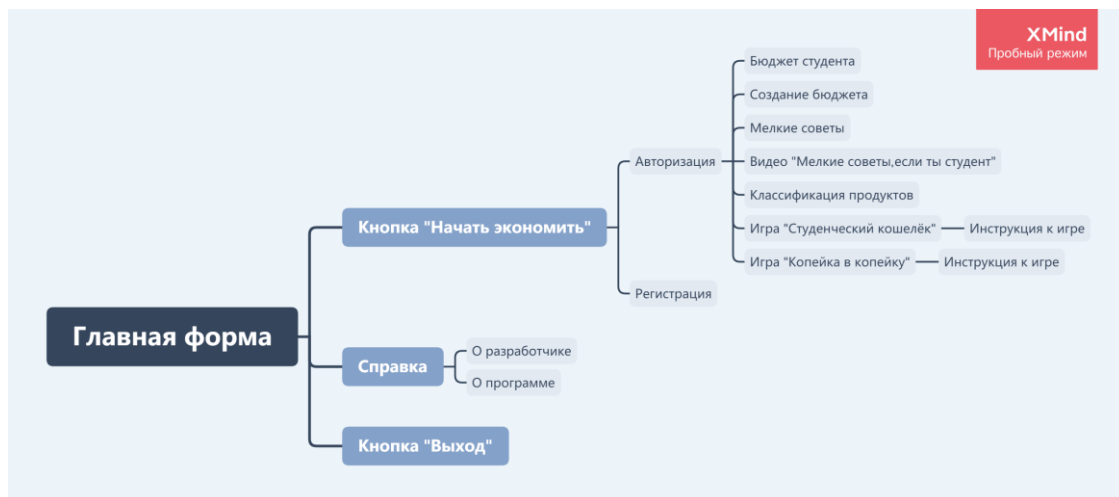


Рисунок 1 – Структура навигации по проекту

Изм.	Коп.	Лист	№ док.	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

19

## 3 Реализация

### 3.1 Структура программы

Данный курсовой проект содержит 7 модулей. Далее рассмотрим назначение каждого модуля:

- 1 Модуль Unit1 - отвечает за вход пользователя в систему, с него пользователь может перейти на модуль регистрации (Form3)
- 2 Модуль Unit2 – приветствующая загрузка проекта, открывающая главное меню(Form1)
- 3 Модуль Unit3 – отвечает за авторизацию пользователя, после успешной авторизации пользователя, открывается форма меню (Unit5)
- 4 Модуль Unit4 – служит для регистрации пользователя в системе.
- 5 В модуле Unit5 - находится вся основная информация программы: информация о данной теме, переход к играм и просмотр видео;
- 6 Модуль Unit6 – содержит WebBrowser, в который подгружаются html-страницы с ознакомительной информацией;
- 7 Модуль Unit7 – содержит видеоролик;

#### 3.1.1 Структура и описание процедур и функций пользователя

Описание разработанных процедур находятся в таблице 1.

Таблица 1 – Процедуры и функции

Имя процедуры (функции)	В каком модуле находится	За каким компонентом закреплена	Назначение
1 procedure TForm1.ExitClick(Sender: TObject);	Unit1	ButtonExit	Выход
2 procedure TForm1.N2Click(Sender: TObject);	Unit1	MainMenu	Открытие проекта «О разработчике»

Продолжение таблицы 1

3	procedure TForm1.N3Click(Sender: TObject);	Unit1	MainMenu	Открытие про- екта «О про- грамме»
4	procedure TForm1.StartClick(Sender: TObject);	Unit1	ButtonStart	Переход на Unit3
5	procedure TForm5.Button3Click(Sender : TObject);	Unit5	ButtonButton3	Открытие Unit6 и отображение ин- формации
6	procedure TForm5.Button4Click(Sender : TObject);	Unit5	ButtonButton4	Переход на Unit7
7	procedure TForm6.Button1Click(Sender : TObject);	Unit6	ButtonButton1	Возврат на Unit5
8	procedure TForm7.Button1Click(Sender : TObject);	Unit7	ButtonButton1	Запуск видеоро- лика
9	procedure TForm7.Button2Click(Sender : TObject);	Unit7	ButtonButton2	Возврат на Unit5
10	ProcedureT- Form1.FormCreate(Sender: TObject);	Unit1	Form1	Запуск музыки в проекте
11	procedure TForm3.BRegistrClick(Sende r: TObject);	Unit3	ButtonBRegistr	Переход на форму регистра- ции(Unit4)
12	procedure TForm3.BVxodClick(Sender: TObject);	Unit3	ButtonBVxod	Процесс автори- зации и переход на Unit5
13	procedure TForm4.BCloseClick(Sender: TObject);	Unit4	-	Возврат на Unit3
14	procedure TForm4.BRegistrClick(Sende r: TObject);	Unit4	ButtonBRegistr	Занесение пользо- вателя в базу дан- ных
15	procedure TForm5.FormShow(Sender: TObject);	Unit5	Unit5	Вывод на экран данных о пользо- вателе
16	procedure TForm5.Button1Click(Sender : TObject);	Unit5	ButtonButton1	Открытие проекта «Копейка в ко- пейку»
17	procedure TForm5.BExitClick(Sender: TObject);	Unit5	ButtonBExit	Выход из проекта
18	procedure TForm5.Button6Click(Sender : TObject);	Unit5	ButtonButton6	Открытие Unit6 и отображение ин- формации
19	procedure TForm5.Button7Click(Sender : TObject);	Unit5	ButtonButton7	Запуск проекта «Студенческий кошелёк»

Изм.	Коп.	Лист	№ док.	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

21

Продолжение таблицы 1

20	procedure TForm5.AconomiuyClick(Sender: TObject);	Unit5	ButtonAconomiuy	Открытие Unit6 и отображение информации
21	procedure TForm5.Button2Click(Sender: TObject);	Unit5	ButtonButton2	Открытие Unit6 и отображение информации
22	procedure TForm5.Button4Click(Sender: TObject);	Unit5	ButtonButton4	Переход на Unit7
23	procedure TForm6.Button1Click(Sender: TObject);	Unit6	ButtonButton1	Возврат на Unit5
24	procedure TForm7.Button1Click(Sender: TObject);	Unit7	ButtonButton1	Запуск видеоролика
25	procedure TForm7.Button2Click(Sender: TObject);	Unit7	ButtonButton2	Возврат на Unit5

### 3.1.2 Описание использованных компонентов

Описание использованных для разработки приложения компонентов приводится в таблице 2.

Таблица 2 – Использованные компоненты

Компонент	На какой форме расположен	Назначение
1 TMainMenu	Unit1 и проекты «Копейка в копейку», «Студенческий кошелёк»	Используется для создания проекта
2 TImage	Unit1, Unit2, Unit3, Unit4, Unit5 и проекты «Копейка в копейку», «Студенческий кошелёк»	Используется как фон, кнопки
3 TTimer	Unit2	Используется для определения длительности загрузочного экрана
4 TWebBrowser	Unit6 и проекты «Копейка в копейку», «Студенческий кошелёк»	Используется для отображения html-страниц с материалами

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

22

Продолжение таблицы 2

5 TLabel	Unit3, Unit4, Unit5 и проекты «Копейка в копейку», «Студенческий кошелёк»	Отображение надписей на форме
6 TButton	Unit1, Unit3, Unit4, Unit5, Unit6, Unit7 и проекты «Копейка в копейку», «Студенческий кошелёк»	Используется для взаимодействия пользователя с программой: выполнить переход, отменить действие, очистить поля ввода и т.д.
7 TPanel	Unit7 и проекты «Копейка в копейку», «Студенческий кошелёк»	Используется для масштабирования интерфейса
8 TADOConnection	Unit3	Используется для связи с БД
9 TADODataSource	Unit3, Unit5	Используется для нахождения пользователя при входе в программу
10 TADOQuery	Unit3, Unit4, Unit5	Используется для выполнения sql-запросов: добавление пользователя в БД
11 TComboBox	Проекты «Копейка в копейку», «Студенческий кошелёк»	Используется для выбора значения из ограниченных вариантов
12 TEdit	Unit3, Unit4 и проекты «Копейка в копейку», «Студенческий кошелёк»	Используется для заполнения данных

### 3.2 Спецификация программы

Точное название проекта и его состав приводится в Таблице 3.

Таблица 3 – Спецификация основной программы

Имя файла	Назначение
1 Student_Wallet.exe	Исполняемый файл основного проекта
2 Student_Wallet.dpr	Файл проекта, связывает все файлы, из которых состоит приложение
3 Unit1.pas	Файл программного модуля для формы Unit1
4 Unit2.pas	Файл программного модуля для формы Unit2
5 Unit3.pas	Файл программного модуля для формы Unit3
6 Unit 4.pas	Файл программного модуля для формы Unit4
7 Unit5.pas	Файл программного модуля для формы Unit5
8 Unit6.pas	Файл программного модуля для формы Unit6
9 Unit7.pas	Файл программного модуля для формы Unit7
10 Game.exe	Исполняемый файл проекта «Студенческий кошелёк»
11 SuperMarket.exe	Исполняемый файл проекта «Копейка в копейку»
12 Unit1.dfm	Главная форма
13 Unit2.dfm	Форма заставки
14 Unit3.dfm	Форма авторизации
15 Unit 4.dfm	Форма регистрации
16 Unit5.dfm	Форма меню
17 Setup.exe	Установочный файл
18 Help.chm	Файл справки, содержит помощь по работе с программой
19 Unit6.dfm	Форма информации
20 Unit7.dfm	Форма видео
21 Бюджет Студента.htm, 22 Категория продуктов.htm, 23 Мелкие советы.htm, 24 Создание Бюджета.htm.	Html- странички с информацией

Таблица 4 – Спецификация программы «Студенческий кошелёк»

Имя файла	Назначение
1 Game.exe	Исполняемый файл проекта «Студенческий кошелёк»
2 Game.dpr	Файл проекта, связывает все файлы, из которых состоит приложение
3 Unit1.pas	Файл программного модуля для формы Unit1
4 Unit2.pas	Файл программного модуля для формы Unit2
5 Unit1.dfm	Главная форма
6 Unit 2.dfm	Форма инструкции

Таблица 5 – Спецификация программы «Копейка в копейку»

Имя файла		Назначение
1	SuperMarket.exe	Исполняемый файл проекта «Копейка в копейку»
2	SuperMarket.dpr	Файл проекта, связывает все файлы, из которых состоит приложение
3	Unit1.pas	Файл программного модуля для формы Unit1
4	Unit2.pas	Файл программного модуля для формы Unit2
5	Unit1.dfm	Главная форма
6	Unit 2.dfm	Форма инструкции

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

25

## 4 Тестирование

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Отчёт о результатах тестирования предоставлен в Таблице 6.

Таблица 6 – Отчёт о результатах тестирования

Тест	Ожидаемый результат	Физический результат	Результат тестирования
1 Проверка пункта меню «О разработчике»	Открытие проекта «О разработчике»	Открытие проекта «О разработчике»	Выполнено
2 Проверка пункта меню «О программе»	Открытие проекта «О программе»	Открытие проекта «О программе»	Выполнено
3 Проверка пункта меню «Тест»	Окно с регистрацией	Окно с регистрацией	Выполнено
4 Проверка кнопки «Выход»	Закрытие проекта	Закрытие проекта	Выполнено
5 Проверка кнопки «Начать «Экономить»»	Открытие формы «Авторизация»	Открытие формы «Авторизация»	Выполнено
6 Проверка входа пользователя в систему	Открытие главной формы	Открытие главной формы	Выполнено
7 Проверка кнопки «Регистрация»	Переход на форму «Регистрации»	Переход на форму «Регистрации»	Выполнено
8 Проверка регистрации и добавления пользователя в БД	Новая запись в БД с пользователем	Новая запись в БД с пользователем	Выполнено
9 Проверка отображения информационного материала	Открытие новой формы и в WebBrowser отображается выбранный материал	Открытие новой формы и в WebBrowser отображается выбранный материал	Выполнено
10 Проверка кнопки «Как тратить меньше, если ты студент»	Открытие новой формы и запуск видео	Открытие новой формы и запуск видео	Выполнено
11 Проверка кнопки «Студенческий кошелек»	Открытие другого проекта	Открытие другого проекта	Выполнено
12 Проверка игры «Студенческий кошелек»	Оптимальная работа всех Edit, Button и CheckBox	Оптимальная работа всех Edit, Button и CheckBox	Выполнено



Продолжение таблицы 6

13 Проверка кнопки «Копейка в копейку»	Открытие другого проекта	Открытие другого проекта	Выполнено
14 Проверка игры «Копейка в копейку»	Оптимальная работа всех Edit, Button и Check-Box	Оптимальная работа всех Edit, Button и Check-Box	Выполнено

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

При использовании данного программного обеспечения пользователь не столкнется с ошибками системы.

## 5 Руководство пользователя

### 5.1 Общие сведения о программном продукте

Разработанная программа является обучающим приложением по рациональному распределению доходов и расходов.

Данный программный продукт преимущественно предназначен для учащихся УО, которые живут в общежитии или на съёмной квартире. Также может быть использован людьми, которые хотят узнать, что такое бюджет, и начать рационально распределять свои доходы и расходы.

Быстродействие любой программы во многом зависит от характеристик выбранного персонального компьютера: рабочей частоты процессора, объема оперативной памяти и т.д. Несмотря на все реализованные в ней задачи, она легко запускается и функционирует на любых машинах.

Тестирование проводилось на разных классах ЭВМ и работать с данной программой было комфортно. Программа разработана на ПК со следующими характеристиками:

- Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz;
- объем ОЗУ 16Гб;
- графический адаптер NVIDIA GeForce GTX 1650 8gb;
- операционная система Windows 10.

### 5.2 Инсталляция

Для того, чтобы установить программу необходимо запустить файл Setup.exe. Появится окно установки программного обеспечения по рациональному распределению доходов и расходов «Студенческий кошелёк».

При нажатии на кнопку «Далее» появляется окно выбора каталога установки.

Следуем инструкции и устанавливаем приложение.

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

28

## 5.3 Выполнение программы

### 5.3.1 Запуск программы

Данную программу можно запустить различными способами.

Первым из них является запуск с помощью ярлыка на рабочем столе. Необходимо дважды щелкнуть левой кнопкой мыши на ярлыке с названием Student Wallet.exe.

Вторым способом является запуск из каталога, в который устанавливалось приложение (по умолчанию C:\Program Files\ Student Wallet\ Student Wallet.exe).

По подготовленным тестам будет осуществляться функциональное и полное тестирование программного продукта. Отчет о результатах тестирования будет представлен в 4 разделе пояснительной записки.

### 5.3.2 Инструкции по работе с программой

После запуска на экране появляется заставка (рисунок 2)



Рисунок 2 – Форма заставки

Изм.	Коп.	Лист	№ док	Подпись	Дата

После появляется главная форма (рисунок 3), где пользователь может почитать о программе, о разработчике, справка, перейти на форму авторизации и выйти из программы.

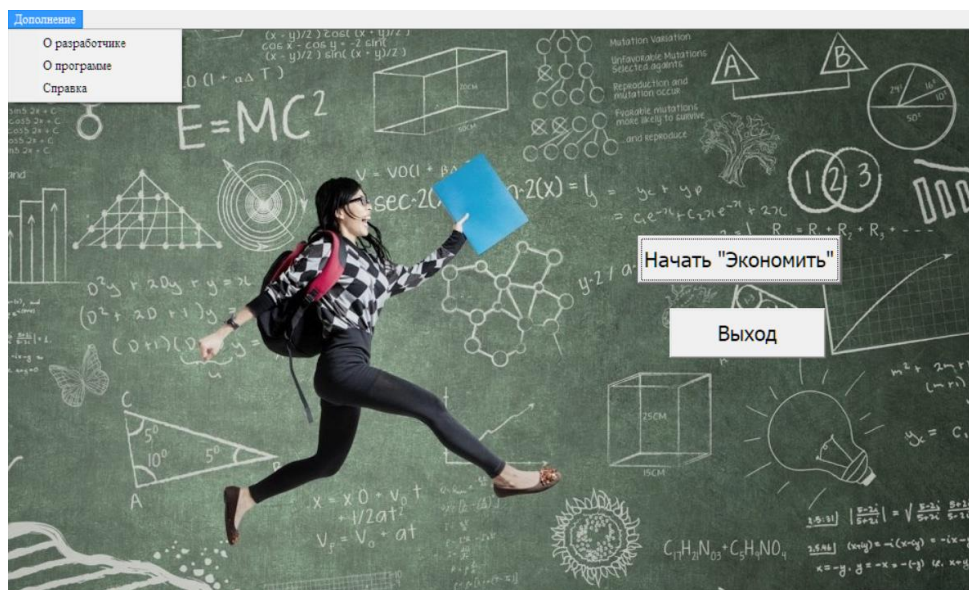


Рисунок 3 – Главная форма

После нажатия кнопки «Начать «Экономить»» открывается форма авторизации (рисунок 4), где пользователь может войти в систему, вернуться на главную форму или перейти на форму регистрации (рисунок 5), где пользователь может занести свои данные в базу данных.

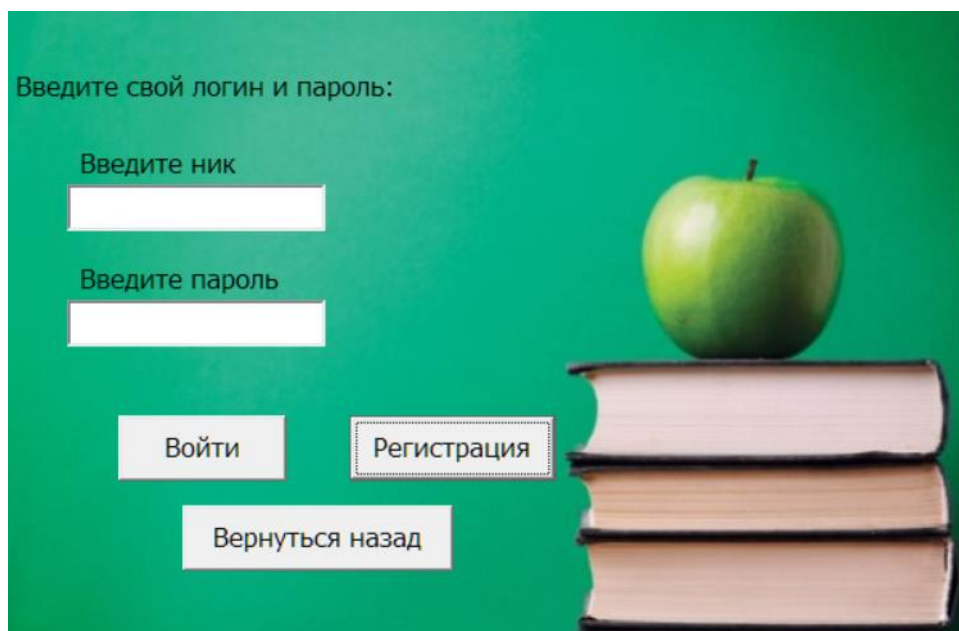


Рисунок 4 – Форма авторизации

Введите ник:  Введите пароль:

Введите фамилию:  Введите имя:  Введите отчество:

Выберите пол:  Введите дату рождения:

Рисунок 5 – Форма регистрации

После авторизации открывается форма меню (рисунок 6), где находится информация по данной тематике и игры «Студенческий кошелёк» и «Копейка в копейку»

Имя: Александр      Фамилия: Муханюк      Отчество: Васильевич

Дата рождения: 13.04.2004 17:28:41      Пол: Мужской

Рисунок 6 – Форма меню

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ



После нажатия кнопок «Бюджет студента», «Создание бюджета», «Мелкие советы», «Классификация продуктов» будет открываться форма информации (рисунок 7, рисунок 8, рисунок 9, рисунок 10), на которой будет отображаться информация по каждой из этих тем.



Рисунок 7 – Форма информация «Бюджет студента»

**Как мне создать бюджет?**

Создание бюджета довольно просто и начинается с этого простого уравнения: Что вы зарабатываете (ваш доход) за вычетом того, что вы тратите (ваши расходы). Этапы создания бюджета включают в себя:

- 1. Запишите ваш общий доход за месяц**  
Это ваш общий доход. Не забудьте указать все: работу с полным рабочим днем, подработку, стипендия, чеки социального обеспечения и любые другие источники постоянного дохода.
- 2. Перечислите все ваши расходы**  
Подумайте о своих регулярных расходах. И нерегулярных расходах, которые должны быть оплачены в следующем месяце.  
После этого просуммируйте другие расходы, такие как еда и развлечения. Следует учитывать каждый потраченный рубль.
- 3. Вычтите расходы из дохода до нуля**  
Это называется бюджетом с нулевой базой, то есть ваш доход за вычетом расходов должен равняться нулю. Если у вас больше или меньше, проверьте свои математические

Назад

Рисунок 8 – Форма информации «Создание бюджета»

**Правильно разделите деньги.** Попробуйте правило 50/30/20. Оно гласит, что 50% ежемесячного дохода должны уходить на необходимые траты: оплата за общежитие или квартиру(комнату), питание, плату за коммуналку, транспорт и прочие важные вещи. 20% нужно в начале каждого месяца откладывать. На оставшиеся 30% можно развлекаться — сходить на шопинг, в ресторан, салон красоты.

Конечно, проценты могут отличаться: у кого-то только на аренду квартиры уходит половина заработка. Но стремитесь к тому, чтобы 20% удавалось откладывать ежемесячно.

**Создайте резервный фонд.** 10% из тех средств, что вы отложили, пойдут на создание «подушки безопасности». В идеале она должна быть такой, чтобы в экстренной ситуации вы могли от 1 до 2 месяцев поддерживать тот уровень жизни, который был раньше. Эти деньги нельзя тратить, чтобы развлечься или купить кому-то подарок. Единственная ситуация, в которой их можно использовать — случилось что-то действительно серьезное.



Назад

Рисунок 9 – Форма информации «Мелкие советы»

#### Классификация товаров

**Классификация товаров** — это систематизированное их распределение на группы по наиболее общим признакам. В товароведении применяют различные виды классификации: биологическую, производственную, учебную, торговую и др. Наиболее часто пользуются учебной и торговой классификацией.

В основу классификации товаров могут быть положены такие признаки, как происхождение, химический состав, степень обработки сырья, назначение товаров и т.д. Так, в зависимости от происхождения пищевые продукты можно подразделить на продукты животного, растительного и минерального происхождения; по химическому составу — на белковые, жировые, углеводные, минеральные; по степени обработки — на сырые, полуфабрикаты, готовые изделия; по назначению — на пищевые и вкусовые.

**По учебной классификации** пищевые продукты подразделяют на следующие группы. Зерномучные товары (зерно, мука, крупы, крупяные, макаронные и хлебобулочные изделия) характеризуются высоким содержанием углеводов.



Назад

Рисунок 10 – Форма информации «Классификация товаров»

После нажатия кнопки «Как тратить меньше, если ты студент» открывается форма видео (рисунок 11), на которой пользователь может просмотреть видео по данной теме.

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

33

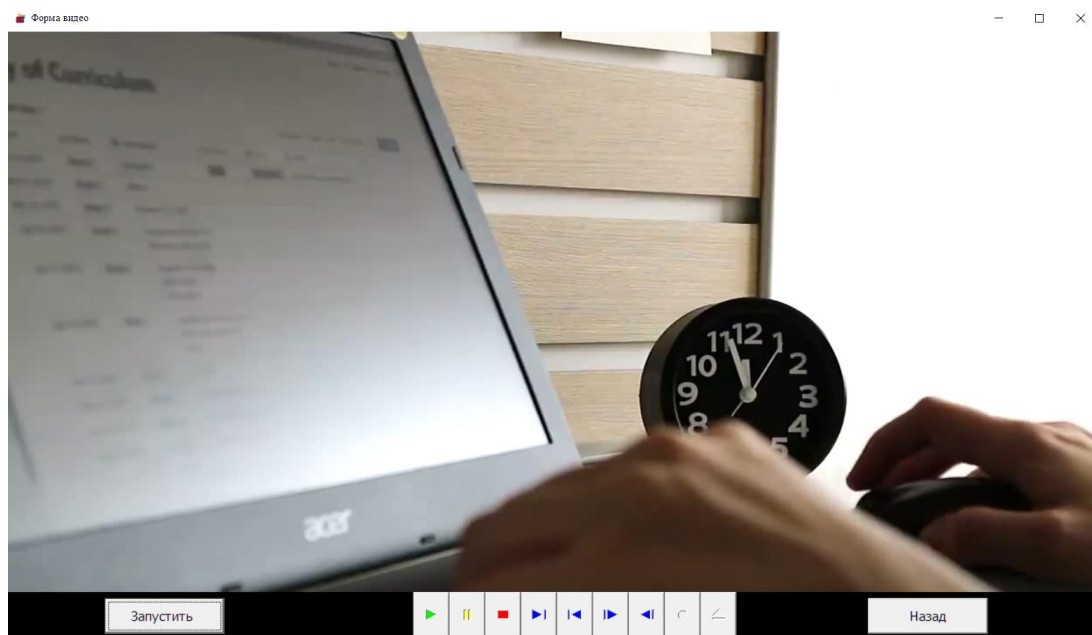


Рисунок 11 – Форма видео «Как тратить меньше, если ты студент»

Так же с формы меню можно перейти на игры «Студенческий кошелек» (рисунок 12) и «Копейка в копейку» (рисунок 13), нажав на кнопки соответственно: «Студенческий кошелек», «Копейка в копейку».

 The screenshot shows the main interface of the "Student Wallet" game. At the top, there are tabs for "Инструкция к игре" and "Выход". Below them are input fields for "Введите ваш средний бал" and "Начислите стипендию". A section for "Ваш бюджет:" and "Вы потратили:" includes a "Вг" (input) field and a "Подсчитать затраты" button. A family icon and a "У кого?" dropdown are also present, along with a "Попросить денег" button. The main area features various expense categories with input fields: "Плата за общежитие" (8,60), "Мобильная связь", "Магазин" (with 1, 2, 3, 4 week options), "Транспорт", "Развлечения", "Отдых на природе", "Канцелярские товары", and "Продуктовый магазин" (with 1, 2, 3, 4 week options). Each category is accompanied by a relevant icon.

Рисунок 12 – Главная форма игры «Студенческий кошелек»

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ





Рисунок 13 – Главная форма меню «Копейка в копейку»

#### 5.4 Использование системы справочной информации

Справочную систему можно запустить с помощью пункта меню Дополнение - Справка (рисунок 14)

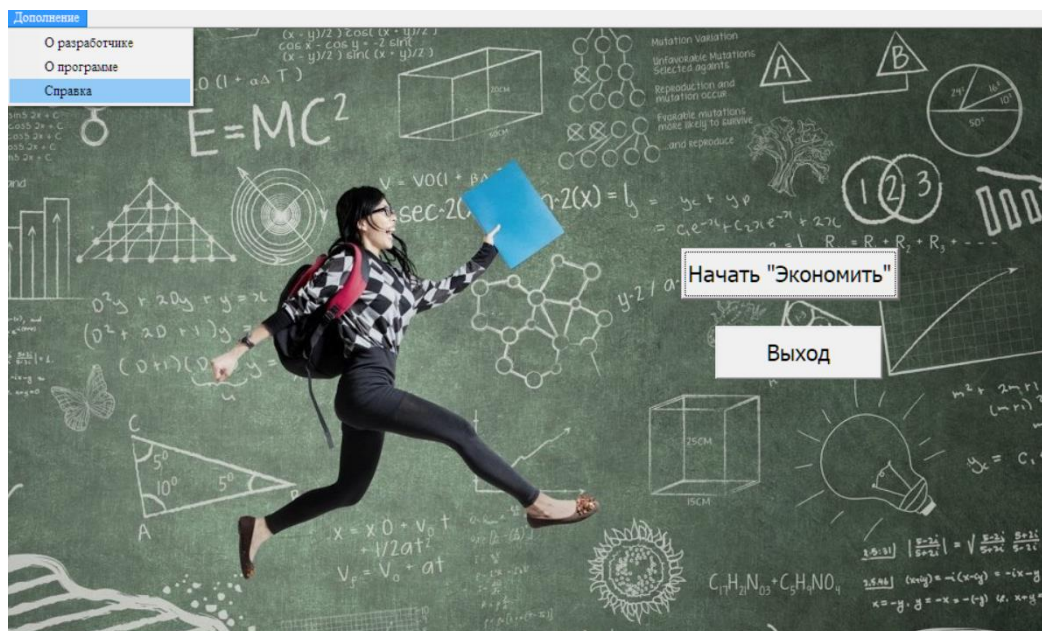


Рисунок 14 – Запуск справки.

После нажатия на данный пункт меню появится форма с помощью пользователю.

## Заключение

Целью данного проекта было разработать программное обеспечение по рациональному распределению доходов и расходов «Студенческий кошелек».

В поставленной задаче был реализован простой и понятный пользовательский интерфейс.

В ходе тестирования все исключительные ситуации были обработаны. Проект работает без сбоев и ошибок.

Продукт находится в законченной стадии. В связи с этим, данный программный продукт имеет законченный вид.

Исходя из этого, можно сделать вывод, что программа реализована успешно.

						КП 2-40 01 01 01.35.32.06.21 ПЗ	Лист
Изм.	Коп.	Лист	№ док	Подпись	Дата		36

## Список использованных источников

- 1 Евдокимов, Н.М. Как правильно управлять своими личными финансами. - СПб.: Невский проспект; Вектор, 2007.
- 2 Осипенкова, О.Ю. Учет затрат, калькулирование и бюджетирование – М : МГИУ, 2007 – 124 с.
- 3 Добровольский, Е.Ю. Бюджетирование: шаг за шагом – СПб.: Питер, 2013. – 480 с.

						КП 2-40 01 01 01.35.32.06.21 ПЗ	Лист
Изм.	Коп.	Лист	№ док	Подпись	Дата		37

**Приложение А**  
**Листинг программы**

						КП 2-40 01 01 01.35.32.06.21 ПЗ	Лист
Изм.	Коп.	Лист	№ док.	Подпись	Дата		38

```

unit Unit1;//Главная форма
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, StdCtrls, Menus,ShellAPI,MMSystem;
type
  TForm1 = class(TForm)
    Start: TButton;
    Exit: TButton;
    MainMenu: TMainMenu;
    C1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    FonMenu: TImage;
    N1: TMenuItem;
    procedure ExitClick(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure N3Click(Sender: TObject);
    procedure StartClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
uses Unit3;
{$R *.dfm}
procedure TForm1.ExitClick(Sender: TObject); //Выход из программы
begin
  Close;
end;
procedure TForm1.N2Click(Sender: TObject);//Открытие о разработчике
var developer:string;
    p:PChar;
begin
  developer:='О разработчике\Project1.exe';
  p:=PChar(developer);
  ShellExecute(Form1.Handle,'Open',p,nil,nil,SW_SHOW);
end;
procedure TForm1.N3Click(Sender: TObject);// Открытие о программе
var programm:string;
    p:PChar;
begin
  programm:='О программе\Project1.exe';
  p:=PChar(programm);
  ShellExecute(Form1.Handle,'Open',p,nil,nil,SW_SHOW);
end;
procedure TForm1.StartClick(Sender: TObject);// Переход на форму авторизации
begin
  Form3.Show;
  hide;
end;
procedure TForm1.FormCreate(Sender: TObject);// Запуск музыки в проекте
begin
  PlaySound(PChar('Музыка\Фоновая музыка.wav'),0,SND_ASYNC);
end;
procedure TForm1.N1Click(Sender: TObject);// Открытие справки

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

```

begin
  ShellEXeCUTE(HANDLE,'open','Справка\Справка.chm',nil,nil,sw_shownormal);
end;
end.
unit Unit2; //Форма заставки
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, jpeg;
type
  TForm2 = class(TForm)
    FonZagruzki: TImage;
    TimerZagruzki: TTimer;
    TextZagruzki: TImage;
    procedure TimerZagruzkiTimer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form2: TForm2;
implementation
{$R *.dfm}
procedure TForm2.TimerZagruzkiTimer(Sender: TObject); //Настройка таймера загрузки
begin
  TimerZagruzki.Enabled:=False;
end;
end.
unit Unit3; //Форма авторизации
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, StdCtrls, DB, ADODB;
type
  TForm3 = class(TForm)
    Fon_registr: TImage;
    BRegistr: TButton;
    BVxod: TButton;
    EName: TEdit;
    EPassword: TEdit;
    LName: TLabel;
    LPassword: TLabel;
    ADOConnection: TADOConnection;
    ADODataset: TADODataset;
    DataSource: TDataSource;
    Proverka: TLabel;
    TExit: TButton;
    procedure BRegistrClick(Sender: TObject);
    procedure BVxodClick(Sender: TObject);
    procedure TExitClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form3: TForm3;
  UserID: integer;
implementation

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

40

```

uses Unit4, Unit5, Unit1;
{$R *.dfm}
procedure TForm3.BRegistrClick(Sender: TObject);// Переход на форму регистрации
begin
    Form4.Show;
    Form3.Close;
end;
procedure TForm3.BVxodClick(Sender: TObject);// Проверка есть ли в базе данных данный пользова-
тель и переход на форму меню
var LogPassword:array of string;
I,J,count:Integer;
Login,Password,ResultLogPass:string;
begin
    SetLength(LogPassword,800);
    ADODataset.Active:=True;
    ADODataset.First;
    for i:=0 to ADODataset.RecordCount-1 do
    begin
        LogPassword[i]:=ADODataset.FieldValues['UserName'] + '+' ADODataset.FieldValues['UserPassword'];
        ADODataset.Next;
        Inc(count);
    end;
    ResultLogPass:=Trim(ENAME.Text)+'+Trim(EPASSWORD.Text);
    for j:=0 to count-1 do
    begin
        if ResultLogPass = LogPassword[j] then
        begin
            UserID:=j;
            hide;
            Form5.Show;
        end
        else
            Proverka.caption:='Неверный логин или пароль'
        end;
    end;
procedure TForm3.TExitClick(Sender: TObject);// Переход на главную форму
begin
    Form3.Hide;
    Form1.Show;
end;
end.
unit Unit4;//Форма регистрации
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, jpeg, ExtCtrls, ComCtrls, DB, ADODB;
type
    TForm4 = class(TForm)
        ENicName: TEdit;
        EPassword: TEdit;
        EName: TEdit;
        EFamily: TEdit;
        EOtchestvo: TEdit;
        FonRegistr: TImage;
        LNicName: TLabel;
        LPassword: TLabel;
        LFamily: TLabel;
        Label2: TLabel;
        LOtchestvo: TLabel;
        CBPol: TComboBox;

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

```

DTPBirthday: TDateTimePicker;
BClose: TButton;
BRegistr: TButton;
LBirthday: TLabel;
ADOQuery: TADOQuery;
procedure BCloseClick(Sender: TObject);
procedure BRegistrClick(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form4: TForm4;
implementation
uses Unit3;
{$R *.dfm}
procedure TForm4.BCloseClick(Sender: TObject);// Переход на форму авторизации
begin
Form3.Show;
Form4.Hide;
end;
procedure TForm4.BRegistrClick(Sender: TObject);// Занесение данных о пользователе в Базу Данных
begin
if (ENic-
Name.Text<>")and(EPasswd.text<>")and(EFamily.Text<>")and(ENAME.text<>")and(EOtchestvo.Text<>")and(C
BPol.ItemIndex<>-1)and(DTPBirthday.Date<>0) then
begin
ADOQuery.Parameters.ParamByName('Username').Value:=ENicName.Text;
ADOQuery.Parameters.ParamByName('Userpassword').Value:=EPasswd.Text;
ADOQuery.Parameters.ParamByName('UserNamePersonal').Value:=ENAME.Text;
ADOQuery.Parameters.ParamByName('Userfamilypersonal').Value:=EFamily.Text;
ADOQuery.Parameters.ParamByName('Userlastpersonal').Value:=EOtchestvo.Text;
ADOQuery.Parameters.ParamByName('Usersexpersonal').Value:=CBPol.ItemIndex;
ADOQuery.Parameters.ParamByName('Userbirthdaypersonal').Value:=DTPBirthday.Date;
ADOQuery.ExecSQL;
ShowMessage('Регистрация прошла успешно. ');
Form4.Hide;
Form3.Show;
end
else
ShowMessage('Вы не зарегистрированы! Пожалуйста заполните все поля!');
end;
end.
unit Unit5; //Форма меню
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, jpeg, ExtCtrls, StdCtrls, DB, ADODB, DBCtrls, ShellAPI;
type
TForm5 = class(TForm)
Fon_programms: TImage;
Aconomiuy: TButton;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
ADOQuery1: TADOQuery;

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ



```

DataSource1: TDataSource;
dbtxtUserNamePersonal: TDBText;
dbtxtUserFamilyPersonal: TDBText;
dbtxtUserLastPersonal: TDBText;
dbtxtUserBirthdayPersonal: TDBText;
dbtxtUserSexPersonal: TDBText;
Button1: TButton;
Button2: TButton;
Button3: TButton;
Button6: TButton;
BExit: TButton;
Button7: TButton;
Button4: TButton;
procedure FormShow(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure BExitClick(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure AconomiuyClick(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form5: TForm5;
implementation
uses Unit3, Unit1, Unit6, Unit7;
{$R *.dfm}
procedure TForm5.FormShow(Sender: TObject);// Отображение информации из БД на форме меню
begin
ADOQuery1.Active:=False;
ADOQuery1.Parameters.ParamByName('UserInt').Value:=Unit3.UserID+1;
ADOQuery1.ExecSQL;
ADOQuery1.Active:=True;
if dbtxtUserSexPersonal.Caption='0' then
begin
Label6.Caption:='Мужской';
end
else
begin
Label6.Caption:='Женский';
end;
end;
procedure TForm5.Button1Click(Sender: TObject);// Запуск игры «Копейка в копейку»
var game:string;
    p:PChar;
begin
    game:='Игра Магазин\SuperMarket.exe';
    p:=PChar(game);
    ShellExecute(Form5.Handle,'Open',p,nil,nil,SW_SHOW);
end;
procedure TForm5.BExitClick(Sender: TObject);// Выход из профиля и переход на главную форму
begin
Form5.Hide;
Form1.Show;
Form3.ENAME.Text:="";

```

Изм.	Коп.	Лист	№ док.	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

```

Form3.EPassword.Text:="";
end;
procedure TForm5.Button6Click(Sender: TObject);// Переход на форму информации и загрузка html-
страницы
begin
form6.WebBrowser1.Navigate(ExtractFilePath(ParamStr(0))+ 'Информация\Категория продуктов.htm');
Form6.Show;
Form5.Hide;
end;
procedure TForm5.Button7Click(Sender: TObject);// Запуск игры «Студенческий кошелёк»
var game1:string;
    p:PChar;
begin
    game1:='Игра Студенческий кошелёк\game.exe';
    p:=PChar(game1);
    ShellExecute(Form5.Handle,'Open',p,nil,nil,SW_SHOW);
end;
procedure TForm5.AconomiuyClick(Sender: TObject);// Переход на форму информации и загрузка html-
страницы
begin
form6.WebBrowser1.Navigate(ExtractFilePath(ParamStr(0))+ 'Информация\Бюджет Студента.htm');
Form6.Show;
Form5.Hide;
end;
procedure TForm5.Button2Click(Sender: TObject);// Переход на форму информации и загрузка html-
страницы
begin
form6.WebBrowser1.Navigate(ExtractFilePath(ParamStr(0))+ 'Информация\Создание Бюджета.htm');
Form6.Show;
Form5.Hide;
end;
procedure TForm5.Button3Click(Sender: TObject);// Переход на форму информации и загрузка html-
страницы
begin
    form6.WebBrowser1.Navigate(ExtractFilePath(ParamStr(0))+ 'Информация\Мелкие советы.htm');
Form6.Show;
Form5.Hide;
end;
procedure TForm5.Button4Click(Sender: TObject);// Переод на форму видео
begin
Form5.Hide;
form7.show;
end;
end.
unit Unit6;// Форма информации
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, OleCtrls, SHDocVw;
type
    TForm6 = class(TForm)
        WebBrowser1: TWebBrowser;
        Button1: TButton;
        procedure Button1Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

44

```

Form6: TForm6;
implementation
uses Unit5;
{$R *.dfm}
procedure TForm6.Button1Click(Sender: TObject);// Переход на форму меню
begin
Form5.Show;
Hide;
end;
end.
unit Unit7;//Форма видео
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, Mplayer, StdCtrls,MMSystem;
type
Tform7 = class(Tform)
MediaPlayer1: TmediaPlayer;
Panel1: Tpanel;
Button1: Tbutton;
Button2: Tbutton;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form7: Tform7;
implementation
uses Unit1, Unit5;
{$R *.dfm}
Procedure Tform7.Button1Click(Sender: TObject);// Подзагрузка и запуск видео, ставит на паузу музыку
begin
MediaPlayer1.FileName:='КАК-ТРАТИТЬ-МЕНЬШЕ_ЕСЛИ-ТЫ-СТУДЕНТ.wmv';
MediaPlayer1.Display:=Panel1;
MediaPlayer1.Open;
MediaPlayer1.Play;
PlaySound(nil,0,SND_ASYNC);
end;
procedure TForm7.Button2Click(Sender: TObject);// Переход на форму меню и запуск музыки
begin
Form7.Hide;
PlaySound(PChar('Музыка\Фоновая музыка.wav'),0,SND_ASYNC);
Form5.Show;
end;
end.
unit Unit1;//Программа «Копейка в копейку»
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, jpeg, ExtCtrls, StdCtrls, Menus;
type
TForm1 = class(TForm)
Fon: TImage;
Panel1: TPanel;
Label1: TLabel;
LBudget: TLabel;
Label2: TLabel;

```

						КП 2-40 01 01 01.35.32.06.21 ПЗ	Лист
Изм.	Коп.	Лист	№ док	Подпись	Дата		45

Label3: TLabel;  
 LZatrat: TLabel;  
 Label4: TLabel;  
 Image1: TImage;  
 Label5: TLabel;  
 Image2: TImage;  
 Label6: TLabel;  
 Image3: TImage;  
 Label7: TLabel;  
 Image4: TImage;  
 Label8: TLabel;  
 Image5: TImage;  
 Label9: TLabel;  
 Image6: TImage;  
 Label10: TLabel;  
 Image7: TImage;  
 Label11: TLabel;  
 Image8: TImage;  
 Label12: TLabel;  
 Image9: TImage;  
 Label13: TLabel;  
 Image10: TImage;  
 Label14: TLabel;  
 Image11: TImage;  
 Label15: TLabel;  
 Image12: TImage;  
 Label16: TLabel;  
 Image13: TImage;  
 Label17: TLabel;  
 Image14: TImage;  
 Label18: TLabel;  
 Image15: TImage;  
 Label19: TLabel;  
 Image16: TImage;  
 Label20: TLabel;  
 Image17: TImage;  
 Label21: TLabel;  
 Image18: TImage;  
 Label22: TLabel;  
 Image19: TImage;  
 Label23: TLabel;  
 Image20: TImage;  
 Label24: TLabel;  
 Image21: TImage;  
 Label25: TLabel;  
 Image22: TImage;  
 Label26: TLabel;  
 Image23: TImage;  
 Label27: TLabel;  
 Image24: TImage;  
 Label28: TLabel;  
 Image25: TImage;  
 Label29: TLabel;  
 Image26: TImage;  
 Label30: TLabel;  
 Image27: TImage;  
 Label31: TLabel;  
 Image28: TImage;  
 Label32: TLabel;  
 Image29: TImage;

						КП 2-40 01 01 01.35.32.06.21 ПЗ	Лист
Изм.	Коп.	Лист	№ док	Подпись	Дата		46

Label33: TLabel;  
 Image30: TImage;  
 Label34: TLabel;  
 Image31: TImage;  
 Label35: TLabel;  
 Image32: TImage;  
 Label36: TLabel;  
 Image33: TImage;  
 Label37: TLabel;  
 Image34: TImage;  
 Label38: TLabel;  
 Image35: TImage;  
 Label39: TLabel;  
 Image36: TImage;  
 Label40: TLabel;  
 Image37: TImage;  
 Label41: TLabel;  
 Image38: TImage;  
 Label42: TLabel;  
 Image39: TImage;  
 Label43: TLabel;  
 Image40: TImage;  
 Label44: TLabel;  
 Image41: TImage;  
 Label45: TLabel;  
 Image42: TImage;  
 Label46: TLabel;  
 Image43: TImage;  
 Label47: TLabel;  
 Image44: TImage;  
 Label48: TLabel;  
 Image45: TImage;  
 Label49: TLabel;  
 Image46: TImage;  
 Label50: TLabel;  
 Image47: TImage;  
 Label51: TLabel;  
 Image48: TImage;  
 Label52: TLabel;  
 CheckBox1: TCheckBox;  
 CheckBox2: TCheckBox;  
 CheckBox3: TCheckBox;  
 CheckBox4: TCheckBox;  
 CheckBox5: TCheckBox;  
 CheckBox6: TCheckBox;  
 CheckBox7: TCheckBox;  
 CheckBox8: TCheckBox;  
 CheckBox9: TCheckBox;  
 CheckBox10: TCheckBox;  
 CheckBox11: TCheckBox;  
 CheckBox12: TCheckBox;  
 CheckBox13: TCheckBox;  
 CheckBox14: TCheckBox;  
 CheckBox15: TCheckBox;  
 CheckBox16: TCheckBox;  
 CheckBox17: TCheckBox;  
 CheckBox18: TCheckBox;  
 CheckBox19: TCheckBox;  
 CheckBox20: TCheckBox;  
 CheckBox21: TCheckBox;

						КП 2-40 01 01 01.35.32.06.21 ПЗ	Лист
Изм.	Коп.	Лист	№ док.	Подпись	Дата		47

```

CheckBox22: TCheckBox;
CheckBox23: TCheckBox;
CheckBox24: TCheckBox;
CheckBox25: TCheckBox;
CheckBox26: TCheckBox;
CheckBox27: TCheckBox;
CheckBox28: TCheckBox;
CheckBox29: TCheckBox;
CheckBox30: TCheckBox;
CheckBox31: TCheckBox;
CheckBox32: TCheckBox;
CheckBox33: TCheckBox;
CheckBox34: TCheckBox;
CheckBox35: TCheckBox;
CheckBox36: TCheckBox;
CheckBox37: TCheckBox;
CheckBox38: TCheckBox;
CheckBox39: TCheckBox;
CheckBox40: TCheckBox;
CheckBox41: TCheckBox;
CheckBox42: TCheckBox;
CheckBox43: TCheckBox;
CheckBox44: TCheckBox;
CheckBox45: TCheckBox;
CheckBox46: TCheckBox;
CheckBox47: TCheckBox;
CheckBox48: TCheckBox;
Button1: TButton;
Label53: TLabel;
Number: TLabel;
Button2: TButton;
Button3: TButton;
LTime: TLabel;
Timer1: TTimer;
MainMenu1: TMainMenu;
N1: TMenuItem;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure N1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
uses Unit2;
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);// Проверка выбранного товара, подсчет суммы и
кол-во товаров
var summa:Double;
    n:Integer;
begin
  n:=0;
  summa:=0;
  if CheckBox1.Checked then

```

Изм.	Коп.	Лист	№ док.	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

```

begin
summa:=summa+2.76;
n:=n+1;
end;
if CheckBox2.Checked then
begin
summa:=summa+2.54;
n:=n+1;
end;
if CheckBox3.Checked then
begin
summa:=summa+2.15;
n:=n+1;
end;
if CheckBox4.Checked then
begin
summa:=summa+3.11;
n:=n+1;
end;
if CheckBox5.Checked then
begin
summa:=summa+2.26;
n:=n+1;
end;
if CheckBox6.Checked then
begin
summa:=summa+0.40;
n:=n+1;
end;
if CheckBox7.Checked then
begin
summa:=summa+25;
n:=n+1;
end;
if CheckBox8.Checked then
begin
summa:=summa+0.70;
n:=n+1;
end;
if CheckBox9.Checked then
begin
summa:=summa+0.87;
n:=n+1;
end;
if CheckBox10.Checked then
begin
summa:=summa+1.25;
n:=n+1;
end;
if CheckBox11.Checked then
begin
summa:=summa+0.93;
n:=n+1;
end;
if CheckBox12.Checked then
begin
summa:=summa+0.98;
n:=n+1;
end;
if CheckBox13.Checked then

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

49

```

begin
summa:=summa+0.56;
n:=n+1;
end;
if CheckBox14.Checked then
begin
summa:=summa+3.12;
n:=n+1;
end;
if CheckBox15.Checked then
begin
summa:=summa+5;
n:=n+1;
end;
if CheckBox16.Checked then
begin
summa:=summa+1.46;
n:=n+1;
end;
if CheckBox17.Checked then
begin
summa:=summa+2.19;
n:=n+1;
end;
if CheckBox18.Checked then
begin
summa:=summa+1.76;
n:=n+1;
end;
if CheckBox19.Checked then
begin
summa:=summa+3.27;
n:=n+1;
end;
if CheckBox20.Checked then
begin
summa:=summa+1.79;
n:=n+1;
end;
if CheckBox21.Checked then
begin
summa:=summa+1.98;
n:=n+1;
end;
if CheckBox22.Checked then
begin
summa:=summa+0.86;
n:=n+1;
end;
if CheckBox23.Checked then
begin
summa:=summa+2.69;
n:=n+1;
end;
if CheckBox24.Checked then
begin
summa:=summa+5.73;
n:=n+1;
end;
if CheckBox25.Checked then

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

50



```

begin
summa:=summa+1.63;
n:=n+1;
end;
if CheckBox26.Checked then
begin
summa:=summa+0.54;
n:=n+1;
end;
if CheckBox27.Checked then
begin
summa:=summa+3.24;
n:=n+1;
end;
if CheckBox28.Checked then
begin
summa:=summa+1.38;
n:=n+1;
end;
if CheckBox29.Checked then
begin
summa:=summa+1.25;
n:=n+1;
end;
if CheckBox30.Checked then
begin
summa:=summa+2.30;
n:=n+1;
end;
if CheckBox31.Checked then
begin
summa:=summa+3;
n:=n+1;
end;
if CheckBox32.Checked then
begin
summa:=summa+2.59;
n:=n+1;
end;
if CheckBox33.Checked then
begin
summa:=summa+1.27;
n:=n+1;
end;
if CheckBox34.Checked then
begin
summa:=summa+3.20;
n:=n+1;
end;
if CheckBox35.Checked then
begin
summa:=summa+1.57;
n:=n+1;
end;
if CheckBox36.Checked then
begin
summa:=summa+1.34;
n:=n+1;
end;
if CheckBox37.Checked then

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

51

```

begin
summa:=summa+2.49;
n:=n+1;
end;
if CheckBox38.Checked then
begin
summa:=summa+2.78;
n:=n+1;
end;
if CheckBox39.Checked then
begin
summa:=summa+2.34;
n:=n+1;
end;
if CheckBox40.Checked then
begin
summa:=summa+0.97;
n:=n+1;
end;
if CheckBox41.Checked then
begin
summa:=summa+5.69;
n:=n+1;
end;
if CheckBox42.Checked then
begin
summa:=summa+11.87;
n:=n+1;
end;
if CheckBox43.Checked then
begin
summa:=summa+9.34;
n:=n+1;
end;
if CheckBox44.Checked then
begin
summa:=summa+3.12;
n:=n+1;
end;
if CheckBox45.Checked then
begin
summa:=summa+2.34;
n:=n+1;
end;
if CheckBox46.Checked then
begin
summa:=summa+0.57;
n:=n+1;
end;
if CheckBox47.Checked then
begin
summa:=summa+1.79;
n:=n+1;
end;
if CheckBox48.Checked then
begin
summa:=summa+1.09;
n:=n+1;
end;
LZatrat.Caption:=FloatToStr(summa);

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

```

Number.Caption:=IntToStr(n)
end;
procedure TForm1.Button2Click(Sender: TObject);// Проверка главного условия игры
begin
if StrToFloat(LZatrat.Caption)<StrToFloat(LBudget.Caption) then
  if (CheckBox9.Checked or CheckBox10.Checked or CheckBox12.Checked or CheckBox13.Checked or Check-
Box16.Checked) and (CheckBox1.Checked or CheckBox2.Checked or CheckBox3.Checked or Check-
Box4.Checked or CheckBox5.Checked or CheckBox6.Checked or CheckBox8.Checked or CheckBox22.Checked or
CheckBox23.Checked or CheckBox30.Checked or CheckBox40.Checked) and (CheckBox7.Checked or Check-
Box24.Checked or CheckBox31.Checked) and (CheckBox19.Checked or CheckBox17.Checked or Check-
Box20.Checked or CheckBox21.Checked) and ( CheckBox18.Checked or CheckBox15.Checked or Check-
Box11.Checked or CheckBox25.Checked or CheckBox26.Checked or CheckBox27.Checked or Check-
Box28.Checked or CheckBox46.Checked or CheckBox47.Checked or CheckBox48.Checked) and (Check-
Box33.Checked or CheckBox34.Checked or CheckBox35.Checked or CheckBox36.Checked or Check-
Box37.Checked or CheckBox38.Checked or CheckBox39.Checked) and (CheckBox41.Checked or Check-
Box42.Checked or CheckBox43.Checked or CheckBox44.Checked) and CheckBox45.Checked then
  begin
    ShowMessage('Ты выиграл.')
  end
else
  ShowMessage('Ты проиграл. Молодец ты вложился в бюджет, но не смог выбрать товары из всех групп.')
end;
ShowMessage('Вы проиграли. Так как потратили денег больше чем ваш бюджет.');
```

```

end;
procedure TForm1.Button3Click(Sender: TObject);// Выход из игры
begin
Hide;
end;
procedure TForm1.Timer1Timer(Sender: TObject);// Отображение реального времени
begin
LTime.Caption:=TimeToStr(Time);
end;
procedure TForm1.N1Click(Sender: TObject);Открытие формы инструкции и подгрузка html-
странички
begin
Form2.WebBrowser1.Navigate(ExtractFilePath(ParamStr(0))+ 'Инструкция к игре\Инструкция.htm');
Form2.Show;
end;
end.
unit Unit1;// Программа «Студенческий кошелёк»
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, jpeg, Menus;
type
  TForm1 = class(TForm)
    Fon: TImage;
    Panel1: TPanel;
    SRBall: TEdit;
    Step: TButton;
    Label1: TLabel;
    LBudget: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    LZatrat: TLabel;
    Label4: TLabel;
    MainMenu1: TMainMenu;
    B1: TMenuItem;
    Image1: TImage;
    Obschaga: TCheckBox;

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

```

Button1: TButton;
Edit1: TEdit;
Image2: TImage;
Magazin: TCheckBox;
EMagazin: TEdit;
Image3: TImage;
ETransport: TCheckBox;
Edit2: TEdit;
Image4: TImage;
ComboBox1: TComboBox;
edt1: TEdit;
Help: TButton;
Image5: TImage;
Koncelur: TCheckBox;
Edit3: TEdit;
Image6: TImage;
CheckBox1: TCheckBox;
Image7: TImage;
Club: TCheckBox;
Edit5: TEdit;
Image8: TImage;
Otdeh: TCheckBox;
Edit6: TEdit;
Edit7: TEdit;
Edit8: TEdit;
Edit9: TEdit;
Edit10: TEdit;
Image9: TImage;
Promtovar: TCheckBox;
Edit4: TEdit;
Edit14: TEdit;
Edit15: TEdit;
Edit16: TEdit;
N1: TMenuItem;
procedure StepClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure HelpClick(Sender: TObject);
procedure SRBallClick(Sender: TObject);
procedure Edit4Click(Sender: TObject);
procedure Edit14Click(Sender: TObject);
procedure Edit15Click(Sender: TObject);
procedure Edit16Click(Sender: TObject);
procedure B1Click(Sender: TObject);
procedure N1Click(Sender: TObject);
procedure MagazinClick(Sender: TObject);
procedure ETransportClick(Sender: TObject);
procedure KoncelurClick(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure ClubClick(Sender: TObject);
procedure OtdehClick(Sender: TObject);
procedure PromtovarClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
  summa: Double;
implementation

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

54

```

uses Unit2;
{$R *.dfm}
procedure TForm1.StepClick(Sender: TObject);// Проверка стипендии и начисление бюджета
var ball:double;
begin
    ball:=StrToFloat(SRBall.Text);
    if ball<5.0 then
        Lbudget.Caption:='45';
    if (ball>=5.0) and(ball<6.0) then
        Lbudget.Caption:='56,54';
    if (ball>=6.0) and(ball<8.0) then
        Lbudget.Caption:='89,23';
    if (ball>=8.0) and(ball<9.0) then
        Lbudget.Caption:='102,31';
    if (ball>=9.0) and(ball<10.0) then
        Lbudget.Caption:='113,69';
end;

procedure TForm1.Button1Click(Sender: TObject);// Проверка выбранного вида затрат и подсчет потраченной суммы
begin
    summa:=0;
    if Obschaga.Checked then
        summa:=summa+StrToFloat(Edit1.Text);
    if Magazin.Checked then
        begin
            summa:=summa+StrToFloat(EMagazin.Text);
            summa:=summa+StrToFloat(Edit7.Text);
            summa:=summa+StrToFloat(Edit8.Text);
            summa:=summa+StrToFloat(Edit9.Text);
        end;
    if ETransport.Checked then
        summa:=summa+StrToFloat(Edit2.Text);
    if Koncelur.Checked then
        summa:=summa+StrToFloat(Edit3.Text);
    if CheckBox1.Checked then
        begin
            summa:=summa+StrToFloat(Edit10.Text);
        end;
    if Club.Checked then
        summa:=summa+StrToFloat(Edit5.Text);
    if Otdeh.Checked then
        summa:=summa+StrToFloat(Edit6.Text);
    if Promtovar.Checked then
        begin
            summa:=summa+StrToFloat(Edit4.Text);
            summa:=summa+StrtoFloat(Edit14.Text);
            summa:=summa+StrToFloat(Edit15.Text);
            summa:=summa+StrToFloat(Edit16.Text);
        end;
    LZatrat.Caption:= FloatToStr(summa);
    if StrToFloat(LBudget.Caption)<StrToFloat(LZatrat.Caption)then
        ShowMessage('Ты не правильно распределил бюджет. Нужно что-то сократить и пересмотреть свои затраты! => ');
end;

procedure TForm1.HelpClick(Sender: TObject);// Получение денег от кого-то из родственников
begin
    if ComboBox1.ItemIndex = 0 then
        begin
            LBudget.Caption:=FloatToStr(StrToFloat(LBudget.Caption)+StrToFloat(edt1.Text));
        end;
    end;

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

```

    ShowMessage('Вот денюжка.Ты главное не голодай и трать с умом.');
```

end;

```

    if ComboBox1.ItemIndex = 1 then
    begin
        LBudget.Caption:=FloatToStr(StrToFloat(LBudget.Caption)+StrToFloat(edt1.Text));
        ShowMessage('Мы что с мамой их печатаем!');
```

end;

```

    if ComboBox1.ItemIndex = 2 then
    begin
        LBudget.Caption:=FloatToStr(StrToFloat(LBudget.Caption)+StrToFloat(edt1.Text));
        ShowMessage('Через маму отправила. Может тебе надо привезти еду, мы с бабушкой выезжаем!');
```

end;

```

    if ComboBox1.ItemIndex = 3 then
    begin
        ShowMessage('Вся пенсия у бабушки звони ей!');
```

end;

```

    if ComboBox1.ItemIndex = 4 then
    begin
        ShowMessage('У самого мышь в кошельке повесилась. Жду степендии!');
```

end;

```

    if ComboBox1.ItemIndex = 5 then
    begin
        LBudget.Caption:=FloatToStr(StrToFloat(LBudget.Caption)+StrToFloat(edt1.Text));
        ShowMessage('Работать иди уже, а не на моей с родителями шее сидишь!');
```

end;

```

    if ComboBox1.ItemIndex = 6 then
    begin
        LBudget.Caption:=FloatToStr(StrToFloat(LBudget.Caption)+StrToFloat(edt1.Text));
        ShowMessage('Работать иди уже, а не на моей с родителями шее сидишь!');
```

end;

end;

```

procedure TForm1.SRBallClick(Sender: TObject);// Очищение Edit
begin
    SRBall.Text:="";
end;
```

```

procedure TForm1.Edit4Click(Sender: TObject); // Очищение Edit
begin
    Edit4.Text:="";
end;
```

```

procedure TForm1.Edit14Click(Sender: TObject); // Очищение Edit
begin
    Edit14.Text:="";
end;
```

```

procedure TForm1.Edit15Click(Sender: TObject); // Очищение Edit
begin
    Edit15.Text:="";
end;
```

```

procedure TForm1.Edit16Click(Sender: TObject); // Очищение Edit
begin
    Edit16.Text:="";
end;
```

```

procedure TForm1.B1Click(Sender: TObject);//Открытие формы инструкции и подзагрузка html-
страницы
begin
    form2.WebBrowser1.Navigate(ExtractFilePath(ParamStr(0))+'\Инструкция к игре\Инструкция.htm');
    Form2.Show;
end;
```

```

procedure TForm1.N1Click(Sender: TObject);//Закрытие игры «Студенческий кошелёк»
begin
    Close;
```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ

Лист

56

```

end;
procedure TForm1.MagazinClick(Sender: TObject);// При выборе автоматически заполняет поля ну-
лями
begin
EMagazin.Text:='0,0';
Edit7.Text:='0,0';
Edit8.Text:='0,0';
Edit9.Text:='0,0';
end;
procedure TForm1.ETransportClick(Sender: TObject); // При выборе автоматически заполняет поля
нулями
begin
Edit2.Text:='0,0';
end;
procedure TForm1.KoncelurClick(Sender: TObject); // При выборе автоматически заполняет поля ну-
лями
begin
Edit3.Text:='0,0';
end;
procedure TForm1.CheckBox1Click(Sender: TObject); // При выборе автоматически заполняет поля ну-
лями
begin
Edit10.Text:='0,0';
end;
procedure TForm1.ClubClick(Sender: TObject); // При выборе автоматически заполняет поля нулями
begin
Edit5.Text:='0,0';
end;
procedure TForm1.OtdehClick(Sender: TObject); // При выборе автоматически заполняет поля нулями
begin
Edit6.Text:='0,0';
end;
procedure TForm1.PromptovarClick(Sender: TObject); // При выборе автоматически заполняет поля ну-
лями
begin
Edit4.Text:='0,0';
Edit14.Text:='0,0';
Edit15.Text:='0,0';
Edit16.Text:='0,0';
end;
end.

```

Изм.	Коп.	Лист	№ док	Подпись	Дата

КП 2-40 01 01 01.35.32.06.21 ПЗ