Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Институт информационных технологий

Кафедра «Микропроцессорные системы и сети»

К защите допустить

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_ *В.А. Полубок*

«\_\_\_»\_\_\_\_\_\_\_\_\_2016 г.

**Пояснительная записка**

**К дипломному проекту**

**На тему:**

**«Веб-ориентированное приложение  
«Справочник банков и банковских услуг для физических лиц»»**

Дипломник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.С. Мячков

(дата подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.И. Гламаздин

(дата подпись)

Консультанты:  
по ЕСПД и ЕСКД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Н. Мухаметов

(дата подпись)

по специальности \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Москалев

(дата подпись)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата подпись)

Минск 2016

Содержание

[Введение 4](#_Toc451347898)

[1 Анализ существующих решений получения справочной информации при выборе розничных банковских продуктов и услуг, постановка задачи 7](#_Toc451347899)

[1.1 Характеристика банковских продуктов и услуг, варианты получения справочной информации о них 7](#_Toc451347900)

[1.2 Анализ существующих приложений для получения справочной информации о банках, их продуктах и услугах 12](#_Toc451347901)

[1.3 Требования к разрабатываемой системе и постановка задачи 19](#_Toc451347902)

[2 Методы и модели, положенные в основу проекта 20](#_Toc451347903)

[2.1 Методы определения дохода по вкладу (депозиту) 20](#_Toc451347904)

[2.2 Методы расчета процентов по кредиту 21](#_Toc451347905)

[3 Разработка проекта программного продукта 24](#_Toc451347906)

[3.1 Функциональный анализ предметной области 24](#_Toc451347907)

[3.2 Выбор архитектуры приложения, языка программирования и системы управления базой данных 29](#_Toc451347908)

[3.3 Разработка модели данных 32](#_Toc451347909)

[4 Разработка алгоритмов и их программная реализация 37](#_Toc451347910)

[4.1 Карта сайта, меню разрабатываемого веб-приложения 37](#_Toc451347911)

[4.2 Реализация функционала проекта, программирование алгоритмов 38](#_Toc451347912)

[4.2.1 Отображение справочной информации о банках и розничных продуктах 38](#_Toc451347913)

[4.2.2. Отображение информации о курсах обмена валют 39](#_Toc451347914)

[4.2.3 Отображение информации о структурных подразделениях банка 41](#_Toc451347915)

[4.2.4 Расчет дохода по вкладам (депозитам) по заданным условиям 41](#_Toc451347916)

[4.2.5 Расчет платежей по кредиту по заданным условиям 42](#_Toc451347917)

[5 Тестирование разработанного программного продукта 43](#_Toc451347918)

[5.1 Выбор методов тестирования 43](#_Toc451347919)

[5.2 Разработанные тесты 44](#_Toc451347920)

[5.3 Результаты тестирования 48](#_Toc451347921)

[Заключение 49](#_Toc451347922)

[Список использованных источников 51](#_Toc451347923)

[ПРИЛОЖЕНИЕ А Текст программы 53](#_Toc451347924)

# Введение

На современном этапе развития общества финансовые услуги, банковские инструменты, иностранная валюта, платежные карты, депозиты, кредиты тесно связаны с повседневной жизнью каждого человека. Сегодня на финансовом рынке Беларуси представлено около трех десятков банков, а также большое количество небанковских кредитно-финансовых учреждений. Все они предоставляют широкий спектр розничных продуктов и услуг для потребителя. Однако, большое разнообразие предложений, привлекательные рекламные слоганы, большое количество специальных терминов, частые изменения в законодательстве и иные факторы не позволяют рядовому обывателю без специального образования или особых знаний свободно ориентироваться в этом многообразии.

В сентябре 2015 было проведено исследование оценки финансовой доступности в Беларуси [3]. Как показало исследование, большинство граждан страны являются потребителями лишь базовых финансовых услуг, среди которых наиболее распространенными являются текущий банковский счет и дебетовая карта. В группу наиболее востребованных услуг также вошли интернет-банкинг и страхование автотранспортных средств.

Среди услуг кредитования лидируют кредиты на потребительские нужды. Для сбережений и инвестирования белорусы чаще всего пользуются сберегательными счетами и срочными депозитами.

Общий индекс финансовой доступности (общий уровень охвата финансовыми услугами) в 2015 году составил 90,3%. Доля населения, полностью либо практически полностью исключенная из финансового рынка страны, остается достаточно большой – 9,7% граждан вообще не используют финансовые услуги, 22,6% являются пользователями базового уровня.

Уровень использования финансовых продуктов и услуг зависит от социально-демографических характеристик, уровня образования, социального положением, наличия постоянной работы и стабильного дохода, общего уровня финансовой грамотности потребителей.

Ранее Институтом социологии Национальной академии наук Беларуси был проведен опрос граждан Беларуси по теме: «Финансовая грамотность населения Республики Беларусь» [4].

Согласно результатам проведенного исследования, портрет белорусского пользователя финансовых услуг выглядит примерно следующим образом. Типичный пользователь финансовых услуг плохо ориентируется в финансовых продуктах, пользуется только несколькими основными услугами (оплата коммунальных платежей, обмен валют, платежи через платежные терминалы, кредиты и текущий банковский счет), предоставляемыми финансовым сектором, и слабо информирован о механизмах защиты своих прав в финансовой сфере.

Как правило, респонденты выше оценивали свой уровень владения финансовыми вопросами по сравнению с тем, насколько на деле были способны ориентироваться в финансовых вопросах. Так, около 60% опрошенных охарактеризовали свой уровень финансовой грамотности как «удовлетворительный», «хороший» либо «отличный». В то же время, ответы на тестовые вопросы по финансовой грамотности показали, что в реальности таких среди опрошенных оказалось чуть более 40% согласно тесту по финансовой математике и около 35%, если судить по тесту на финансовую осведомленность. Практически на каждый из тестовых вопросов в области финансов более половины респондентов дали неверные ответы либо затруднились ответить. Лучше всего респонденты справились с вопросами о скидках и об ответственности поручителя: на них верные ответы дало около 3/4 опрошенных. Хуже всего белорусы ориентируются в информации, касающейся доходов по облигации, страховой франшизы, ценных бумаг, гарантий банковских вкладов и сложного процента по депозиту: правильные ответы на эти вопросы смогло дать менее четверти опрошенных. Особую обеспокоенность вызывает тот факт, что белорусы довольно слабо информированы о различиях между кредитной и дебетовой платежными картами и гарантиях по банковским вкладам (соответственно 31,6% и 23,5% правильных ответов), несмотря на то, что эти виды финансовых услуг достаточно широко распространены среди белорусского населения.

Около 17% опрошенных имеют негативный опыт получения финансовых услуг. Чаще всего этот опыт связан с получением банковского вклада/депозита и потребительского кредита. Наиболее часто респонденты, у которых до получения следующего дохода оставались неистраченные деньги, предпочитали хранить их в наличной форме либо тратить на потребительские товары. Значительно реже респонденты размещали сэкономленные средства на банковский вклад. При этом почти 3/4 опрошенных имеет желание и возможность сберегать денежные средства. В качестве причины, не позволяющей сберегать деньги 8,8% опрошенных называют недоверие финансовым организациям.

Об изменениях на финансовых рынках граждане чаще всего узнают из специализированных изданий – более 45%. Около 40% получают эту информацию от друзей и знакомых или из Интернета. Около 30% опрошенных о динамике процессов, протекающих на финансовых рынках, узнают из рекламы, примерно четверть – от сотрудников банков, страховых компаний и иных финансовых организаций.

Итак, как показывают результаты проведенного исследования, граждане Беларуси отличаются сравнительно низким уровнем осведомленности в финансовых вопросах. Эту ситуацию можно назвать тревожной, учитывая, что люди, обладающие низким уровнем финансовой грамотности, как правило, демонстрируют невысокий уровень доверия финансовым институтам и значительно реже являются пользователями финансовых услуг. Принимая во внимание тот факт, что доля белорусского населения, отличающегося низким уровнем знаний и навыков в области финансов, сегодня весьма велика, становится понятно, что значительная часть белорусов на сегодняшний момент практически исключена из финансовой сферы.

Целью дипломного проекта является повышение финансовой грамотности населения. Для достижения поставленной цели необходимо разработать и создать веб-ориентированное приложение. Данное приложение должно обеспечивать доступ к справочной информации о банках Республики Беларусь, об установленных курсах обмена иностранных валют и об основных услугах для физических лиц, а также должно позволять производить расчет суммы дохода по вкладу (депозиту), расчет платежей по кредиту, расчет эквивалента при обмене валюты с помощью конвертера валют.

# 1 Анализ существующих решений получения справочной информации при выборе розничных банковских продуктов и услуг, постановка задачи

## 1.1 Характеристика банковских продуктов и услуг, варианты получения справочной информации о них

Как отмечено выше, большинство граждан страны является потребителями лишь базовых финансовых услуг, отказаться от использования которых нельзя. Например, именно обязательный характер использования услуги является для значительной части белорусского населения основной причиной пользования текущим счетом, на который начисляется заработная плата и социальные выплаты, дебетовой карточкой, которая в обязательном порядке прилагается к текущему счету. При этом необходимо сказать, что на третьем месте по распространенности находятся кредитные услуги. Среди услуг кредитования лидируют кредиты на потребительские нужды. В Национальном банке хранится больше 4,5 миллиона кредитных историй белорусов. Это около 60% взрослого населения страны. В меньшей степени белорусы заинтересованы в сберегательных продуктах. В данный сегмент финансового рынка вовлечен лишь каждый пятый респондент. Граждане, предпочитающие делать сбережения или инвестировать свои деньги, чаще всего для этого пользуются сберегательными счетами и срочными депозитами. Отдельно необходимо отметить востребованный у населения вид банковских услуг – это обмен валют.

Взаимоотношения банков с клиентами строятся на основе банковского законодательства и заключенных договоров. При этом банки самостоятельно определяют условия заключения сделок с клиентами, в том числе размер процентов по денежным средствам, привлекаемым, например, во вклады (депозиты), размер процентов за пользование кредитами, проводят свою тарифную политику относительно взимаемой платы (вознаграждения) за оказание банковских услуг. В соответствии с законодательством вкладчики и кредитополучатели свободны в выборе обслуживающего банка. В настоящее время в Республике Беларусь на рынке банковских услуг существует высокий уровень конкуренции между банками (по состоянию на 01.01.2016 право на осуществление банковских операций по привлечению денежных средств физических лиц во вклады (депозиты) имели 22 банка), что заставляет банки постоянно совершенствовать свою работу по формированию и оптимизации стоимости банковских услуг, повышению доступности их для клиентов с точки зрения как качественных, так и стоимостных параметров. Таким образом, потребитель вправе выбрать банк, спектр оказываемых услуг которого и условия по предлагаемым банковским продуктам будут для него наиболее выгодными.

Что необходимо знать об услугах банков по кредитованию? Кредит – это привлеченные или собственные денежные средства, предоставленные банком другому лицу (кредитополучателю) в размере и на условиях, предусмотренных кредитным договором.

От вида кредитования зависят основные правила их предоставления и погашения, перечень необходимых документов, которые должен представить в банк потенциальный кредитополучатель и процедура их рассмотрения банками, максимально возможная сумма кредита, величина процентной ставки и многие другие условия кредитных договоров.

Банки и небанковские финансово-кредитные учреждения осуществляют выдачу кредитов на основании Постановления Правления Национального банка Республики Беларусь № 226 от 30.12.2003 г. «О порядке предоставления (размещения) банками денежных средств в форме кредита и их возврата». Правила определяют основные принципы и подходы при кредитовании и обязательны для исполнения банками, небанковскими кредитно-финансовыми организациями и кредитополучателями.

Гражданин может получить в банке кредит на финансирование недвижимости либо кредит на потребительские нужды. Получение кредита обязывает кредитополучателя вернуть в установленные кредитным договором сроки основную сумму долга (сумму, которая была получена от банка), а также уплатить проценты за пользование кредитом. Как правило, кредит предоставляется гражданину при условии его кредитоспособности (способности в полном объеме и в срок погасить сумму долга по кредитному договору и уплатить проценты). Условия кредитования и требования к кредитоспособности банки определяют самостоятельно. Часто при кредитовании кредитным договором могут быть предусмотрены дополнительные услуги банка или третьих лиц (страховых компаний, отделений связи, органов, совершающих нотариальные действия и т.п.). Как правило, такие услуги являются платными. Поэтому необходимо обратить внимание на условия оказания дополнительных услуг и их стоимость до принятия решения о подписании кредитного договора. В случае ненадлежащего исполнения кредитополучателем условий кредитного договора, банк может требовать уплату процентов за пользование кредитом в повышенном размере, а также уплату неустойки (штрафа, пени). О праве банка предъявлять такое требование сказано в кредитном договоре. Банк обязан предоставить данную информацию до заключения кредитного договора или дополнительного соглашения к нему, связанного с изменением суммы (размера) денежных обязательств кредитополучателя и (или) срока их уплаты. Исчерпывающая информация об условиях кредитования должна быть представлена в письменной форме и содержать сведения о:

* сумме кредита с указанием валюты кредита;
* сроке, на который кредит может быть получен;
* размере процентов за пользование кредитом, порядке определения размера процентов (с использованием фиксированной либо переменной годовой процентной ставки);
* целях, на которые кредит может быть использован, если кредитный договор предусматривает условие о целевом использовании кредита;
* способах обеспечения исполнения обязательств по кредитному договору, если условиями кредитного договора предусмотрено условие об обеспечении исполнения обязательств по кредитному договору;
* сумме платежей кредитополучателя по срокам уплаты в соответствии с кредитным договором (платежей по возврату (погашению) кредита и уплате процентов за пользование им);
* ответственности сторон за неисполнение (ненадлежащее исполнение) условий кредитного договора;
* возможности и условиях досрочного погашения кредита;
* иных условиях предоставления и погашения (возврата) кредита, а также уплаты процентов за пользование им.

Сроки уплаты платежей по кредиту могут быть указаны в виде конкретных дат либо периодов уплаты платежей.

При принятии решения о получении кредита в банке необходимо предварительно тщательно изучить условия предоставления кредитов и оценить свои возможности по погашению основного долга по кредиту и уплате причитающихся платежей, обращая особое внимание на:

* размер процентов за пользование кредитом;
* срок исполнения обязательств;
* возможность и порядок досрочного погашения кредита;
* необходимость заключения дополнительных договоров, связанных с получением кредита (например, договора страхования).

Потенциальным кредитополучателям следует учитывать то обстоятельство, что чем лучше обеспечен кредит (наличие поручителей, залога и др.), тем выгоднее условия кредитования, то есть ниже размер процентов за пользование кредитом.

На что должен обращать внимание потребитель при выборе банковского вклада (депозита)? Согласно статье 185 Банковского кодекса Республики Беларусь вкладчики свободны в выборе банка для размещения во вклады (депозиты) принадлежащих им денежных средств и могут иметь вклады (депозиты) в одном либо нескольких банках. Банками страны предлагается достаточно широкий выбор видов вкладов (депозитов) как в белорусских рублях, так и в иностранной валюте, с различными сроками и условиями хранения. Граждане вправе распоряжаться собственными сбережениями по своему усмотрению. Выбрав ту или иную форму хранения своих сбережений, каждый вкладчик несет определенные риски в части получения максимального уровня доходности от размещения своих денежных средств [2].

В соответствии с Декретом Президента Республики Беларусь от 11 ноября 2015 г. № 7 «О привлечении денежных средств во вклады (депозиты)» договоры банковского вклада (депозита) подразделяются на безотзывные и отзывные. Главной отличительной особенностью этих двух видов договоров являются условия досрочного возврата вклада (депозита). При заключении отзывного договора предусматривается возможность досрочного возврата вклада (депозита) по инициативе вкладчика. При этом конкретные сроки и условия возврата вклада (депозита) будут прописываться в договоре между банком и вкладчиком. Если для потенциального вкладчика приоритетом является возможность досрочного снятия вклада (депозита) при минимальной потере доходности, то следует обращать внимание на наличие такого условия в договоре банковского вклада (депозита) и на минимальное время хранения денежных средств во вкладе (депозите), обеспечивающее сохранение доходности. Безотзывным же договором досрочный возврат вклада (депозита) по инициативе вкладчика не предусматривается. Вернуть вклад (депозит) до истечения срока действия договора возможно только с согласия банка.

Кроме того, договором на размещение денежных средств во вклад (депозит) могут быть предусмотрены любые не противоречащие закону условия возврата вклада (депозита). Условиями банковского вклада (депозита) может быть предусмотрена возможность пополнения вклада (депозита) и расхода части средств вклада (депозита) в период его действия.

При выборе того или иного вида банковского вклада (депозита) для размещения денежных средств потребителю можно порекомендовать рассматривать условия предлагаемых вкладов (депозитов) с учетом своих ожиданий от хранения денежных средств в том или ином виде вклада (депозита) и обращать внимание не только на размер процентов по нему, но и на другие условия. При этом если наиболее предпочтительным условием является максимальная доходность и денежные средства в ближайшее время вкладчику, скорее всего, не понадобятся, в таком случае ориентиром может выступать размер процентов по вкладу (депозиту) и способ их начисления (такое условие, как капитализация процентов, обеспечивает большую доходность вклада (депозита)). Следует отметить, что многими банками предлагаются виды срочных вкладов (депозитов), предусматривающих возможность получения дополнительного дохода в виде различных бонусов, премий, выигрышей в зависимости от фактического срока хранения вклада (депозита) и соблюдения условий договора банковского вклада (депозита).

При валютно-обменной операции для потребителя определяющим фактором при выборе банка, как правило, является курс обмена. Согласно инструкции «О порядке осуществления валютно-обменных операций с участием физических лиц и организации работы обменных пунктов, касс при проведении таких операций» № 35 от 01.02.2011 г., банки самостоятельно определяют порядок установления обменных курсов для осуществления валютно-обменных операций и вправе осуществлять валютно-обменные операции со всеми иностранными валютами, по отношению к которым Национальный банк устанавливает официальный курс белорусского рубля.

Принятию решения приобрести банковский продукт или воспользоваться той или иной банковской услугой должно предшествовать изучение информации о ней, выяснение возникших вопросов и понимание своей ответственности, в том числе финансовой. Выбору оптимального банковского продукта или услуги может способствовать изучение предложений нескольких банков. Здесь есть несколько альтернативных подходов, каждый из которых имеет свои достоинства и недостатки.

Можно обратиться за советом к родственникам, друзьям, коллегам или соседям, узнать продуктами и услугами каких банков они уже пользуются. Главным минусом такого выбора является то, что фактически потребитель принимает решение, полагаясь на мнение постороннего человека, при таком подходе выбор не будет оптимальным.

Другой вариант, обратиться в ближайшее к месту работы или месту жительства подразделение какого-либо банка. В этом случае гражданин получит необходимый банковский продукт или услугу. В центрах банковских услуг всегда есть специалист, в обязанности которого входит предоставление справочной информации и консультирование клиентов – физических лиц, в том числе не обладающих специальными финансовыми знаниями, в доступной для понимания форме. Однако, потребитель не получит информации о предложениях других банков, и сделает оптимальный выбор продукта или услуги лишь в рамках продуктовой линейки конкретного банка. Обращение за получением справочной информации и консультацией в подразделения разных банков, безусловно, позволит сделать выбор более выгодного продукта или услуги из всех предложений, но на посещение нескольких офисов придется затратить много времени, чего потребитель часто просто не может себе позволить.

Чтобы не тратить время на посещение отделений банков можно воспользоваться возможностью получения справочной информации о продуктах и услугах по телефону. У этого пути также есть ряд недостатков. Как отмечено выше, в Беларуси обслуживание населения осуществляют около двух десятков банков, и в короткий срок обзвонить все для получения справочной информации не представляется возможным. При разговоре со специалистом банка потребителю будет необходимо фиксировать получаемую информацию о продукте или услуге, затем самостоятельно осуществлять анализ полученных данных. В call-центре банка, как правило, предоставляют общую информацию по продуктам и услугам и за более детальной информацией предлагают лично обратиться в подразделение банка.

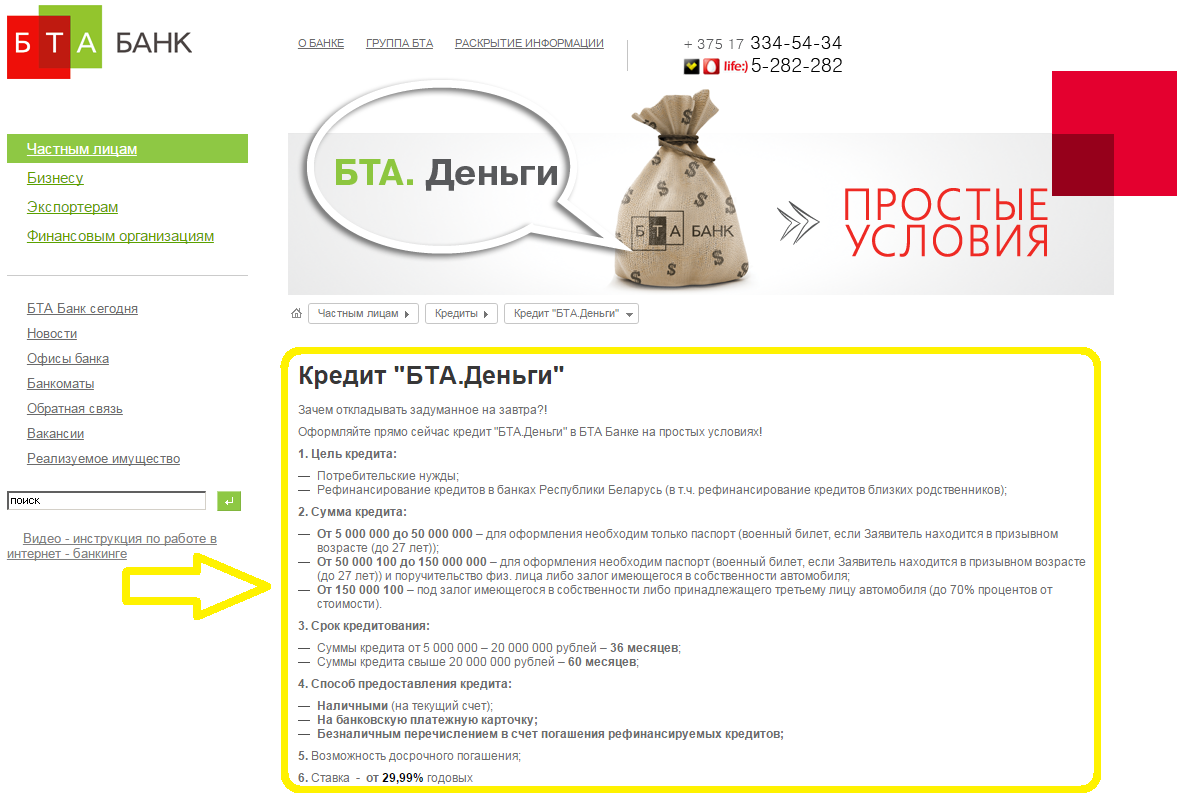
## 1.2 Анализ существующих приложений для получения справочной информации о банках, их продуктах и услугах

Наиболее эффективным для потребителя с точки зрения охвата предложений всех, представленных на белорусском рынке розничных банковских продуктов и услуг, за минимальное время является анализ предложений банков в сети Интернет. Сегодня каждый белорусский банк имеет официальный сайт. Банк самостоятельно определяет функциональность сайта, содержание и форму предоставления информации потребителю. При этом для частных лиц вся информация о розничных услугах и продуктах, как правило, собрана в отдельный раздел. Например, так выглядит главная страница белорусского БТА Банка (www.btabank.by) Рисунок 1:

Рисунок 1 – Главная страница ЗАО «БТА Банк» (Беларусь)

Посетителю представлены вход в Интернет-банкинг, номера телефонов контакт-центра, новости и события банка, курсы обмена наиболее востребованных валют, а также возможность перейти по ссылке непосредственно к информации об интересующем виде банковского продукта или услуги, либо к информации обо всех предложениях в банке, для каждой категории клиентов.

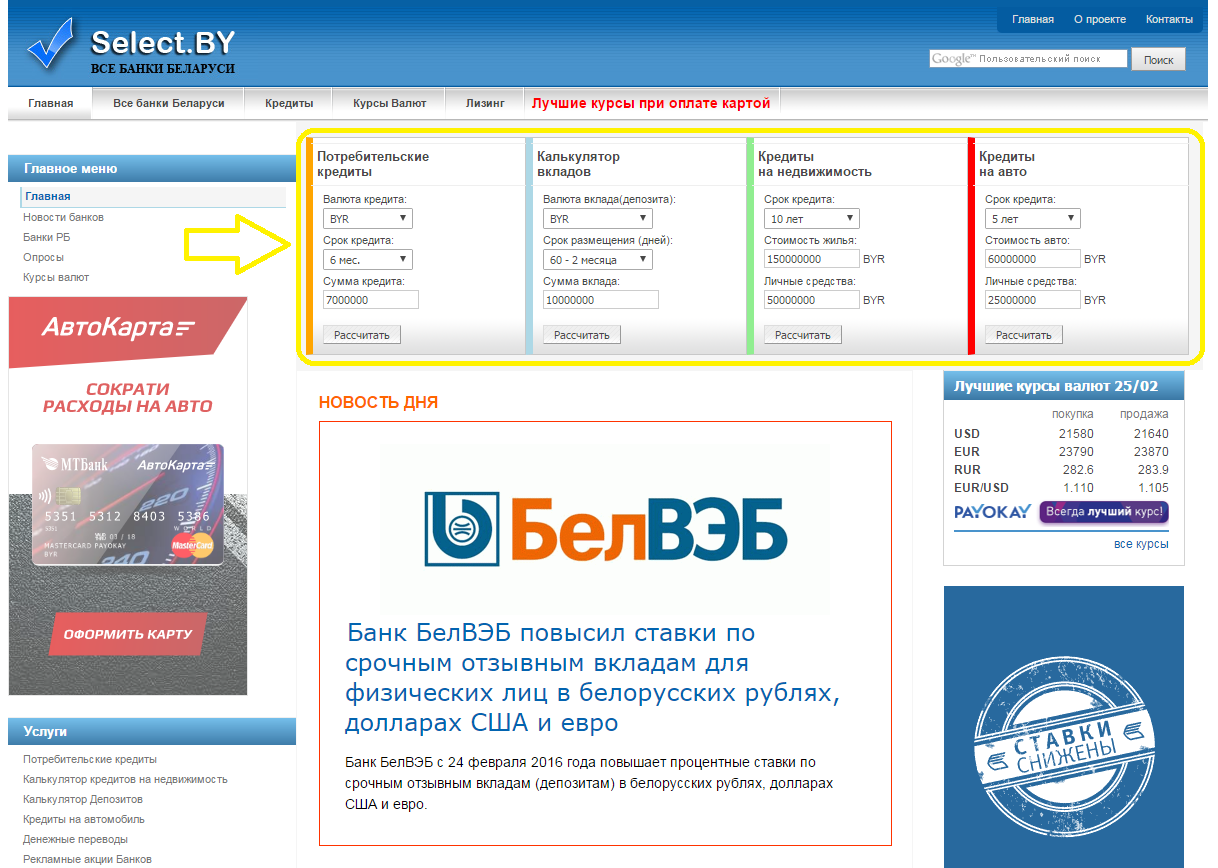
Рассмотрим для примера форму предоставления справочной информации на данном сайте о потребительских кредитах для физических лиц. Потенциальный клиент имеет возможность ознакомиться с условиями получения кредита, суммой, величиной процентов за пользование денежными средствами, способами обеспечения возврата (Рисунок 2 – Информация об условиях кредитования). Информация представлена в текстовом виде, отсутствует возможность сделать предварительный расчет суммы кредита в зависимости от дохода, а также рассчитать примерную сумму ежемесячных платежей [8].

Рисунок 2 – Информация об условиях кредитования.

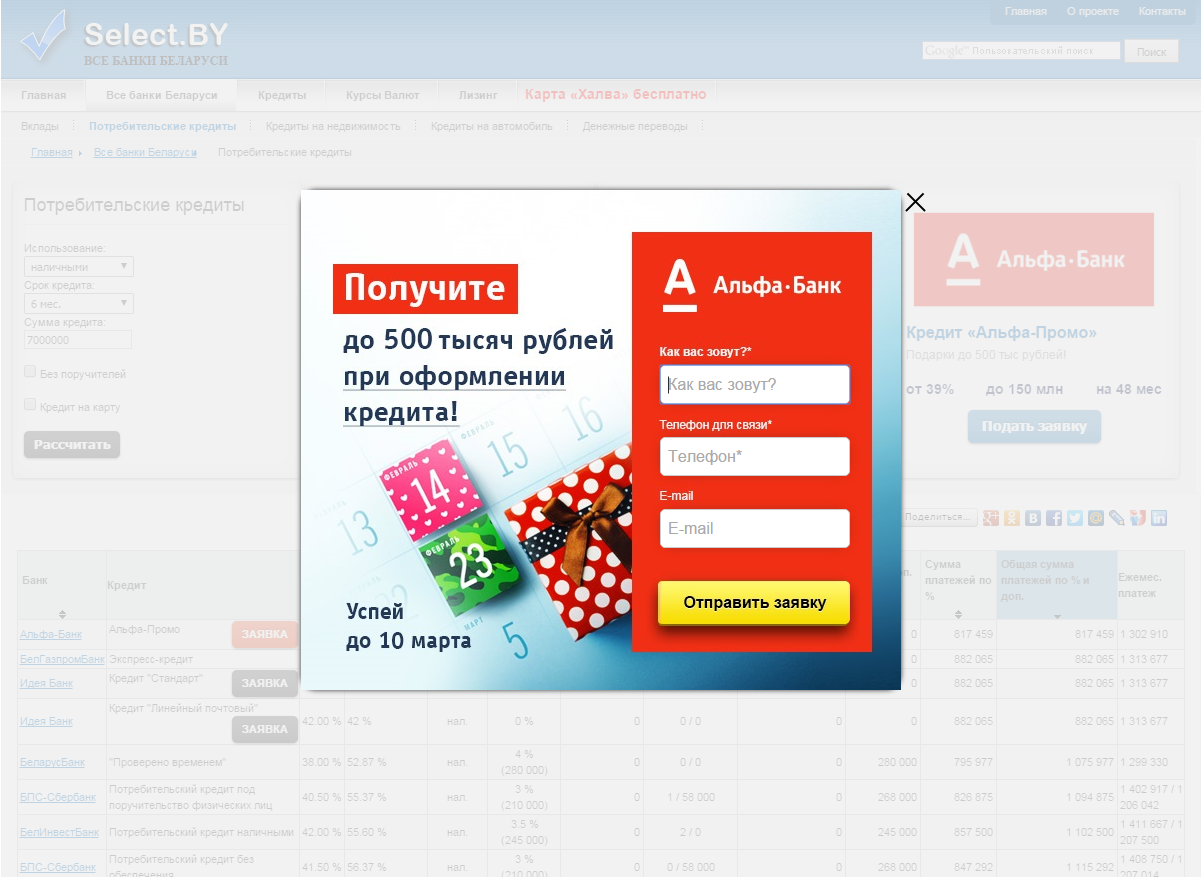
Чтобы не изучать и не анализировать самостоятельно предложения на всех официальных интернет-страницах более двух десятков белорусских банков, справочную информацию о банках, их продуктах и услугах для населения можно получить на тематических информационно-аналитических порталах. Такие ресурсы не только предоставляют посетителям справочную информацию, позволяют сравнивать различные услуги и продукты, получая объективные сведения. По заявлению собственников подобных сайтов, вся информация о банковских продуктах публикуется совершенно беспристрастно, чтобы у посетителей всегда была возможность свободного выбора на основе полных данных о рынке банковских услуг.

На сегодняшний день одним из самых посещаемых сайтов финансово-экономической тематики в Беларуси является сайт Select.by – Все банки Беларуси (www.select.by). Согласно информации, размещенной в разделе «О проекте», ежемесячно на сайт приходит до 900 тысяч посетителей, ежедневно – до 30 тысяч уникальных посетителей, доля посетителей из Беларуси более 94% [7].

Данный ресурс позволяет получить справочную информацию о каждом банке, представленном на рынке Беларуси, а именно контактные и адресные данные, перечень операций, которые банк вправе совершать и краткую историческую сводку. Непосредственно на главной странице (Рисунок 3 – Главная страница Select.by.) посетитель может воспользоваться кредитным или депозитным калькуляторами, которые позволяют отобрать предложения банковских продуктов по заданным параметрам (валюта, срок, сумма).

Рисунок 3 – Главная страница Select.by

Главным недостатком такого рода ресурсов, на мой взгляд, является большое количество рекламы, для отображения которой отведена значительная часть площади страницы. Рекламные баннеры отвлекают и затрудняют поиск необходимой потребителю информации, либо вовсе ее закрывают. Для получения необходимых данных пользователю нужно закрыть всплывающее окно с рекламой. Например, страница с информацией о кредитах для частных лиц сразу после перехода на нее принимает следующий вид (Рисунок 4 – Страница «Кредиты» на сайте select.by):

Рисунок 4 – Страница «Кредиты» на сайте select.by

Еще один популярный и востребованный информационно-аналитический портал – это Инфобанк.бай (www.infobank.by), который является одним из крупнейших банковских ресурсов белорусского сегмента сети Интернет. Основная цель проекта – предоставление объективной и достоверной информации о банках и финансах, а также поддержание постоянного конструктивного общения, между представителями банков и клиентами. Чтобы объективно представить информацию о банках, сотрудники портала сами собирают сведения о банковских продуктах и всегда выступают на стороне потребителя [6].

К полезным для потенциального клиента банка особенностям данного портала можно отнести следующие проекты:

* «[народный рейтинг](http://infobank.by/330/default.aspx)» банков, по отзывам посетителей на портале;
* регулярный [публичный рейтинг белорусских банков](http://infobank.by/2636/default.aspx) по показателям деятельности на основании данных официальной статистики Национального банка Республики Беларусь;
* проект «Глас народа» – опрос на улицах города Минска об отношении населения к банкам и банковским услугам

Функции, которые предоставлены пользователю портала, схожи с рассмотренным выше сайтом Select.by.

Рисунок 5 – Главная страница Infobank.by

Потребитель имеет возможность получить справочную информацию о каждом банке, представленном на белорусском рынке, подобрать вклад или кредит, получить оперативные сведения о курсах валют в банках. В качестве главного недостатка, вновь можно отметить большое количество рекламы, что затрудняет объективное восприятие предоставленной информации. Так, например, форма для подбора кредита, расположенная на главной странице портала, оформлена в фирменных цветах одного из банков и подбор кредитных продуктов по заданным параметрам производится только из продуктовой линейки этого банка (Рисунок 5 – Главная страница Infobank.by). Очевидно, что это может ввести пользователя в заблуждение.

Рассмотренные выше варианты можно оценить по ключевым параметрам. Сравнительный анализ веб-приложений представлен в таблице 1.

Таблица 1 – Сравнение веб-приложений для получения информации о банковских продуктах и услугах для физических лиц

|  |  |  |  |
| --- | --- | --- | --- |
| Параметры/Приложения | Официальная страница ЗАО «БТА Банк» | Информационно-аналитический портал Select.by | Информационно-аналитический портал Infobank.by |
| Справочная информация о банках Беларуси | **-** | **+** | **+** |
| Информация о вкладах (депозитах) | **Только ЗАО «БТА Банк»** | **+** | **+** |
| Информация о кредитах | **Только ЗАО «БТА Банк»** | **+** | **+** |
| Курсы валют | **Только ЗАО «БТА Банк»** | **+** | **+** |
| Депозитный калькулятор | **-** | **+** | **+** |
| Кредитный калькулятор | **-** | **+** | **+** |
| Конвертер валют | **-** | **+** | **+** |
| Без рекламы | **+** | **-** | **-** |

Проанализировав большинство доступных потребителю путей выбора того или иного вида банковского продукта или услуги можно сделать вывод, что на конечный выбор потребителя влияние оказывает множество факторов. Зачастую самостоятельно найти оптимальный вариант простому обывателю сложно. Банки предлагают большое количество розничных продуктов и услуг, дополнительно привлекают клиентов при помощи акций и бонусов. Наиболее эффективный способ для ознакомления с большим количеством предложений – это обращение к информации, размещенной на страницах информационно-аналитических порталов в сети Интернет. Это позволяет выбрать два-три варианта продукта или услуги для подробного изучения и обращения в финансовое учреждение за детальной консультацией. Главным недостатком таких ресурсов является большое количество рекламы, что затрудняет объективное восприятие информации потребителем.

## 1.3 Требования к разрабатываемой системе и постановка задачи

На основе анализа существующих решений, наиболее близким аналогом для разрабатываемого приложения можно признать информационно-аналитический портал Infobank.by. К приложению, создаваемому в рамках данного дипломного проекта, предъявляются следующие требования:

* приложение должно обеспечивать пользователю доступ к справочной информации о:

1. белорусских банках, с обязательным отображением наименования, краткого описания, адреса официальной страницы в сети Интернет;
2. отделениях и филиалах белорусских банков с обязательным отображением наименования подразделения его адреса и контактных телефонов;
3. основных продуктах и услугах для физических лиц, таких как вклады (депозиты), кредиты, курсы обмена валюты;

* приложение должно предоставлять возможность рассчитывать:

1. сумму дохода по вкладу (депозиту) по заданным условиям (валюта сбережения, сумма, процентная ставка и срок хранения);
2. платежи по кредиту по заданным условиям (срок кредитования, процент вознаграждения за пользование кредитом, сумма кредита);
3. сумму эквивалента при обмене валюты.

Исходя из задания на проектирование и задач, которые позволит решать приложение необходимо:

* выполнить анализ и сформировать функциональные требования;
* разработать проект программного средства;
* выполнить разработку схемы данных реляционной модели и структуры классов;
* разработать основные алгоритмы и их программную реализацию;
* выполнить тестирование разработанного программного средства.

# 2 Методы и модели, положенные в основу проекта

## 2.1 Методы определения дохода по вкладу (депозиту)

Важным условием банковского вклада является величина процентной ставки. Процентная ставка по банковскому вкладу отражается в процентах годового дохода, независимо от суммы вклада.

Начисление процентов может осуществляться по формулам простого или сложного процента.

Простой процент – процент, начисляемый на сумму вклада исходя из срока вклада с определенной договором банковского вклада периодичностью без учета ранее начисленных на вклад процентов.

Сумма денежных средств, причитающаяся к выплате вкладчику при начислении простых процентов по вкладу, рассчитывается по формуле:

S = P + (P \* I \* t) / (K \* 100), (1)

где S — сумма денежных средств, причитающихся к возврату;

P – первоначальная сумма привлеченных в депозит денежных средств;

I – годовая процентная;

t – количество дней начисления процентов по вкладу;

K – количество дней в календарном году (365 или 366);

Чтобы рассчитать только сумму простых процентов, причитающихся к выплате используется следующая формула:

Sp = (P \* I \* t) / (K \* 100), (2)

где Sp — сумма процентов (доход);

P – первоначальная сумма привлеченных в депозит денежных средств;

I – годовая процентная;

t – количество дней начисления процентов по вкладу;

K – количество дней в календарном году (365 или 366);

Сложный процент (капитализация) – процент, начисляемый на сумму вклада и сумму ранее начисленных по вкладу процентов с учетом срока вклада с определенной договором банковского вклада периодичностью.

Наличие данного условия в договоре означает, что начисленные проценты будут автоматически (ежемесячно, ежегодно), без участия вкладчика, добавлены к основной сумме вклада и тоже будут приносить доход.

Сумма денежных средств, причитающихся к возврату вкладчику по окончании срока вклада при начислении сложных процентов рассчитывается по формуле:

(3)

где S — сумма денежных средств, причитающихся к возврату;

P – первоначальная сумма привлеченных в депозит денежных средств;

I – годовая процентная ставка;

j – количество календарных дней в периоде;

K – количество дней в календарном году (365 или 366);

n — количество операций по капитализации начисленных процентов в течение общего срока привлечения денежных средств.

Чтобы рассчитать только сумму простых процентов, причитающихся к выплате используется следующая формула:

(4)

где Sp — сумма процентов (доход);

P – первоначальная сумма привлеченных в депозит денежных средств;

I – годовая процентная ставка;

j – количество календарных дней в периоде;

K – количество дней в календарном году (365 или 366);

n — количество операций по капитализации начисленных процентов в течение общего срока привлечения денежных средств.

Сегодня большинство банков предлагают вклады (депозиты) с ежемесячной капитализацией.

## 2.2 Методы расчета процентов по кредиту

Методика расчета платежей по кредиту зависит от схемы погашения задолженности. Наиболее распространенные с ежемесячным погашением задолженности дифференцированными (убывающими) или аннуитетными (равными) платежами. Есть также кредиты с разовым погашением задолженности, когда основной долг и проценты погашаются в конце его срока. Некоторые банки выдают кредиты с индивидуальными схемами погашения задолженности, которые указываются в кредитном договоре.

Применения того или иного вида платежа влияет на конечную сумму погашения долга.

Ежемесячный платёж, при аннуитетной схеме погашения кредита состоит из двух частей. Первая часть платежа идёт на погашение процентов за пользование кредитом. Вторая часть идёт на погашение долга. Аннуитетная схема погашения отличается тем, что в начале кредитного периода проценты составляют большую часть платежа. Тем самым сумма основного долга уменьшается медленно, соответственно переплата процентов при такой схеме погашения кредита получается больше. Тем ни менее, такой способ погашения кредита пользуется популярностью в народе из-за лёгкости вычисления ежемесячного платежа, потому что с ним нет путаницы.

Рассчитать ежемесячный аннуитетный платеж по кредиту можно по следующей формуле:

(5)

где x — месячный платеж;

S – первоначальная сумма кредита;

P – 1/12 процентной ставки;

N – количество месяцев.

Формула, для определения того, какая часть платежа пошла на погашение кредита, а какая на оплату процентов является достаточно сложной и без специальных математических знаний простому обывателю будет сложно ей воспользоваться. Поэтому мы рассчитаем данные величины простым способом, дающим такой же результат.

Для расчета процентной составляющей аннуитетного, нужно остаток кредита на указанный период умножить на годовую процентную ставку и всё это поделить на 12 (количество месяцев в году).

(6)

где Pn — начисленные проценты;

Sn – остаток задолженности на период;

P – годовая процентная ставка по кредиту.

Что бы определить часть, идущую на погашение долга, необходимо из месячного платежа вычесть начисленные проценты:

(7)

где s — часть выплаты, направляемая на погашение долга;

x – сумма месячного платежа;

Pn – начисленные проценты на момент n-ой выплаты.

Поскольку часть, идущая на погашение основного долга, зависит от предыдущих платежей, поэтому расчёт графика, по данной методике вычислять последовательно, начиная с первого платежа.

Платеж по фактическому остатку. Здесь формула вычисления намного проще, в сравнении с предыдущим способом. Платеж складывается из процентного объема и погашения ежемесячного кредита:

Sn = K / n + K \* P (8)

где Sn — платеж по кредиту в n-й период;

K – первоначальная сумма кредита;

n – количество периодов оплаты;

P – процентная ставка.

В отличие от аннуитетного платежа, здесь величина выплат каждый месяц будет разной.

# 3 Разработка проекта программного продукта

## 3.1 Функциональный анализ предметной области

Анализ требований – это процесс сбора требований к программному обеспечению, их систематизации, документирования, анализа, выявления противоречий, неполноты, разрешения конфликтов в процессе разработки программного обеспечения.

Исходя из поставленной задачи, а также перечня необходимых к реализации расчетов были выявлены следующие основные бизнес-процессы:

* получение справочной информации о белорусских банках и их продуктах и услугах для частных лиц, а именно вкладах (депозитах), кредитах и курсах обмена иностранных валют;
* расчет платежей по кредиту по заданным пользователем параметрам (сумма основного долга, процент за пользование заемными средствами);
* расчет суммы дохода по вкладу (депозиту) по заданным пользователем параметрам (валюта вклада (депозита), сумма, процент вознаграждения)
* получение справочной информации о курсах обмена иностранных валют в белорусских банках;
* расчет эквивалента обмена с помощью конвертера валют.

Для автоматизации выявленных бизнес-процессов необходимо разработать программный продукт, который позволит пользователю удаленно посредством доступа в сеть Интернет получить справочную информацию о банках и об условиях наиболее востребованных у населения банковских продуктов и услуг.

На стадии определения требований к системе была выявлена структура предметной области, позволившая выделить сущности и их взаимосвязь.

На рисунке 6 бизнес-сущности и их связь представлены в виде UML-диаграммы:

Рисунок 6 − Бизнес-сущности предметной области и их связь

В таблице 2 описаны выделенные сущности и их назначение.

Таблица 2 — Описание сущностей

| Название сущности | Назначение и атрибуты |
| --- | --- |
| 1 | 2 |
| Банк | Отображение информации о банке. Хранение справочной информации о каждом банке  Атрибуты: наименование, описание, веб-страница, логотип. |
| Подразделение банка | Отображение подробной информации о структурном подразделении конкретного банка. Хранение справочной информации о структурных подразделениях  Атрибуты: наименование, адрес, номер телефона. |
| Депозит | Отображение банковского продукта депозит. Хранение справочной информации о каждом продукте.  Атрибуты: наименование, валюта, срок, процентная ставка. |
| Кредит | Отображение банковского продукта кредит. Хранение справочной информации о каждом продукте.  Атрибуты: наименование, срок, процент вознаграждения. |
| Валюта | Хранение информации о каждой валюте  Атрибуты: наименование, мнемоника, код. |
| Курс обмена | Отображение условий обменной операции. Хранение информации о курсах валют  Атрибуты: наименование валюты, курс покупки, курс продажи |

После определения внешних сущностей, с которыми разрабатываемая система должна взаимодействовать, необходимо определить возможных кандидатов, которые обращаются к системе для выполнения своих задач или запускают первичные, или вторичные функции системы, а также внешние устройства, программные средства и другие системы.

Далее определяется область ответственности действующего лица и цели, которые действующее лицо пытается достичь при использовании системы. Те кандидаты на действующее лицо, которые не имеют никаких целей, исключаются.

Участник бизнеса/бизнес-процесса (business actor) – это штатная единица или группа, исполняющая свои функциональные обязанности в данном бизнесе или бизнес-процессе. Чаще всего ими могут быть штатные работники, занимающие определенные должности и исполняющие должностные обязанности, либо структурные подразделения. Этот круг лиц включает в себя пользователей программной системы. Эти лица моделируются при помощи стереотипа Business Actor, а их функциональные обязанности с помощью стереотипа Business Use-Case. Действующее лицо (actor), синонимы актёр, актант – абстрактное понятие характеризует внешнего пользователя (или группу пользователей), непосредственно взаимодействующих с программной системой.

Можно выделить действующих лиц, которые будут использовать программный продукт: администратор и пользователь (посетитель сайта).

Администратор – уполномоченное лицо, наделенное необходимыми правами, которое поддерживает работоспособность проекта, занимается вопросами обновления материала на сайте, распределяет роли и права доступа.

Рисунок 7 − Бизнес-функции администратора

В качестве пользователя выступает любой посетитель сайта, который имеет возможность использовать основные функции разрабатываемого веб-приложения, среди которых:

* просмотр справочной информации о банках;
* расчет платежей по кредиту;
* расчет дохода по депозиту;
* просмотр обменных курсов в банках;
* расчет эквивалента при обмене валюты.

Рисунок 8 − Бизнес-функции пользователя.

На основании сформированных Use case разработана диаграмма вариантов использования (use case diagram) приложения пользователем (Рисунок 9 – Диаграмма вариантов использования), которая представляет собой модель, с изображенными связями между пользователем (actor) и вариантами использования программного продукта (use case).

Рисунок 9 – Диаграмма вариантов использования

Функциональные требования к программному средству представлены в трассировочной таблице (Таблица 3).

Таблица 3 – Трассировочная таблица

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Требование заказчика | № | Функции системы | Use case |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Просматривать справочную информацию о банках, структурных подразделениях, основных продуктах и услугах | 1.1 | Чтение из БД и отображение представления со справочной информацией о банках, структурных подразделениях, основных продуктах и услугах | Просмотр справочной информации о банке |
| 2 | Производить расчет платежей по кредиту по заданным параметрам (срок, сумма, процент вознаграждения) | 2.1 | Расчет платежей по кредиту | Расчет платежей по кредиту |
| 2.2 | Ввод суммы кредита | Ввод суммы кредита |
| 2.3 | Ввод срока кредита | Ввод срока кредита |
| 2.4 | Ввод % за пользование кредитом | Ввод % за пользование кредитом |

Продолжение таблицы 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Производить расчет дохода по депозиту по заданным условиям (валюта, срок, сумма, процент вознаграждения) | 3.1 | Расчет дохода по депозиту | Расчет дохода по депозиту |
| 3.2 | Выбор валюты вклада | Выбор валюты вклада |
| 3.3 | Ввод суммы вклада | Ввод суммы вклада |
| 3.4 | Ввод срока вклада | Ввод срока вклада |
| 3.5 | Ввод процента вознаграждения | Ввод процента вознаграждения |
| 4 | Просматривать курсы обмена иностранных валют в банках | 4.1 | Чтение из БД и отображение представления со справочной информацией о курсах обмена валют | Просмотр информации об обменных курсах |
| 5 | Производить расчет эквивалента при обмене иностранной валюты | 5.1 | Расчет эквивалента обмена иностранной валюты | Расчет эквивалента обмена иностранной валюты |
| 5.2 | Выбор валюты №1 | Выбор валюты №1 |
| 5.3 | Выбор валюты №2 | Выбор валюты №2 |
| 5.4 | Ввод суммы операции | Ввод суммы операции |

## 3.2 Выбор архитектуры приложения, языка программирования и системы управления базой данных

Создание архитектуры приложения – это процесс формирования структурированного решения, отвечающего всем техническим и операционным требованиям и обеспечивающего оптимальные общие атрибуты качества, такие как производительность, безопасность и управляемость. Он включает принятие ряда решений на основании широкого диапазона факторов. Каждое из этих решений может иметь существенное влияние на качество, производительность, удобство обслуживания и общий успех приложения. Как и любая другая сложная структура, программное средство должно строиться на прочном фундаменте. Неправильное определение ключевых сценариев, неправильное проектирование общих вопросов или неспособность выявить долгосрочные последствия основных решений могут поставить под угрозу все приложение. Современные инструменты и платформы упрощают задачу по созданию приложений, но не устраняют необходимости в тщательном их проектировании на основании конкретных сценариев и требований. Неправильно выработанная архитектура обусловливает нестабильность программного обеспечения, невозможность поддерживать существующие или будущие бизнес-требования, сложности при развертывании или управлении в среде производственной эксплуатации.

Основное назначение архитектуры – описание использования или взаимодействия основных элементов и компонентов приложения. При продумывании архитектуры приложения, разрабатываемого в рамках данного дипломного проекта, я предполагаю, что дизайн будет эволюционировать со временем и что совершенно невозможно наперед знать все то, что необходимо для проектирования системы. Как правило, дизайн изменяется и дорабатывается в ходе реализации приложения по мере выявления новых сведений и в ходе тестирования на соответствие требованиям реального окружения. Таким образом, архитектура ориентирована на такие изменения, на обеспечение возможности адаптировать их к требованиям, которые в начале процесса проектирования известны не в полном объеме [16].

Для разрабатываемого в рамках дипломного проекта приложения выбрана клиент-серверная архитектура. Такой подход разделяет функции приложения пользователя (называемого клиентом) и сервера. Приложение-клиент формирует запрос к серверу, сервер принимает запрос, обрабатывает и выполняет его, при этом ресурсы клиентского компьютера не участвуют в физическом выполнении запроса. Результат выполнения запроса возвращается приложению-клиенту. Все это повышает быстродействие системы и снижает время ожидания результата запроса. При выполнении запросов сервером существенно повышается степень безопасности данных, поскольку правила целостности данных определяются в базе данных на сервере и являются едиными для всех приложений. Таким образом, исключается возможность одновременного изменения одних и тех же данных различными пользователями.

В качестве шаблона для разрабатываемого приложения мною выбран паттерн MVC (Model-View-Controller) − это архитектурный принцип, согласно которому веб-приложение делится на компоненты. Разделение веб-приложения на компоненты упрощает его разработку, тестирование и сопровождение.

Основные критерии, предъявляемые к современным приложениям: удобство интерфейса и гибкость использования, эффективность работы, компактность и функциональность приложения.

Для разработки программного обеспечения выбрана среда разработки Visual Studio Сommunity 2015, обеспечивающая высокое качество кода на протяжении всего цикла разработки программного обеспечения, от проектирования до разработки.

В качестве языка программирования выбран C#. На сегодняшний момент язык программирования C# одни из самых мощных, быстро развивающихся и востребованных языков. В настоящий момент на нем пишутся самые различные приложения: от небольших десктопных программок до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей.

C# является объектно-ориентированным, поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений. C# продолжает активно развиваться, и с каждой новой версией появляется все больше интересных функциональностей [14]. Разрабатываемое приложение построено на платформе ASP.NET. Она является составной частью платформы [Microsoft .NET](https://ru.wikipedia.org/wiki/Microsoft_.NET), которая позволяет с помощью языка программирования C# создавать веб-приложения. Хостинг приложений ASP.NET в сети Интернет осуществляется веб-сервером IIS (Internet Information Services) от компании Microsoft.

Visual Studio поддерживает C# с полнофункциональным редактором кода, компилятором, шаблонами проектов, конструкторами, мастерами кода, мощным и простым в использовании отладчиком и многими другими средствами. В состав библиотеки .NET Framework входит платформа ADO.NET, которая также задействована в курсовом проекте.

В дипломном проекте использована ADO.NET Entity Framework — объектно-ориентированная технология доступа к данным. Entity Framework является [object-relational mapping](https://ru.wikipedia.org/wiki/ORM) (ORM) решением для [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework) от [Microsoft](https://ru.wikipedia.org/wiki/Microsoft). Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища [9].

Центральной концепцией Entity Framework является понятие сущности или entity. Сущность представляет набор данных, ассоциированных с определенным объектом. Поэтому данная технология предполагает работу не с таблицами, а с объектами и их наборами. Любая сущность, как и любой объект из реального мира, обладает рядом свойств. Сущности могут быть связаны ассоциативной связью один-ко-многим, один-ко-одному и многие-ко-многим.

Для обеспечения высокой надежности хранения данных была выбрана система управления баз данных – Microsoft SQL Server 2014. Она обеспечивает доступ к обширным ресурсам, ведущую в отрасли производительность и масштабируемость корпоративного класса, высочайший уровень безопасности, высочайший уровень доступности и высочайший уровень надежности [5].

Схема ресурсов системы представлена на чертеже РТДП 4032.052.001.

## 3.3 Разработка модели данных

Одной из основных задач, при проектировании системы, является разработка модели данных для хранения информации.

Первый этап процесса проектирования модели данных называется концептуальным проектированием. Он заключается в создании концептуальной модели данных для анализируемой предметной области. Эта модель данных основывается на семантической модели и не зависит от любых физических аспектов ее представления.

Созданная концептуальная модель данных является источником информации для этапа логического проектирования базы данных.

Цель этапа логического проектирования – преобразование концептуальной модели данных в логическую модель, не зависимую от особенностей используемой в дальнейшем СУБД для физической реализации базы данных. Первым шагом при создании логической модели является построение диаграммы ERD (Entity Relationship Diagram). Целью построения логической модели является получение графического представления логической структуры исследуемой предметной области [10].

Логическая модель предметной области иллюстрирует сущности, а также их взаимоотношения между собой.

Сущности описывают объекты, являющиеся предметом деятельности предметной области, и субъекты, осуществляющие деятельность в рамках предметной области. Свойства объектов и субъектов реального мира описываются с помощью атрибутов. Взаимоотношения между сущностями иллюстрируются с помощью связей.

Графическое отображение логической модели базы данных представлено на Рисунке 10:

Рисунок 10 – Логическая модель базы данных



Построение модели данных в приложении будет производится на основе подхода Code First, когда создаются классы модели данных, которые будут храниться в базе данных, а затем Entity Framework по этой модели генерирует базу данных и ее таблицы.

Поэтому, следующий этап проектирования – это разработка структуры классов. Описание классов приведено в таблице 4.

Таблица 4 – Описание классов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Назначение | Атрибуты |
| 1 | 2 | 3 | 4 |
| 1 | Банк (Bank) | Описывает Банк | Наименование  Описание  Веб-страница  Логотип |
| 2 | Подразделение банка (Branch) | Описывает обособленное структурное подразделение конкретного банка | Наименование  Город  Улица  Дом  Номер телефона |

Продолжение таблицы 4

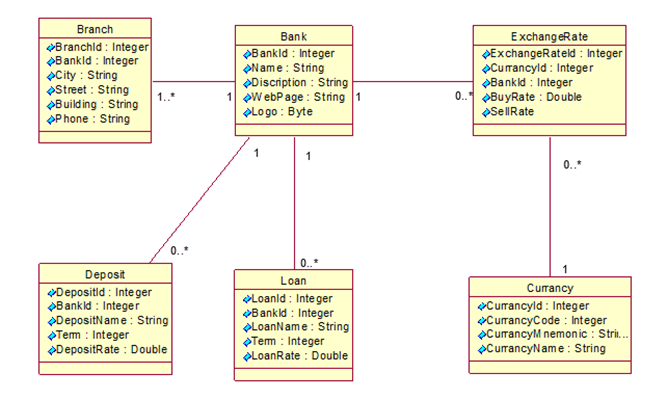
|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 3 | Вклад (депозит) (Deposit) | Описывает банковский продукт – вклад (депозит) | Наименование  Срок  Валюта  Процентная ставка |
| 4 | Кредит (Loan) | Описывает банковский продукт – кредит | Наименование  Срок  Процент за пользование |
| 5 | Обменный курс (ExchangeRate) | Описывает конкретные значения обменного курса по валютно-обменной операции | Валюта  Курс покупки  Курс продажи |
| 6 | Валюта (Currency) | Описывает валюту | Наименование  Мнемоника валюты  Код валюты |

Классы связаны между собой. Связи описаны в таблице 5.

Таблица 5 – Связи классов

|  |  |  |
| --- | --- | --- |
| № | Связь | Описание |
| 1 | 2 | 3 |
| 1 | Банк – Подразделение Банка | Банк включает в свой состав обособленные структурные подразделения. Структурное подразделение входит в состав только одного конкретного банка. Связь «один-ко-многим». |
| 2 | Банк – Вклад (депозит) | Банк включает в состав продуктовой линейки такой вид продукта, как вклад (депозит). Конкретный вклад (депозит) входит в продуктовую линейку только одного банка Связь «один-ко-многим». |
| 3 | Банк – Кредит | Банк включает в состав продуктовой линейки такой вид продукта, как кредит. Конкретный кредит входит в продуктовую линейку только одного банка Связь «один-ко-многим». |
| 4 | Банк – Обменный курс | Банк устанавливает курсы обмена валюты. Установленный обменный курс действует в пределах банка. Связь «один-ко-многим». |
| 5 | Валюта – Обменный курс | Обменный курс установлен по отношению к конкретной валюте. Валюта в разных банках может обмениваться по разным курсам. Связь «один-ко-многим». |

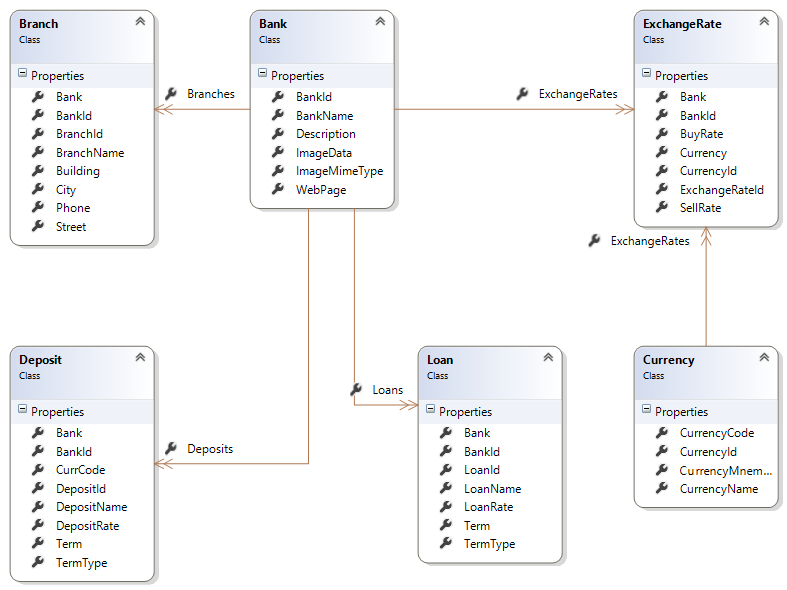
Описанные выше классы и их связь представлены на Рисунке 11 в виде диаграммы классов. Диаграммы классов являются центральным звеном методологии объектно-ориентированных анализа и проектирования.

Рисунок 11 – Диаграмма классов

На стадии анализа диаграммы классов используются, чтобы выделить общие роли и обязанности сущностей, обеспечивающих требуемое поведение системы. На стадии проектирования диаграммы классов используются, чтобы передать структуру классов, формирующих архитектуру системы.

Атрибуты класса определяют состав и структуру данных, хранимых в объектах этого класса. Каждый атрибут имеет имя и тип, определяющий, какие данные он представляет. При реализации объекта в программном коде для атрибутов будет выделена память, необходимая для хранения всех атрибутов, и каждый атрибут будет иметь конкретное значение в любой момент времени работы программы. Объектов одного класса в программе может быть сколь угодно много, все они имеют одинаковый набор атрибутов, описанный в классе, но значения атрибутов у каждого объекта свои и могут изменяться в ходе выполнения программы.

Классы, разработанные в Visual Studio на основе диаграммы классов, созданной на этапе проектирования, представлены на Рисунке 12:

Рисунок 12 – Структура классов модели данных

Таким образом, в проекте разработана модель данных, которая обеспечит хранение в базе данных и доступ к данным при выполнении функций программы.

# 4 Разработка алгоритмов и их программная реализация

## 4.1 Карта сайта, меню разрабатываемого веб-приложения

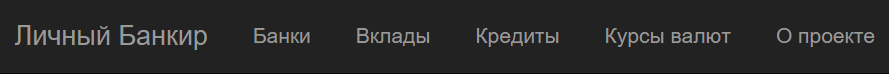
Разработка веб-приложения производится с применением адаптивного дизайна веб-страниц. Целью адаптивного веб-дизайна является универсальность отображения содержимого веб-сайта для устройств различных [разрешений](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0)) и форматов, обеспечение динамически подстраивающегося отображения веб-страниц под заданные размеры окна браузера. Например, главное меню сайта меняет вид в зависимости от ширины экрана. В обычном режиме главное меню сайта представлено в виде расположенных линейно кнопок:

Рисунок 13 – Главное меню сайта (обычный режим)

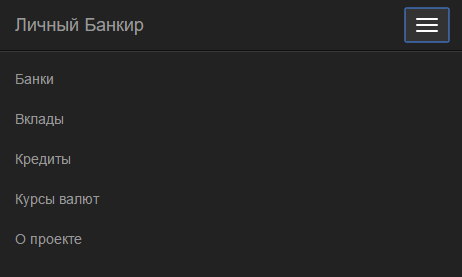
Если ширина экрана менее 600 px, то главное меню сайта принимает вид раскрывающегося списка:

Рисунок 14 – Главное меню сайта (ширина экрана менее 600 px)

При входе на сайт, пользователь имеет возможность выбрать в главном меню сайта один из следующих разделов.

Банки – страница отображает краткие сведения (название, краткое описание, адрес официального сайта банка в сети Интернет) обо всех белорусских банках, информация о которых есть в базе данных проекта. Здесь пользователь имеет возможность выбрать конкретный банк и перейти по ссылке к подробной информации о банке (адреса и контакты всех отделений и филиалов банка).

Вклады – на странице представлена информация (наименование банка, название вклада, срок, процентная ставка, валюта вклада) обо всех предложениях по вкладам, записи о которых есть в базе данных проекта. Здесь пользователь имеет возможность выбрать конкретную валюту вклада, задать срок и процент вознаграждения и получить расчет дохода по вкладу по заданным критериям.

Кредиты – на странице представлена информация (наименование банка, название кредита, срок, процент за пользование) обо всех предложениях по кредитам для физических лиц, записи о которых есть в базе данных проекта. Здесь пользователь имеет возможность задать конкретные условия кредитования (срок, сумма, процент по кредиту), выбрать вид погашения (равными долями, либо убывающими платежами) и получить расчет платежей по кредиту по заданным критериям.

Курсы валют – на странице представлена информация о покупке и продаже банками иностранной валюты (доллары США, евро, российские рубли).

О проекте – на странице представлена информация о приложении и разработчике.

Схема работы системы представлена на чертеже РТДП 4032.052.002.

## 4.2 Реализация функционала проекта, программирование алгоритмов

### 4.2.1 Отображение справочной информации о банках и розничных продуктах

Справочная информация о банках, банковских вкладах (депозитах) или кредитах отображается в соответствующих разделах приложения. Сведения для отображения отбираются из таблиц базы данных. Отобранные записи сортируются по наименованию банка и объединяются в список, который передается в представление для отображения пользователю. Если записи для отображения не найдены, то выводится сообщение об отсутствии информации в данном разделе. Схема алгоритма представлена на Рисунке 17:

Рисунок 17 – Схема алгоритма поиска и отбора данных



### 4.2.2. Отображение информации о курсах обмена валют

Справочная информация о курсах обмена валют отображается в разделе «Курсы валют». Сведения для отображения отбираются из таблиц базы данных. Отобранные записи сортируются по видам операции (покупка валюты банком или продажа валюты банком) и группируются по наименованию валюты, затем данные сортируются по наименованию банка, и объединяются в список, который передается в представление для отображения пользователю. Если записи для отображения не найдены, то выводится сообщение об отсутствии информации в данном разделе. Схема алгоритма представлена на Рисунке 18:

Рисунок 18 – Схема алгоритма формирования списка курсов обмена валют



### 4.2.3 Отображение информации о структурных подразделениях банка

В качестве входного параметра выступает целочисленное значение id, которое представляет собой идентификатор конкретного банка. По входящему параметру производится отбор данных об отделениях и филиалах банка из таблицы базы данных. Если данные не найдены, выводится сообщение об отсутствии информации по данному банку. Если информация о подразделениях банка найдена, то отобранные данные объединяются в результирующий список и передаются в представление для отображения.

Схема алгоритма отображения подробной информации о структурных подразделениях банка представлена на чертеже РТДП 4032.052.003.

### 4.2.4 Расчет дохода по вкладам (депозитам) по заданным условиям

Условия расчета задаются пользователем. Входными параметрами для расчетов выступают:

* валюта сбережения;
* максимальный срок хранения средств;
* минимальный размер вознаграждения;
* сумма.

На основании входных параметров из базы данных отбираются все записи о вкладах (депозитах), условия которых соответствуют заданным требованиям, а именно срок хранения не превышает максимальный срок сбережения, процентная ставка не менее установленного размера вознаграждения, валюта вклада соответствует выбранной пользователем валюте. Если записи не найдены, то формируется сообщение об отсутствии в базе данных сведений о вкладах (депозитах), условия которых соответствуют заданным параметрам. Если записи найдены, то по каждой найденной записи производится расчет вознаграждения исходя из заданной суммы вклада. Расчет по каждой записи добавляется в список. После обработки всех отобранных из базы данных записей список с результатами расчетов сортируется по наименованию банка и передается в представление для отображения.

Схема алгоритма расчета дохода по вкладам (депозитам) по заданным условиям представлена на чертеже РТДП 4032.052.004.

### 4.2.5 Расчет платежей по кредиту по заданным условиям

Условия расчета задаются пользователем. Входными параметрами для расчетов выступают:

* сумма кредита;
* процент вознаграждения за пользование заемными средствами;
* срок кредитования;
* способ погашения кредита (равными долями, убывающими платежами).

На основании входных параметров за каждый платежный период осуществляется расчет:

* суммы платежа по основному долгу;
* суммы процентов за пользование кредитом;
* суммы остатка основного долга;
* месяца, в который производится погашение.

Расчет за каждый платежный период добавляется в список. После завершения расчетов, платежи за каждый платежный период суммируются, и рассчитывается сумма переплаты. Затем список данных о платежах передается в представление для отображения.

Схема алгоритма расчета платежей по кредиту по заданным условиям представлена на чертеже РТДП 4032.052.005.

# 5 Тестирование разработанного программного продукта

## 5.1 Выбор методов тестирования

Тестирование [программного обеспечения](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) – это процесс исследования, испытания программного продукта, имеющий две различные цели:

* продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
* выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

Сущностью тестирования являются проверочные действия, которые демонстрируют выполнение всех представленных требований к программному обеспечению. При тестировании также надо удостовериться, что все представленные для исправления дефекты могут быть исправлены, и исправления, в свою очередь, не станут основой новых ошибок.

Существует несколько типов тестирования:

* модульное тестирование;
* интеграционное тестирование;
* системное тестирование;
* регрессионное тестирование;
* тестирование производительности и нагрузочные тесты.

В случае модульного тестирования тестируют конкретные программные модули - одну подсистему из всей системы. Тестирование проводит в общем случае разработчик, занимающийся реализацией модуля. Целью тестирования модуля является все же нахождение ошибок, а не установление соответствия требованиям пользователя.

В случае интеграционного тестирования тестируют совместную работу между модулями - проверяют, работают ли объединенные друг с другом модули, и не генерируют ли самостоятельно работающие без ошибок модули совместные ошибки.

В случае системного тестирования тестируют работу системы как единое целое. При тестировании системы методом «черного ящика» рассматривают части системы, непосредственно доступные пользователю (пользовательский интерфейс), без углубления в код. Этот тип тестирования также называют функциональным тестированием.

Регрессионное тестирование – это тестирование любых типов программного обеспечения, которое используют после введенных изменений кода.

Тестирование производительности и нагрузочные тесты предназначены для проверки соответствия данной системы техническим требованиям. Цель тестов производительности - распознать критические места, где может возникнуть перегрузка и потратить время на оптимизацию этих мест.

Для данного программного продукта выбран системный тип тестирования методом «черного ящика». Проведем «Smoke» тест, который содержит набор тестов на выявление явных ошибок. Цель такого тестирования проверить, что после сборки программного продукта нет явных грубых ошибок.

Вывод о работоспособности основных функций делается на основании результатов поверхностного тестирования наиболее важных модулей приложения на предмет возможности выполнения требуемых задач и наличия быстро находимых критических и блокирующих дефектов.

## 5.2 Разработанные тесты

Набор «Smoke» тестов, разработан для того чтобы проверить работу основных функций программы, основанных на разработанной ранее спецификации требований.

Результаты тестирования разработанного программного средства представлены в таблице 6.

Таблица 6 – Тестовый сценарий с набором «Smoke» тестов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Идентификатор | Требование заказчика | Описание теста | Ожидаемый результат | Статус |
| 1 | 2 | 3 | 4 | 5 |
| BA\_01 | Просмотр справочной информации о банках | **Переход на страницу «Банки»**  1. Кликнуть по кнопке «Банки» в главном меню сайта. | 1. В браузере отображается страница «Банки». | Выполнено успешно |
| BA\_02 | Просмотр справочной информации о банковских продуктах | **Переход на страницу «Вклады»**  1. Кликнуть по кнопке «Вклады» в главном меню сайта. | 1. В браузере отображается страница «Вклады». | Выполнено успешно |

Продолжение таблицы 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| BA\_03 | Просмотр справочной информации о банковских продуктах | **Переход на страницу «Кредиты»**  1. Кликнуть по кнопке «Кредиты» в главном меню сайта. | 1. В браузере отображается страница «Кредиты». | Выполнено успешно |
| BA\_04 | Просмотр справочной информации о курсах валют | **Переход на страницу «Курсы валют»**  1. Кликнуть по кнопке «Курсы валют» в главном меню сайта. | 1. В браузере отображается страница «Курсы валют». | Выполнено успешно |
| BA\_05 | Информация о проекте | **Переход на страницу «О проекте»**  1. Кликнуть по кнопке «О проекте» в главном меню сайта. | 1. В браузере отображается страница «О проекте». | Выполнено успешно |
| BA\_06 | Просмотр справочной информации о структурных подразделениях банков | **Просмотр информации об отделениях и филиалах «Альфа-банка»**  1. Кликнуть по кнопке «Банки» в главном меню сайта.  2. Кликнуть на ссылку «Подробнее» в столбце «Описание». | 1. В браузере отображается страница «Банки».  2. В браузере отображается страница «Отделения и филиалы банка». | Выполнено успешно |
| BA\_07 | Просмотр справочной информации о структурных подразделениях банков | **Переход на официальную страницу «Альфа-банка» в сети Интернет**  1. Кликнуть по кнопке «Банки» в главном меню сайта.  2. Кликнуть на ссылку www.alfabank.by в столбце «Вэб-страница». | 1. В браузере отображается страница «Банки».  2. В браузере отображается страница «Альфа-банк». | Выполнено успешно |

Продолжение таблицы 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| BA\_08 | Расчет дохода по вкладам по заданным условиям | **Выбрать валюту вклада**  1. Кликнуть по кнопке «Вклады» в главном меню сайта.  2. В форме выбора условий вклада кликнуть по полю «Валюта вклада».  3. В раскрывающемся списке выбрать «Евро». | 1. В браузере отображается страница «Вклады».  2. Поле «Валюта вклада» приняло вид раскрытого списка.  3. В поле «Валюта вклада» надпись «Евро». | Выполнено успешно |
| BA\_09 | Расчет дохода по вкладам по заданным условиям | **Ввести сумму вклада**  1. Кликнуть по кнопке «Вклады» в главном меню сайта.  2. В форме выбора условий вклада ввести в поле «Сумма» 1 000 000 | 1. В браузере отображается страница «Вклады».  2. В поле «Сумма» отображается число 1 000 000. | Выполнено успешно |
| BA\_10 | Расчет дохода по вкладам по заданным условиям | **Выбрать срок вклада**  1. Кликнуть по кнопке «Вклады» в главном меню сайта.  2. В форме выбора условий вклада кликнуть по полю «Срок».  3. В раскрывающемся списке выбрать «1 год». | 1. В браузере отображается страница «Вклады».  2. Поле «Срок» приняло вид раскрытого списка.  3. В поле «Срок» надпись «1 год». | Выполнено успешно |
| BA\_11 | Расчет дохода по вкладам по заданным условиям | **Ввести процентную ставку по вкладу**  1. Кликнуть по кнопке «Вклады» в главном меню сайта.  2. В форме выбора условий вклада с помощью кнопки увеличения значения счетчика задать значение ставки равное 30 процентов. | 1. В браузере отображается страница «Вклады».  2. В поле «Ставка» отображается число 30. | Выполнено успешно |

Продолжение таблицы 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| BA\_12 | Расчет платежей по кредитам по заданным условиям | **Ввести сумму кредита**  1. Кликнуть по кнопке «Кредиты» в главном меню сайта.  2. В форме условий кредита ввести в поле «Сумма» 10 000 000. | 1. В браузере отображается страница «Кредиты».  2. В поле «Сумма» отображается число 10 000 000. | Выполнено успешно |
| BA\_13 | Расчет платежей по кредитам по заданным условиям | **Выбрать срок кредита**  1. Кликнуть по кнопке «Кредиты» в главном меню сайта.  2. В форме условий кредита кликнуть по полю «Срок».  3. В раскрывающемся списке выбрать «3 года». | 1. В браузере отображается страница «Кредиты».  2. Поле «Срок» приняло вид раскрытого списка.  3. В поле «Срок» надпись «3 года». | Выполнено успешно |
| BA\_14 | Расчет платежей по кредитам по заданным условиям | **Ввести процентную ставку по кредиту**  1. Кликнуть по кнопке «Кредиты» в главном меню сайта.  2. В форме условий кредита с помощью кнопки увеличения значения счетчика задать значение ставки равное 50 процентов. | 1. В браузере отображается страница «Кредиты».  2.В поле «Ставка» отображается число 50. | Выполнено успешно |
| BA\_15 | Просмотр справочной информации о курсах валют | **Переход на страницу «Курсы валют» с помощью кнопки «Все курсы» на главной странице**  1. Кликнуть по кнопке «Все куры» на главной странице в области «Курсы валют». | 1. В браузере отображается страница «Курсы валют». | Выполнено успешно |
| BA\_16 | Просмотр справочной информации о банках | **Переход на страницу «Банки»**  1. Кликнуть по кнопке «Все банки» на главной странице в области «Банки». | 1. В браузере отображается страница «Банки». | Выполнено успешно |

Продолжение таблицы 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| BA\_17 | Расчет дохода по вкладам по заданным условиям | **Рассчитать доход по вкладам**  1. Кликнуть по кнопке «Вклады» в главном меню сайта.  2. В форме выбора условий задать условия для расчета (Валюта вклада – белорусские рубли, Ставка от 20%, Срок до 1 года, Сумма 1 000 000).  3. Нажать кнопку «Рассчитать». | 1. В браузере отображается страница «Вклады».  2. Все поля формы выбора условий по вкладу заполнены заданными значениями.  3. В браузере отображается таблица, содержащая выбранные по заданным условиям вклады с указанием суммы дохода | Выполнено успешно |
| BA\_18 | Расчет платежей по кредитам по заданным условиям | **Отображение рассчитанных платежей по кредитам**  1. Кликнуть по кнопке «Кредиты» в главном меню сайта.  2. В форме выбора условий кредита задать условия для расчета (Ставка от 50%, Срок 2 года, Сумма 10 000 000).  3. Нажать кнопку «Рассчитать». | 1. В браузере отображается страница «Кредиты».  2. Все поля формы выбора условий по кредиту заполнены заданными значениями.  3. В браузере отображается таблица, содержащая суммы платежей по погашению кредита за каждый месяц | Выполнено успешно |

## 5.3 Результаты тестирования

В результате проведения «Smoke» тестов явных ошибок основных функциональных возможностей наиболее важных модулей разработанного приложения не выявлено. Приложение является работоспособным и соответствует требованиям.

# Заключение

В процессе работы над данным дипломным проектом мною изучены теоретические основы проектирования и разработки веб-ориентированных приложений на основе .NET Framework. При реализации проекта выполнены следующие задачи:

* проведен анализ вариантов получения справочной информации о банковских продуктах и услугах для физических лиц;
* разработана модель данных для хранения информации о банках и их продуктах в базе данных;
* спроектировано и разработано программное средство, обеспечивающее доступ к справочной информации о белорусских банках и их продуктах услугах для физических лиц;
* в разработанном приложении реализована возможность для потребителя самостоятельно рассчитать доход по вкладу (депозиту) и сумму ежемесячных взносов в счет погашения кредита;
* проведено тестирование разработанного программного средства.

Программное средство представляет собой веб-приложение. Для хранения и обработки данных используется СУБД Microsoft SQL Server. При разработке приложения, для доступа к данным использована платформа Entity Framework, которая является объектно-ориентированной технологией доступа к данным и позволяет эффективно взаимодействовать данными.

Воспользоваться информацией и функциями приложения можно посредством доступа в сеть Интернет.

Внедрение программного средства позволит повысить осведомленность потребителей о банках и их продуктах и услугах для физических лиц, а также снизить затраты времени на получение справочной информации и поиск подходящих предложений белорусских банков для потенциального клиента.

Дальнейшее развитие разработанного программного продукта заключается в расширении его функциональности. Программное средство может быть использовано как инструмент, направленный на повышение финансовой грамотности населения. Эту задачу можно решить путем увеличения перечня предоставляемой пользователю справочной информации, размещения аналитических обзоров и статей на тему развития розничных банковских продуктов и услуг. Обеспечение возможности зарегистрироваться на сайте позволит расширить функциональность программного продукта для категории зарегистрированных пользователей. Такой подход обеспечит возможность сбора отзывов пользователей и получения обратной связи о работе приложения и направлении его развития.

# Список использованных источников

1. Постановление Правления Национального Банка Республики Беларусь от 01февраля 2011 №35 «Об утверждении Инструкции о порядке осуществления валютно-обменных операций с участием физических лиц и организации работы обменных пунктов, касс при проведении таких операций.

2. Банковский кодекс Республики Беларусь 25 октября 2000 г. №441-3.

3. Отчет о выполнении научно-исследовательской работы по теме: «Оценка и анализ доступа населения к финансам в Республике Беларусь» // М: Национальная академия наук Беларуси. Государственное научное учреждение «Институт социологии НАН Беларуси», 2015 г.

4. Результаты национального исследования по теме: «Финансовая грамотность населения Республики Беларусь: факты и выводы» // М: Национальная академия наук Беларуси. Государственное научное учреждение «Институт социологии НАН Беларуси», 2013 г.

5. Compare Microsoft SQL Server to MySQL [Электронный ресурс]. – Microsoft, 2016. – Режим доступа: http://www.microsoft.com/sqlserver/2008/ru-ru/compare-mysql.aspx.

6. «Информационный банк «Все о финансах». [Электронный ресурс]. – Режим доступа: http:// <http://infobank.by/> Дата доступа: 10.03.2016.

7. «Все банки Беларуси». [Электронный ресурс]. – Режим доступа: http:// <http://select.by/> Дата доступа: 12.03.2016.

8. Официальный сайт ЗАО «БТА Банк». [Электронный ресурс]. – Режим доступа: http:// <http://www.btabank.by/> Дата доступа: 10.03.2016.

9. Библиотека MSDN (по-русски) [Электронный ресурс] / ADO.NET. – Microsoft, 2016. – Режим доступа: http://msdn.microsoft.com/ru-ru/library/-e80y5yhx.aspx.

10. Горев, А. Эффективная работа с СУБД / А. Горев, С. Макашарипов, Р.Ахаян. – СПб.: Питер, 2005. – 704 с.

11. Виейра, Р. Программирование баз данных Microsoft SQL Server 2005. Базовый курс / Р. Виейра. Мосвка: Вильямс, 2008 – 833 с.

12. Википедия – свободная энциклопедия [Электронный ресурс] / Представление (базы данных). – Microsoft, 2016. – Режим доступа: http://ru.wikipedia.org/wiki/Представление\_(базы\_данных).

13. Малик, С. Microsoft ADO.NET 2.0 для профессионалов / С. Малик. – Москва: Вильямс, 2006. – 560 с.

14. Библия C#. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011. – 560 с.

15. Освой самостоятельно SQL. 10 минут на урок, 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005 – 288 с.

16. С. Сомасегар, С. Гатри Руководство Microsoft по проектированию архитектуры приложения. 2-е изд. 2009г – 259с.

# ПРИЛОЖЕНИЕ А (обязательное) Текст программы

namespace AllBanksDiploma.Models

{

public class BankContext : DbContext

{

public DbSet<Bank> Banks { get; set; }

public DbSet<Branch> Branches { get; set; }

public DbSet<Deposit> Deposits { get; set; }

public DbSet<Loan> Loans { get; set; }

public DbSet<ExchangeRate> ExchangeRates { get; set; }

public DbSet<Currency> Currency { get; set; }

}

}

namespace AllBanksDiploma.Models

{

public class Bank

{

public int BankId { get; set; } //id Банка

public string BankName { get; set; } //Название

public string Description { get; set; } //Описание

public string WebPage { set; get; } //Вэб-страница

public byte[] ImageData { set; get; } // Изображение

public string ImageMimeType { set; get; }

public virtual ICollection<Branch> Branches { get; set; }

public virtual ICollection<Deposit> Deposits { get; set; }

public virtual ICollection<Loan> Loans { get; set; }

public virtual ICollection<ExchangeRate> ExchangeRates { get; set; }

}

}

namespace AllBanksDiploma.Models

{

public class Branch

{

public int BranchId { get; set; }

public int BankId { get; set; }

public string BranchName { get; set; }

public string City { get; set; }

public string Street { get; set; }

public string Building { get; set; }

public string Phone { get; set; }

public virtual Bank Bank { get; set; }

}

}

namespace AllBanksDiploma.Models

{

public class Currency

{

public int CurrencyId { get; set; }

public int CurrencyCode { get; set; }

public string CurrencyMnemonic { get; set; }

public string CurrencyName { get; set; }

public virtual ICollection<ExchangeRate> ExchangeRates { get; set; }

}

}

namespace AllBanksDiploma.Models

{

public class ExchangeRate

{

public int ExchangeRateId { get; set; }

public int CurrencyId { get; set; }

public int BankId { get; set; }

public double BuyRate { get; set; }

public double SellRate { get; set; }

public virtual Currency Currency { get; set; }

public virtual Bank Bank { get; set; }

}

}

namespace AllBanksDiploma.Models

{

public class Deposit

{

public int DepositId { get; set; }

public int BankId { get; set; }

public string DepositName { get; set; }

public string TermType { get; set; }

public int Term { get; set; }

public double DepositRate { get; set; }

public int CurrCode { get; set; }

public virtual Bank Bank { get; set; }

}

}

namespace AllBanksDiploma.Models

{

public class Loan

{

public int LoanId { get; set; }

public int BankId { get; set; }

public string LoanName { get; set; }

public string TermType { get; set; }

public int Term { get; set; }

public double LoanRate { get; set; }

public virtual Bank Bank { get; set; }

}

}

namespace AllBanksDiploma.Controllers

{

public class BankController : Controller

{

BankContext db = new BankContext();

public ActionResult List()

{

return View(db.Banks);

}

[HttpGet]

public ActionResult Details(int id)

{

var branches = db.Branches.Include(b => b.Bank).Where(i => i.Bank.BankId == id);

ViewBag.Branches = branches.ToList();

foreach(var g in branches.GroupBy(b => b.Bank.BankName))

{

ViewBag.Bank = g.Key;

}

return View();

}

}

}

namespace AllBanksDiploma.Controllers

{

public class ProductController : Controller

{

BankContext db = new BankContext();

public ActionResult Deposits()

{

var deposits = db.Deposits.Include(b => b.Bank).OrderBy(o => o.BankId);

ViewBag.Deposits = deposits.ToList();

return View();

}

public ActionResult Loans()

{

var loans = db.Loans.Include(b => b.Bank).OrderBy(o => o.BankId);

ViewBag.Loans = loans.ToList();

return View();

}

public ActionResult CurrencyExchange()

{

var groups = db.ExchangeRates.Include(b => b.Currency).Include(b => b.Bank);

List<CurrencyEx> currencyExchange = new List<CurrencyEx>();

foreach (var g in groups.GroupBy(b => b.Bank.BankName))

{

CurrencyEx ce = new CurrencyEx();

ce.Bank = g.Key;

foreach (var p in g)

{

if (p.CurrencyId == 2)

{

ce.BuyUSD = (decimal)p.BuyRate;

ce.SellUSD = (decimal)p.SellRate;

}

else if (p.CurrencyId == 3)

{

ce.BuyEUR = (decimal)p.BuyRate;

ce.SellEUR = (decimal)p.SellRate;

}

else if (p.CurrencyId == 1)

{

ce.BuyRUB = (decimal)p.BuyRate;

ce.SellRUB = (decimal)p.SellRate;

}

}

currencyExchange.Add(ce);

}

ViewBag.CurrencyExchange = currencyExchange.ToList();

return View();

}

[HttpPost]

public ActionResult CalcDepositIncome(DepositConditions condition)

{

List<DepositIncome> Result = new List<DepositIncome>();

var deposits = db.Deposits

.Include(b => b.Bank)

.Where(d => d.CurrCode == condition.Currency)

.Where(d => d.DepositRate >= condition.Rate)

.Where(d => d.Term <= condition.Term);

foreach (var i in deposits.ToList())

{

DepositIncome tmp = new DepositIncome();

tmp.BankName = i.Bank.BankName;

tmp.DepositName = i.DepositName;

tmp.Term = i.Term;

tmp.Rate = i.DepositRate;

tmp.Sum = condition.Sum;

tmp.Income = CalcDepositIncome(condition.Sum, i.Term, i.DepositRate);

Result.Add(tmp);

}

return View(Result);

}

[HttpPost]

public ActionResult CalcLoanPayments(LoanCondition condition)

{

List<LoanPayments> Result = new List<LoanPayments>();

decimal sum = condition.Sum;

for (int i = 1; i < condition.Term + 1; i++)

{

LoanPayments payment = new LoanPayments();

var date = DateTime.Today;

date = date.AddMonths(i);

payment.Date = string.Format("{0}-{1}", date.Month, date.Year);

Result.Add(payment);

}

return View(Result);

}

private decimal CalcDepositIncome(decimal sum, int term, double rate)

{

decimal income = 0;

double deposit = (double)sum;

double q = rate / 100 / 12 + 1;

for (int i = 0; i < term; i++)

{

deposit \*= q;

}

return income=(decimal)(deposit-(double)sum);

}

private decimal CalcLoanPayments (decimal sum, int term, double rate)

{

return sum;

}

}

}

@{

ViewBag.Title = "Главная страница";

}

<div id="myCarousel" class="carousel slide" data-ride="carousel">

<!-- Indicators -->

<ol class="carousel-indicators">

<li data-target="#myCarousel" data-slide-to="0" class="active"></li>

<li data-target="#myCarousel" data-slide-to="1"></li>

<li data-target="#myCarousel" data-slide-to="2"></li>

<li data-target="#myCarousel" data-slide-to="3"></li>

<li data-target="#myCarousel" data-slide-to="4"></li>

</ol>

<!-- Wrapper for slides -->

<div class="carousel-inner" role="listbox">

<div class="item active">

<img src="~/Content/images/banks.jpg" />

<div class="carousel-caption">

<h1><span style="color: azure">БАНКИ</span></h1>

<h3><span style="color: azure">Подробная информация о банках.</span></h3>

</div>

</div>

<div class="item">

<img src="~/Content/images/deposits.jpg" />

<div class="carousel-caption">

<h1><span style="color: darkslategray">ВКЛАДЫ</span></h1>

<h3><span style="color: darkslategray">Получить дополнителный доход легко!</span></h3>

</div>

</div>

<div class="item">

<img src="~/Content/images/loans.jpg" />

<div class="carousel-caption">

<h1><span style="color: azure">КРЕДИТЫ</span></h1>

<h3><span style="color: azure">Хватит копить, пора покупать!</span></h3>

</div>

</div>

<div class="item">

<img src="~/Content/images/curency.jpg" />

<div class="carousel-caption">

<h1><span style="color: darkslategray">КУРСЫ ВАЛЮТ</span></h1>

<h3><span style="color: darkslategray">Лучшие курсы здесь!</span></h3>

</div>

</div>

<div class="item">

<img src="~/Content/images/card.jpg" />

<div class="carousel-caption">

<h1><span style="color: azure">ПЛАТЕЖНЫЕ КАРТЫ</span></h1>

<h3><span style="color: azure">Плати удобно!</span></h3>

</div>

</div>

</div>

<!-- Left and right controls -->

<a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev">

<span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="right carousel-control" href="#myCarousel" role="button" data-slide="next">

<span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>

<span class="sr-only">Next</span>

</a>

</div>

<div class="row">

<div class="col-md-4">

<h1>Курсы валют</h1>

<table class="table">

<thead class="currensy">

<tr>

<th>Валюта</th>

<th>Покупка</th>

<th>Продажа</th>

</tr>

</thead>

<tbody>

<tr>

<td>USD</td>

<td>19100</td>

<td>19250</td>

</tr>

<tr>

<td>EUR</td>

<td>21700</td>

<td>21950</td>

</tr>

<tr>

<td>RUB</td>

<td>293.00</td>

<td>300.00</td>

</tr>

</tbody>

</table>

<p><a class="btn btn-info" href="">Все курсы &raquo;</a></p>

</div>

<div class="col-md-4">

<h1>Банки</h1>

<p>Здесь все банки Ркспублики Беларусь - как на ладони. Все, что Вы как клиент хотите узнать

о финансовом учреждении Вы найдете именно здесь. Вы можете увидеть краткую справку о всех

белорусских банках: узнать историю, ознакомиться с перечнем совершаемых операций</p>

<p><a class="btn btn-info" href="">Выбрать банк &raquo;</a></p>

</div>

<div class="col-md-4">

<h1>Продукты и услуги</h1>

<p>Вклады (Депозиты) и Кредиты</p>

<p><a class="btn btn-info" href="">Подробнее &raquo;</a></p>

</div>

</div>

@using AllBanksDiploma.Models

@model IEnumerable<Bank>

@{

ViewBag.Title = "Банки";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="title" align="center">

<h1><b>Информация о банках Беларуси</b></h1>

</div>

<div class="table-responsive">

<table class="table table-hover">

<thead style="font-size: 14pt">

<tr>

<th></th>

<th>Наименование Банка</th>

<th>Описание</th>

<th>Официальный сайт</th>

</tr>

</thead>

<tbody>

@foreach (Bank b in Model)

{

<tr>

<td>@b.ImageData</td>

<td>@b.BankName</td>

<td>@b.Description<a href="/Bank/Details/@b.BankId"> Подробнее...</a></td>

<td align="center"><a href="@b.WebPage">@b.WebPage</a></td>

</tr>

}

</tbody>

</table>

</div>

@{

ViewBag.Title = "Подробно о банке";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="container">

<h1><b>@ViewBag.Bank</b></h1>

</div>

<div class="table-responsive">

<table class="table table-hover">

@\*<thead style="font-size: 14pt">

<tr>

<th>Наименование подразделения</th>

<th>Адрес</th>

<th>Телефон</th>

</tr>

</thead>\*@

<tbody>

@foreach (var i in ViewBag.Branches)

{

<tr>

<td>@i.BranchName</td>

<td>@i.City, @i.Street, дом @i.Building</td>

<td>@i.Phone</td>

</tr>

}

</tbody>

</table>

</div>

@{

ViewBag.Title = "Вклады(Депозиты)";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="title" align="center">

<h1><b>Вклады в белорусских рублях и иностранной валюте</b></h1>

</div>

<div class="container">

@Html.Partial("CalcDepositIncomePartial")

</div>

<div class="table-responsive">

@Html.Partial("DepositsPartial")

</div>

@{

ViewBag.Title = "Кредиты";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="title" align="center">

<h1><b>Кредиты для частных лиц</b></h1>

</div>

<div class="container">

@Html.Partial("CalcLoanPaymentsPartial")

</div>

<hr />

<div class="table-responsive">

@Html.Partial("LoansPartial")

</div>

@{

ViewBag.Title = "Currency";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="title" align="center">

<h1><b>Курсы обмена валют</b></h1>

</div>

<hr />

<div class="table-responsive">

<table class="table table-hover">

<thead style="font-size: 14pt">

<tr>

<th rowspan="2" , style="text-align:center">Банки</th>

<th colspan="2" , style="text-align:center">USD</th>

<th colspan="2" , style="text-align:center">EUR</th>

<th colspan="2" , style="text-align:center">RUB</th>

</tr>

<tr>

<th>Банк покупает</th>

<th>Банк продает</th>

<th>Банк покупает</th>

<th>Банк продает</th>

<th>Банк покупает</th>

<th>Банк продает</th>

</tr>

</thead>

<tbody>

@foreach (var i in ViewBag.CurrencyExchange)

{

<tr>

<td>@i.Bank</td>

<td>@i.BuyUSD</td>

<td>@i.SellUSD</td>

<td>@i.BuyEUR</td>

<td>@i.SellEUR</td>

<td>@i.BuyRUB</td>

<td>@i.SellRUB</td>

</tr>

}

</tbody>

</table>

</div>

@model IEnumerable<AllBanksDiploma.Models.DepositIncome>

@{

ViewBag.Title = "Доход по вкладу";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="title" align="center">

<h1><b>Вклады в белорусских рублях и иностранной валюте</b></h1>

</div>

<div class="container">

@Html.Partial("CalcDepositIncomePartial")

</div>

<div class="table-responsive">

<table class="table table-hover">

<thead style="font-size: 13pt">

<tr>

<th>Банк</th>

<th>Название вклада</th>

<th>Срок</th>

<th>Ставка</th>

<th>Сумма вклада</th>

<th>Доход</th>

</tr>

</thead>

<tbody>

@foreach (var item in Model)

{

<tr>

<td>@Html.DisplayFor(modelItem => item.BankName)</td>

<td>@Html.DisplayFor(modelItem => item.DepositName)</td>

<td>@Html.DisplayFor(modelItem => item.Term)</td>

<td>@Html.DisplayFor(modelItem => item.Rate)</td>

<td>@Html.DisplayFor(modelItem => item.Sum)</td>

<td style="font-size: 13pt">@Html.DisplayFor(modelItem => item.Income)</td>

</tr>

}

</tbody>

</table>

</div>

@model IEnumerable<AllBanksDiploma.Models.LoanPayments>

@{

ViewBag.Title = "CalcLoanPayments";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>CalcLoanPayments</h2>

<p>

@Html.ActionLink("Create New", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.Date)

</th>

<th>

@Html.DisplayNameFor(model => model.Debt)

</th>

<th>

@Html.DisplayNameFor(model => model.Interest)

</th>

<th>

@Html.DisplayNameFor(model => model.Total)

</th>

<th>

@Html.DisplayNameFor(model => model.Balance)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Date)

</td>

<td>

@Html.DisplayFor(modelItem => item.Debt)

</td>

<td>

@Html.DisplayFor(modelItem => item.Interest)

</td>

<td>

@Html.DisplayFor(modelItem => item.Total)

</td>

<td>

@Html.DisplayFor(modelItem => item.Balance)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { /\* id=item.PrimaryKey \*/ }) |

@Html.ActionLink("Details", "Details", new { /\* id=item.PrimaryKey \*/ }) |

@Html.ActionLink("Delete", "Delete", new { /\* id=item.PrimaryKey \*/ })

</td>

</tr>

}

</table>

<table class="table table-hover">

<thead style="font-size: 14pt">

<tr>

<th>Банк/<p>Название вклада</p></th>

<th>Валюта</th>

<th>Срок</th>

<th>Процентная ставка</th>

</tr>

</thead>

<tbody>

@foreach (var i in ViewBag.Deposits)

{

<tr>

<td><b>@i.Bank.BankName/</b><p>"@i.DepositName"</p></td>

@if (i.CurrCode == 974)

{

<td>Белорусские рубли</td>

}

else if (i.CurrCode == 840)

{

<td>Доллары США</td>

}

else if (i.CurrCode == 978)

{

<td>Евро</td>

}

else if (i.CurrCode == 643)

{

<td>Российские рубли</td>

}

@if (i.TermType == "m")

{

<td>@i.Term мес.</td>

}

else if (i.TermType == "d")

{

<td>@i.Term дн.</td>

}

<td>@i.DepositRate %</td>

</tr>

}

</tbody>

</table>

<form class="form-horizontal" role="form" method="post" action="/Product/CalcLoanPayments">

<div class="form-group">

<label class="control-label col-sm-2" for="sum">Сумма кредита</label>

<div class="col-sm-10">

<input class="form-control" name="sum" id="sum" value="1000000" pattern="^[0-9]+$" />

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="rate">Процентная ставка</label>

<div class="col-sm-10">

<input class="form-control" name="rate" id="rate" type="number" min="0" max="100" value="37" />

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="term">Срок кредита</label>

<div class="col-sm-10">

<select class="form-control" name="term" id="term">

<option value="6" selected>6 мес.</option>

<option value="12">1 год</option>

<option value="24">2 года</option>

<option value="36">3 года</option>

<option value="60">5 лет</option>

<option value="84">7 лет</option>

</select>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<div class="radio">

<label class="radio"><input type="radio" name="calctype" value="1">Равными платежами</label>

</div>

<div class="radio">

<label class="radio"><input type="radio" name="calctype" value="0" checked>По фактическому остатку</label>

</div>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<button type="submit" class="btn btn-default">Рассчитать платежи</button>

</div>

</div>

</form>

<table class="table table-hover">

<thead style="font-size: 14pt">

<tr>

<th>Банк/<p>Название кредита</p></th>

<th>Срок</th>

<th>Процентная ставка</th>

</tr>

</thead>

<tbody>

@foreach (var i in ViewBag.Loans)

{

<tr>

<td><b>@i.Bank.BankName/</b><p>"@i.LoanName"</p></td>

@if (i.TermType == "m")

{

<td>@i.Term мес.</td>

}

else if (i.TermType == "d")

{

<td>@i.Term дн.</td>

}

<td>@i.LoanRate %</td>

</tr>

}

</tbody>

</table>

<form class="form-horizontal" role="form" method="post" action="/Product/CalcLoanPayments">

<div class="form-group">

<label class="control-label col-sm-2" for="sum">Сумма кредита</label>

<div class="col-sm-10">

<input class="form-control" name="sum" id="sum" value="1000000" pattern="^[0-9]+$" />

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="rate">Процентная ставка</label>

<div class="col-sm-10">

<input class="form-control" name="rate" id="rate" type="number" min="0" max="100" value="37" />

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="term">Срок кредита</label>

<div class="col-sm-10">

<select class="form-control" name="term" id="term">

<option value="6" selected>6 мес.</option>

<option value="12">1 год</option>

<option value="24">2 года</option>

<option value="36">3 года</option>

<option value="60">5 лет</option>

<option value="84">7 лет</option>

</select>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<div class="radio">

<label class="radio"><input type="radio" name="calctype" value="1">Равными платежами</label>

</div>

<div class="radio">

<label class="radio"><input type="radio" name="calctype" value="0" checked>По фактическому остатку</label>

</div>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<button type="submit" class="btn btn-default">Рассчитать платежи</button>

</div>

</div>

</form>