Министерство науки и высшего образования Российской Федерации

Рубцовский индустриальный институт (филиал) ФГБОУ ВО

«Алтайский государственный технический университет

им. И. И. Ползунова»

Факультет Технический

*наименование подразделения*

Кафедра Прикладная математика

*наименование кафедры*

Отчёт защищён с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

*подпись руководителя инициалы, фамилия*

«\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_\_\_\_\_г.

ОТЧЁТ

о прохождении учебной практики

*наименование вида практики*

на РИИ АлтГТУ

*наименование организации*

Обучающегося гр. ИВТ-81 Е.А. Беляков

. *индекс группы подпись инициалы, фамилия*

Руководитель от

кафедры доцент, к. т. н. И.Б. Шульман

*должность, ученое звание подпись инициалы, фамилия*

2020

СОДЕРЖАНИЕ

Введение

Задача необходима различным предприятиям для планирования календаря на год вперед. Она позволяет указывать праздничные, рабочие и выходные дни. Составлять список дел и назначать мероприятие на конкретный день.

Так же хранить данные в архиве и переносить список дел на следующий год.

Цель работы: Создать программу для базы данных производственный календарь.

Задачи:

* Ознакомиться с предметной областью.
* Составить инфологическую модель.
* Составить даталогическую модель.
* Описать входные и выходные данные.
* Подготовить алгоритм решения.
* Написать и отладить программу.

1. ПОСТАНОВКА ЗАДАЧИ
   1. Характеристика задачи
      1. Программа «Производственный календарь» предназначена для составления календаря на текущий год с указанием рабочих, выходных и праздничных дней.

Программа предоставляет пользователю следующие возможности:

* Ввод данных о праздничных днях;
* Ввод данных о списке дел;
* Ввод данных о календаре;
* Ввод данных о мероприятиях на конкретный день;
* Перенос списка дел на другой год;
* Перенос мероприятий в архивную таблицу;
* Изменение и удаление введенных данных;
* Создание отчётов и их печать;
* Вывод данных из таблиц на экран;
* Вывод справки о программе.
  + 1. Пользоваться задачей будут сотрудники различных предприятий.
    2. Периодичность решения задачи – по запросу пользователя.

Ограничений на продолжительность работы нет

* + 1. Связь данной задачи с другими не предполагается.
    2. Программа функционирует в диалоговом режиме.
    3. Прекращение автоматизированного решения задачи происходит при отключении источника электропитания или в случае поломки ЭВМ.
  1. **Инфологическая модель**

Инфологическая модель представлена на рисунке 1.

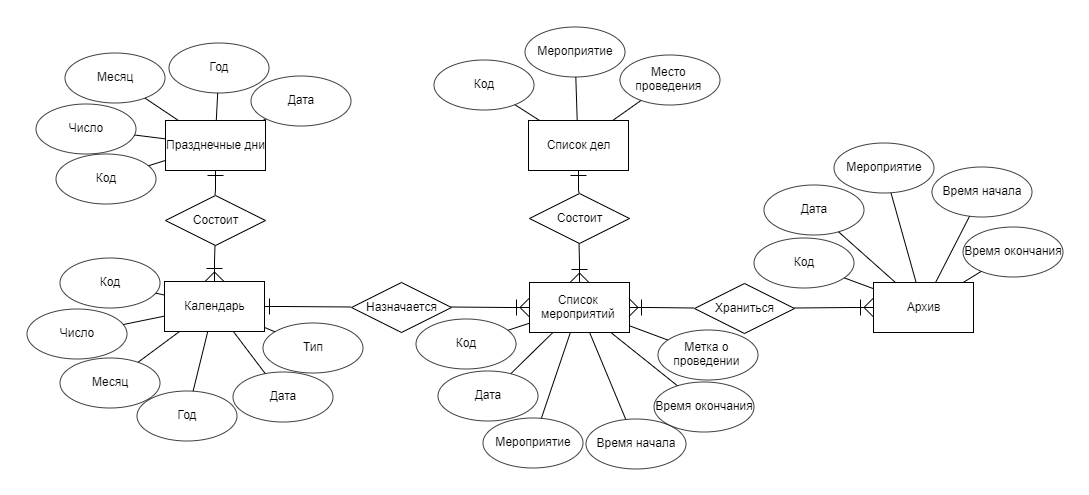


Рисунок 1 – ER-диаграмма

* 1. Даталогическая модель

Даталогическая модель представлена на рисунке 2



Рисунок 2 - Даталогическая модель

* 1. Выходная информация
     1. Перечень выходных данных содержится в таблице 1

Таблица 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Идентификатор | Форма представления | Наименование | Периодичность | Получатель | Число строк |
| F001 | Файл БД | DBProductionCalendar.pdb | По запросу | Пользователь | - |
| F002 | Текстовой файл | CasesDate.txt | По запросу | Пользователь | - |
| V001 | Видеограмма | Список праздничных дней | По запросу | Пользователь | 62 |
| V002 | Видеограмма | Список дел | По запросу | Пользователь | 62 |
| V003 | Видеограмма | Календарь | По запросу | Пользователь | 365 |
| V004 | Видеограмма | Список мероприятий | По запросу | Пользователь | 62 |
| V005 | Видеограмма | Архив | По запросу | Пользователь | 62 |
| V010 | Видеограмма | Параметры | По запросу | Пользователь | 10 |

* + 1. Описание структурных единиц файла F001 содержится в таблице 2

Таблица 2

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| DBProductionCalendar.pdb | | F001 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Список праздничных дней | Holidays | Х(30) | «А»- «Я»; «а»- «я»; «0»–«9» |
| Список дел | ListofCases | Х(80) | «А»- «Я»; «а»- «я»; «0»–«9»; «.» |
| Календарь | Calendar | Х(40) | «А»- «Я»; «а»- «я» «0»–«9» |
| Список мероприятий | EventDate | Х(120) | «А»- «Я»; «а»- «я» «0»–«9»; «.» |
| Архив | ArchiveEventDate | Х(110) | «А»- «Я»; «а»- «я» «0»–«9»; «.» |

* + 1. Описание структурных единиц видеограммы V001 содержится в таблице 3

Таблица 3

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| Список праздничных дней | | V001 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Код | Id | 9(4) | «0»–«9» |
| Число | NumDay | 9(2) | «0»-«9» |
| Месяц | NameMonth | A(8) | «А»-«Я»; «а»- «я» |
| Год | Year | 9(4) | «0»-«9» |
| Полная дата | FullDate | X(16) | «А»-«Я»; «а»-«я»;  «0»-«9» |

* + 1. Описание структурных единиц видеограммы V002 содержится в таблице 4

Таблица 4

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| Список дел | | V002 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Код | Id | 9(4) | «0»–«9» |
| Мероприятие | NameEvent | X(50) | «А»- «Я»; «а»- «я»,  «0»–«9»; «.» |
| Место проведения | PlaceEvent | X(30) | «А»- «Я»; «а»- «я»,  «0»–«9»; «.» |

* + 1. Описание структурных единиц видеограммы V003 содержится в таблице 5

Таблица 5

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| Календарь | | V003 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Код | Id | 9(4) | «0»–«9» |
| Число | NumDay | 9(2) | «0»-«9» |
| Месяц | NameMonth | A(8) | «А»- «Я»; «а»- «я» |
| Год | Year | 9(4) | «0»-«9» |
| Полная дата | FullDate | X(16) | «А»- «Я»; «а»- «я»,  «0»-«9» |
| Тип | TypeofDay | A(15) | А»- «Я»; «а»- «я» |

* + 1. Описание структурных единиц видеограммы V004 содержится в таблице 6

Таблица 6

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| Список мероприятий | | V004 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Код | Id | 9(4) | «0»–«9» |
| Дата | vCalendar | X(16) | «А»- «Я»; «а»- «я»; «0»–«9» |
| Мероприятие | vListofCases | X(80) | «А»- «Я»; «а»- «я»,  «0»–«9»; «.» |
| Время начала | StartTime | X(5) | «0»–«9», «:» |
| Время окончания | EndTime | X(5) | «0»–«9»; «:» |
| Метка о проведении | MarkEvent | 9(1) | «0 – 1» |

* + 1. Описание структурных единиц видеограммы V005 содержится в таблице 7

Таблица 7

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| Архив | | V005 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Код | Id | 9(4) | «0»–«9» |
| Дата | vCalendar | X(16) | «А»-«Я»; «а»-«я»;  «0»–«9» |
| Мероприятие | vListofCases | X(80) | «А»-«Я»; «а»-«я»,  «0»–«9»; «.» |
| Время начала | StartTime | X(5) | «0»–«9»; «:» |
| Время окончания | EndTime | X(5) | «0»–«9»; «:» |

* 1. Входная информация
     1. Перечень входных сообщений содержится в таблице 8

Таблица 8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Идентификатор | Форма представления | Наименование | Периодичность | Получатель | Число строк |
| F001 | Файл БД | DBProductionCalendar.pdb | По запросу | Пользователь | - |
| V006 | Видеокадр | Данные о праздничных днях | По запросу | Пользователь | 26 |
| V007 | Видеокадр | Данные о списке дел | По запросу | Пользователь | 26 |
| V008 | Видеокадр | Данные о календаре | По запросу | Пользователь | 60 |
| V009 | Видеокадр | Данные о мероприятие на конкретный день | По запросу | Пользователь | 46 |
| V011 | Видеокадр | Данные для печати | По запросу | Пользователь | 60 |

* + 1. Описание структурных единиц файла F001 содержится в таблице 1
    2. Описание структурных единиц видеокадра V006 содержится в таблице 9

Таблица 9

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| Данные о праздничных днях | | V006 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Число | NumDay | 9(2) | «0»–«9» |
| Месяц | Month | A(8) | «А»- «Я»; «а»- «я»; |
| Год | Year | 9(4) | «0»–«9» |

* + 1. Описание структурных единиц видеокадра V007 содержится в таблице 10

Таблица 10

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| Данные о списке дел | | V007 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Мероприятие | NameEvent | X(50) | «А»- «Я»; «а»- «я»;  «0»–«9»; «.» |
| Место проведения | PlaceEvent | X(30) | «А»- «Я»; «а»- «я»;  «0»–«9»; «.» |

* + 1. Описание структурных единиц видеокадра V008 содержится в таблице 11

Таблица 11

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| Данные о календаре | | V008 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Число | NumDay | 9(2) | «0»–«9» |
| Месяц | Month | A(8) | «А»- «Я»; «а»- «я»; |
| Год | Year | 9(4) | «0»–«9» |
| Тип | TypeofDay | A(15) | «А»- «Я»; «а»- «я»; |

* + 1. Описание структурных единиц видеокадра V009 содержится в таблице 12

Таблица 12

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| Данные о мероприятиях на конкретный день | | V009 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Дата | vCalendar | X(16) | «А»- «Я»; «а»- «я»; «0»–«9» |
| Мероприятие | vListofCases | X(80) | «А»- «Я»; «а»- «я»,  «0»–«9»; «.» |
| Время начала | StartTime | X(5) | «0»–«9», «:» |
| Время окончания | EndTime | X(5) | «0»–«9»; «:» |
| Метка о провидении | MarkEvent | 9(1) | «0 – 1» |

* + 1. Описание структурных единиц видеокадра V010 содержится в таблице 13

Таблица 13

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование выходного сообщения | | Идентификатор | |
| Архивные данные | | V010 | |
| Наименование | Обозначение | Длина в знаках | Диапазон |
| Дата | vCalendar | X(16) | «А»- «Я»; «а»- «я»; «0»–«9» |
| Мероприятие | vListofCases | X(80) | «А»- «Я»; «а»- «я»,  «0»–«9»; «.» |
| Время начала | StartTime | X(5) | «0»–«9», «:» |
| Время окончания | EndTime | X(5) | «0»–«9»; «:» |

1. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ
   1. Назначение и характеристика

Алгоритм решения задачи предназначен для хранения, обработки данных и позволяет: вводить данные в таблицы, выводить данные на экран, сортировать по нескольким критериям, совершать поиск по нескольким критериям, выводить справку о программе.

* 1. Исходные данные
* DBProductionCalendar.pdb
* Данные о праздничных днях
* Данные о списке дел
* Данные о календаре
* Данные о мероприятиях на конкретный день
* Архивные данные
  1. Результаты решения

В результате работы формируется:

Видеограммы:

DBProductionCalendar.pdb – файл БД;

V001 – Список праздничных дней;

V002 – Список дел;

V003 – Календарь;

V004 – Список мероприятий;

V005 – Архив.

* 1. Диаграмма деятельности

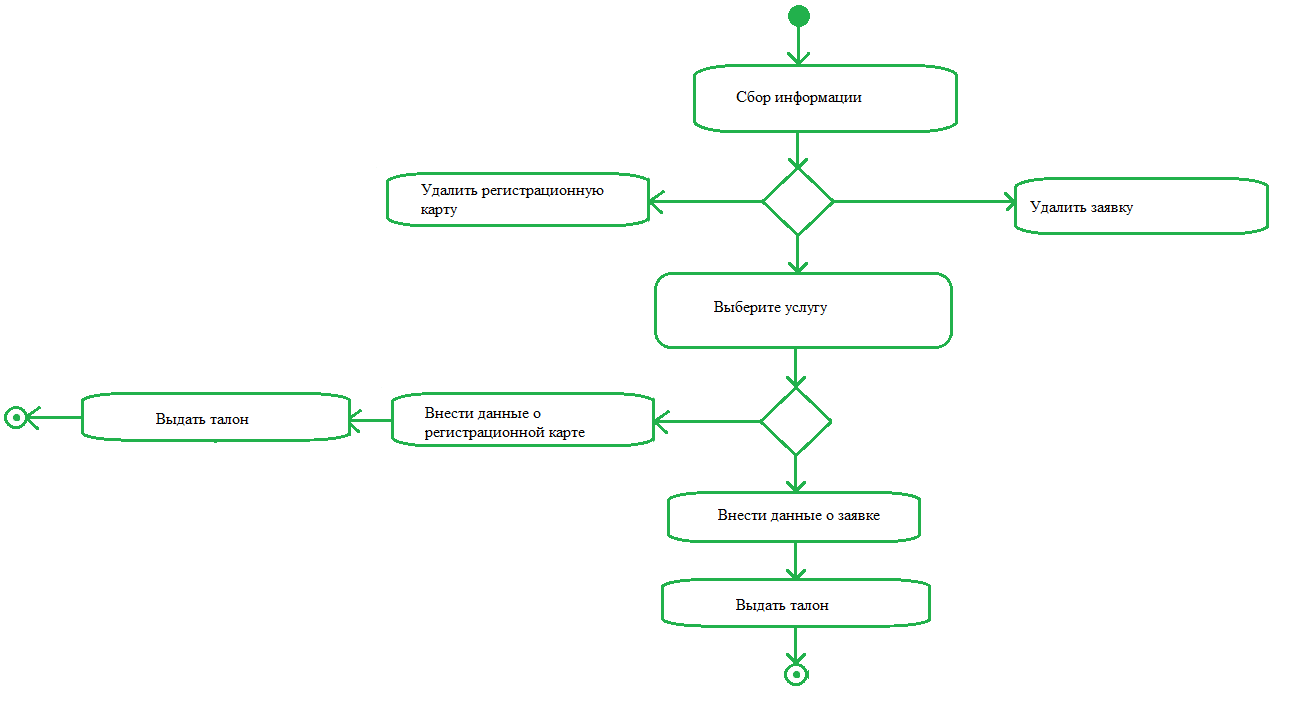


Рисунок 3 - Диаграмма деятельности

* 1. Алгоритм решения задачи

1. Создать или подключить БД на основе разработанной структуры классов.
2. Вывести главную форму, содержащую пункты меню.
3. Выбрать пункт меню.
4. Анализировать, выбранный пункт меню.
5. Если выбран пункт меню «Календарь» - «Праздничные дни»
   1. Вывести видеограмму V001.
   2. Если выбран пункт «Добавить».
      1. Вывести видеокадр V006.
      2. Ввести даты.
      3. Если выбран пункт «ОК».
         1. Сохранить введённые данные.
         2. Закрыть видеокадр V006.
      4. Если выбран пункт «Отмена».
         1. Закрыть видеокадр V006.
   3. Если выбран пункт «Удалить».
      1. Удалить выбранную дату.
   4. Если выбран пункт меню «Найти».
      1. Вывести список по введенной дате.
6. Если выбран пункт меню «Календарь» - «Календарь»
   1. Вывести на экран видеограмму V003.
   2. Если выбран пункт «Создать».
      1. Вывести видеокадр V008.
      2. Ввести дату начала, выбрать выходные дни в неделе.
      3. Если выбран пункт «ОК».
         1. Сохранить введённые данные.
         2. Закрыть видеокадр V008.
      4. Если выбран пункт «Отмена».
         1. Закрыть видеокадр V008.
   3. Если выбран пункт «Изменить тип».
      1. Изменить тип выбранного дня.
   4. Если выбран пункт «Найти».
      1. Вывести список по введенной дате.
7. Если выбран пункт меню «Мероприятия» - «Список дел»
   1. Вывести на экран видеограмму V002.
   2. Если выбран пункт «Добавить».
      1. Вывести видеокадр V007.
      2. Ввести мероприятие и место проведения.
      3. Если выбран пункт «ОК».
         1. Сохранить введённые данные.
         2. Закрыть видеокадр V007.
      4. Если выбран пункт «Отмена».
         1. Закрыть видеокадр V007.
   3. Если выбран пункт «Изменить».
      1. Вывести видеокадр V007.
      2. Изменить данные.
      3. Если выбран пункт «ОК».
         1. Сохранить измененные данные.
         2. Закрыть видеокадр V007.
      4. Если выбран пункт «Отмена».
         1. Закрыть видеокадр V007.
   4. Если выбран пункт «Удалить».
      1. Удалить выбранное дело.
   5. Если выбран пункт «Выгрузить данные».
      1. Создать файл F002.
      2. Загрузить в него данные списка дел.
      3. Сохранить и закрыть файл F002.
   6. Если выбран пункт «Загрузить данные»
      1. Открыть, если существует, файл F002.
      2. Считать данные.
      3. Добавить данные в таблицу.
      4. Сохранить добавленные данные.
      5. Закрыть файл F002.
8. Если выбран пункт меню «Мероприятия» - «Список мероприятий».
   1. Вывести на экран видеограмму V004.
   2. Если выбран пункт «Добавить».
      1. Вывести видеокадр V009.
      2. Ввести Дату, Мероприятие, Время начала и Время окончания.
      3. Если выбран пункт «ОК».
         1. Сохранить введённые данные.
         2. Закрыть видеокадр V009.
      4. Если выбран пункт «Отмена».
         1. Закрыть видеокадр V009.
   3. Если выбран пункт «Изменить».
      1. Вывести видеокадр V009.
      2. Изменить данные.
      3. Если выбран пункт «ОК».
         1. Сохранить введённые данные.
         2. Закрыть видеокадр V009.
      4. Если выбран пункт «Отмена».
         1. Закрыть видеокадр V009.
   4. Если выбран пункт «Отметить».
      1. Изменить значение Метки о проведении на true.
   5. Если выбран пункт «Найти»
      1. Если выбран критерий «По мероприятию».
         1. Вывести список по введенному мероприятию.
      2. Если выбран критерий «По дате».
         1. Вывести список по введенной дате.
   6. Если выбран пункт «Архивировать».
      1. Переместить все запланированные мероприятия в Архив.
      2. Удалить мероприятия из списка мероприятий.
      3. Сохранить изменения.
9. Если выбран пункт меню «Архив».
   1. Вывести на экран видеограмму V005.
   2. Если выбран пункт «Вернуть в список».
      1. Вывести видеокадр V009.
      2. Изменить имеющиеся данные.
      3. Если выбран пункт «ОК».
         1. Сохранить введённые данные.
         2. Закрыть видеокадр V009.
         3. Передать данные в таблицу «Список мероприятий».
         4. Удалить данные из архива.
         5. Сохранить изменения.
      4. Если выбран пункт «Отмена».
         1. Закрыть видеокадр V009.
   3. Если выбран пункт «Удалить».
      1. Удалить выбранный объект.
   4. Если выбран пункт «Найти».
      1. Если выбран критерий «По мероприятию».
         1. Вывести список по введенному мероприятию
      2. Если выбран критерий «По дате»
         1. Вывести список по введенной дате.
10. Если выбран пункт меню «Параметры».
    1. Вывести видеограмму V010.
    2. Если выбран пункт «Справки».
       1. Если выбран пункт «СПРАВКА: о запланированных мероприятиях на прошедший день».
          1. Вывести видеокадр V011.
          2. Если выбран пункт «Печать».
             1. Сделать печать видеокадра V011.
             2. Закрыть видеокадр V011.
       2. Если выбран пункт «СПРАВКА: о запланированных мероприятиях на текущий год».
          1. Вывести видеокадр V011.
          2. Если выбран пункт «Печать».
             1. Сделать печать видеокадра V011.
             2. Закрыть видеокадр.
    3. Если выбран пункт «Отчеты».
       1. Если выбран пункт «ОТЧЕТ: о запланированных мероприятиях на будущий день».
          1. Вывести видеокадр V011.
          2. Если выбран пункт «Печать».
             1. Сделать печать видеокадра V011.
             2. Закрыть видеокадр.
       2. Если выбран пункт «ОТЧЕТ: о мероприятиях, проведенных за интересующий период».
          1. Указать даты.
          2. Если выбран пункт «Продолжить»
             1. Вывести видеокадр V011.
             2. Если выбран пункт «Печать».

Сделать печать видеокадра V011.

Закрыть видеокадр.

* 1. Если выбран пункт меню «Выход».
  2. Закрыть программу.

1. Если выбран пункт «Справка»
   1. Вывести текст справки.
2. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ
   1. Характеристика задачи

В программе предусмотрены следующие возможности:

* Свободный пользовательский выбор пункта меню;
* Ввод данных в таблицы БД;
* Вывод данных на экран;
* Назначение мероприятий на конкретную дату;
* Формирование отчета о запланированных мероприятиях на будущий день;
* Формирование отчета о мероприятиях, проведенных за интересующий период;
* Вывод справки о мероприятиях за прошедший день;
* Перенос списка дел на следующий год.
* Перенос данных в архивную таблицу.
* Вывод справки о программе.
  1. Назначение и условия применения

Программа создана для хранения и обработки данных, изменения данных, ввода данных в таблицы, вывода данных на экран.

Программа функционирует в ОС Windows 10.

* 1. Подготовка к работе

Для работы Базы данных необходимы файлы:

eBvel.Praktic.OtchetVersion.exe;

DBProdactionCalendar.mdf;

DBProdactionCalendar\_log.idf;

EntityFramework.dll;

EntityFramework.SqlServer.dll;

* 1. Описание операций
     1. Функция создания списка праздничных дней

Чтобы создать список, выберите в командном меню пункт «Календарь» и подпункт «Праздничные дни». Затем, нажмите кнопку «Добавить» и выберите даты, которые хотите пометить как выходной.

* + 1. Функция добавления дела в список.

Чтобы добавить новое дело, выберите в командном меню пункт «Мероприятия» и подпункт «Список дел». Затем, нажмите кнопку «Добавить» и введите нужные данные, после нажмите «Ок».

* + 1. Функция создания календаря.

Выберите пункт командного меню «Календарь» и подпункт «Календарь». Затем, нажмите кнопку «Создать» - в сплывшем окне выберите дату отсчета и настройте рабочую неделю(Серый – рабочий день, Белый - выходной). И нажмите «Ок».

* + 1. Функция назначения мероприятия на конкретную дату.

Чтобы назначить мероприятие, выберите в командном меню пункт «Мероприятия» и подпункт «Список мероприятий». Затем, нажмите кнопку «Добавить» - в сплывшем окне выберите дело из списка, дату из списка, установите время начала и окончания мероприятия и нажмите «Ок».

* + 1. Функция переноса данных в архив.

Чтобы перенести данные, выберите в командном меню пункт «Мероприятия» и подпункт «Список мероприятий». Затем, нажмите на кнопку «Архивировать» - в сплывшем уведомлении нажмите «Да», если хотите перенести данные в архив или «Нет», если – нет.

* + 1. Функция переноса списка данных на следующий год.

Чтобы перенести список, выберите в командном меню пункт «Мероприятия» и подпункт «Список дел». Затем, нажмите кнопку «Выгрузить данные», после чего данные из списка буду сохранены в файл. Когда спланируете календарь на новый год, в том же окне, нажмите кнопку «Загрузить данные», после чего, данные из файла считаются и добавятся в список.

* + 1. Функция для печати справок

Чтобы распечатать справку, выберите в командном меню пункт «Параметры». Затем, нажмите «Справки», из отобразившегося списка выберите справку, которая вас интересует – в сплывшем окне нажмите «Печать».

* + 1. Функция для печати отчетов

Чтобы распечатать отчет, выберите в командном меню пункт «Параметры». Затем, нажмите «Отчеты», их отобразившегося списка выберите отчет, который вас интересует – в сплывшем окне нажмите «Печать».

* + 1. Функция вывода справки о программе

Чтобы вызвать справку, выберите в командном меню пункт «Справка». После чего выведется текст о программе.

* + 1. Функция выход

Чтобы выйти из программы, выберите в командном меню пункт «Параметры», а затем нажмите «Выход». Программа закроется.

* 1. Аварийные ситуации

Программа выдаёт ошибку, если ввели не все необходимые данные. Файлы могут быть повреждены, рекомендуется сделать резервную копию на другие носители.

1. РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА
   1. Общие сведения

Программа создана для хранения и обработки данных, ввода новых данных, вывода данных на экран, формирования отчётов и их печати, формирования талонов и их печати.

* 1. Технические условия программы
* Компьютер семейство IBM PC или совместимый;
* 100 Мбайт свободного пространства на жёстком диске или другом носители.
* Для запуска программы необходимы следующие характеристики:
* Операционная система Windows 10;
* Среда программирования Visual Studio 2017 и выше;
* Программа написана на языке программирования C#.
  1. Описание переменных и свойств

Диаграмма классов показана на рисунке 4.

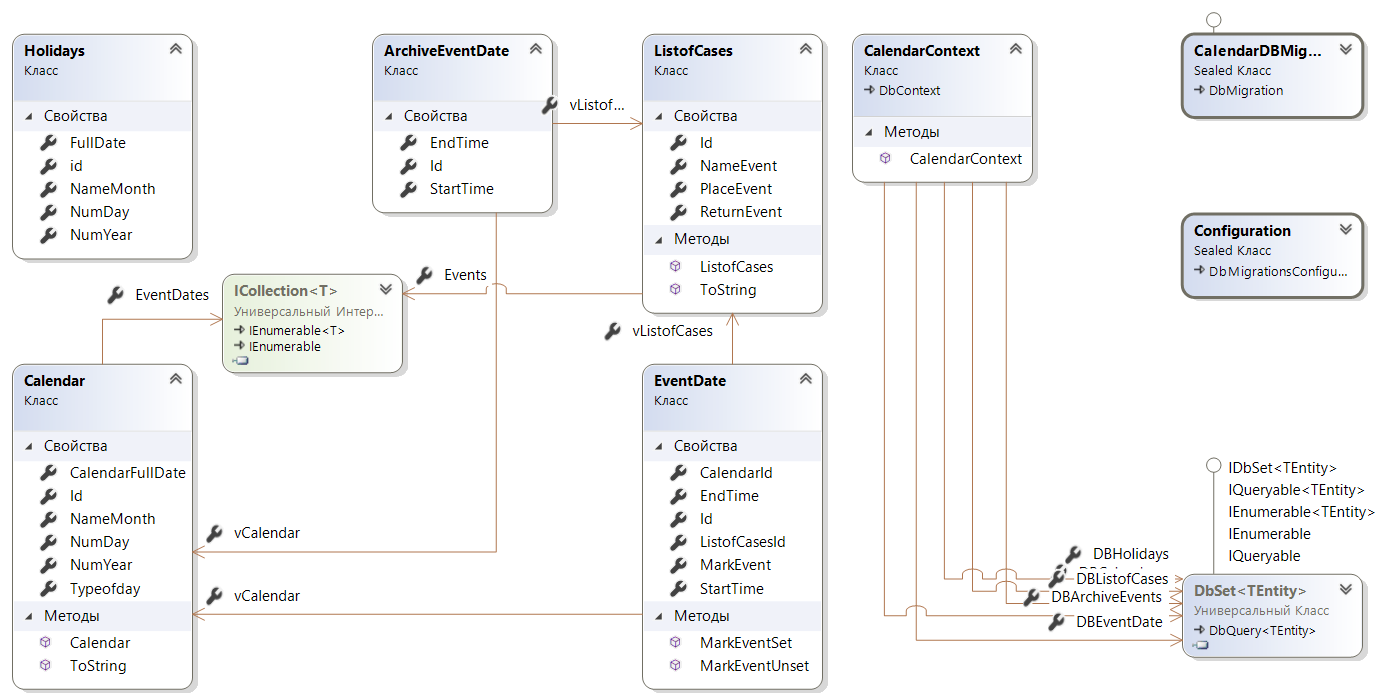


Рисунок 4 – Диаграмма классов

**Holiday** – Класс, содержит свойства для ввода данных о праздничном дне.

Id – Свойство, целое число, идентификатор класса

NumDay – Свойство, целое число, содержит число праздничного дня.

NameMonth – Свойство, строка, содержит наименование месяца праздничного дня.

Year – Свойство, целое число, содержит год праздничного дня.

FullDate – Свойство, строка, полная дата праздничного дня.

**ListofCases** – Класс, содержит свойства для ввода данных о деле.

Id – Свойство, целое число, идентификатор класса.

NameEvent – Свойство, строка, содержит данные о мероприятие.

PlaceEvent – Свойство, строка, содержит данные о месте проведения.

**Calendar** – Класс, содержит свойства для ввода данных о дне.

Id – Свойство, целое число, идентификатор класса.

NumDay – Свойство, целое число, содержит число даты.

NameMonth – Свойство, строка, содержит наименование месяца даты.

Year – Свойство, целое число, содержит год даты.

CalendarFullDate – Свойство, строка, содержит полную дату.

TypeofDay – Свойство, строка, содержит тип дня.

**EventDate** – Класс, содержит свойства для ввода мероприятия.

Id – Свойство, целое число, идентификатор класса.

CalendarId – Свойство, целое число, идентификатор класса Calendar.

ListofCasesId – Свойство, целое число, идентификатор класса ListofCases.

vCalendar – Виртуальное свойство, классовый тип, содержит данные о дне.

cListofCases – Виртуальное свойство, классовый тип, содержит данные о деле.

StartTime – Свойство, строка, содержит данные о начале проведения мероприятия.

EndTime – Свойство, строка, содержит данные об окончание мероприятия.

MarkEvent – Свойство, логического типа, содержит данные было ли проведено мероприятие.

**ArchiveEventDate** – Класс, содержит свойства для хранения данных.

Id – Свойство, целое число, идентификатор класса.

vCalendar – Виртуальное свойство, классового типа, содержит дату.

vListofCases – Виртуальное свойство, классового типа, содержит дело.

StartTime – Свойство, строка, содержит данные о начале проведения мероприятия.

EndTime – Свойство, строка, содержит данные об окончание проведения мероприятия.

**CalendarContext** – модель ADO.NET, служит для создания базы данных

DBHolidays – База данных, содержит список праздничных дней.

DBListofCases – База данных, содержит список дел.

DBCalendars – База данных, содержит календарь на год.

DBEventDate – База данных, содержит список назначенных мероприятий на конкретный день.

DBArchiveEvents – База данных, содержит список архивных мероприятий.

SearchingObject() – Метод, для поиска объекта в базе данных.

**AddHolidaysControl** – Элемент управления, выводит данные о списке праздничных дней.

Add\_Button\_Click() – Кнопка, добавляет данные в список праздничных дней.

Delete\_Button\_Click() – Кнопка, удаляет праздничный день.

Search\_Button\_Click() – Кнопка, совершает поиск элемента в списке.

SearchTextBox\_KeyPress() – Событие, переводит фокус на кнопку «Найти».

**AddCasesControl** – Элемент управления, выводит данные о списке дел.

Add\_Button\_Click() – Кнопка, добавляет данные в список дел.

Edit\_Button\_Click() – Кнопка, позволяет изменять данные в списке.

Delete\_Button\_Click() – Кнопка, удаляет выбранное дело.

Unload\_Button\_Click() – Кнопка, выгружает список в файл.

Load\_Button\_Click() – Кнопка, загружает список из файла.

**CreateCalendarControl** – Элемент управления, выводит календарь.

Create\_Button\_Click() – Кнопка, создает календарь.

Holiday\_Button\_Click() – Кнопка, изменяет тип выбранного дня.

Search\_Button\_Click() – Кнопка, совершает поиск по дню.

SearchTextBox\_KeyPress() – Событие, переводит фокус на кнопку «Найти».

HighlightHolidays() – Метод, выделяет выходные дни.

CheckHolidays() – Метод, проверяет на ввод новых праздничных дней.

**CreateEventControl** – Элемент управления, выводит данные о запланированных мероприятиях на конкретный день.

LoadDate\_ComboBox() – Метод, загружает данные в combo box’ы.

SaveDate() – Метод, сохраняет введенные данные.

Add\_Button\_Click() – Кнопка, добавляет новое мероприятие.

Edit\_Button\_Click() – Кнопка, позволяет изменить данные.

Mark\_Button\_Click() – Кнопка, отмечает мероприятие, как проведенное.

Archive\_Button\_Click() – Кнопка, переносит список мероприятий в архив.

**ArchiveDBControl** – Элемент управления, выводит архив мероприятий.

LoadDate\_ComboBox() – Метод, загружает данные в combo box’ы.

Transfer\_Button\_Click() – Кнопка, переносит мероприятие из архива в основную таблицу.

Delete\_Button\_Click() – Кнопка, удаляет мероприятие из архива.

**StartPageControl** – Элемент управления, выводит данные для печати.

References\_Label\_Click() – Кнопка, выводит список справок на печать.

Reports\_Label\_Click() – Кнопка, выводит список отчетов на печать.

Exit\_Label\_Click() – Кнопка, выхода из программы.

Reference\_Label1\_Click() – Кнопка, выводит данные на печать.

Reference\_Label2\_Click() – Кнопка, выводит данные на печать.

Report\_Label1\_Click() – Кнопка, выводит данные на печать.

Report\_Label2\_Click() – Кнопка, отображает элементы для ввода дат.

Continue\_Button\_Click() - Кнопка, выводит данные на печать.

HelpControl – Элемент управления, выводит данные о программе.

**AddHolidaysForm** – Форма, для ввода данных о праздничных днях.

Add\_Button\_Click() – Кнопка, для сохранения списка праздничных дней.

Period\_CheckBox\_CheckedChanged() – Кнопка, позволяет вводить последовательность дат.

Delete\_Button\_Click() – кнопка, удаляет дату из списка.

**AddCasesForm** – Форма, для ввода данных о деле.

**AddCalendarForm** – Форма, для ввода данных о календаре.

Pane1\_Down() – Кнопка, отмечает день, как выходной – рабочий.

Pane2\_Down() - Кнопка, отмечает день, как выходной – рабочий.

Pane3\_Down() - Кнопка, отмечает день, как выходной – рабочий.

Pane4\_Down() - Кнопка, отмечает день, как выходной – рабочий.

Pane5\_Down() – Кнопка, отмечает день, как выходной – рабочий.

Pane6\_Down() - Кнопка, отмечает день, как выходной – рабочий.

Pane7\_Down() - Кнопка, отмечает день, как выходной – рабочий.

**AddEventForm** – Форма, для ввода данных о мероприятие на конкретный день.

**PrintForm** – Форма, выводит данные для печати.

CaptureScreen() – Метод, делает скриншот формы.

printFormDocument\_PrintPage() – Событие, делает печать при вызове.

Print\_Button\_Click() – Кнопка, вызывает событие для печати.

* 1. Сообщение системному администратору

После внесения изменений в код программы, необходимо обновить миграцию и сохранить её резервную копию. Если возникла ошибка, то программу нужно восстановить.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ Р 1.5–2012 Стандарты национальные Российской федерации. Правила построения, изложения, оформления и обозначения
2. ГОСТ 2.102–2013 ЕСКД Виды и комплектность конструкторских документов
3. ГОСТ 3.1105–2011 ЕСТД Формы и правила оформления документов общего назначения
4. ГОСТ 19.401–78 ЕСПД Текст программы. Требования к содержанию и оформлению
5. ГОСТ 19.402–78 ЕСПД Описание программы
6. ГОСТ 19.502–78 ЕСПД Описание применения. Требования к содержанию и оформлению
7. ГОСТ 19.105–78 ЕСПД Общие требования к программным документам
8. ГОСТ 19.404–79 ЕСПД Пояснительная записка. Требования к содержанию и оформлению
9. СТО АлтГТУ 12 570–2013. Система качества. Образовательный стандарт высшего профессионального образования АлтГТУ. Общие требования к текстовым, графическим и программным документам
10. Абрамян, Михаил Visual C# на примерах (+ CD-ROM) / Михаил Абрамян. - М.: БХВ-Петербург, 2012. - 496 c.
11. Секунов, Н. Самоучитель C# / Н. Секунов. - М.: БХВ-Петербург, 2017. - 576 c.
12. Емельянов, Н.Е. Введение в СУБД ИНЕС / Н.Е. Емельянов. - М.: Наука, 2011. - 256 c.

Приложение А

Листинг программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.ComponentModel.DataAnnotations;

using System.Threading.Tasks;

namespace AutoHousSelectionModel

{

public class Type\_House

{

public int Id { get; set; }

[Required]

public string Title { get; set; }

public virtual ICollection<Registr\_Card> Registr\_Cards { get; set; }

public virtual ICollection<Statement> Statements { get; set; }

public Type\_House() { Registr\_Cards = new List<Registr\_Card>(); Statements = new List<Statement>(); }

public override string ToString() { return Title; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.ComponentModel.DataAnnotations;

using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

namespace AutoHousSelectionModel

{

public class Statement

{

public int Id { get; set; }

public int Number\_rooms { get; set; }

public double Price { get; set; }

public int Floor { get; set; }

public DateTime Date\_registration { get; private set; }

public string FIO { get; set; }

public int Type\_HouseID { get; set; }

public virtual Type\_House Type\_Houses { get; set; }

public void AddStatement()

{

Date\_registration = DateTime.Today;

}

public bool TodayDate()

{

var a = DateTime.Today;

var b = Date\_registration;

bool c=true;

if (a > b)

{

return c=false;

}

else

{

return c;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.ComponentModel.DataAnnotations;

namespace AutoHousSelectionModel

{

public class Registr\_Card

{

public int Id { get; set; }

public DateTime Date\_registration { get; private set; }

public int Number\_rooms { get; set; }

public int Floor { get; set; }

public double Total\_area{ get; set; }

public double Living\_area { get; set; }

public int Number\_floors { get; set; }

public int Balcony { get; set; }

public int Glazed\_balcony { get; set; }

public int Iron\_door { get; set; }

public string Repair\_information { get; set; }

public string Information\_registration { get; set; }

public int Information\_privatization { get; set; }

public int Telephone\_coverage { get; set; }

public int Gasification { get; set; }

public double Price { get; set; }

public string Telephone { get; set; }

public string Address { get; set; }

public string FIO { get; set; }

public int Reason\_deregistration { get; set; }

public DateTime Date\_deregistration { get; private set; }

public void AddRegistrCard()

{

Date\_registration = DateTime.Today;

Date\_deregistration = Date\_registration.AddDays(21);

}

public bool Deregister()

{

if (Reason\_deregistration == 0)

{

Date\_deregistration = DateTime.Today;

Reason\_deregistration = 1;

return true;

}

return false;

}

public void Removed\_register()

{

Date\_deregistration = DateTime.Today.AddDays(21);

if (Reason\_deregistration == 1)

{

Reason\_deregistration = 0;

}

}

public string Note { get; set; }

public int Part\_CityId { get; set; }

public int Type\_HouseID { get; set; }

public int House\_MaterialId { get; set; }

public virtual House\_Material House\_Material { get; set; }

public virtual Part\_City Part\_Cities { get; set; }

public virtual Type\_House Type\_Houses { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AutoHousSelectionModel

{

public class Part\_City

{

public int Id { get; set; }

[Required]

public string Title { get; set; }

public virtual ICollection<Registr\_Card> Registr\_Cards { get; set; }

public Part\_City() { Registr\_Cards = new List<Registr\_Card>(); }

public override string ToString() { return Title; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.ComponentModel.DataAnnotations;

using System.Threading.Tasks;

namespace AutoHousSelectionModel

{

public class House\_Material

{

public int Id { get; set; }

[Required]

public string Title { get; set; }

public virtual ICollection<Registr\_Card> Registr\_Cards { get; set; }

public House\_Material() { Registr\_Cards = new List<Registr\_Card>(); }

public override string ToString() { return Title; }

}

}

namespace AutoHousSelectionModel

{

using System;

using System.Data.Entity;

using System.Linq;

public class AutoHousContext : DbContext

{

public AutoHousContext()

: base("name=AutoHous\_DB\_Connection")

{

}

public DbSet<House\_Material> House\_Materials { get; set; }

public DbSet<Part\_City> Part\_Cities { get; set; }

public DbSet<Registr\_Card> Registr\_Cards { get; set; }

public DbSet<Statement> Statements { get; set; }

public DbSet<Type\_House> Type\_Houses { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.ComponentModel.DataAnnotations;

namespace AutoHousSelection

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection.Forms

{

public partial class Type\_House\_Edit : Form

{

public Type\_House\_Edit()

{

InitializeComponent();

}

private void Title(object sender, KeyPressEventArgs e)

{

char charCode = e.KeyChar;

if (!char.IsLetter(charCode) && charCode != 8)// Разрешить вводить только буквы и символ backspace (код равен 8)

e.Handled = true;

textBox1.MaxLength = 15;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection.Forms

{

public partial class Statement\_Edit : Form

{

public Statement\_Edit()

{

InitializeComponent();

textBox1.Text = "0";

textBox3.Text = "0";

textBox5.Text = "0";

}

private void Number(object sender, KeyPressEventArgs e)

{

char charCode = e.KeyChar;

if (!char.IsDigit(charCode) && charCode != 8) // Разрешить вводить только цифры и символ backspace

e.Handled = true;

textBox1.MaxLength = 1;

textBox5.MaxLength = 2;

}

private void Price(object sender, KeyPressEventArgs e)

{

char charCode = e.KeyChar;

if (!char.IsDigit(charCode) && charCode != 8 && charCode != 44) // Разрешить вводить только цифры и символ backspace и запятую

e.Handled = true;

textBox3.MaxLength = 9;

}

private void FIO(object sender, KeyPressEventArgs e)// Разрешить вводить буквы и символ backspace (код равен 8)

{

char charCode = e.KeyChar;

if (!char.IsLetter(charCode) && charCode != 8 && charCode != 46 && charCode != 32) // Space и точку

e.Handled = true;

textBox7.MaxLength = 20;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection.Forms

{

public partial class Report\_hous\_sale : Form

{

public Report\_hous\_sale()

{

InitializeComponent();

string s = DateTime.Today.Month.ToString();

if (s == "1") s = "январь";

if (s == "2") s = "февраль";

if (s == "3") s = "март";

if (s == "4") s = "апрель";

if (s == "5") s = "май";

if (s == "6") s = "июнь";

if (s == "7") s = "июль";

if (s == "8") s = "август";

if (s == "9") s = "сентябрь";

if (s == "10") s = "октябрь";

if (s == "11") s = "ноябрь";

if (s == "12") s = "декабрь";

label8.Text = DateTime.Today.Day.ToString();

label9.Text = s;

label10.Text = DateTime.Today.Year.ToString();

}

[System.Runtime.InteropServices.DllImport("gdi32.dll")]

public static extern long BitBlt(IntPtr hdcDest, int nXDest, int nYDest,

int nWidth, int nHeight, IntPtr hdcSrc, int nXSrc, int nYSrc, int dwRop);

private Bitmap memoryImage;

//

//Метод делает Скрин формы.

//

private void CaptureScreen()

{

Graphics mygraphics = this.CreateGraphics();

Size s = this.Size;

memoryImage = new Bitmap(s.Width, s.Height, mygraphics);

Graphics memoryGraphics = Graphics.FromImage(memoryImage);

IntPtr dc1 = mygraphics.GetHdc();

IntPtr dc2 = memoryGraphics.GetHdc();

BitBlt(dc2, 0, 0, this.ClientRectangle.Width,

this.ClientRectangle.Height, dc1, 0, 0, 13369376);

mygraphics.ReleaseHdc(dc1);

memoryGraphics.ReleaseHdc(dc2);

}

//

//Метод кнопки, печать формы.

//

private void PrintDocButton\_Click(object sender, System.EventArgs e)

{

button1.Visible = false;

CaptureScreen();

printDocument1.Print();

button1.Visible = true;

}

//

//Обработчик события PrintDocument.

//

private void printDocument1\_PrintPage\_1(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

e.Graphics.DrawImage(memoryImage, 0, 0);

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection.Forms

{

public partial class Registr\_Card\_Edit : Form

{

public Registr\_Card\_Edit()

{

InitializeComponent();

textBox10.MaxLength = 10;

textBox15.MaxLength = 10;

}

private void Number(object sender, KeyPressEventArgs e)// Разрешить вводить только цифры и символ backspace

{

char charCode = e.KeyChar;

if (!char.IsDigit(charCode) && charCode != 8)

e.Handled = true;

textBox1.MaxLength = 1;

textBox5.MaxLength = 2;

textBox4.MaxLength = 2;

}

private void Double(object sender, KeyPressEventArgs e)// Разрешить вводить только цифры и символ backspace и запятую

{

char charCode = e.KeyChar;

if (!char.IsDigit(charCode) && charCode != 8 && charCode != 44)

e.Handled = true;

textBox2.MaxLength = 7;

textBox3.MaxLength = 7;

textBox14.MaxLength = 9;

}

private void Input\_1\_or\_2(object sender, KeyPressEventArgs e) // ввод только 1 или 0 и backspace

{

char charCode = e.KeyChar;

if (charCode != 48 && charCode != 49 && charCode != 8)

e.Handled = true;

textBox6.MaxLength = 1;

textBox7.MaxLength = 1;

textBox8.MaxLength = 1;

textBox11.MaxLength = 1;

textBox12.MaxLength = 1;

textBox13.MaxLength = 1;

}

private void String(object sender, KeyPressEventArgs e) // Разрешить вводить только буквы и символ backspace (код равен 8)

{

char charCode = e.KeyChar;

if (!char.IsLetter(charCode) && charCode != 8 && charCode != 46 && charCode != 32) // и точка и space

e.Handled = true;

textBox9.MaxLength = 10;

textBox17.MaxLength = 20;

textBox18.MaxLength = 20;

}

private void Adress(object sender, KeyPressEventArgs e) // Разрешить вводить буквы, цифры и символ backspace (код равен 8)

{

char charCode = e.KeyChar;

if (!char.IsLetter(charCode) && charCode != 8 && charCode != 46 && charCode != 32 && !char.IsDigit(charCode)) // и точка и space

e.Handled = true;

textBox16.MaxLength = 20;

}

}

}

using AutoHousSelectionModel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.Entity;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection.Forms

{

public partial class Reference : Form

{

public Reference()

{

InitializeComponent();

string s = DateTime.Today.Month.ToString();

if (s == "1") s = "января";

if (s == "2") s = "февраля";

if (s == "3") s = "марта";

if (s == "4") s = "апреля";

if (s == "5") s = "мая";

if (s == "6") s = "июня";

if (s == "7") s = "июля";

if (s == "8") s = "августа";

if (s == "9") s = "сентября";

if (s == "10") s = "октября";

if (s == "11") s = "ноября";

if (s == "12") s = "декабря";

label8.Text = DateTime.Today.Day.ToString();

label9.Text = s;

label10.Text = DateTime.Today.Year.ToString();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection.Forms

{

public partial class Part\_City\_Edit : Form

{

public Part\_City\_Edit()

{

InitializeComponent();

}

private void Title(object sender, KeyPressEventArgs e)

{

char charCode = e.KeyChar;

if (!char.IsLetter(charCode) && charCode != 8)// Разрешить вводить только буквы и символ backspace (код равен 8)

e.Handled = true;

textBox1.MaxLength = 20;

}

}

}

using AutoHousSelection.Controls;

using AutoHousSelection.Forms;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

private void House\_Material\_Click(object sender, EventArgs e)

{

var uc = new House\_Material\_Control();

if (panel1.Controls.Count > 0)

panel1.Controls.RemoveAt(0);

panel1.Controls.Add(uc);

}

private void Part\_City\_Click(object sender, EventArgs e)

{

var uc = new Part\_City\_Control();

if (panel1.Controls.Count > 0)

panel1.Controls.RemoveAt(0);

panel1.Controls.Add(uc);

}

private void Type\_House\_Click(object sender, EventArgs e)

{

var uc = new Type\_House\_Control();

if (panel1.Controls.Count > 0)

panel1.Controls.RemoveAt(0);

panel1.Controls.Add(uc);

}

private void Registr\_Card\_Click(object sender, EventArgs e)

{

var uc = new Registr\_Card\_Control();

if (panel1.Controls.Count > 0)

panel1.Controls.RemoveAt(0);

panel1.Controls.Add(uc);

}

private void Statement\_Click(object sender, EventArgs e)

{

var uc = new Statement\_Control();

if (panel1.Controls.Count > 0)

panel1.Controls.RemoveAt(0);

panel1.Controls.Add(uc);

}

private void HelpClick(object sender, EventArgs e)

{

var uc = new HelpControl();

if (panel1.Controls.Count > 0)

panel1.Controls.RemoveAt(0);

panel1.Controls.Add(uc);

}

private void Exit(object sender, EventArgs e)

{

this.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection.Forms

{

public partial class House\_Material\_Edit : Form

{

public House\_Material\_Edit()

{

InitializeComponent();

}

private void Title(object sender, KeyPressEventArgs e)

{

char charCode = e.KeyChar;

if (!char.IsLetter(charCode) && charCode != 8)// Разрешить вводить только буквы и символ backspace (код равен 8)

e.Handled = true;

textBox1.MaxLength = 10;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection.Forms

{

public partial class CouponSell : Form

{

public CouponSell()

{

InitializeComponent();

string s = DateTime.Today.Month.ToString();

if (s == "1") s = "января";

if (s == "2") s = "февраля";

if (s == "3") s = "марта";

if (s == "4") s = "апреля";

if (s == "5") s = "мая";

if (s == "6") s = "июня";

if (s == "7") s = "июля";

if (s == "8") s = "августа";

if (s == "9") s = "сентября";

if (s == "10") s = "октября";

if (s == "11") s = "ноября";

if (s == "12") s = "декабря";

label8.Text = " " + DateTime.Today.Day.ToString() + " " + s + " " + DateTime.Today.Year.ToString() + " ";

}

[System.Runtime.InteropServices.DllImport("gdi32.dll")]

public static extern long BitBlt(IntPtr hdcDest, int nXDest, int nYDest,

int nWidth, int nHeight, IntPtr hdcSrc, int nXSrc, int nYSrc, int dwRop);

private Bitmap memoryImage;

//

//Метод делает Скрин формы.

//

private void CaptureScreen()

{

Graphics mygraphics = this.CreateGraphics();

Size s = this.Size;

memoryImage = new Bitmap(s.Width, s.Height, mygraphics);

Graphics memoryGraphics = Graphics.FromImage(memoryImage);

IntPtr dc1 = mygraphics.GetHdc();

IntPtr dc2 = memoryGraphics.GetHdc();

BitBlt(dc2, 0, 0, this.ClientRectangle.Width,

this.ClientRectangle.Height, dc1, 0, 0, 13369376);

mygraphics.ReleaseHdc(dc1);

memoryGraphics.ReleaseHdc(dc2);

}

//

//Метод кнопки, печать формы.

//

private void PrintDocButton\_Click(object sender, System.EventArgs e)

{

button1.Visible = false;

CaptureScreen();

printDocument1.Print();

button1.Visible = true;

}

//

//Обработчик события PrintDocument.

//

private void printDocument1\_PrintPage\_1(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

e.Graphics.DrawImage(memoryImage, 0, 0);

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection.Forms

{

public partial class Coupon : Form

{

public Coupon()

{

InitializeComponent();

string s = DateTime.Today.Month.ToString();

if (s == "1") s = "января";

if (s == "2") s = "февраля";

if (s == "3") s = "марта";

if (s == "4") s = "апреля";

if (s == "5") s = "мая";

if (s == "6") s = "июня";

if (s == "7") s = "июля";

if (s == "8") s = "августа";

if (s == "9") s = "сентября";

if (s == "10") s = "октября";

if (s == "11") s = "ноября";

if (s == "12") s = "декабря";

label8.Text = " " + DateTime.Today.Day.ToString() + " " + s + " " + DateTime.Today.Year.ToString() + " ";

}

[System.Runtime.InteropServices.DllImport("gdi32.dll")]

public static extern long BitBlt(IntPtr hdcDest, int nXDest, int nYDest,

int nWidth, int nHeight, IntPtr hdcSrc, int nXSrc, int nYSrc, int dwRop);

private Bitmap memoryImage;

//

//Метод делает Скрин формы.

//

private void CaptureScreen()

{

Graphics mygraphics = this.CreateGraphics();

Size s = this.Size;

memoryImage = new Bitmap(s.Width, s.Height, mygraphics);

Graphics memoryGraphics = Graphics.FromImage(memoryImage);

IntPtr dc1 = mygraphics.GetHdc();

IntPtr dc2 = memoryGraphics.GetHdc();

BitBlt(dc2, 0, 0, this.ClientRectangle.Width,

this.ClientRectangle.Height, dc1, 0, 0, 13369376);

mygraphics.ReleaseHdc(dc1);

memoryGraphics.ReleaseHdc(dc2);

}

//

//Метод кнопки, печать формы.

//

private void PrintDocButton\_Click(object sender, System.EventArgs e)

{

button1.Visible = false;

CaptureScreen();

printDocument1.Print();

button1.Visible = true;

}

//

//Обработчик события PrintDocument.

//

private void printDocument1\_PrintPage\_1(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

e.Graphics.DrawImage(memoryImage, 0, 0);

}

}

}

using AutoHousSelectionModel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.Entity;

using AutoHousSelection.Forms;

namespace AutoHousSelection.Forms

{

public partial class Additional\_info : Form

{

public Additional\_info()

{

InitializeComponent();

}

[System.Runtime.InteropServices.DllImport("gdi32.dll")]

public static extern long BitBlt(IntPtr hdcDest, int nXDest, int nYDest,

int nWidth, int nHeight, IntPtr hdcSrc, int nXSrc, int nYSrc, int dwRop);

private Bitmap memoryImage;

//

//Метод делает Скрин формы.

//

private void CaptureScreen()

{

Graphics mygraphics = this.CreateGraphics();

Size s = this.Size;

memoryImage = new Bitmap(s.Width, s.Height, mygraphics);

Graphics memoryGraphics = Graphics.FromImage(memoryImage);

IntPtr dc1 = mygraphics.GetHdc();

IntPtr dc2 = memoryGraphics.GetHdc();

BitBlt(dc2, 0, 0, this.ClientRectangle.Width,

this.ClientRectangle.Height, dc1, 0, 0, 13369376);

mygraphics.ReleaseHdc(dc1);

memoryGraphics.ReleaseHdc(dc2);

}

//

//Метод кнопки, печать формы.

//

private void PrintDocButton\_Click(object sender, System.EventArgs e)

{

button1.Visible = false;

CaptureScreen();

printDocument1.Print();

button1.Visible = true;

}

//

//Обработчик события PrintDocument.

//

private void printDocument1\_PrintPage\_1(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

e.Graphics.DrawImage(memoryImage, 0, 0);

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using AutoHousSelectionModel;

using System.Data.Entity;

using AutoHousSelection.Forms;

namespace AutoHousSelection.Controls

{

public partial class Type\_House\_Control : UserControl

{

AutoHousContext db;

public Type\_House\_Control()

{

InitializeComponent();

db = new AutoHousContext();

db.Type\_Houses.Load();

dataGridView1.DataSource = db.Type\_Houses.Local.ToBindingList();

}

private void AddClick(object sender, EventArgs e)

{

Type\_House\_Edit type\_house\_Edit = new Type\_House\_Edit();

//type\_house\_Edit.button1.Text = "Добавить";

Type\_House type\_house = new Type\_House();

DialogResult result = type\_house\_Edit.ShowDialog(this);

SaveData(type\_house\_Edit, result, type\_house, "добавлены");

}

private void SaveData(Type\_House\_Edit type\_house\_Edit, DialogResult result, Type\_House type\_house, string str)

{

if (result == DialogResult.OK)

{

try

{

type\_house.Title = type\_house\_Edit.textBox1.Text;

if (str == "добавлены") db.Type\_Houses.Add(type\_house); db.SaveChanges();

MessageBox.Show("Данные о типе дома " + str);

}

catch (Exception exept) { MessageBox.Show("Данные не " + str + ".\n " + exept.Message); }

}

}

private void DeleteClick(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out int id);

if (converted == false) return;

Type\_House type\_house = db.Type\_Houses.Find(id);

db.Type\_Houses.Remove(type\_house);

db.SaveChanges();

MessageBox.Show("Объект удален");

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using AutoHousSelectionModel;

using System.Data.Entity;

using AutoHousSelection.Forms;

namespace AutoHousSelection.Controls

{

public partial class Statement\_Control : UserControl

{

AutoHousContext db;

public Statement\_Control()

{

InitializeComponent();

db = new AutoHousContext();

AutoDelete();

db.Statements.Load();

dataGridView1.DataSource = db.Statements.Local.ToBindingList();

}

private void AddClick(object sender, EventArgs e)

{

Statement\_Edit statement\_Edit = new Statement\_Edit();

List<Type\_House> type\_houses = db.Type\_Houses.ToList();

statement\_Edit.comboBox1.DataSource = type\_houses;

statement\_Edit.comboBox1.ValueMember = "Id";

statement\_Edit.comboBox1.DisplayMember = "Title";

Statement statement = new Statement();

DialogResult result = statement\_Edit.ShowDialog(this);

SaveData(statement\_Edit, result, statement, "добавлены");

}

private void SaveData(Statement\_Edit statement\_Edit, DialogResult result, Statement statement, string str)

{

if (result == DialogResult.OK)

{

try

{

statement.Type\_Houses = (Type\_House)statement\_Edit.comboBox1.SelectedItem;

statement.Number\_rooms = int.Parse(statement\_Edit.textBox1.Text);

statement.Price = int.Parse(statement\_Edit.textBox3.Text);

statement.Floor = int.Parse(statement\_Edit.textBox5.Text);

statement.FIO = statement\_Edit.textBox7.Text;

statement.AddStatement();

if (str == "добавлены") db.Statements.Add(statement); db.SaveChanges();

dataGridView1.Refresh();

MessageBox.Show("Данные о заявке " + str);

var coupon = new Coupon();

coupon.dataGridView1.DataSource = db.Statements.Local.Where(p => p.Id == statement.Id).ToList();

coupon.label6.Text = "Покупка";

DialogResult resulte = coupon.ShowDialog(this);

}

catch (Exception exept) { MessageBox.Show("Данные не " + str + ".\n " + exept.Message); }

}

}

private void EditClick(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out int id);

if (converted == false) return;

Statement statement = db.Statements.Find(id);

Statement\_Edit statement\_Edit = new Statement\_Edit();

List<Type\_House> type\_houses = db.Type\_Houses.ToList();

statement\_Edit.comboBox1.DataSource = type\_houses;

statement\_Edit.comboBox1.ValueMember = "Id";

statement\_Edit.comboBox1.DisplayMember = "Title";

statement\_Edit.comboBox1.SelectedItem = statement.Type\_Houses;

statement\_Edit.textBox1.Text = statement.Number\_rooms.ToString();

statement\_Edit.textBox3.Text = statement.Price.ToString();

statement\_Edit.textBox5.Text = statement.Floor.ToString();

statement\_Edit.textBox7.Text = statement.FIO;

DialogResult result = statement\_Edit.ShowDialog(this); SaveData(statement\_Edit, result, statement, "изменены");

}

}

private void DeleteClick(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out int id);

if (converted == false) return;

Statement statement = db.Statements.Find(id);

db.Statements.Remove(statement); db.SaveChanges();

MessageBox.Show("Заявка удалена");

}

}

public void AutoDelete()

{

foreach (var i in db.Statements)

{

if (i.Date\_registration <= DateTime.Today.AddDays(-14))

{

db.Statements.Remove(i);

db.SaveChanges();

}

}

}

private void Search(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out int id);

if (converted == false) return;

Statement statement = db.Statements.Find(id);

var report\_hous = new Report\_hous\_sale();

foreach (var i in db.Registr\_Cards)

{

if ((statement.Floor != 0) && (statement.Price == 0) && (statement.Number\_rooms == 0))

{

report\_hous.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Floor == statement.Floor && p.Type\_Houses == statement.Type\_Houses).ToList();

}

if ((statement.Floor == 0) && (statement.Price != 0) && (statement.Number\_rooms == 0))

{

report\_hous.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Price == statement.Price && p.Type\_Houses == statement.Type\_Houses).ToList();

}

if ((statement.Floor == 0) && (statement.Price == 0) && (statement.Number\_rooms != 0))

{

report\_hous.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Number\_rooms == statement.Number\_rooms && p.Type\_Houses == statement.Type\_Houses).ToList();

}

if ((statement.Floor != 0) && (statement.Price != 0) && (statement.Number\_rooms == 0))

{

report\_hous.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Price == statement.Price && p.Floor == statement.Floor &&

p.Type\_Houses == statement.Type\_Houses).ToList();

}

if ((statement.Floor != 0) && (statement.Price == 0) && (statement.Number\_rooms != 0))

{

report\_hous.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Floor == statement.Floor && p.Number\_rooms == statement.Number\_rooms &&

p.Type\_Houses == statement.Type\_Houses).ToList();

}

if ((statement.Floor == 0) && (statement.Price != 0) && (statement.Number\_rooms != 0))

{

report\_hous.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Number\_rooms == statement.Number\_rooms && p.Price == statement.Price &&

p.Type\_Houses == statement.Type\_Houses).ToList();

}

if ((statement.Floor != 0) && (statement.Price != 0) && (statement.Number\_rooms != 0))

{

report\_hous.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Number\_rooms == statement.Number\_rooms && p.Floor == statement.Floor &&

p.Price == statement.Price && p.Type\_Houses == statement.Type\_Houses).ToList();

}

}

DialogResult result = report\_hous.ShowDialog(this);

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using AutoHousSelectionModel;

using System.Data.Entity;

using AutoHousSelection.Forms;

namespace AutoHousSelection.Controls

{

public partial class Part\_City\_Control : UserControl

{

AutoHousContext db;

public Part\_City\_Control()

{

InitializeComponent();

db = new AutoHousContext();

db.Part\_Cities.Load();

dataGridView1.DataSource = db.Part\_Cities.Local.ToBindingList();

}

private void AddClick(object sender, EventArgs e)

{

Part\_City\_Edit part\_city\_Edit = new Part\_City\_Edit();

//part\_city\_Edit.button1.Text = "Добавить";

Part\_City part\_city = new Part\_City();

DialogResult result = part\_city\_Edit.ShowDialog(this);

SaveData(part\_city\_Edit, result, part\_city, "добавлены");

}

private void SaveData(Part\_City\_Edit part\_city\_Edit, DialogResult result, Part\_City part\_city, string str)

{

if (result == DialogResult.OK)

{

try

{

part\_city.Title = part\_city\_Edit.textBox1.Text;

if (str == "добавлены") db.Part\_Cities.Add(part\_city); db.SaveChanges();

MessageBox.Show("Данные о районе города " + str);

}

catch (Exception exept) { MessageBox.Show("Данные не " + str + ".\n " + exept.Message); }

}

}

private void DeleteClick(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out int id);

if (converted == false) return;

Part\_City part\_city = db.Part\_Cities.Find(id);

db.Part\_Cities.Remove(part\_city);

db.SaveChanges();

MessageBox.Show("Объект удален");

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using AutoHousSelectionModel;

using System.Data.Entity;

using AutoHousSelection.Forms;

using Microsoft.Reporting.WebForms;

namespace AutoHousSelection.Controls

{

public partial class Registr\_Card\_Control : UserControl

{

AutoHousContext db;

public Registr\_Card\_Control()

{

InitializeComponent();

db = new AutoHousContext();

db.Registr\_Cards.Load();

dataGridView1.DataSource = db.Registr\_Cards.Local.ToBindingList();

}

private void AddClick(object sender, EventArgs e)

{

Registr\_Card\_Edit registr\_card\_Edit = new Registr\_Card\_Edit();

List<Part\_City> part\_cities = db.Part\_Cities.ToList();

registr\_card\_Edit.comboBox1.DataSource = part\_cities;

registr\_card\_Edit.comboBox1.ValueMember = "Id";

registr\_card\_Edit.comboBox1.DisplayMember = "Title";

List<Type\_House> type\_houses = db.Type\_Houses.ToList();

registr\_card\_Edit.comboBox2.DataSource = type\_houses;

registr\_card\_Edit.comboBox2.ValueMember = "Id";

registr\_card\_Edit.comboBox2.DisplayMember = "Title";

List<House\_Material> house\_materials = db.House\_Materials.ToList();

registr\_card\_Edit.comboBox3.DataSource = house\_materials;

registr\_card\_Edit.comboBox3.ValueMember = "Id";

registr\_card\_Edit.comboBox3.DisplayMember = "Title";

Registr\_Card registr\_card = new Registr\_Card();

DialogResult result = registr\_card\_Edit.ShowDialog(this);

SaveData(registr\_card\_Edit, result, registr\_card, "добавлены");

}

private void SaveData(Registr\_Card\_Edit registr\_card\_Edit, DialogResult result, Registr\_Card registr\_card, string str)

{

if (result == DialogResult.OK)

{

try

{

registr\_card.Part\_Cities = (Part\_City)registr\_card\_Edit.comboBox1.SelectedItem;

registr\_card.Type\_Houses = (Type\_House)registr\_card\_Edit.comboBox2.SelectedItem;

registr\_card.House\_Material = (House\_Material)registr\_card\_Edit.comboBox3.SelectedItem;

registr\_card.Number\_rooms = int.Parse(registr\_card\_Edit.textBox1.Text);

registr\_card.Floor = int.Parse(registr\_card\_Edit.textBox5.Text);

registr\_card.Total\_area = double.Parse(registr\_card\_Edit.textBox2.Text);

registr\_card.Living\_area = double.Parse(registr\_card\_Edit.textBox3.Text);

registr\_card.Number\_floors = int.Parse(registr\_card\_Edit.textBox4.Text);

registr\_card.Balcony = int.Parse(registr\_card\_Edit.textBox6.Text);

registr\_card.Glazed\_balcony = int.Parse(registr\_card\_Edit.textBox7.Text);

registr\_card.Iron\_door = int.Parse(registr\_card\_Edit.textBox8.Text);

registr\_card.Repair\_information = registr\_card\_Edit.textBox9.Text;

registr\_card.Information\_registration = registr\_card\_Edit.textBox10.Text;

registr\_card.Information\_privatization = int.Parse(registr\_card\_Edit.textBox11.Text);

registr\_card.Telephone\_coverage = int.Parse(registr\_card\_Edit.textBox12.Text);

registr\_card.Gasification = int.Parse(registr\_card\_Edit.textBox13.Text);

registr\_card.Price = double.Parse(registr\_card\_Edit.textBox14.Text);

registr\_card.Telephone = registr\_card\_Edit.textBox15.Text;

registr\_card.Address = registr\_card\_Edit.textBox16.Text;

registr\_card.FIO = registr\_card\_Edit.textBox17.Text;

registr\_card.Note = registr\_card\_Edit.textBox18.Text;

registr\_card.AddRegistrCard();

if (str == "добавлены") db.Registr\_Cards.Add(registr\_card); db.SaveChanges();

dataGridView1.Refresh();

MessageBox.Show("Данные о регистрационной карте " + str);

var couponsell = new CouponSell();

couponsell.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Id == registr\_card.Id).ToList();

couponsell.label6.Text = "Продажа";

DialogResult resulte = couponsell.ShowDialog(this);

}

catch (Exception exept) { MessageBox.Show("Данные не " + str + ".\n " + exept.Message); }

}

}

private void EditClick(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out int id);

if (converted == false) return;

Registr\_Card registr\_card = db.Registr\_Cards.Find(id);

Registr\_Card\_Edit registr\_card\_Edit = new Registr\_Card\_Edit();

List<Part\_City> part\_cities = db.Part\_Cities.ToList();

registr\_card\_Edit.comboBox1.DataSource = part\_cities;

registr\_card\_Edit.comboBox1.ValueMember = "Id";

registr\_card\_Edit.comboBox1.DisplayMember = "Title";

List<Type\_House> type\_houses = db.Type\_Houses.ToList();

registr\_card\_Edit.comboBox2.DataSource = type\_houses;

registr\_card\_Edit.comboBox2.ValueMember = "Id";

registr\_card\_Edit.comboBox2.DisplayMember = "Title";

List<House\_Material> house\_materials = db.House\_Materials.ToList();

registr\_card\_Edit.comboBox3.DataSource = house\_materials;

registr\_card\_Edit.comboBox3.ValueMember = "Id";

registr\_card\_Edit.comboBox3.DisplayMember = "Title";

registr\_card\_Edit.textBox1.Text = registr\_card.Number\_rooms.ToString();

registr\_card\_Edit.textBox5.Text = registr\_card.Floor.ToString();

registr\_card\_Edit.textBox2.Text = registr\_card.Total\_area.ToString();

registr\_card\_Edit.textBox3.Text = registr\_card.Living\_area.ToString();

registr\_card\_Edit.textBox4.Text = registr\_card.Number\_floors.ToString();

registr\_card\_Edit.textBox6.Text = registr\_card.Balcony.ToString();

registr\_card\_Edit.textBox7.Text = registr\_card.Glazed\_balcony.ToString();

registr\_card\_Edit.textBox8.Text = registr\_card.Iron\_door.ToString();

registr\_card\_Edit.textBox9.Text = registr\_card.Repair\_information;

registr\_card\_Edit.textBox10.Text = registr\_card.Information\_registration;

registr\_card\_Edit.textBox11.Text = registr\_card.Information\_privatization.ToString();

registr\_card\_Edit.textBox12.Text = registr\_card.Telephone\_coverage.ToString();

registr\_card\_Edit.textBox13.Text = registr\_card.Gasification.ToString();

registr\_card\_Edit.textBox14.Text = registr\_card.Price.ToString();

registr\_card\_Edit.textBox15.Text = registr\_card.Telephone;

registr\_card\_Edit.textBox16.Text = registr\_card.Address;

registr\_card\_Edit.textBox17.Text = registr\_card.FIO;

registr\_card\_Edit.textBox18.Text = registr\_card.Note;

registr\_card\_Edit.comboBox1.SelectedItem = registr\_card.Part\_Cities;

registr\_card\_Edit.comboBox3.SelectedItem = registr\_card.House\_Material;

DialogResult result = registr\_card\_Edit.ShowDialog(this); SaveData(registr\_card\_Edit, result, registr\_card, "изменены");

}

}

private void DeleteClick(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out int id);

if (converted == false) return;

Registr\_Card registr\_card = db.Registr\_Cards.Find(id);

db.Registr\_Cards.Remove(registr\_card); db.SaveChanges();

MessageBox.Show("Регистрационная карта удалена");

}

}

private void Deregister(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out int id);

if (converted == false) return;

Registr\_Card registr\_card = db.Registr\_Cards.Find(id);

if (registr\_card.Deregister())

{

db.SaveChanges();

dataGridView1.Refresh();

MessageBox.Show("Регистрационная карта снята с учёта");

}

else MessageBox.Show("Данные не изменены");

}

}

private void Addition\_info(object sender, EventArgs e)

{

dataGridView1.DataSource = db.Registr\_Cards.Local.Where(x => x.Id.ToString().Contains(textBox1.Text)).OrderBy(x => x.Id).ToList();

var element = FindElement();

var addInfoForm = new Additional\_info();

addInfoForm.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Id == element.Id).ToList();

DialogResult result = addInfoForm.ShowDialog(this);

dataGridView1.DataSource = db.Registr\_Cards.Local.ToBindingList();

textBox1.Text = "";

}

public Registr\_Card FindElement()

{

if (dataGridView1.SelectedRows.Count > 0)

{

int x = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, x].Value.ToString(), out int id);

if (converted == false) return null;

return db.Registr\_Cards.Find(id);

}

else return null;

}

private void Removed\_register(object sender, EventArgs e)

{

var reference = new Reference();

reference.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Date\_deregistration >= DateTime.Today.AddMonths(-1) && p.Date\_deregistration <= DateTime.Today).ToList();

DialogResult result = reference.ShowDialog(this);

SaveDataRemoved(result);

}

private void SaveDataRemoved(DialogResult result)

{

if (result == DialogResult.OK)

{

try

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out int id);

if (converted == false) return;

Registr\_Card registr\_card = db.Registr\_Cards.Find(id);

registr\_card.Removed\_register();

db.SaveChanges();

MessageBox.Show("Данные о регистрационной карте изменены ");

dataGridView1.Refresh();

}

}

catch (Exception exept) { MessageBox.Show("Данные не изменены" + ".\n " + exept.Message); }

}

}

private void Number(object sender, KeyPressEventArgs e)

{

char charCode = e.KeyChar;

if (!char.IsDigit(charCode) && charCode != 8) // Разрешить вводить только цифры и символ backspace

e.Handled = true;

textBox5.MaxLength = 2;

textBox6.MaxLength = 2;

}

private void Price(object sender, KeyPressEventArgs e)

{

char charCode = e.KeyChar;

if (!char.IsDigit(charCode) && charCode != 8 && charCode != 44) // Разрешить вводить только цифры и символ backspace и запятую

e.Handled = true;

textBox7.MaxLength = 9;

textBox8.MaxLength = 9;

}

private void Type\_house\_Part\_city(object sender, KeyPressEventArgs e)

{

char charCode = e.KeyChar;

if (!char.IsLetter(charCode) && charCode != 8)// Разрешить вводить только буквы и символ backspace (код равен 8)

e.Handled = true;

textBox2.MaxLength = 20;

textBox9.MaxLength = 15;

}

private void Search(object sender, EventArgs e)

{

int num\_rooms\_min, num\_rooms\_max, num\_floor\_min, num\_floor\_max;

double price\_min, price\_max;

string part\_city, type\_house;

if (textBox3.Text == "")

{

num\_rooms\_min = 0;

}

else num\_rooms\_min = int.Parse(textBox3.Text);

if (textBox4.Text == "")

{

num\_rooms\_max = 9;

}

else num\_rooms\_max = int.Parse(textBox4.Text);

if (textBox5.Text == "")

{

num\_floor\_min = 0;

}

else num\_floor\_min = int.Parse(textBox5.Text);

if (textBox6.Text == "")

{

num\_floor\_max = 99;

}

else num\_floor\_max = int.Parse(textBox6.Text);

if (textBox7.Text == "")

{

price\_min = 0;

}

else price\_min = int.Parse(textBox7.Text);

if (textBox8.Text == "")

{

price\_max = 999999999.99;

}

else price\_max = int.Parse(textBox8.Text);

part\_city = textBox2.Text;

type\_house = textBox9.Text;

var report\_hous = new Report\_hous\_sale();

foreach(var i in db.Registr\_Cards)

{

if ((part\_city != "") && (type\_house == ""))

{

report\_hous.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Part\_Cities.ToString() == part\_city && num\_rooms\_min <= p.Number\_rooms && p.Number\_rooms <= num\_rooms\_max &&

p.Number\_floors >= num\_floor\_min && p.Number\_floors <= num\_floor\_max && p.Price >= price\_min && p.Price <= price\_max).ToList();

}

if ((part\_city == "") && (type\_house != ""))

{

report\_hous.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => p.Type\_Houses.ToString() == type\_house && num\_rooms\_min <= p.Number\_rooms && p.Number\_rooms <= num\_rooms\_max &&

p.Number\_floors >= num\_floor\_min && p.Number\_floors <= num\_floor\_max && p.Price >= price\_min && p.Price <= price\_max).ToList();

}

if ((part\_city == "") && (type\_house == ""))

{

report\_hous.dataGridView1.DataSource = db.Registr\_Cards.Local.Where(p => num\_rooms\_min <= p.Number\_rooms && p.Number\_rooms <= num\_rooms\_max &&

p.Number\_floors >= num\_floor\_min && p.Number\_floors <= num\_floor\_max && p.Price >= price\_min && p.Price <= price\_max).ToList();

}

}

DialogResult result = report\_hous.ShowDialog(this);

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using AutoHousSelectionModel;

using System.Data.Entity;

using AutoHousSelection.Forms;

namespace AutoHousSelection.Controls

{

public partial class House\_Material\_Control : UserControl

{

AutoHousContext db;

public House\_Material\_Control()

{

InitializeComponent();

db = new AutoHousContext();

db.House\_Materials.Load();

dataGridView1.DataSource = db.House\_Materials.Local.ToBindingList();

}

private void AddClick(object sender, EventArgs e)

{

House\_Material\_Edit house\_material = new House\_Material\_Edit();

House\_Material material = new House\_Material();

DialogResult result = house\_material.ShowDialog(this);

SaveData(house\_material, result, material, "добавлены");

}

private void SaveData(House\_Material\_Edit house\_material, DialogResult result, House\_Material material, string str)

{

if (result == DialogResult.OK)

{

try

{

material.Title = house\_material.textBox1.Text;

if (str == "добавлены") db.House\_Materials.Add(material); db.SaveChanges();

MessageBox.Show("Данные о материале дома " + str);

}

catch (Exception exept) { MessageBox.Show("Данные не " + str + ".\n " + exept.Message); }

}

}

private void DeleteClick(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out int id);

if (converted == false) return;

House\_Material material = db.House\_Materials.Find(id);

db.House\_Materials.Remove(material);

db.SaveChanges();

MessageBox.Show("Объект удален");

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AutoHousSelection.Controls

{

public partial class HelpControl : UserControl

{

public HelpControl()

{

InitializeComponent();

}

}

}

Приложение Б

Результаты работы

