

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Тверской государственный университет»
Факультет прикладной математики и кибернетики
Кафедра информационных технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Направление: 02.04.02 - Фундаментальная информатика и информационные
технологии

Магистерская программа: Информационные технологии в управлении и
принятии решений

Тема: «Автоматизированное составление расписания факультета на основе
эвристик и методов интеллектуальной оптимизации»

Допущен к защите

«__» _____ 2018 г.

Руководитель ООП

Выполнил:

магистрант 2 курса

Бонь Алексей Михайлович

Научный руководитель:

кандидат физ.-мат. Наук, доцент

Солдатенко Илья Сергеевич

Тверь 2018

Оглавление.

1. Введение	3
2. Построение математической модели	5
3. Генетический алгоритм и программный комплекс реализации построенной модели.....	12
Представление базы данных	14
Представление реализации информационной системы.....	20
4. Построение расписания.....	24
5. Заключение	24
6. Список литературы.....	26
Приложение 1	27
Приложение 2	31
Приложение 3	40

1. Введение

Актуальность исследования

Оптимизационные задачи возникают в различных сферах деятельности человека, таких как проектирование сложных технических систем, ведение экономической деятельности, прогнозирование природных явлений, исследования в биологии и физике и многое другое. В нашем случае это оптимизация процесса составления расписания.

Почему же задачи такого рода столь актуальны. Ручной способ составления расписания довольно сложен и включает в себя некоторые особенности, от которых можно избавиться:

- 1) излишние временные затраты;
- 2) наличие грамотного специалиста с большим опытом в составлении расписания;
- 3) хранение информации в бумажном виде;
- 4) составление происходит с помощью интуитивного подбора, соответственно не исключены ошибки, которые вынуждают начать составление расписания заново;
- 5) любые изменения в учебном плане, требует повторного составления расписания.

Возможным решением данных проблем является разработка информационной системы. Данная информационная система предоставит следующие возможности:

- 1) удобное хранение данных;
 - 1.1) о факультетах;
 - 1.2) о преподавателях;
 - 1.3) о студентах;
 - 1.4) об аудиторном фонде;
 - 1.5) о группах;
 - 1.6) об учебном плане;

- 1.7) о расписании;
- 2) удобный и быстрый доступ к данным;
- 3) расчет нескольких вариантов расписаний (если позволяют ограничения);
- 4) изменение исходных данных и перерасчет текущего расписания;
- 5) выбор оптимального расписания.

Обзор литературы

Важной частью магистерской диссертации является изучение литературы. Для написания данной работы были использованы научная и учебно-методическая литература, а так же научные статьи.

Для изучения генетического алгоритма было рассмотрено несколько статей, таких как С.А. Исаев «Генетические алгоритмы в задачах оптимизации» [5], В.А Попов и другие «Оптимизационные задачи на основе генетического поиска» [6], Д.И. Батищев и другие, Коробкин А.А. «Использование агрегативного генетического алгоритма для составления расписания» [1], Панченко Т.В. «Генетические алгоритмы» [2].

Также, для ознакомления была изучена иностранная литература. Основные понятия и определения были рассмотрены в книгах Prakash A, Chan FTS, Deshmukh SG (2011) FMS scheduling with knowledge based genetic algorithm approach. [8] и Xing KYU, Han LB, Zhou MC, Wang F (2012) Deadlock-free genetic scheduling algorithm for automated manufacturing systems based on deadlock control policy. [14].

На основе статьи С.А. Исаева «Генетические алгоритмы в задачах оптимизации» [5], были изучены инструментарий генетического алгоритма, т.к. в общем случае это NP-полная задача.

Цели и задачи

В рамках магистерской работы передо мной были поставлены следующие цели и задачи:

- 1) исследование предметной области:
 - 1.1. изучение необходимой литературы;
 - 1.2. общение с сотрудниками привлеченными к составлению расписания;
- 2) построение модели задачи составления оптимального расписания;
- 3) проектирование базы данных;
- 4) разработка информационной системы для ввода параметров модели и их последующего расчета.

2. Построение математической модели

Исследование предметной области

Основной целью исследования предметной области, был сбор данных и их анализ. В рамках сбора данных, было выполнено следующее:

- 1) изучена необходимая литература;
- 2) опрошен специалист в области составления расписания;
- 3) получены данные и принцип, по которым составляется расписание;
- 4) получены сведения об ограничениях, которые создают преподаватели и естественные ограничения налагаемых на расписание (например, одна группа не может в один момент времени находиться на двух занятиях);

Ограничения, которые учитываются при составлении расписания, заданные аудиторным фондом:

- 1) вместимость аудитории;
- 2) тип аудитории;
- 3) наличие доски;
- 4) наличие розеток;
- 5) наличие технического оборудования (компьютеры, проектор, полотно для проектора);

- б) деление аудитории на несколько факультетов, т.е. доступность этой аудитории для разных факультетов.

Под естественными ограничениями подразумевается следующее:

- 1) группа студентов «А» не может в один момент времени находиться на занятии «Б» и «В»;
- 2) в один момент времени у группы студентов «А», не может проводиться занятие в аудитории «А1», если в этот же момент времени у группы «Б» также запланировано занятие в аудитории «А1»;
- 3) в один момент времени, преподаватель «А», не может проводить занятие у группы «Б» и группы «В»;

Ограничения, которые задаются преподавателями, обширны и могут быть довольно специфичными, например: наличие в аудитории жалюзи. Но есть ограничения, которые встречаются чаще всего, например это время проведения занятий. У каждого преподавателя свои временные рамки, в которые имеется возможность проводить занятия.

Составление математической модели задачи

Рассматривается учебное заведение, в котором выделяются следующие группы объектов:

- 1) $\exists F$ – множество **факультетов**, где $(f_0, f_1, \dots, f_n) \in F$
, f_i – это выбранный факультет, имеющий набор параметров указанный в **таблице 1**;
- 2) $\exists Sa$ – множество **расписаний звонков** факультета, где $(sa_0, sa_1, \dots, sa_n) \in Sa$
, sa_i
– это выбранный звонок для факультета, имеющий набор параметров указанный в **таблице 9**
, $sa_i \in f_j$;
- 3) $\exists T$ – множество **преподавателей**, где $(t_0, t_1, \dots, t_n) \in T$
, t_i
– это выбранный преподаватель, имеющий набор параметров указанный в **таблице 4**
, $t_i \in f_j$;
- 4) $\exists R$ – множество **званий преподавателей**, где $(r_0, r_1, \dots, r_n) \in R$

- , r_i
 – это выбранное звание преподавателя, имеющее набор параметров указанный в **таблице 11**
- , $r_i \in t_j$;
- 5) $\exists G$ – множество **учебных групп**, где $(g_0, g_1, \dots, g_n) \in G$
- , g_i
 – это выбранная учебная группа, имеющая набор параметров указанный в **таблице 3**
- , $g_i \in f_j$;
- 6) $\exists S$ – множество **учебных предметов**, где $(s_0, s_1, \dots, s_n) \in S$
- , s_i
 – это выбранный учебный предмет, имеющий набор параметров указанный в **таблице 5**
- , $s_i \in g_j$;
- 7) $\exists H$ – множество **аудиторий**, где $(h_0, h_1, \dots, h_n) \in H$
- , h_i – это выбранная аудитория, имеющая набор параметров указанный в **таблице 2**
- , $h_i \in f_j$;
- 8) $\exists St$ – множество **предметов назначенных преподавателю**, где $(st_0, st_1, \dots, st_n) \in St$
- , st_i – это выбранный предмет и преподаватель
 , имеющие набор параметров указанный в **таблице 6**
- , $st_i \in f_j, st_i \in s_k, st_i \in t_m$;
- 9) $\exists L$ – множество **ограничений**, где $(l_0, l_1, \dots, l_n) \in L$
- , l_i – это выбранное ограничение, имеющее набор параметров указанный в **таблице 7**;
- 10) $\exists Wt$ – множество **ограничений выбранных преподавателем для предмета**
 , где $(wt_0, wt_1, \dots, wt_n) \in Wt$
- , wt_i – это выбранное пожелание преподавателя
 , имеющее набор параметров указанный в **таблице 8**
- , $wt_i \in st_j, wt_i \in l_k$;
- 11) $\exists W$ – множество **учебных дней в неделе**
 , где $(w_0, w_1, \dots, w_n) \in W$
- , w_i –
 это выбранный учебный день, имеющий набор параметров указанный в **таблице 10**.

Необходимо найти:

$\exists Rs$ – множество **расписаний**, где $(rs_0, rs_1, \dots, rs_n) \in Rs$

, rs_i – это выбранная часть расписания, имеющая набор параметров указанный в **таблице 130**

, $rs_i \in f_j, rs_i \in st_k, rs_i \in h_m, rs_i \in sa_u$.

При создании расписания необходимо учитывать множество требований, и ограничений. Условно, весь ряд ограничений разбивается на обязательные ограничения и желательные. К обязательным относятся [3,4]:

- исключение накладок различного вида;
- минимизация не заполненного времени в течение учебного дня;
- соответствие типов проводимого занятия, аудитории проведения;
- ограничение на количество проводимых занятий в течение учебного дня;
- соответствие количества учебных часов в расписании, количеству запланированных часов в учебном плане.

Желательные требования представляются следующим образом:

- создание благоприятных условий для обучения, к примеру, аудитории по типу соответствуют типу занятия, возможность проведения совместных занятий для групп, спортивные предметы проводятся в конце дня и т.д.;
- ограничения от преподавателей;
- равномерное распределение нагрузки на студентов в течение семестра и конкретного учебного дня.

Математическая модель ограничений накладываемых на расписание занятий выглядит следующим образом [3-4]:

- 1) $\forall (h_i, s_j): h_i \in H, s_j \in S (\exists! rs_k: (h_i = h_k) \cap (rs_k \in Rs^{s_j})) \cup (\nexists rs_k: (h_i = h_k) \cap (rs_k \in Rs^{s_j}))$, где $R_s^{s_j}$ – множество блоков занятий, проводимых во время пары s_j .

Данное ограничение исключает пересечения по аудиториям. Для каждой упорядоченной двойки элементов: аудитория и пара, у аудитории существует либо единственный блок занятий из множества R_s , что означает

проведение занятия этого блока в этой аудитории в момент данной пары, либо отсутствие блока занятия, указывающее на то, что аудитория свободна.

$$2) \forall (t_i, s_j): t_i \in T, s_j \in S (\exists! rs_k: (t_i = t_k) \cap (rs_k \in Rs^{s_j})) \cup (\nexists rs_k: (t_i = t_k) \cap (rs_k \in Rs^{s_j})), \text{ где } Rs^{s_j} -$$

множество блоков занятий, проводимых во время пары s_j .

Данное ограничение исключает для преподавателя пересечения в занятиях: существует либо единственный блок занятий, которые проводит данный преподаватель во время заданной пары, либо этого блока не существует вообще.

$$3) \forall (g_n, s_j): g_n \in G, s_k \in S \sum rs_i^e \leq 1, i \in Rs^{g_n} \cap Rs^{t_j}, \text{ где } Rs^{g_n} -$$

множество блоков занятий, в которых присутствует группа g_n , а Rs^{t_j} – множество блоков занятий, проводящихся во время пары t_k .

Данное ограничение исключает накладки для учебных групп, т.е. для каждой пары элементов: группа и пара, сумма компонент rs_i^e вектора rs_i блоков из множества $Rs^{g_n} \cap Rs^{t_j}$ не превышает единицы. Во время конкретной пары группа находится на одном занятии, или проводится занятие только у одной из подгрупп, либо у обеих, либо занятий нет вообще.

$$4) \forall rs_i \in Rs \ h_i \in H^{rs_i^h}, \text{ данное ограничение обеспечивает соответствие типа аудитории типу проводимого занятия, т.е. для каждого блока занятия } rs_i, rs_i \in RS \text{ аудитория выбирается из допустимого подмножества аудиторий, код этого подмножества хранит компонента } rs_i^h$$

$$5) \forall (w_j, g_n): w_j \in W, g_n \in G \sum rs_i^e \leq N_{max}, i \in I_{g_n}^{w_j}.$$

Данное ограничение, исключает превышение количества занятий проводимых в течение учебного дня, означает, что для каждой пары элементов: группа и день, число проводимых пар не превышает максимально допустимого – N_{max} .

б) $\forall (w_\tau, g_n): w_\tau \in W, g_n \in$

$$G(\sum_{i \in I_{g_n}^{w_\tau}} rs_i^e = t_max_num_{g_n}^{w_\tau} - t_min_num_{g_n}^{w_\tau} + 1) \cap (\forall st: t_min l_{g_n}^{w_\tau} \leq st \leq t_max l_{g_n}^{w_\tau} \sum_{i: rs_i \in Rs^{g_n} \cap st_j^p = t} rs_i^e = 1)$$

$t_max_num_{g_n}^{w_\tau}$ - максимальный номер пары в течение дня w_τ , для группы g_n ,

$t_min_num_{g_n}^{w_\tau}$ - минимальный номер пары в течение дня w_τ , для группы g_n ,

$I_{g_n}^{w_\tau} = \{i: (rs_i \in Rs^{g_n}) \cap (st_j^d = w_\tau)\}$ - множество номеров блоков занятий, проводимых для группы g_n во время дня w_τ .

Данное ограничение, исключает наличие окон для учебных групп, т.е. для каждой пары элементов: день и группа, количество пар, проводимых в этой группе в текущий день должно равняться величине: K - критерий качества основывается на желательных требованиях и имеет следующий вид:

$K = \varphi(rs) = \sum_{i=1}^N c_i \omega_i(rs)$, где c_i - значение штрафного коэффициента за невыполнение i -го требования, а ω_i - оценка степени невыполнения i -го желательного требования.

K наиболее значимым желательным требованиям относятся:

1. Ограничения от преподавательского состава: Рассмотрим 2 матрицы для описания данного требования. Первая матрица называется матрицей запретов и выглядит следующим образом:

$$M_{запретов} = \begin{cases} 1, \text{запрет проведения занятия} \\ \text{для } i - \text{го преподавателя} \\ \text{во время } k - \text{ой пары} \\ 0, \text{отсутствие запрета} \end{cases}$$

Вторая матрица - матрица занятости формируется следующим образом:

$$M_{занятости} = \begin{cases} 1, \text{ проведение занятия} \\ \text{для } i - \text{го преподавателя} \\ \text{во время } k - \text{ой пары} \\ 0, \text{ отсутствие занятий} \end{cases}$$

$$\sum_{t_i \in T} \sum_{st_j \in St} (\overline{M_{запретов} \cap M_{занятости}}) \rightarrow \min$$

Необходимо учесть пожелания преподавателей.

2. $\sum \sum (t_{max}_{t_i}^{w_\tau} - t_{min}_{t_i}^{w_\tau}) - N_{блоков} \rightarrow \min$, где $t_{max}_{t_i}^{w_\tau}$ – максимальный номер пары в день w_τ у преподавателя t_i , а $t_{min}_{t_i}^{w_\tau}$ – минимальный номер пары в день w_τ у преподавателя t_i .

Это ограничение уменьшения количества пустых промежутков между парами.

3. $S_n^{cp} = \frac{1}{N_{дней}} \sum_{w_\tau} (M_n^{cp} - |I_{g_n}^{w_\tau}|)^2$ – желательное требование равномерности занятий.

Если среднее отклонение количества занятий для группы g_n

$$M_n^{cp} = \frac{1}{N_{дней}} \sum_{w_\tau \in W} |I_{g_n}^{w_\tau}|, \text{ отсюда требование выглядит так:}$$

$$\sum_{n=1, N_{групп}} S_n^{cp} \rightarrow \min$$

С помощью описанных требований строится целевая функция на основе минимизации штрафных показателей. Каждое нарушение ограничения или желательного требования увеличивает значение целевой функции в соответствии с коэффициентом значимости требования. В результате целевая функция в общем виде описывается формулой:

$$F_\mu = \left(\sum_{n=1, N_{\text{количество ограничений}}} og_i * k_{og_i} \right) + K$$

На основе математической модели расписания, описанной в пунктах 1–3, производится реализация эволюционного поиска оптимального расписания с использованием генетического алгоритма.

3. Генетический алгоритм и программный комплекс реализации построенной модели

Алгоритм разработанный для составления расписания строится на основе следующих шагов [1]:

- 1) создание начальной популяции;
- 2) слияние особей случайными значениями функции пригодности;
- 3) мутация потомства;
- 4) отбор особей в новую популяцию;
- 5) оценка штрафов по ограничениям;
- 6) выбор наилучшей особи.

1. Формирование начальной популяции

На первом шаге, начальная популяция состоящая из заданной числа N особей, заполняется случайными значениями, где особь популяции это отдельный вариант учебного дня с случайно заданным, днем недели, парой и аудиторией. [1][2]

2. Слияние

Вторым шагом генетического алгоритма является слияние. На данном шаге основную функцию выполняет оператор кроссинговера. Это языковая конструкция, позволяющая на основе скрещивания хромосом родителей создавать хромосомы потомков. [1][2]

3. Мутация потомства

К некоторым из особей созданным после скрещивания применяется оператор мутации. Мутация несет в себе важную роль в работе генетического алгоритма: вносит дополнительное разнообразие в текущую популяцию и тем самым предоставляет возможность поиска оптимального решения. [1][2]

4. Оператор отбора

Данный оператор выполняет функции фильтрующего инструмента, который выделяет в составе популяции особи, имеющие низкое значение функции пригодности. Выделенные найденные слабые особи удаляются из популяции до тех пор, пока численность не становится исходной. [1][2]

5. Оценка популяции на основе ограничений

После использования отбора, слияния, мутации, получаем новую популяцию, популяцию потомков, она же заменит родительскую популяцию. После этого происходит оценка популяции согласно заданным ограничениям, и начисляются штрафы за невыполнение данных ограничений. Но если при оценке, не получает ни одного штрафа, значит данная популяция считается "лучшей" и работа генетического алгоритма прекращается. Если "лучшая" популяция не была найдена, то алгоритм начинает заново свою работу, но уже с шага отбора и процесс поиска продолжается. [1][2]

6. Выбор «лучшего» решения

На данном шаге, из полученных особей, выбирается лучшая, она и является решением задачи. Лучшая особь это та, у которой значение оценки на основе ограничений равняется 0 или является минимальным.

Описанный алгоритм был реализован с использованием языка C++/Qt. Для отладки данного алгоритма использовались реальные данные, предоставленные Академической гимназией Тверского Государственного Университета. В испытаниях принимало участие порядка ста предметов. Между описанными дисциплинами были определены связи, которые характеризуют очередность следования предметов друг за другом. Информация о дисциплинах и связях между ними хранится в базе данных, работа с которой реализуется через интерфейс. [1][2]

Проектирование базы данных

Для хранения всей информации о факультетах, преподавателях, студентах, аудиторном фонде, учебном плане, нам потребуется база данных. При проектировании информационной системы, было решено использовать **PostgreSQL**, т.к. данная разновидность БД имеет бесплатный доступ, удобный и практичный интерфейс.

Представление базы данных

udb_faculty					
Поле	Тип	Параметры	PK	FK	Комментарий
idFaculty	SERIAL	UNIQUE	+	-	идентификатор факультета
nameFaculty	VARCHAR(256)	NOT NULL	-	-	наименование факультета
codeFaculty	INT	NOT NULL	-	-	код факультета
dean	VARCHAR(56)	-	-	-	декан
adrFaculty	VARCHAR(256)	-	-	-	адрес факультета
phoneNumber	VARCHAR(28)	-	-	-	телефон факультета

Табл. 1 – представление таблицы udb_faculty о факультетах.

Столбцы codeFaculty, nameFaculty в интерфейсе программы будут определяться из таблицы-справочника табл.12. Так же при расширении программы до размеров АИС, имеет смысл добавить такие столбцы как: адрес сайта, электронная почта и пр.

udb_lecturehalls					
Поле	Тип	Параметры	PK	FK	Комментарий
idHall	SERIAL	UNIQUE	+	-	идентификатор аудитории
idFaculty	INT	NOT NULL	-	+	идентификатор факультета
numberHall	INT	NOT NULL	+	-	номер
typeHall	INT	NOT NULL	-	-	тип
capacityHall	INT	NOT NULL	-	-	вместимость
codePriority	INT	DEFAULT '107'	-	-	приоритет факультета
haveBoard	BOOLEAN	DEFAULT 'true'	-	-	есть доска
haveProjector	BOOLEAN	DEFAULT 'true'	-	-	есть проектор
haveSocket	BOOLEAN	DEFAULT 'true'	-	-	есть розетка
haveWindows	BOOLEAN	DEFAULT 'true'	-	-	есть окна
X_SUBJECT_TYPE_CODE	VARCHAR(10)	DEFAULT 'All'	-	-	тип аудитории для физ-ры/обж/пр.

Табл. 2 – представление таблицы udb_lecturehalls об аудиторном фонде.

Столбец X_SUBJECT_TYPE_CODE в интерфейсе программы будет определяться из таблицы-справочника табл. 12.

udb_groups					
Поле	Тип	Параметры	PK	FK	Комментарий
idGroup	SERIAL	UNIQUE	+	-	идентификатор группы
idFaculty	INT	NOT NULL	-	+	идентификатор факультета
numberGroup	INT	NOT NULL	-	-	номер
quantityStudents	INT	NOT NULL	-	-	кол-во студентов
trend	VARCHAR(256)	NOT NULL	-	-	направление

Табл. 3 – представление таблицы udb_groups о группах студентов/учеников.

udb_teachers					
Поле	Тип	Параметры	PK	FK	Комментарий
idTeacher	SERIAL	UNIQUE	+	-	идентификатор преподавателя
idRank	INT	NOT NULL	-	+	идентификатор звания
idFaculty	INT	NOT NULL	-	+	идентификатор факультета
middleNameTeacher	VARCHAR(32)	NOT NULL	-	-	отчество
lastNameTeacher	VARCHAR(32)	NOT NULL	-	-	фамилия
firstNameTeacher	VARCHAR (32)	NOT NULL	-	-	имя
initialTeacher	VARCHAR (64)	NOT NULL	-	-	инициалы

Табл. 4 – представление таблицы udb_teachers преподавателях/учителях.

Так же при расширении программы до размеров АИС, можно дополнить таблицу, столбцами, отвечающими за контактную информацию о преподавателе.

udb_subject					
Поле	Тип	Параметры	PK	FK	Комментарий
idSubject	SERIAL	UNIQUE	+	-	идентификатор предмета
idGroup	INT	NOT NULL	-	+	идентификатор группы
nameSubject	VARCHAR (128)	NOT NULL	-	-	наименование предмета
quantityHours	INT	-	-	-	кол-во часов
lecture	INT	-	-	-	кол-во лекций
practice	INT	-	-	-	кол-во практик
quantityStudents	INT	-	-	-	кол-во студентов
quantityGroups	INT	-	-	-	кол-во групп
quantitySubgroups	INT	-	-	-	кол-во подгрупп

Табл. 5 – представление таблицы udb_subject об учебных предметах.

udb_subjectteacher					
Поле	Тип	Параметры	PK	FK	Комментарий
idST	SERIAL	UNIQUE	+	-	идентификатор предмета преподавателя
idFaculty	INT	NOT NULL	-	+	идентификатор факультета
idTeacher	INT	NOT NULL	-	+	идентификатор преподавателя
idSubject	INT	NOT NULL	-	+	идентификатор предмета
inactive	BOOLEAN	DEFAULT false	-	-	неактивная запись
X_SUBJECT_TYPE_CODE	VARCHAR(10)	DEFAULT 'All'	-	-	необходимый тип аудитории физ-ра/обж/пр.

Табл. 6 – представление таблицы udb_subjectteacher о учебном плане.

Столбец X_SUBJECT_TYPE_CODE в интерфейсе программы будет определяться из таблицы-справочника табл.12.

udb_wishesteacher					
Поле	Тип	Параметры	PK	FK	Комментарий
idWishes	SERIAL	UNIQUE	+	-	идентификатор пожелания преподавателя
idSubTeach	INT	NOT NULL	-	+	идентификатор предмета преподавателя
idLimitation	INT	NOT NULL	-	+	идентификатор ограничения

Табл. 7 – представление таблицы udb_wishesteacher о пожеланиях преподавателей/учителей.

udbc_limitations					
Поле	Тип	Параметры	PK	FK	Комментарий
idLimitation	SERIAL	UNIQUE	+	-	идентификатор ограничения
codeLimitation	INT	NOT NULL	-	-	код ограничения
nameLimitation	VARCHAR(128)	NOT NULL	-	-	наименование ограничения

Табл. 8 – представление таблицы udbc_limitations об ограничениях.

udbc_schedulealarm					
Поле	Тип	Параметры	PK	FK	Комментарий
idAlarm	SERIAL	UNIQUE	+	-	идентификатор звонка
idFaculty	INT	NOT NULL	-	+	идентификатор факультета
pairNumber	INT	NOT NULL	-	-	номер пары
timeBegin	TIME	NOT NULL	-	-	время начала пары
timeEnd	TIME	NOT NULL	-	-	время окончания пары

Табл. 9 – представление таблицы udbc_schedulealarm о расписании звонков.

udbc_week					
Поле	Тип	Параметры	PK	FK	Комментарий
idday	SERIAL	UNIQUE	+	-	идентификатор дня
dayname	VARCHAR(16)	NOT NULL	-	-	наименование дня
cutname	VARCHAR(4)	NOT NULL	-	-	сокращенное наименование дня
daynumber	INT	NOT NULL	-	-	номер дня
idfaculty	INT	NOT NULL	-	+	идентификатор факультета

Табл. 10 – представление таблицы udbc_week о расписании учебных дней факультета.

udbc_ranks					
Поле	Тип	Параметры	PK	FK	Комментарий

idrank	SERIAL	UNIQUE	+	-	идентификатор звания
rankname	VARCHAR(32)	NOT NULL	-	-	наименование звания
cutname	VARCHAR(16)	NOT NULL	-	-	сокращенное наименование звания
importance	INT	NOT NULL	-	-	важность/приоритет звания

Табл. 11 – представление таблицы udbc_ranks о званиях преподавателей/учителей.

udb_list_val					
Поле	Тип	Параметры	PK	FK	Комментарий
idlist	SERIAL	UNIQUE	+	-	идентификатор списка
type	VARCHAR(256)	NOT NULL	-	-	тип/имя списка
value	VARCHAR(256)	NOT NULL	-	-	код значения в списке
name	VARCHAR(56)	NOT NULL	-	-	наименование значения в списке
descr	VARCHAR(56)	-	-	-	описание значения
X_INACTIVE	BOOLEAN	DEFAULT false	-	-	неактивная запись

Табл. 12 – представление таблицы udb_list_val о списках значений.

res_Schedule					
Поле	Тип	Параметры	PK	FK	Комментарий
idSchedule	SERIAL	UNIQUE	+	-	идентификатор расписания
codeSchedule	INT	NOT NULL	-	+	код расписания
idFaculty	INT	NOT NULL	-	+	идентификатор факультета
idDay	INT	NOT NULL	-	+	идентификатор дня
idSubjectTeacher	INT	NOT NULL	-	+	идентификатор предмета преподавателя
idHall	INT	NOT NULL	-	+	идентификатор аудитории
idAlarm	INT	NOT NULL	-	+	идентификатор номера пары

Табл. 13 – представление таблицы res_Schedule о составленном расписании.

На рис. 14 представлена схема спроектированной базы данных.

Разработка информационной системы для ввода параметров модели

Информационная система реализуется на языке программирования C/C++, в среде разработки Qt. На текущий момент реализованы следующие возможности: подключение к БД, просмотр, редактирование записей, добавление или удаление записей, динамически расширяемая загрузка таблиц, экспорт данных, фильтрация записей [4].

Представление реализации информационной системы

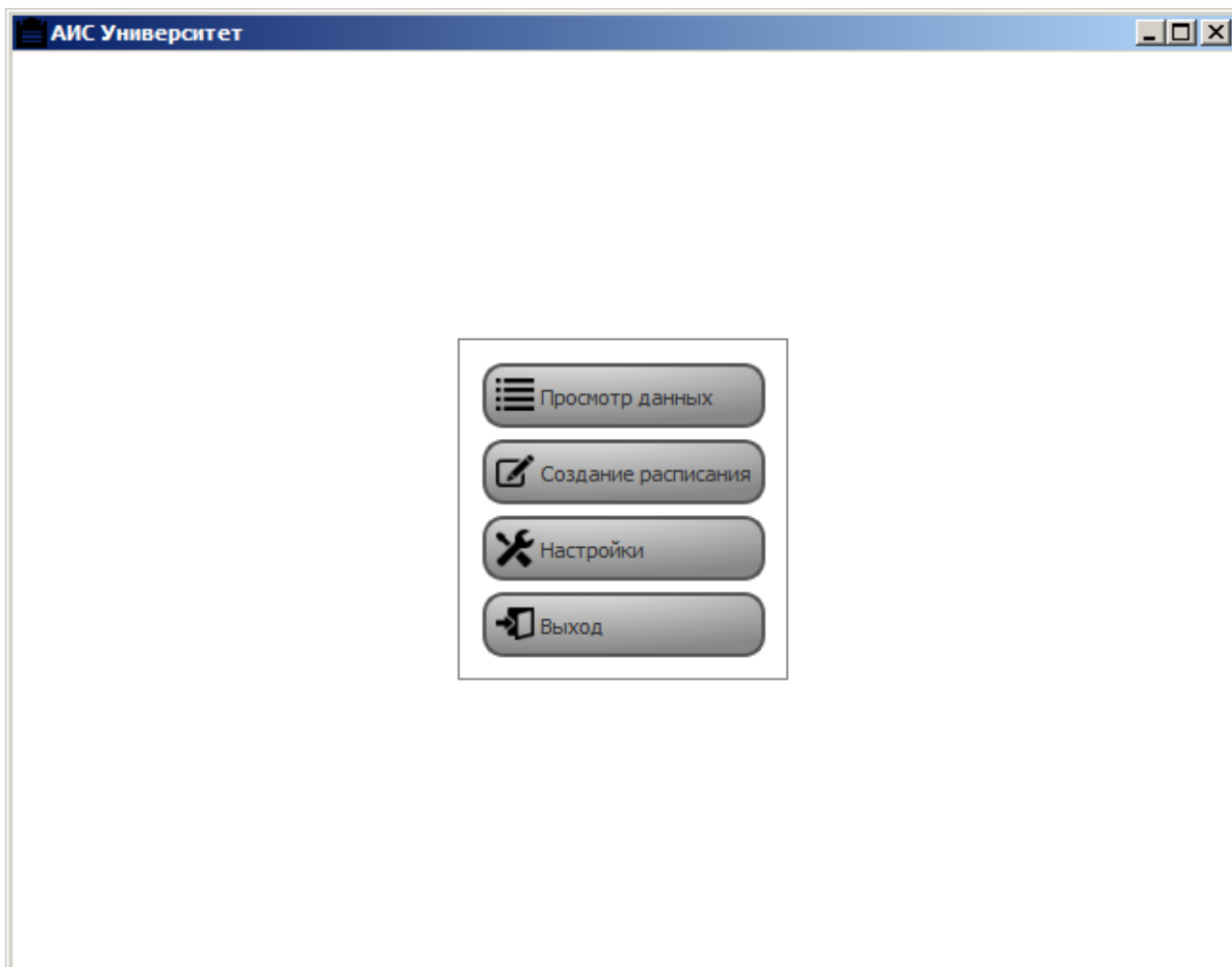


Рис 12. – стартовое меню программы.

В стартовом меню программы (рис. 12) можно увидеть стандартный набор кнопок. Назначение которых, интуитивно понятно.

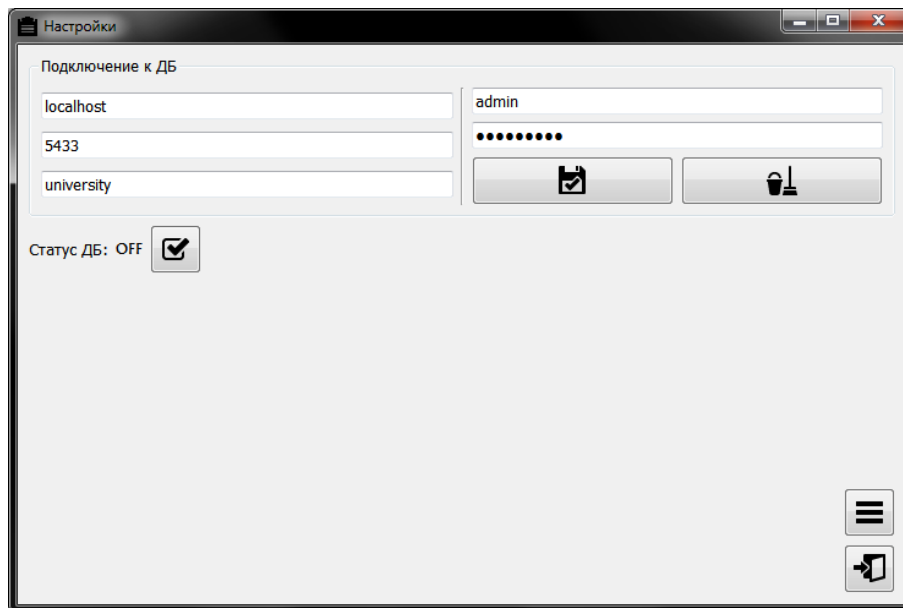


Рис. 13 – меню настроек.

Перейдя по кнопке «Настройки», откроется меню настроек (рис. 13). Данное окно предназначено для настройки подключения к базе данных. Здесь мы видим все тот же интуитивно понятный набор кнопок и полей. Для каждой кнопки добавлен параметр ToolTip (всплывающая подсказка), на случай, если возникает недопонимание, для чего необходима кнопка или поле. Для каждого поля ввода добавлен Placeholder (маркер вставки), для четкого понимания, какое поле, для каких данных предназначается (рис. 14).

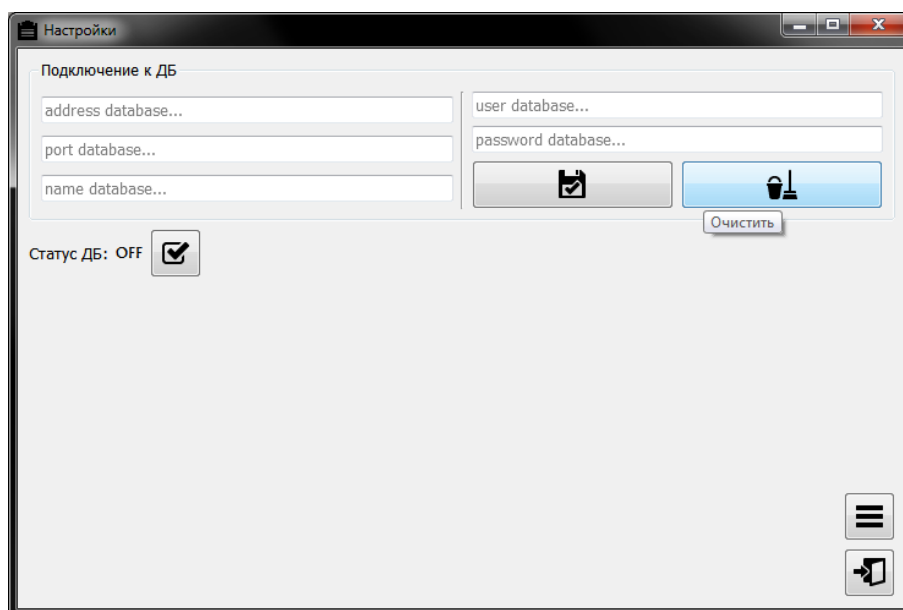


Рис. 14 – меню настроек (Placeholder + ToolTip).

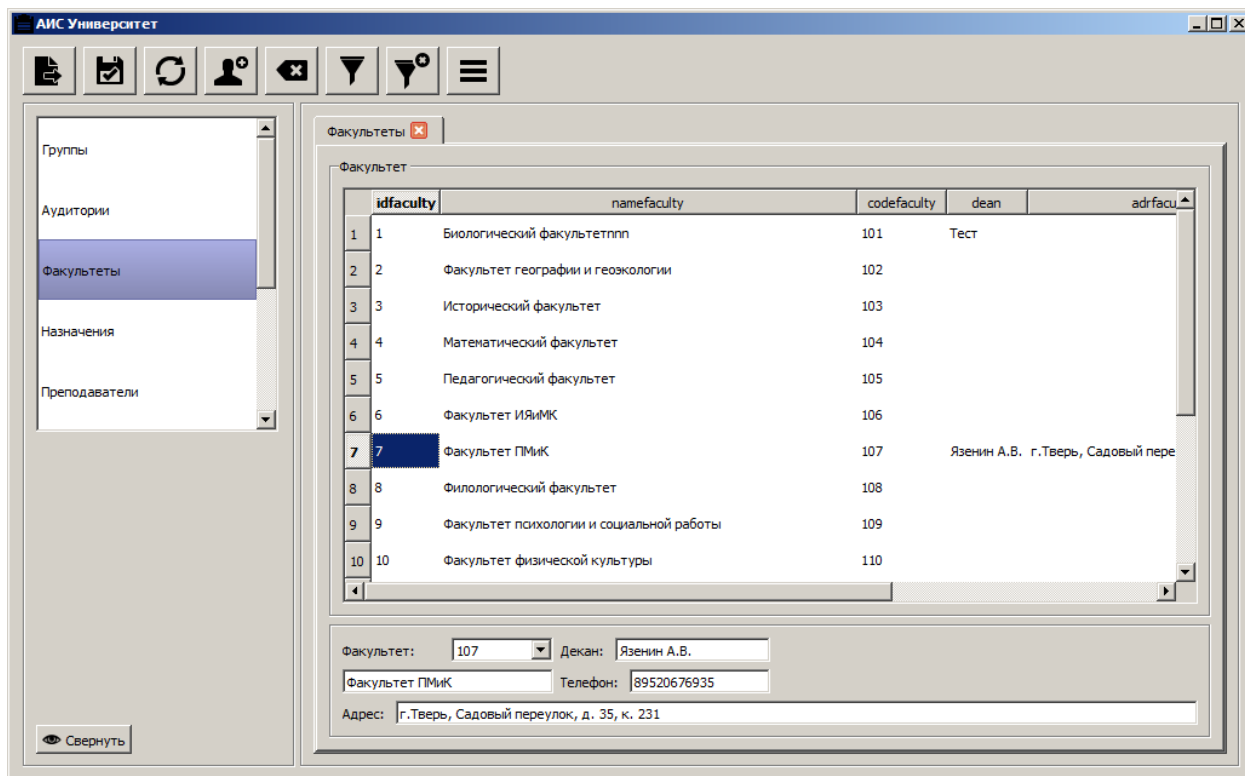


Рис. 15 – окно просмотра базы данных.

Перейдя по кнопке «Просмотр данных», откроется окно просмотра базы данных (рис. 15). Верхнюю часть окна занимаю вспомогательные кнопки для работы с данными.



- кнопка экспорта данных в формате CSV.



- кнопка сохранения произведенных изменений с данными.



- кнопка обновления данных.



- кнопка добавления новой записи в выбранной таблице.



- кнопка удаления выбранной записи.



- кнопка добавления фильтрации данных. Вызывает вспомогательное окно в котором вводятся необходимые значения для фильтра (рис. 16).

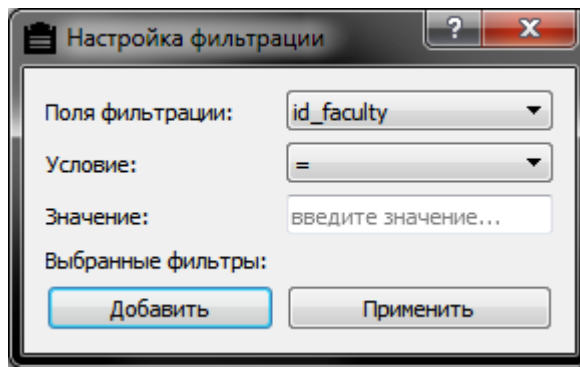


Рис. 16 – окно ввода значений фильтра.



- кнопка удаления всех фильтров.



- кнопка перехода в главное меню программы.

Перейдя по кнопке «CreateSchedule», откроется меню создания расписания (рис. 17). Данное окно предназначено для настройки параметров расписания. Здесь мы видим все тот же интуитивно понятный набор кнопок и полей. Для каждой кнопки добавлен параметр ToolTip (всплывающая подсказка), на случай, если возникает недопонимание, для чего необходима данная кнопка. Также для наглядности добавлен ProgressBar, отображающий процесс построения расписания

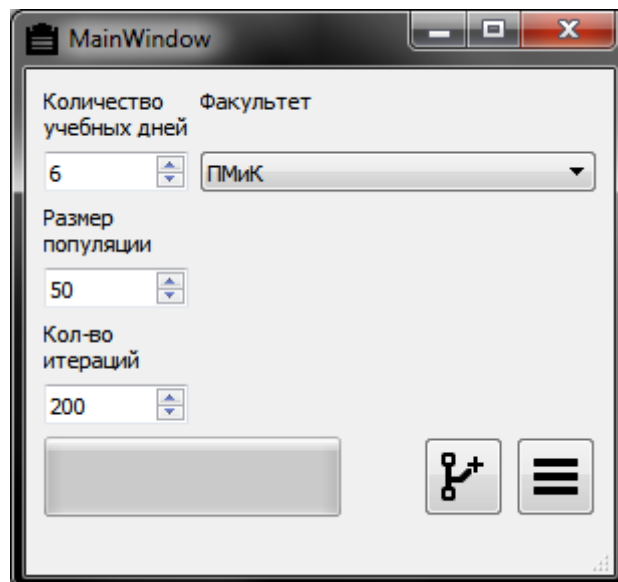


Рис. 17 – окно создания расписания



- кнопка для начала построения расписания.



- кнопка перехода в главное меню программы.

4. Построение расписания

На основе математической модели построения расписания и математических моделях описывающих ограничения накладываемых на расписание, описанных в **главе 2**, с использованием базы данных и информационной системы описанной в **главе 3**, в связке с **приложением 1**, было построено расписание.



Рис. 18 – изменение суммы штрафов.

На рисунке 18, изображено колебание суммы штрафов поитерационно.

Из графика можно увидеть, что 2-ое расписание, на 673-ей итерации достигла значения = 0, по сумме всех штрафов на основе ограничений, а это значит, что построенное расписание удовлетворяет всем ограничениям. Результат построенного расписания можно увидеть в **приложении 2**.

5. Заключение

В ходе выполнения работы была изучена литература на темы:

- генетические алгоритмы;
- построение расписания на основе генетических алгоритмов;
- задачи необходимой оптимизации.

Также была описана математическая модель оптимального расписания и математическая модель ограничений накладываемых на данное расписание.

Для полученной модели был разработан программный комплекс для ввода/вывода данных и расчета оптимального расписания.

6. Список литературы

1. Коробкин А.А. Использование агрегативного генетического алгоритма для составления расписания // Коробкин А.А. //Вестник Воронежского государственного технического университета, том 5, № 11,2009. –С. 184–186.
2. Панченко Т.В. Генетические алгоритмы // Издательский дом «Астраханский университет» 2007г.
3. Солдатенко И.С., Пильщиков Д.Е., Язенин А.В. О методе решения одной задачи необходимостной оптимизации // Вестник Тверского гос. ун-та. Сер. Прикладная математика. №26(86).
4. Шлее М. Qt 4.8. Профессиональное программирование на C++. — СПб.: БХВ-Петербург, 2012. — 912 с.
5. С.А. Исаев, Популярно о генетических алгоритмах.
6. В.А. Попов, А.В. Бердочник, Оптимизационные задачи на основе генетического поиска.
7. https://ru.wikipedia.org/wiki/Генетический_алгоритм. Генетический алгоритм.
8. Prakash A, Chan FTS, Deshmukh SG (2011) FMS scheduling with knowledge based genetic algorithm approach. Expert Syst Appl 38(4):3161–3171.
9. Xing KYU, Han LB, Zhou MC, Wang F (2012) Deadlock-free genetic scheduling algorithm for automated manufacturing systems based on deadlock control policy. IEEE Trans Syst Man Cybern Part B Cybern 42(3):603–615.

Приложение 1

Аудитории

№п/п	За кем закреплен	Вместимость, чел.
204		20
205	5А класс	10
206	5Б класс	10
208	6А класс	10
211	7А класс	16
212	6Б класс	10
213	кабинет физики	14
219	кабинет ОБЖ	20
302	мал.спортзал	20
305	8А класс	18
306	8Б класс	18
307		20
309		10
310	7Б класс	12
312	конференц-зал	50
408		14
409	кабинет технологии для 5-7 кл.	12
411	9Б класс	12
301 корпБ		18
49 корп2		30
корпус Б	тренажерный зал для 7-9 кл.	30
корпус №3	спортивный зал	

Учебный план 5-9 классы

№	Учебный предмет	Количество учебных часов в неделю									
		5а	5б	6а	6б	7а	7б	8а	8б	9а	9б
1	Русский язык	5	5	6	6	4	4	3	3	2	2
2	Литература	3	3	3	3	2	2	2	2	3	3
3	Английский язык	3	3	3	3	3	3	3	3	3	3
4	Немецкий язык	2	2								
5	Математика	5	5	5	5			6	6	5	5
	Алгебра					3	3				
	Геометрия					2	2				
6	Информатика и ИКТ					2	2	1	1	1	2
7	География	1	1	1	1	2	2	2	2	2	2
8	Биология	1	1	1	1	1	1	2	2	2	2
9	Физика					2	2	2	2	2	2
10	Химия							2	2	2	2
11	История	2	2	2	2	2	2	2	2	2	2
12	Искусство (МХК)							1	1	1	1
13	Обществознание			1	1	1	1	1	1	1	1
14	Музыка	1	1	1	1	1	1				
15	ИЗО	1	1	1	1	1	1				
16	Физкультура	3	3	3	3	3	3	3	3	3	3
17	Технология	2	2	2	2	1	1	1	1	1	
18	ОБЖ							1	1		
19	Математика (компонент ОУ)							1	1		
20	Путешествия в слова, электив			1	1						
21	В мире чисел, электив					1	1				
22	Трудные вопросы орфографии, электив					1	1				
23	Тайный мир текста, электив							1	1		
24	Теоретические основы русского языка, электив									1,5	1,5
25	Трудные вопросы математики, решение задач в формате ОГЭ, электив									1,5	1,5
	Всего:	29	29	30	30	32	32	33	33	33	33

Расписание звонков

№ урока	Время проведения
1	09.00-09-45
2	09.55-10.40
3	10.50-11.35
4	12.05-12.50
5	13.00-13-45
6	13.55-14.40
7	14.50-15.35
8	15.45-16.30

Учебный план 5-9 классы

№	Учебный предмет	Количество учебных часов в неделю									
		5а	5б	6а	6б	7а	7б	8а	8б	9а	9б
1	Русский язык	Ганжина И.М.	Андреева Т.Н.	Капуш Е.В.	Капуш Е.В.	Тихонов а С.А.	Тихонов а С.А.	Кузнецов а Л.М.	Кузнецов а Л.М.	Кузнецов а Л.М.	Кузнецов а Л.М.
2	Литература	Ганжина И.М.	Андреева Т.Н.	Капуш Е.В.	Капуш Е.В.	Тихонов а С.А.	Тихонов а С.А.	Химочки на В.В.	Химочки на В.В.	Химочки на В.В.	Химочки на В.В.
3	Английский язык	Зимоглад С.В.	Зимоглад С.В.	Блохинова О.Ю.	Блохинова О.Ю.	Рыжкова А.Г.	Блохинова О.Ю.	Головина Е.В.	Головина Е.В.	Зимоглад С.В.	Бойцова Н.В.
4	Немецкий язык	Сафонов а А.В.	Рыжова М.В.								
5	Математика	Баданова А.В.	Баданова А.В.	Преснякова Е.В.	Баданова А.В.			Преснякова Е.В.	Голубева Е.Н.	Бойцова К.Н.	Бойцова К.Н.
	Алгебра					Гошева П.Б.	Гошева П.Б.				
	Геометрия					Гошева П.Б.	Гошева П.Б.				
6	Информатика и ИКТ					Егужокова Р.М.	Егужокова Р.М.	Егужокова Р.М.	Егужокова Р.М.	Калугина О.Н.	Егужокова Р.М.
7	География	Сухарева Н.М.	Сухарева Н.М.	Сальникова Е.А.	Сальникова Е.А.	Сальникова Е.А.	Сальникова Е.А.	Сальникова Е.А.	Сухарева Н.М.	Сухарева Н.М.	Сухарева Н.М.
8	Биология	Игнатеев Д.И.	Игнатеев Д.И.	Игнатеев Д.И.	Игнатеев Д.И.	Игнатеев Д.И.	Игнатеев Д.И.	Копкарева О.О.	Копкарева О.О.	Копкарева О.О.	Копкарева О.О.
9	Физика					Егужокова Р.М.	Егужокова Р.М.	Егужокова Р.М.	Егужокова Р.М.	Калугина О.Н.	Егужокова Р.М.
10	Химия							???	???	Баранова Н.В.	???
11	История	Петров К.А.	Петров К.А.	Петров К.А.	Петров К.А.	Петров К.А.	Петров К.А.	Лисицына О.И.	Лисицына О.И.	Иванова А.А.	Иванова А.А.
12	Искусство (МХК)							Иванова А.А.	Иванова А.А.	Иванова А.А.	Иванова А.А.
13	Обществознание			Чицова В.В.	Чицова В.В.	Виноградова Э.С.	Виноградова Э.С.	Лисицына О.И.	Лисицына О.И.	Иванова А.А.	Иванова А.А.
14	Музыка	Мачульская М.А.	Мачульская М.А.	Мачульская М.А.	Мачульская М.А.	Мачульская М.А.	Мачульская М.А.				
15	ИЗО	Креславская Т.А.	Креславская Т.А.	Креславская Т.А.	Креславская Т.А.	Креславская Т.А.	Креславская Т.А.				
16	Физкультура	Клунко В. В.	Клунко В. В.	Клунко В. В.	Клунко В. В.	Клунко В. В.	Клунко В. В.	Озеров П.П.	Озеров П.П.	Озеров П.П.	Озеров П.П.
17	Технология	Баканова Е.О.	Баканова Е.О.	Креславская Т.А.	Креславская Т.А.	Баканова Е.О.	Баканова Е.О.	Егужокова Р.М.	Егужокова Р.М.		
18	ОБЖ							Кузахметов Д.О.	Кузахметов Д.О.		
19	Математика (компонент ОУ)							!1	!1		
20	Путешествия в слова, электив)			Капуш Е.В.	Капуш Е.В.						
21	В мире чисел, электив					Гошева П.Б.	Гошева П.Б.				
22	Трудные вопросы орфографии, электив					Тихонов а С.А.	Тихонов а С.А.				
23	Тайный мир текста, электив							Кузнецов а Л.М.	Кузнецов а Л.М.		
24	Теоретические основы русского языка, электив									Кузнецов а Л.М.	Кузнецов а Л.М.
25	Трудные вопросы математики Б решение задач в формате ОГЭ									Бойцова	Бойцова

Приложение 2

время /класс	5а	5б	6а	6б	7а	7б	8а	8б	9а	9б	
П Н	09:00-09:45	Английский язык Зимоглад С.В. 205	Русский язык Андреева Т.Н. 206	Технология Креславская Т.А. 409	Русский язык Капуш Е.В. 212	В мире чисел, электив Гошева П.Б. 211	Литература Тихонова С.А. 310	Биология Копкарева О.О. 305	Русский язык Кузнецова Л.М. 406	Общественное Иванова А.А. 411	
	09:55-10:40	Математика Баданова А.В. 205	Музыка Мачульняк М.А. 206	Математика Преснякова Е.В. 208	Технология Креславская Т.А. 409	Русский язык Тихонова С.А. 211	Физика Егужкова Р.М. 213	Литература Химочкина В.В. 305	Математика Голубева Е.Н. 312	Русский язык Кузнецова Л.М. 406	
	10:05-11:35	Русский язык Ганжина И.М. 205	Немецкий язык Рыжова М.В. 206		Путешествие в слова, электив Капуш Е.В. 212	Русский язык Тихонова С.А. 211	Английский язык Блохина О.Ю. 310	География Сальникова Е.А. 305	Литература Химочкина В.В. 307	Английский язык Зимоглад С.В. 408	Математика Бойцова К.Н. 312
	12:05-12:50	География Сухарева Н.М. 205	История Петров К.А. 206	Русский язык Капуш Е.В. 208	Обществознание Чижилова К.А. 212		Алгебра Гошева П.Б. 310	Технология Егужокова Р.М. 409	Математика Бойцова К.Н. 406	Теоритические основы русского языка, электив Кузнецова Л.М. 309	

	13:00-13:45	Технология Баканова Е.О. 409	Математика Баданова А.В. 206	Литература Капуш Е.В. 208		История Петров К.А. 211	Информатика и ИКТ Егужкова Р.М. 49 корп 2	Математика Преснякова Е.В. 305	Химия Баранов Н.В. 407		Английский язык Бойцова Н.В. 408
	13:55-14:40	Физкультура Клунков В.В. Корпус Б тр. зал	Английский язык Блохина О.Ю. Зимогляд С.В. 206	Английский язык Блохина О.Ю. 208	Математика Баданова А.В. 212	Информатика и ИКТ Егужкова Р.М. 204	Технология Баканова Е.О. 409	Математика Преснякова Е.В. 305	История Лисицына О.И. 411	Физкультура Озеров П.П. Корпус №3 спорт зал	Биология Копкарева О.О. 49 корп 2
	14:50-15:35			Физкультура Клунков В.В. Корпус Б тр. зал		ИЗО Креславская Т.А. 211	География Сальникова Е.А. 310	Английский язык Головина Е.В. 305	Русский язык Кузнецова Л.М. 407	Обществознание Иванова А.А. 411	Физика Егужкова Р.М. 213
	15:45-16:30							Физкультура Озеров П.П.302	Физкультура Озеров П.П.302		
ВТ	09:00-09:45	История Петров К.А. 205	Биология Игнатьева Д.И. 206	Английский язык Блохина О.Ю. 208	Русский язык Капуш Е.В. 212	Русский язык Тихонова С.А. 211	Физика Егужкова Р.М. 213		Русский язык Кузнецова Л.М. 312		

09:55-10:40	Литература Ганжина И.М. 205	Математика Баданова А.В. 206	Математика Преснякова Е.В. 208	Русский язык Капуш Е.В. 212	Английский язык Рыжкова А.Г. 211	Русский язык Тихонова С.А. 310	Химия Баранов Н.В. 305	ОБЖ Кузахметов Д.О. 219	Английский язык Зимоглаз С.В. 49 корп 2	Искусство (МХК) Иванова А.А. 406
10:50-11:35		Русский язык Андреева Т.Н. 206	История Петров К.А. 208	Математика Баданова А.В. 212	Геометрия Гошева П.Б. 211			Литература Химочкина В.В. 309	Математика Бойцова К.Н. 312	История Иванова А.А. 406
12:05-12:50	Музыка Мачульская М.А. 205	Русский язык Андреева Т.Н. 206	Литература Капуш Е.В. 208	Английский язык Блохина О.Ю. 212	Литература Тихонова С.А. 211	История Петров К.А. 310	Тайный мир текста, электив Кузнецова Л.М. 305	Технология Егужкова Р.М. 409	Математика Бойцова К.Н. 408	Литература Химочкина В.В. 411
13:00-13:45	Английский язык Зимоглаз С.В. 205	Физкультура Клункова В.В. Корпус №3 спорт зал	Русский язык Капуш Е.В. 208	ИЗО Креславская Т.А. 409	История Петров К.А. 211	Литература Тихонова С.А. 310	Русский язык Кузнецова Л.М. 305	Физика Егужкова Р.М. 219	Искусство (МХК) Иванова А.А. 406	Информатика и ИКТ Гошева П.Б. 309

	13:55-14:40	Математика Баданова А.В. 205	Технология Баканова Е.О. 409	Русский язык Капуш Е.В. 208	Физкультура Клунков В.В. Корпус №3 спорт зал		Музыка Мачульская М.А. 310	Математика Преснякова Е.В. 305	География Сухарева Н.М. 49 корпус 2	Литература Химочкина В.В. 406	Физкультура Озеров П.П. Корпус Б тр. зал
	14:50-15:35				Физкультура Клунков В.В. Корпус Б тр. зал	ИЗО Креславская Т.А. 310	Математика Преснякова Е.В. 305	Химия Баранов Н.В. 406	Биология Копкарева О.О. 49 корпус 2	Английский язык Бойцова Н.В. 312	
	15:44-16:30					Физкультура Клунков В.В. 302	Физкультура Озеров П.П. Корпус Б тр. зал				
	19:00-19:45	Математика Баданова А.В. 205	Литература Андреева Т.Н. 206	ИЗО Креславская Т.А. 208	География Сальникова Е.А. 212	Физика Егужокова Р.М. 213	Биология Игнатьева Д.И. 310	Русский язык Кузнецова Л.М. 305		История Иванова А.А. 49 корпус 2	Математика Бойцова К.Н. 307
СР	19:55-20:40	Немецкий язык Сафонова А.В. 205	Литература Андреева Т.Н. 206	Математика Преснякова Е.В. 208	Музыка Мачульская М.А. 309	Геометрия Гошева П.Б. 211	Обществознание Лисицына О.И. 305	Английский язык Головина Е.В. 407		Химия Баранов Н.В. 307	История Иванова А.А. 411

1 0: 0- 1: 3 5	История Петров К.А. 205	Технология Бака нов а Е.О. 409	Общество знание Чи жо ва К.А. 208	Русский язык Кау пу ш Е.В. 212	Информатика и ИКТ Егужо ва Р.М. 301 корп Б	Алгебра Гошев а П.Б. 310	Английский язык Голови на Е.В. 305	История Лисиц ына О.И. 408	Литература Химоч кина В.В. 309	Химия Барано в Н.В. 411
1 2: 0 5- 1: 2: 5 0	Биология Игнать ева Д.И. 205	Немец кий язык Рыж ова М.В. 206	Литература Кау пу ш Е.В. 208	История Петро в К.А. 212	Технология Бакано ва Е.О. 409	Русский язык Тихоно ва С.А. 310	Математика Пресня кова Е.В. 305	Физика Егужо ва Р.М. 219	Английский язык Зимогл яд С.В. 312	География Сухаре ва Н.М. 49 корп 2
1 3: 0 0- 1: 3: 4 5	Русский язык Ганжи на И.М. 205	Математика Бад ано ва А.В. 206	Технология Кресл авска я Т.А. 409	Литература Кау пу ш Е.В. 212	Биология Игнать ева Д.И. 211	Информатика и ИКТ Егужо ва Р.М. 312		Математика Голубе ва Е.Н. 309	Физика Калуги на О.Н. 213	География Сухаре ва Н.М. 49 корп 2
1 3: 5 5- 1: 4: 4 0	Русский язык Ганжи на И.М. 205		Русский язык Кау пу ш Е.В. 208	Физкультура Клунк о В.В. Корпу с Б тр. зал	География Сальни кова Е.А. 211	В мире чисел, электи в Гошев а П.Б. 310	ОБЖ Кузахме тов Д.О. 219	Математика Голубе ва Е.Н. 408	Математика Бойцов а К.Н. 406	Литература Химоч кина В.В. 309

Т Ч	14:50-15:35	Физкультура Клунко В.В. Корпус №3 спорт зал	Физкультура Клунко В.В. Корпус №3 спорт зал	Английский язык Рыжкова А.Г. 211	Английский язык Блохина О.Ю. 310			Литература Химочкина В.В. 305	Физкультура Озеров П.П. 302	Трудные вопросы математики, решение задач в форме ОГЭ, электрив Бойцова К.Н. 307	Информатика и ИКТ Гошева П.Б. 406
	15:45-16:30					Физкультура Клунко В.В. Корпус №3 спорт зал	Физкультура Клунко В.В. Корпус №3 спорт зал		Физкультура Озеров П.П. 302		
	09:00-09:45	Литература Ганжина И.М. 205	География Сухарева Н.М. 206	История Петров К.А. 208	Математика Баданова А.В. 212	Русский язык Тихонова С.А. 211	Геометрия Гошева П.Б. 310	История Лисицына О.И. 305		Биология Копкарева О.О. 406	Математика Бойцова К.Н. 312
09:55-10:40		Английский язык Преснякова Е.В. Зимогляд С.В. 206	Математика Преснякова Е.В. 208	Математика Баданова А.В. 212	Алгебра Гошева П.Б. 211		Искусство (МХК) Иванова А.А. 305	Биология Копкарева О.О. 407	Информатика и ИКТ Калугина О.Н. 411	Математика Бойцова К.Н. 408	
10:55-11:15	Математика Баданова А.В.	ИЗО Креславская Т.А.	Русский язык Капуш Е.В.	Английский язык Блохина	Алгебра Гошева П.Б.	Русский язык Тихонова С.А.	Биология Копкарева О.О.	Обществознание Лисицына	География Сухарева Н.М.	Русский язык Кузнецова Л.М.	

3 5	205	206	208	О.Ю. 212	211	310	305	О.И. 312	407	307
1 2: 0 5- 1 2: 5 0	Немец кий язык Саффо ва А.В. 205	Исто рия Пет ров К.А. 206		Литер атура Капу ш Е.В. 212	Музык а Мачул ьяная М.А. 211	Трудн ые вопрос ы орфогра фии, электи в Тихоно ва С.А. 310	Инфор матика и ИКТ Егужоко ва Р.М. 312	Тайны й мир текста, электи в Кузнец ова Л.М. 406	Химия Барано в Н.В. 411	Англий ский язык Бойцов а Н.В. 407
1 3: 0 0- 1 3: 4 5	ИЗО Кресла вская Т.А. 409	Русск ий язык Игнат ьева Д.И. Андре ева Т.Н. 206	Биоло гия Игнат ьева Д.И. 208	Русск ий язык Капу ш Е.В. 212	Трудн ые вопрос ы орфогра фии, электи в Тихоно ва С.А. 211	Общес твозна ние Виногра дова Э.С. 310	Матема тика Пресня кова Е.В. 305	Геогра фия Сухаре ва Н.М. 411		Англий ский язык Бойцов а Н.В. 407
1 3: 5 5- 1 4: 4 0	Русски й язык Ганжи на И.М. 205	Матема тика Бадано ва А.В. 206	Англи йский язык Блохи на О.Ю. 208	Биоло гия Игнат ьева Д.И. 212		Истори я Петров К.А. 310	Инфор матика и ИКТ Егужоко ва Р.М. 406	Матем атика Бойцов а К.Н. 309	Химия Барано в Н.В. 408	
1 4: 5 0- 1 5: 3 5			Физкул ьтура Клунк о В.В. Корпус Б тр. зал	Физкул ьтура Клунк о В.В. Корпус Б тр. зал		Геогра фия Сальни кова Е.А. 310	Физика Егужоко ва Р.М. 213	Матем атика Голубе ва Е.Н. 406	Физкул ьтура Озеров П.П. Корпус №3 спорт зал	Биолог ия Копка рева О.О. 49 корп 2

П Т	1 5: 4 5- 1 6: 3 0						Физкул ьтураК лунко В.В.Ко рпус №3 спорт зал			Физкул ьтураО зеров П.П.Ко рпус Б тр. зал	Физкул ьтураО зеров П.П.Ко рпус Б тр. зал
	0 9: 0 0- 0 9: 4 5	Техно логия Бакано ва Е.О. 409	Лит ерат ура Анд реев а Т.Н. 206	Геогр афия Сальн икова Е.А. 208	Литер атура Каупу ш Е.В. 212		Англий ский язык Блохин а О.Ю. 310	История Лисицы на О.И. 305	Матем атика Голубе ва Е.Н. 307	Русски й язык Кузнец ова Л.М. 408	Англий ский язык Бойцов а Н.В. 312
	0 9: 5 5- 1 0: 4 0	Литера тура Ганжи на И.М. 205	Русск ий язык Анд реев а Т.Н. 206	Музы ка Мачу льная М.А. 208	Русск ий язык Каупу ш Е.В. 212	Геогр афия Сальни кова Е.А. 211		Физика Егужоко ва Р.М. 213	Матем атика Голубе ва Е.Н. 407		Литера тура Химоч кина В.В. 408
	1 0: 5 0- 1 1: 3 5		Англ ийск ий язык Зим огля д С.В. 206	Мате матик а Пресн якова Е.В. 208	Техно логия Кресл авска я Т.А. 409	Литера тура Тихоно ва С.А. 211	Геомет рия Гошев а П.Б. 310	Химия Барано в Н.В. 305	Англий ский язык Голови на Е.В. 411	Литера тура Химоч кина В.В. 312	Физика Егужок ова Р.М. 213

1 2: 0 5- 1 2: 5 0	Русский язык Ганжина И.М. 205	Математика Баданова А.В. 206	Русский язык Капуш Е.В. 208	Английский язык Блохина О.Ю. 212	Обществознание Виноградова Э.С. 211	Алгебра Гошева П.Б. 310	Русский язык Кузнецова Л.М. 305	Английский язык Головина Е.В. 406	География Сухарева Н.М. 411	Трудные вопросы математики, решение задач в формате ОГЭ, электрон Бойцова К.Н. 307
1 3: 0 0- 1 3: 4 5	Английский язык Зимоглав С.В. 205	Физкультура Клунков В.В. Корпус №3 спорт зал	Путешествия в слова, электрон Капуш Е.В. 208	Математика Баданова А.В. 212	Физика Егужкова Р.М. 213	Русский язык Тихонова С.А. 310	Английский язык Головина Е.В. 305	Русский язык Кузнецова Л.М. 312	История Иванова А.А. 407	Математика Бойцова К.Н. 411
1 3: 5 5- 1 4: 4 0	Математика Баданова А.В. 205		Физкультура Клунков В.В. 302	История Петров К.А. 212	Английский язык Рыжкова А.Г. 211			Биология Копкарева О.О. 307	Теоритические основы русского языка, электрон Кузнецова Л.М. 312	

1 4: 5 0- 1 5: 3 5	Физкул ьтураК лунко В.В.302				Алгебр аГоше ва П.Б.21 1		Географ ияСаль никова Е.А.305	Искус ство (МХК)И ванова А.А.41 1	Физика Калуги на О.Н.21 3	Физкул ьтураО зеров П.П.Ко рпус Б тр. зал
1 5: 4 5- 1 6: 3 0					Физкул ьтура Клунко В.В. Корпус Б тр. зал		Физкул ьтура Озеров П.П. 302			

Приложение 3

#mainmenu.ui

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>MainMenu</class>
  <widget class="QMainWindow" name="MainMenu">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>640</width>
        <height>480</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>АИС Университет</string>
    </property>
    <property name="styleSheet">
      <string notr="true">QMainWindow {
border: none;
background: white;
}</string>
    </property>
    <widget class="QWidget" name="centralWidget">
      <layout class="QHBoxLayout" name="horizontalLayout_3">
        <item>
          <layout class="QHBoxLayout" name="horizontalLayout_2">
            <item>
              <spacer name="horizontalSpacer">
                <property name="orientation">
                  <enum>Qt::Horizontal</enum>
                </property>
                <property name="sizeHint" stdset="0">
                  <size>
                    <width>40</width>
                    <height>20</height>
                  </size>
                </property>
              </spacer>
            </item>
          </layout>
        </item>
      </layout>
    </widget>
  </widget>
</ui>
```



```

    </property>
  </spacer>
</item>
<item>
  <layout class="QVBoxLayout" name="verticalLayout_2">
    <item>
      <spacer name="verticalSpacer">
        <property name="orientation">
          <enum>Qt::Vertical</enum>
        </property>
        <property name="sizeHint" stdset="0">
          <size>
            <width>20</width>
            <height>40</height>
          </size>
        </property>
      </spacer>
    </item>
    <item>
      <widget class="QGroupBox" name="groupBox">
        <property name="title">
          <string/>
        </property>
        <layout class="QHBoxLayout" name="horizontalLayout">
          <item>
            <layout class="QVBoxLayout" name="verticalLayout">
              <item>
                <widget class="QPushButton" name="lookDataBase">
                  <property name="styleSheet">
                    <string notr="true">QPushButton {
color: #333;
border: 2px solid #555;
border-radius: 11px;
padding: 5px;
background: radialgradient(cx: 0.3, cy: -0.4,
fx: 0.3, fy: -0.4,
radius: 1.35, stop: 0 #fff, stop: 1 #888);
min-width: 80px;
text-align: left;
}

QPushButton: hover {
background: radialgradient(cx: 0.3, cy: -0.4,
fx: 0.3, fy: -0.4,
radius: 1.35, stop: 0 #fff, stop: 1 #bbb);
}

QPushButton: pressed {
background: radialgradient(cx: 0.4, cy: -0.1,
fx: 0.4, fy: -0.1,
radius: 1.35, stop: 0 #fff, stop: 1 #ddd);
}</string>
                  </property>
                  <property name="text">
                    <string>Просмотр данных</string>
                  </property>
                  <property name="icon">
                    <iconset resource="image.qrc">
                      <normaloff>:/icon/glyphicons-115-
list.png</normaloff>:/icon/glyphicons-115-list.png</iconset>
                  </property>
                  <property name="iconSize">

```

```

        <size>
            <width>20</width>
            <height>20</height>
        </size>
    </property>
</widget>
</item>
<item>
    <widget class="QPushButton" name="editDataBase">
        <property name="styleSheet">
            <string notr="true">QPushButton {
color: #333;
border: 2px solid #555;
border-radius: 11px;
padding: 5px;
background: radialgradient(cx: 0.3, cy: -0.4,
fx: 0.3, fy: -0.4,
radius: 1.35, stop: 0 #fff, stop: 1 #888);
min-width: 80px;
text-align: left;
}

QPushButton:hover {
background: radialgradient(cx: 0.3, cy: -0.4,
fx: 0.3, fy: -0.4,
radius: 1.35, stop: 0 #fff, stop: 1 #bbb);
}

QPushButton:pressed {
background: radialgradient(cx: 0.4, cy: -0.1,
fx: 0.4, fy: -0.1,
radius: 1.35, stop: 0 #fff, stop: 1 #ddd);
}</string>
        </property>
        <property name="text">
            <string>Создание расписания</string>
        </property>
        <property name="icon">
            <iconset resource="image.qrc">
                <normaloff>:/icon/glyphicons-151-
edit.png</normaloff>:/icon/glyphicons-151-edit.png</iconset>
            </property>
            <property name="iconSize">
                <size>
                    <width>20</width>
                    <height>20</height>
                </size>
            </property>
        </widget>
    </item>
    <item>
        <widget class="QPushButton" name="settingMenu">
            <property name="styleSheet">
                <string notr="true">QPushButton {
color: #333;
border: 2px solid #555;
border-radius: 11px;
padding: 5px;
background: radialgradient(cx: 0.3, cy: -0.4,
fx: 0.3, fy: -0.4,
radius: 1.35, stop: 0 #fff, stop: 1 #888);
min-width: 80px;

```

```

text-align: left;
}

QPushButton:hover {
background: radialgradient(cx: 0.3, cy: -0.4,
fx: 0.3, fy: -0.4,
radius: 1.35, stop: 0 #fff, stop: 1 #bbb);
}

QPushButton:pressed {
background: radialgradient(cx: 0.4, cy: -0.1,
fx: 0.4, fy: -0.1,
radius: 1.35, stop: 0 #fff, stop: 1 #ddd);
}</string>
    </property>
    <property name="text">
    <string>Настройки</string>
    </property>
    <property name="icon">
    <iconset resource="image.qrc">
    <normaloff>:/icon/glyphicons-281-
settings.png</normaloff>:/icon/glyphicons-281-settings.png</iconset>
    </property>
    <property name="iconSize">
    <size>
    <width>20</width>
    <height>20</height>
    </size>
    </property>
</widget>
</item>
<item>
    <widget class="QPushButton" name="exitMainMenu">
    <property name="styleSheet">
    <string notr="true">QPushButton {
color: #333;
border: 2px solid #555;
border-radius: 11px;
padding: 5px;
background: radialgradient(cx: 0.3, cy: -0.4,
fx: 0.3, fy: -0.4,
radius: 1.35, stop: 0 #fff, stop: 1 #888);
min-width: 80px;
text-align: left;
}

QPushButton:hover {
background: radialgradient(cx: 0.3, cy: -0.4,
fx: 0.3, fy: -0.4,
radius: 1.35, stop: 0 #fff, stop: 1 #bbb);
}

QPushButton:pressed {
background: radialgradient(cx: 0.4, cy: -0.1,
fx: 0.4, fy: -0.1,
radius: 1.35, stop: 0 #fff, stop: 1 #ddd);
}</string>
    </property>
    <property name="text">
    <string>Выход</string>
    </property>
    <property name="icon">

```

```

        <iconset resource="image.qrc">
        <normaloff>:/icon/glyphicons-389-
exit.png</normaloff>:/icon/glyphicons-389-exit.png</iconset>
        </property>
        <property name="iconSize">
        <size>
        <width>20</width>
        <height>20</height>
        </size>
        </property>
        </widget>
        </item>
        </layout>
        </item>
        </layout>
        </widget>
        </item>
        <item>
        <spacer name="verticalSpacer_2">
        <property name="orientation">
        <enum>Qt::Vertical</enum>
        </property>
        <property name="sizeHint" stdset="0">
        <size>
        <width>20</width>
        <height>40</height>
        </size>
        </property>
        </spacer>
        </item>
        </layout>
        </item>
        <item>
        <spacer name="horizontalSpacer_2">
        <property name="orientation">
        <enum>Qt::Horizontal</enum>
        </property>
        <property name="sizeHint" stdset="0">
        <size>
        <width>40</width>
        <height>20</height>
        </size>
        </property>
        </spacer>
        </item>
        </layout>
        </item>
        </layout>
        </widget>
        </widget>
        <layoutdefault spacing="6" margin="11"/>
        <resources>
        <include location="image.qrc"/>
        </resources>
        <connections/>
</ui>

```

```
#mainmenu.h
```

```
#ifndef MAINMENU_H
#define MAINMENU_H
```

```

#include <QMainWindow>

namespace Ui {
class MainMenu;
}

class MainMenu : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainMenu(QWidget *parent = 0);
    ~MainMenu();

private slots:
    void slotToMainMenu();

    void on_exitMainMenu_clicked();

    void on_settingMenu_clicked();

    void on_lookDataBase_clicked();

    void on_editDataBase_clicked();

private:
    Ui::MainMenu *ui;
};

#endif // MAINMENU_H

```

#mainmenu.cpp

```

#include "mainmenu.h"
#include "ui_mainmenu.h"
#include "settingmenu.h"
#include "lookdatabase.h"
#include "createschedule.h"
#include <time.h>

MainMenu::MainMenu(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainMenu)
{
    srand(time(NULL));
    ui->setupUi(this);
}

MainMenu::~MainMenu()
{
    delete ui;
}

void MainMenu::on_exitMainMenu_clicked()
{
    this->close();
}

void MainMenu::on_settingMenu_clicked()
{
    this->setVisible(false);
}

```

```

        SettingMenu *settingMenu = new SettingMenu;
        settingMenu->setGeometry(this->geometry());
        connect(settingMenu, SIGNAL(signalSettingToMainMenu()), this,
SLOT(slotToMainMenu()));
        settingMenu->show();
        settingMenu->setFocus();
    }

void MainMenu::slotToMainMenu()
{
    this->setVisible(true);
    this->setFocus();
}

void MainMenu::on_lookDataBase_clicked()
{
    this->setVisible(false);
    LookDataBase *lookBD = new LookDataBase;
    lookBD->setGeometry(this->geometry());
    connect(lookBD, SIGNAL(signalSettingToMainMenu()), this,
SLOT(slotToMainMenu()));
    lookBD->show();
    lookBD->setFocus();
}

void MainMenu::on_editDataBase_clicked()
{
    this->setVisible(false);
    CreateSchedule *editDB = new CreateSchedule;
    connect(editDB, SIGNAL(signalSettingToMainMenu()), this,
SLOT(slotToMainMenu()));
    editDB->setGeometry(this->geometry());
    editDB->show();
    editDB->setFocus();
}

```

#lookdatabase.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>LookDataBase</class>
    <widget class="QMainWindow" name="LookDataBase">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>1045</width>
                <height>766</height>
            </rect>
        </property>
        <property name="windowTitle">
            <string>АИС Университет</string>
        </property>
        <widget class="QWidget" name="centralwidget">
            <layout class="QHBoxLayout" name="horizontalLayout_4">
                <item>
                    <layout class="QVBoxLayout" name="verticalLayout_4">
                        <item>
                            <layout class="QHBoxLayout" name="horizontalLayout">
                                <item>
                                    <widget class="QPushButton" name="exportCSV">
                                        <property name="minimumSize">

```

```

        <size>
            <width>40</width>
            <height>40</height>
        </size>
    </property>
    <property name="toolTip">
        <string>Экспорт в CSV</string>
    </property>
    <property name="text">
        <string/>
    </property>
    <property name="icon">
        <iconset resource="image.qrc">
            <normaloff>:/icon/glyphicons-360-file-
export.png</normaloff>:/icon/glyphicons-360-file-export.png</iconset>
        </property>
    <property name="iconSize">
        <size>
            <width>30</width>
            <height>30</height>
        </size>
    </property>
</widget>
</item>
<item>
    <widget class="QPushButton" name="save">
        <property name="toolTip">
            <string>Сохранить</string>
        </property>
        <property name="text">
            <string/>
        </property>
        <property name="icon">
            <iconset resource="image.qrc">
                <normaloff>:/icon/glyphicons-445-floppy-
saved.png</normaloff>:/icon/glyphicons-445-floppy-saved.png</iconset>
            </property>
        <property name="iconSize">
            <size>
                <width>30</width>
                <height>30</height>
            </size>
        </property>
    </widget>
</item>
<item>
    <widget class="QPushButton" name="refresh">
        <property name="minimumSize">
            <size>
                <width>40</width>
                <height>40</height>
            </size>
        </property>
        <property name="toolTip">
            <string>Обновить</string>
        </property>
        <property name="text">
            <string/>
        </property>
        <property name="icon">
            <iconset resource="image.qrc">

```

```

        <normaloff>:/icon/glyphicons-82-
refresh.png</normaloff>:/icon/glyphicons-82-refresh.png</iconset>
    </property>
    <property name="iconSize">
        <size>
            <width>30</width>
            <height>30</height>
        </size>
    </property>
</widget>
</item>
<item>
<widget class="QPushButton" name="add">
    <property name="minimumSize">
        <size>
            <width>40</width>
            <height>40</height>
        </size>
    </property>
    <property name="toolTip">
        <string>Добавить запись</string>
    </property>
    <property name="text">
        <string/>
    </property>
    <property name="icon">
        <iconset resource="image.qrc">
            <normaloff>:/icon/glyphicons-7-user-
add.png</normaloff>:/icon/glyphicons-7-user-add.png</iconset>
        </property>
        <property name="iconSize">
            <size>
                <width>30</width>
                <height>30</height>
            </size>
        </property>
    </widget>
</item>
<item>
<widget class="QPushButton" name="delete_2">
    <property name="minimumSize">
        <size>
            <width>40</width>
            <height>40</height>
        </size>
    </property>
    <property name="toolTip">
        <string>Удалить выделенную запись</string>
    </property>
    <property name="text">
        <string/>
    </property>
    <property name="icon">
        <iconset resource="image.qrc">
            <normaloff>:/icon/glyphicons-257-
delete.png</normaloff>:/icon/glyphicons-257-delete.png</iconset>
        </property>
        <property name="iconSize">
            <size>
                <width>30</width>
                <height>30</height>
            </size>
        </property>
    </widget>
</item>

```



```

    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="filter">
    <property name="minimumSize">
      <size>
        <width>40</width>
        <height>40</height>
      </size>
    </property>
    <property name="toolTip">
      <string>Фильтрация данных</string>
    </property>
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="image.qrc">
        <normaloff>:/icon/glyphicons-321-
filter.png</normaloff>:/icon/glyphicons-321-filter.png</iconset>
      </property>
    <property name="iconSize">
      <size>
        <width>30</width>
        <height>30</height>
      </size>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="deleteFilter">
    <property name="minimumSize">
      <size>
        <width>40</width>
        <height>40</height>
      </size>
    </property>
    <property name="toolTip">
      <string>Сбросить фильтрацию</string>
    </property>
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="image.qrc">
        <normaloff>:/icon/glyphicons-789-filter-
remove.png</normaloff>:/icon/glyphicons-789-filter-remove.png</iconset>
      </property>
    <property name="iconSize">
      <size>
        <width>30</width>
        <height>30</height>
      </size>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="goMainMenu">
    <property name="minimumSize">
      <size>
        <width>40</width>

```

```

        <height>40</height>
    </size>
</property>
<property name="toolTip">
    <string>Главное меню</string>
</property>
<property name="text">
    <string/>
</property>
<property name="icon">
    <iconset resource="image.qrc">
        <normaloff>:/icon/glyphicons-517-menu-
hamburger.png</normaloff>:/icon/glyphicons-517-menu-hamburger.png</iconset>
    </property>
<property name="iconSize">
    <size>
        <width>30</width>
        <height>30</height>
    </size>
</property>
</widget>
</item>
<item>
    <spacer name="horizontalSpacer">
        <property name="orientation">
            <enum>Qt::Horizontal</enum>
        </property>
        <property name="sizeHint" stdset="0">
            <size>
                <width>40</width>
                <height>20</height>
            </size>
        </property>
    </spacer>
</item>
</layout>
</item>
<item>
    <widget class="QSplitter" name="splitter">
        <property name="orientation">
            <enum>Qt::Horizontal</enum>
        </property>
        <widget class="QGroupBox" name="groupBox">
            <property name="title">
                <string/>
            </property>
            <layout class="QHBoxLayout" name="horizontalLayout_3">
                <item>
                    <layout class="QVBoxLayout" name="verticalLayout_2">
                        <item>
                            <widget class="QListView" name="listView">
                                <property name="sizePolicy">
                                    <sizepolicy hsize="Maximum" vsizetype="Expanding">
                                        <horstretch>0</horstretch>
                                        <verstretch>0</verstretch>
                                    </sizepolicy>
                                </property>
                                <property name="maximumSize">
                                    <size>
                                        <width>200</width>
                                        <height>16777215</height>
                                    </size>
                                </property>
                            </widget>
                        </item>
                    </layout>
                </item>
            </layout>
        </widget>
    </item>
</item>

```



```

        </size>
    </property>
    <property name="text">
        <string>Свернуть</string>
    </property>
    <property name="icon">
        <iconset resource="icon.qrc">
            <normaloff>:/image/icon/glyphicons-52-eye-
open.png</normaloff>:/image/icon/glyphicons-52-eye-open.png</iconset>
        </property>
    </widget>
</item>
<item>
    <spacer name="horizontalSpacer_2">
        <property name="orientation">
            <enum>Qt::Horizontal</enum>
        </property>
        <property name="sizeHint" stdset="0">
            <size>
                <width>40</width>
                <height>20</height>
            </size>
        </property>
    </spacer>
</item>
</layout>
</item>
</layout>
</item>
</layout>
</item>
</layout>
</widget>
<widget class="QGroupBox" name="groupBox_2">
    <property name="title">
        <string/>
    </property>
    <layout class="QVBoxLayout" name="verticalLayout_3">
        <item>
            <widget class="QTabWidget" name="tabWidget">
                <property name="currentIndex">
                    <number>-1</number>
                </property>
            </widget>
        </item>
    </layout>
</widget>
</widget>
</item>
</layout>
</item>
</layout>
</widget>
</widget>
<resources>
    <include location="image.qrc"/>
    <include location="icon.qrc"/>
</resources>
<connections/>
</ui>

```

#lookdatabase.h

```
#ifndef LOOKDATABASE_H
#define LOOKDATABASE_H

#include <QMainWindow>
#include <QShortcut>
#include <QSqlDatabase>
#include <QSqlTableModel>

namespace Ui {
class LookDataBase;
}

class LookDataBase : public QMainWindow
{
    Q_OBJECT

public:
    explicit LookDataBase(QWidget *parent = 0);
    ~LookDataBase();

    void russianColumn(QString input);

    QStringList russianListView(QStringList input, int mode);

private slots:
    void on_comboBox_activated(const QString &arg1);

    void on_goMainMenu_clicked();

    void on_tableView_clicked(const QModelIndex &index);

    void on_save_clicked();

    void on_add_clicked();

    void on_delete_2_clicked();

    void on_listView_clicked(const QModelIndex &index);

    void on_exportCSV_clicked();

    void on_filter_clicked();

    void on_deleteFilter_clicked();

    void on_hideTableMenu_clicked();

    void on_tabWidget_tabCloseRequested(int index);

    void on_refresh_clicked();

private:
    Ui::LookDataBase *ui;
    QSqlDatabase db;
    QSqlTableModel *model;
    int rowForDelete;

    QShortcut * keyCtrlD;
    QShortcut * keyCtrlN;
    QShortcut * keyCtrlS;
    QShortcut * keyCtrlR;
```

```

signals:
    void signalSettingToMainMenu ();
};

#endif // LOOKDATABASE_H

```

lookdatabase.cpp

```

#include "View/facultylistapplet.h"
#include "View/TeacherView.h"
#include "lookdatabase.h"
#include "ui_lookdatabase.h"

#include <QDebug>
#include <QDir>
#include <QFile>
#include <QInputDialog>
#include <QMessageBox>
#include <QPrinter>
#include <QSettings>
#include <QSqlDatabase>
#include <QSqlError>
#include <QSqlQuery>
#include <QSqlRecord>
#include <QSqlTableModel>
#include <QStringListModel>
#include <QTextDocument>
#include <QTime>
#include "filterdialog.h"
#include <QHeaderView>
#include <View/LectureHallsView.h>
#include <View/SubjectTeacherView.h>
#include <View/SubjectView.h>
#include <View/GroupView.h>
#include <View/AlarmView.h>

LookDataBase::LookDataBase (QWidget *parent) :
    QMainWindow (parent),
    ui (new Ui::LookDataBase)
{
    ui->setupUi (this);

    rowForDelete = -1;

    QSettings settings ("GatesB", "MakeMeSchedule");

    this->db = QSqlDatabase::addDatabase ("QPSQL");
    db.setHostName (settings.value ("SETTING/ADDRESS_DB", "").toString ());
    db.setDatabaseName (settings.value ("SETTING/NAME_DB", "").toString ());
    db.setUserName (settings.value ("SETTING/USER_DB", "").toString ());
    db.setPassword (settings.value ("SETTING/PASSWORD_DB", "").toString ());
    db.setPort (settings.value ("SETTING/PORT_DB", "").toInt ());
    db.open ();

    QStringList listTables;
    listTables = db.tables (QSql::Tables);
    qDebug () << listTables;
    if (listTables.isEmpty ())
    {
        QMessageBox q;
        q.setText ("WARNING:\ndata base is empty");
    }
}

```

```

        q.exec();
    }
    else
    {
        //QButtonGroup bg;
        //bg.addButton(new QPushButton(listTables[0], ui->groupBox));
        //bg.setParent(ui->groupBox);
        //bg.setExclusive(false);

        //ui->comboBox->addItem(listTables);
        QStringListModel *qlm = new
QStringListModel(russianListView(listTables,1));
        ui->listView->setModel(qlm);
        ui->listView->show();
        //on_listView_clicked(QModelIndex());
        //ui->listView->activated(qlm->index(0,0));
        QSqlTableModel *model = new QSqlTableModel;
        model->setTable(listTables[0]);
        model->setEditStrategy(QSqlTableModel::OnManualSubmit);
        model->select();
        ui->tableView->setModel(model);

    }
    ui->tabWidget->setTabsClosable(true);
    //----Add
    this->keyCtrlN = new QShortcut(this);
    this->keyCtrlN->setKey(Qt::CTRL + Qt::Key_N);
    connect(this->keyCtrlN, SIGNAL(activated()), this,
SLOT(on_add_clicked()));
    //----Save
    this->keyCtrlS = new QShortcut(this);
    this->keyCtrlS->setKey(Qt::CTRL + Qt::Key_S);
    connect(this->keyCtrlS, SIGNAL(activated()), this,
SLOT(on_save_clicked()));
    //----Delete
    this->keyCtrlD = new QShortcut(this);
    this->keyCtrlD->setKey(Qt::CTRL + Qt::Key_D);
    connect(this->keyCtrlD, SIGNAL(activated()), this,
SLOT(on_delete_2_clicked()));
    //----Refresh
    this->keyCtrlR = new QShortcut(this);
    this->keyCtrlR->setKey(Qt::CTRL + Qt::Key_R);
    connect(this->keyCtrlR, SIGNAL(activated()), this,
SLOT(on_refresh_clicked()));
}

LookDataBase::~LookDataBase()
{
    db.close();
    delete ui;
}

void LookDataBase::on_comboBox_activated(const QString &arg1)
{
    /*!
     *Константа      Значение      Описание
     *QSqlTableModel::OnFieldChange  0      Все изменения модели будут
применены непосредственно к БД
     *QSqlTableModel::OnRowChange      1      Изменения строки будут
применены, когда пользователь выберет другую строку.
     *QSqlTableModel::OnManualSubmit  2      Все изменения будут кэшироваться
в модели пока не будут вызваны submitAll() или revertAll().
    */
}

```

```

    model = new QSqlTableModel;
    model->setTable(arg1);
    model->setSort(0, Qt::AscendingOrder);
    model->setEditStrategy(QSqlTableModel::OnManualSubmit);
    if(!model->select())
    {
        QMessageBox q;
        q.setText(model->lastError().text());
        q.exec();
    }
    russianColumn(arg1);
    ui->tableView->setModel(model);
    ui->tableView->resizeColumnsToContents();
}

void LookDataBase::on_goMainMenu_clicked()
{
    this->db.close();
    this->close();
    emit signalSettingToMainMenu();
}

void LookDataBase::on_tableView_clicked(const QModelIndex &index)
{
    //ui->tableView->selectRow(index.row());
    rowForDelete = index.row();
}

void LookDataBase::on_save_clicked()
{
    if(!this->model->submitAll())
    {
        QMessageBox q;
        q.setText(model->lastError().text());
        q.exec();
    }
    ui->tableView->resizeColumnsToContents();
    ui->tableView->repaint();
}

void LookDataBase::on_add_clicked()
{
    int newRow = model->rowCount();
    if(!model->insertRow(newRow))
    {
        QMessageBox q;
        q.setText(model->lastError().text());
        q.exec();
    }
    model->setData(model->index(newRow, 0), "");
    ui->tableView->resizeColumnsToContents();
    ui->tableView->repaint();
}

void LookDataBase::on_delete_2_clicked()
{
    if(rowForDelete != -1
        && !model->removeRow(this->rowForDelete))
    {
        QMessageBox q;
        q.setText(model->lastError().text());
        q.exec();
    }
}

```



```

else
{
    model->submitAll();
    ui->tableView->resizeColumnsToContents();
    ui->tableView->repaint();
}
}

QStringList LookDataBase::russianListView(QStringList input, int mode)
{
    QStringList output;
    if(mode == 1)
    {
        if(input.contains("udb_groups"))
            output.append("Группы");
        if(input.contains("udb_lecturehalls"))
            output.append("Аудитории");
        if(input.contains("udb_faculty"))
            output.append("Факультеты");
        if(input.contains("udb_subjectteacher"))
            output.append("Назначения");
        if(input.contains("udb_teachers"))
            output.append("Преподаватели");
        if(input.contains("udb_wishesteacher"))
            output.append("Ограничения преподавателей");
        if(input.contains("udbc_schedulealarm"))
            output.append("Расписание звонков");
        if(input.contains("res_schedule"))
            output.append("Расписание");
        if(input.contains("udb_subject"))
            output.append("Предметы");
        return output;
    }
    if(mode == 2)
    {
        if(input.contains("Группы"))
        {
            output.append("udb_groups");
            return output;
        }
        if(input.contains("Аудитории"))
        {
            output.append("udb_lecturehalls");
            return output;
        }
        if(input.contains("Факультеты"))
        {
            output.append("udb_faculty");
            return output;
        }
        if(input.contains("Назначения"))
        {
            output.append("udb_subjectteacher");
            return output;
        }
        if(input.contains("Преподаватели"))
        {
            output.append("udb_teachers");
            return output;
        }
        if(input.contains("Ограничения преподавателей"))
        {
            output.append("udb_wishesteacher");
            return output;
        }
    }
}

```

```

    }
    if(input.contains("Расписание звонков"))
    {
        output.append("udbc_schedulealarm");
        return output;
    }
    if(input.contains("Расписание"))
    {
        output.append("res_schedule");
        return output;
    }
    if(input.contains("Предметы"))
    {
        output.append("udb_subject");
        return output;
    }
}
return input;
}

void LookDataBase::russianColumn(QString input)
{
    if(input == "groups")
    {
        model->setHeaderData(0,Qt::Horizontal, "Идентификатор\nфакультета");
        model->setHeaderData(1,Qt::Horizontal, "Код\nфакультета");
        model->setHeaderData(2,Qt::Horizontal, "Идентификатор\nгруппы");
        model->setHeaderData(3,Qt::Horizontal, "Номер\nгруппы");
        model->setHeaderData(4,Qt::Horizontal, "Количество\nстудентов");
        return;
    }

    if(input == "lecture_halls")
    {
        model->setHeaderData(0,Qt::Horizontal, "Идентификатор\nфакультета");
        model->setHeaderData(1,Qt::Horizontal, "Код\nфакультета");
        model->setHeaderData(2,Qt::Horizontal, "Идентификатор\nаудитории");
        model->setHeaderData(3,Qt::Horizontal, "Номер\nаудитории");
        model->setHeaderData(4,Qt::Horizontal, "Тип\nаудитории");
        model->setHeaderData(5,Qt::Horizontal, "Вместимость\nаудитории");
        model->setHeaderData(6,Qt::Horizontal, "Приоритет\nфакультета");
        model->setHeaderData(7,Qt::Horizontal, "Имеет\nдоску");
        model->setHeaderData(8,Qt::Horizontal, "Имеет\nпроектор");
        model->setHeaderData(9,Qt::Horizontal, "Имеет\nрозетки");
        model->setHeaderData(10,Qt::Horizontal, "Имеет\nокна");
        return;
    }

    if(input == "faculty")

```

```

    {
        model->setHeaderData(0,Qt::Horizontal, "Идентификатор\nфакультета");
        model->setHeaderData(1,Qt::Horizontal, "Название\nфакультета");
        model->setHeaderData(2,Qt::Horizontal, "Код\nфакультета");
        model->setHeaderData(3,Qt::Horizontal, "Декан\nфакультета");
        model->setHeaderData(4,Qt::Horizontal, "Адрес\nфакультета");
        return;
    }
}

void LookDataBase::on_listView_clicked(const QModelIndex &index)
{
    QString selectTable = index.data().toString();
    /*!
    *Константа      Значение      Описание
    *QSqlTableModel::OnFieldChange  0      Все изменения модели будут
применены непосредственно к БД
    *QSqlTableModel::OnRowChange    1      Изменения строки будут
применены, когда пользователь выберет другую строку.
    *QSqlTableModel::OnManualSubmit  2      Все изменения будут кэшироваться
в модели пока не будут вызваны submitAll() или revertAll().
    */
    model = new QSqlTableModel;
    QStringList translate;
    if (index.isValid())
        translate = russianListView(QStringList(index.data().toString()), 2);
    //else
    //    translate.append("groups");
    model->setTable(translate[0]);
    model->setSort(0,Qt::AscendingOrder);
    model->setEditStrategy(QSqlTableModel::OnManualSubmit);
    if(!model->select())
    {
        QMessageBox q;
        q.setText(model->lastError().text());
        q.exec();
    }
    //russianColumn(translate[0]);
    ui->tableView->setModel(model);
    ui->tableView->resizeColumnsToContents();

    int posExistTab;
    int pos = -1;
    for (posExistTab = 0; posExistTab < ui->tabWidget->count();
posExistTab++)
    {
        QString nameTab = ui->tabWidget->tabText(posExistTab);
        if (nameTab.compare("Факультеты") == 0 &&
selectTable.compare("Факультеты") == 0)
            ui->tabWidget->removeTab(posExistTab);
        if (nameTab.compare("Преподаватели") == 0 &&
selectTable.compare("Преподаватели") == 0)
            ui->tabWidget->removeTab(posExistTab);
        if (nameTab.compare("Предметы") == 0 &&
selectTable.compare("Предметы") == 0)
            ui->tabWidget->removeTab(posExistTab);
        if (nameTab.compare("Аудитории") == 0 &&
selectTable.compare("Аудитории") == 0)
            ui->tabWidget->removeTab(posExistTab);
    }
}

```

```

        if (nameTab.compare("Назначения") == 0 &&
selectTable.compare("Назначения") == 0)
            ui->tabWidget->removeTab(posExistTab);
        if (nameTab.compare("Группы") == 0 && selectTable.compare("Группы")
== 0)
            ui->tabWidget->removeTab(posExistTab);
        if (nameTab.compare("Расписание звонков") == 0 &&
selectTable.compare("Расписание звонков") == 0)
            ui->tabWidget->removeTab(posExistTab);
    }
    if (translate[0] == "udb_faculty")
    {
        FacultyListApplet *tabView = new FacultyListApplet();
        tabView->SetTableViewModel(model);
        pos = ui->tabWidget->addTab(tabView, "Факультеты");
    }
    if (translate[0] == "udb_teachers")
    {
        TeacherView *tabView = new TeacherView();
        tabView->SetTableViewModel(model);
        pos = ui->tabWidget->addTab(tabView, "Преподаватели");
    }
    if (translate[0] == "udb_subject")
    {
        SubjectView *tabView = new SubjectView();
        tabView->SetTableViewModel(model);
        pos = ui->tabWidget->addTab(tabView, "Предметы");
    }
    if (translate[0] == "udb_lecturehalls")
    {
        LectureHallsView *tabView = new LectureHallsView();
        tabView->SetTableViewModel(model);
        pos = ui->tabWidget->addTab(tabView, "Аудитории");
    }
    if (translate[0] == "udb_subjectteacher")
    {
        SubjectTeacherView *tabView = new SubjectTeacherView();
        tabView->SetTableViewModel(model);
        pos = ui->tabWidget->addTab(tabView, "Назначения");
    }
    if (translate[0] == "udb_groups")
    {
        GroupView *tabView = new GroupView();
        tabView->SetTableViewModel(model);
        pos = ui->tabWidget->addTab(tabView, "Группы");
    }
    if (translate[0] == "udbc_schedulealarm")
    {
        AlarmView *tabView = new AlarmView();
        tabView->SetTableViewModel(model);
        pos = ui->tabWidget->addTab(tabView, "Расписание звонков");
    }
    if (pos >= 0)
    {
        ui->tabWidget->repaint();
        ui->tabWidget->setCurrentIndex(pos);
        ui->tabWidget->show();
    }
}

void LookDataBase::on_exportCSV_clicked()
{
    QDateTime dt;

```

```

    QDir dir;
    QFile csvFile(dir.currentPath() + "/" + model->tableName() + "_" +
dt.currentDateTime().toString("dd-MM-yyyy_hh-mm") + ".csv");
    // Открываем, или создаём файл, если он не существует
    if(csvFile.open(QIODevice::WriteOnly))
    {
        // Создаём текстовый поток, в который будем писать данные
        QTextStream textStream(&csvFile);
        QStringList stringList; // Вспомогательный объект QStringList,
который сформирует строку
        // Проходимся по всем строкам ...
        for(int row = 0; row < model->rowCount(); row++)
        {
            stringList.clear(); // ... каждый раз очищая stringList
            /* Если обратить внимание в заголовочный файл, то
            * можно увидеть в перечислении Roles, что ролей модели всего 4
            * */
            for(int column = 0; column < model->columnCount(); column++)
            {
                // Записываем в stringList каждый элемент таблицы
                stringList << model->data(model-
>index(row,column)).toString();
            }
            /* После чего отправляем весь stringList в файл через текстовый
поток
            * добавляя разделители в виде ";", а также поставив в конце
символ окончания строки
            * */
            textStream << stringList.join(';')+"\n";
        }
        // Закрываем файл - готово
        csvFile.close();
        QMessageBox msg;
        msg.setText("Экспорт успешно завершен");
        msg.exec();
    }
}

void LookDataBase::on_filter_clicked()
{
    QHeaderView *q = new QHeaderView(Qt::Horizontal, this);
    q->setModel(model);
    FilterDialog *fd = new FilterDialog();
    for(int i = 0; i < q->count(); i++)
    {
        fd->addItem(model->headerData(i,Qt::Horizontal).toString());
    }
    fd->applyList();
    model->setFilter(fd->showDialog());
    ui->tableView->setModel(model);
    ui->tableView->resizeColumnsToContents();
}

void LookDataBase::on_deleteFilter_clicked()
{
    model->setFilter("");
    ui->tableView->setModel(model);
    ui->tableView->resizeColumnsToContents();
}

void LookDataBase::on_hideTableMenu_clicked()
{
    if (ui->hideTableMenu->text() == "Свернуть")

```

```

    {
        ui->hideTableMenu->setText("Развернуть");
        ui->listView->setVisible(false);
        ui->tableView->setVisible(false);
        ui->groupBox->adjustSize();
        ui->groupBox->repaint();
    }
    else
    {
        ui->hideTableMenu->setText("Свернуть");
        ui->listView->setVisible(true);
        ui->tableView->setVisible(false);
        ui->groupBox->adjustSize();
        ui->groupBox->repaint();
    }
}

void LookDataBase::on_tabWidget_tabCloseRequested(int index)
{
    ui->tabWidget->removeTab(index);
}

void LookDataBase::on_refresh_clicked()
{
    ;
}

```

settingmenu.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>SettingMenu</class>
    <widget class="QMainWindow" name="SettingMenu">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>413</width>
                <height>320</height>
            </rect>
        </property>
        <property name="windowTitle">
            <string>Настройки</string>
        </property>
        <widget class="QWidget" name="centralwidget">
            <layout class="QHBoxLayout" name="horizontalLayout_7">
                <item>
                    <layout class="QVBoxLayout" name="verticalLayout_3">
                        <item>
                            <widget class="QGroupBox" name="groupDBConnect">
                                <property name="font">
                                    <font>
                                        <pointsize>10</pointsize>
                                        <weight>50</weight>
                                        <bold>>false</bold>
                                    </font>
                                </property>
                                <property name="title">
                                    <string>Подключение к ДБ</string>
                                </property>
                                <layout class="QHBoxLayout" name="horizontalLayout_5">

```

```

<item>
<layout class="QHBoxLayout" name="horizontalLayout_2">
  <item>
    <layout class="QVBoxLayout" name="verticalLayout">
      <item>
        <widget class="QLineEdit" name="adrDB"/>
      </item>
      <item>
        <widget class="QLineEdit" name="portDB"/>
      </item>
      <item>
        <widget class="QLineEdit" name="nameDB"/>
      </item>
    </layout>
  </item>
  <item>
    <widget class="Line" name="line">
      <property name="orientation">
        <enum>Qt::Vertical</enum>
      </property>
    </widget>
  </item>
  <item>
    <layout class="QVBoxLayout" name="verticalLayout_2">
      <item>
        <widget class="QLineEdit" name="userDB"/>
      </item>
      <item>
        <widget class="QLineEdit" name="passDB">
          <property name="echoMode">
            <enum>QLineEdit::Password</enum>
          </property>
        </widget>
      </item>
    </layout>
  <item>
    <layout class="QHBoxLayout" name="horizontalLayout">
      <item>
        <widget class="QPushButton" name="saveChange">
          <property name="minimumSize">
            <size>
              <width>40</width>
              <height>40</height>
            </size>
          </property>
          <property name="toolTip">
            <string>Сохранить</string>
          </property>
          <property name="styleSheet">
            <string notr="true"/>
          </property>
          <property name="text">
            <string/>
          </property>
          <property name="icon">
            <iconset resource="image.qrc">
              <normaloff>:/icon/glyphicons-445-floppy-
saved.png</normaloff>:/icon/glyphicons-445-floppy-saved.png</iconset>
            </property>
          <property name="iconSize">
            <size>
              <width>30</width>

```

```

        <height>30</height>
    </size>
</property>
</widget>
</item>
<item>
<widget class="QPushButton" name="clearFields">
    <property name="minimumSize">
        <size>
            <width>40</width>
            <height>40</height>
        </size>
    </property>
    <property name="toolTip">
        <string>Очистить</string>
    </property>
    <property name="styleSheet">
        <string notr="true"/>
    </property>
    <property name="text">
        <string/>
    </property>
    <property name="icon">
        <iconset resource="image.qrc">
            <normaloff>:/icon/glyphicons-68-
cleaning.png</normaloff>:/icon/glyphicons-68-cleaning.png</iconset>
        </property>
    <property name="iconSize">
        <size>
            <width>30</width>
            <height>30</height>
        </size>
    </property>
</widget>
</item>
</layout>
</item>
</layout>
</item>
</layout>
</item>
</layout>
</widget>
</item>
<item>
<layout class="QHBoxLayout" name="horizontalLayout_6">
    <item>
    <layout class="QHBoxLayout" name="horizontalLayout_3">
        <item>
        <widget class="QLabel" name="label">
            <property name="font">
                <font>
                    <pointsize>10</pointsize>
                    <weight>50</weight>
                    <bold>false</bold>
                </font>
            </property>
            <property name="text">
                <string>Статус ДБ:</string>
            </property>
        </widget>
    </item>

```



```

<item>
  <widget class="QLabel" name="statusDB">
    <property name="font">
      <font>
        <family>Narkisim</family>
        <pointsize>10</pointsize>
        <weight>50</weight>
        <bold>>false</bold>
      </font>
    </property>
    <property name="text">
      <string>OFF</string>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="checkConnectDB">
    <property name="minimumSize">
      <size>
        <width>40</width>
        <height>40</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>16777215</width>
        <height>16777215</height>
      </size>
    </property>
    <property name="toolTip">
      <string>Проверить подключение</string>
    </property>
    <property name="styleSheet">
      <string notr="true"/>
    </property>
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="image.qrc">
        <normaloff>:/icon/glyphicons-153-
check.png</normaloff>:/icon/glyphicons-153-check.png</iconset>
      </property>
    <property name="iconSize">
      <size>
        <width>30</width>
        <height>30</height>
      </size>
    </property>
  </widget>
</item>
</layout>
</item>
<item>
  <spacer name="horizontalSpacer_2">
    <property name="orientation">
      <enum>Qt::Horizontal</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>40</width>
        <height>20</height>
      </size>
    </property>
  </spacer>
</item>

```

```

        </size>
    </property>
</spacer>
</item>
</layout>
</item>
<item>
<layout class="QHBoxLayout" name="horizontalLayout_4">
    <item>
    <spacer name="horizontalSpacer">
    <property name="orientation">
    <enum>Qt::Horizontal</enum>
    </property>
    <property name="sizeHint" stdset="0">
    <size>
    <width>40</width>
    <height>20</height>
    </size>
    </property>
    </spacer>
    </item>
    <item>
    <layout class="QVBoxLayout" name="verticalLayout_5">
    <item>
    <spacer name="verticalSpacer">
    <property name="orientation">
    <enum>Qt::Vertical</enum>
    </property>
    <property name="sizeHint" stdset="0">
    <size>
    <width>20</width>
    <height>40</height>
    </size>
    </property>
    </spacer>
    </item>
    <item>
    <layout class="QVBoxLayout" name="verticalLayout_4">
    <item>
    <widget class="QPushButton" name="mainMenu">
    <property name="minimumSize">
    <size>
    <width>40</width>
    <height>40</height>
    </size>
    </property>
    <property name="toolTip">
    <string>Главное меню</string>
    </property>
    <property name="styleSheet">
    <string notr="true"/>
    </property>
    <property name="text">
    <string/>
    </property>
    <property name="icon">
    <iconset resource="image.qrc">
    <normaloff>:/icon/glyphicons-517-menu-
hamburger.png</normaloff>:/icon/glyphicons-517-menu-hamburger.png</iconset>
    </property>
    <property name="iconSize">
    <size>

```

```

        <width>30</width>
        <height>30</height>
    </size>
</property>
</widget>
</item>
<item>
    <widget class="QPushButton" name="exitSetting">
        <property name="enabled">
            <bool>true</bool>
        </property>
        <property name="sizePolicy">
            <sizepolicy hstretch="Minimum" vsizetype="Fixed">
                <horstretch>0</horstretch>
                <verstretch>0</verstretch>
            </sizepolicy>
        </property>
        <property name="minimumSize">
            <size>
                <width>40</width>
                <height>40</height>
            </size>
        </property>
        <property name="toolTip">
            <string>Выход</string>
        </property>
        <property name="styleSheet">
            <string notr="true"/>
        </property>
        <property name="text">
            <string/>
        </property>
        <property name="icon">
            <iconset resource="image.qrc">
                <normaloff>:/icon/glyphicons-389-
exit.png</normaloff>:/icon/glyphicons-389-exit.png</iconset>
            </property>
            <property name="iconSize">
                <size>
                    <width>30</width>
                    <height>30</height>
                </size>
            </property>
        </widget>
    </item>
</layout>
</item>
</layout>
</item>
</layout>
</item>
</layout>
</item>
</layout>
</widget>
</widget>
<resources>
    <include location="image.qrc"/>
</resources>
<connections/>
</ui>

```

```

# settingmenu.h

#ifndef SETTINGMENU_H
#define SETTINGMENU_H

#include <QMainWindow>
#include "mainmenu.h"

namespace Ui {
class SettingMenu;
}

class SettingMenu : public QMainWindow
{
    Q_OBJECT

public:
    explicit SettingMenu(QWidget *parent = 0);
    ~SettingMenu();

private slots:
    void on_exitSetting_clicked();

    void on_mainMenu_clicked();

    void on_checkConnectDB_clicked();

    void on_saveChange_clicked();

    void on_clearFields_clicked();

private:
    Ui::SettingMenu *ui;

signals:
    void signalSettingToMainMenu();
};

#endif // SETTINGMENU_H

# #include "settingmenu.h"
#include "ui_settingmenu.h"

#include <QSettings>
#include <QMessageBox>

#include <QSqlDatabase>
#include <QSqlQueryModel>
#include <QSqlError>
#include <QTableView>
#include <QDebug>

SettingMenu::SettingMenu(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::SettingMenu)
{
    ui->setupUi(this);

    QSettings settings("GatesB", "MakeMeSchedule");
    ui->adrDB->setText(settings.value("SETTING/ADDRESS_DB", "").toString());
    ui->portDB->setText(settings.value("SETTING/PORT_DB", "").toString());
    ui->nameDB->setText(settings.value("SETTING/NAME_DB", "").toString());

```

```

    ui->userDB->setText(settings.value("SETTING/USER_DB","").toString());
    ui->passDB->setText(settings.value("SETTING/PASSWORD_DB","").toString());

    ui->adrDB->setPlaceholderText("address database...");
    ui->portDB->setPlaceholderText("port database...");
    ui->nameDB->setPlaceholderText("name database...");
    ui->userDB->setPlaceholderText("user database...");
    ui->passDB->setPlaceholderText("password database...");
}

SettingMenu::~SettingMenu()
{
    delete ui;
}

void SettingMenu::on_exitSetting_clicked()
{
    exit(1);
}

void SettingMenu::on_mainMenu_clicked()
{
    this->close();
    emit signalSettingToMainMenu();
}

void SettingMenu::on_checkConnectDB_clicked()
{
    qDebug() << "Доступные драйверы: ";
    qDebug() << QSqlDatabase::drivers();
    QSqlDatabase db;
    //db.registerSqlDriver("QPSQL", new QSqlDriverCreatorBase<MyQPSQL>);
    qDebug() << "Доступность QPSQL = " + db.isDriverAvailable("QPSQL");

    db = QSqlDatabase::addDatabase("QPSQL");
    QSettings settings("GatesB", "MakeMeSchedule");

    ui->adrDB->setPlaceholderText("address DB...");
    ui->portDB->setPlaceholderText("port DB...");
    ui->userDB->setPlaceholderText("user DB...");
    ui->passDB->setPlaceholderText("password DB");

    db.setHostName(settings.value("SETTING/ADDRESS_DB","").toString());
    db.setDatabaseName(settings.value("SETTING/NAME_DB","").toString());
    db.setUserName(settings.value("SETTING/USER_DB","").toString());
    db.setPassword(settings.value("SETTING/PASSWORD_DB","").toString());
    db.setPort(settings.value("SETTING/PORT_DB","").toInt());
    if (db.open())
    {
        ui->statusDB->setText("ON");
        ui->statusDB->setStyleSheet("QLabel { color : green; }");
    }
    else
    {
        ui->statusDB->setText("OFF");
        ui->statusDB->setStyleSheet("QLabel { color : red; }");
    }
    db.close();
}

void SettingMenu::on_saveChange_clicked()
{

```

```

    QSettings settings("GatesB", "MakeMeSchedule");
    settings.setValue("SETTING/ADDRESS_DB", ui->adrDB->text());
    settings.setValue("SETTING/PORT_DB", ui->portDB->text());
    settings.setValue("SETTING/NAME_DB", ui->nameDB->text());
    settings.setValue("SETTING/USER_DB", ui->userDB->text());
    settings.setValue("SETTING/PASSWORD_DB", ui->passDB->text());
    settings.sync();
}

void SettingMenu::on_clearFields_clicked()
{
    ui->adrDB->clear();
    ui->portDB->clear();
    ui->nameDB->clear();
    ui->userDB->clear();
    ui->passDB->clear();

    QSettings settings("GatesB", "MakeMeSchedule");
    settings.setValue("SETTING/ADDRESS_DB", "");
    settings.setValue("SETTING/PORT_DB", "");
    settings.setValue("SETTING/NAME_DB", "");
    settings.setValue("SETTING/USER_DB", "");
    settings.setValue("SETTING/PASSWORD_DB", "");
    settings.sync();
}

```

createschedule.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>CreateSchedule</class>
    <widget class="QMainWindow" name="CreateSchedule">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>800</width>
                <height>600</height>
            </rect>
        </property>
        <property name="windowTitle">
            <string>MainWindow</string>
        </property>
        <widget class="QWidget" name="centralwidget">
            <layout class="QHBoxLayout" name="horizontalLayout_3">
                <item>
                    <layout class="QVBoxLayout" name="verticalLayout_8">
                        <item>
                            <widget class="QSplitter" name="splitter">
                                <property name="orientation">
                                    <enum>Qt::Horizontal</enum>
                                </property>
                                <widget class="QWidget" name="layoutWidget">
                                    <layout class="QHBoxLayout" name="horizontalLayout_2">
                                        <item>
                                            <layout class="QVBoxLayout" name="verticalLayout_7">
                                                <item>
                                                    <layout class="QVBoxLayout" name="verticalLayout_5">
                                                        <item>
                                                            <layout class="QVBoxLayout" name="verticalLayout">
                                                                <item>

```

```

        <widget class="QLabel" name="label">
        <property name="text">
        <string>Количество
учебных дней</string>
        </property>
        </widget>
    </item>
    <item>
        <widget class="QSpinBox" name="quantityDay">
        <property name="minimum">
        <number>1</number>
        </property>
        <property name="maximum">
        <number>7</number>
        </property>
        <property name="value">
        <number>6</number>
        </property>
        </widget>
    </item>
</layout>
</item>
<item>
    <layout class="QVBoxLayout" name="verticalLayout_2">
    <item>
        <widget class="QLabel" name="label_2">
        <property name="text">
        <string>Размер
популяции</string>
        </property>
        </widget>
    </item>
    <item>
        <widget class="QSpinBox" name="sizePopulation">
        <property name="minimum">
        <number>1</number>
        </property>
        <property name="maximum">
        <number>1000</number>
        </property>
        <property name="value">
        <number>1</number>
        </property>
        </widget>
    </item>
</layout>
</item>
<item>
    <layout class="QVBoxLayout" name="verticalLayout_3">
    <item>
        <widget class="QLabel" name="label_4">
        <property name="text">
        <string>Кол-во
итераций</string>
        </property>
        </widget>
    </item>
    <item>
        <widget class="QSpinBox" name="countIteration">
        <property name="minimum">
        <number>1</number>
        </property>

```

```

        <property name="maximum">
            <number>1000000</number>
        </property>
    </widget>
</item>
</layout>
</item>
</layout>
</item>
<item>
    <spacer name="verticalSpacer_2">
        <property name="orientation">
            <enum>Qt::Vertical</enum>
        </property>
        <property name="sizeHint" stdset="0">
            <size>
                <width>20</width>
                <height>40</height>
            </size>
        </property>
    </spacer>
</item>
</layout>
</item>
<item>
    <layout class="QVBoxLayout" name="verticalLayout_6">
        <item>
            <layout class="QVBoxLayout" name="verticalLayout_4">
                <item>
                    <widget class="QLabel" name="label_3">
                        <property name="text">
                            <string>Факультет
</string>
                        </property>
                    </widget>
                </item>
                <item>
                    <widget class="QComboBox" name="selectFaculty">
                        <property name="currentIndex">
                            <number>6</number>
                        </property>
                        <item>
                            <property name="text">
                                <string comment="101">Биологический</string>
                            </property>
                        </item>
                        <item>
                            <property name="text">
                                <string comment="102">Географии и геоэкологии</string>
                            </property>
                        </item>
                        <item>
                            <property name="text">
                                <string comment="103">Исторический</string>
                            </property>
                        </item>
                        <item>
                            <property name="text">
                                <string comment="104">Математический</string>
                            </property>
                        </item>
                    </widget>
                </item>
            </layout>
        </item>
    </layout>
</item>

```



```

        <property name="text">
            <string comment="105">Педагогический</string>
        </property>
    </item>
</item>
<item>
    <property name="text">
        <string comment="106">ИЯиМК</string>
    </property>
</item>
<item>
    <property name="text">
        <string comment="107">ПМиК</string>
    </property>
</item>
<item>
    <property name="text">
        <string comment="108">Филологический</string>
    </property>
</item>
<item>
    <property name="text">
        <string comment="109">Психологии и социальной
работы</string>
    </property>
</item>
<item>
    <property name="text">
        <string comment="110">Физической культуры</string>
    </property>
</item>
<item>
    <property name="text">
        <string comment="111">Физико-технический</string>
    </property>
</item>
<item>
    <property name="text">
        <string comment="112">Химико-технологический</string>
    </property>
</item>
<item>
    <property name="text">
        <string comment="113">Юридический</string>
    </property>
</item>
<item>
    <property name="text">
        <string comment="114">Экономический</string>
    </property>
</item>
<item>
    <property name="text">
        <string comment="115">Управления и социологии</string>
    </property>
</item>
</widget>
</item>
</layout>
</item>
<item>
    <spacer name="verticalSpacer">

```

```

        <property name="orientation">
            <enum>Qt::Vertical</enum>
        </property>
        <property name="sizeHint" stdset="0">
            <size>
                <width>20</width>
                <height>40</height>
            </size>
        </property>
    </spacer>
</item>
</layout>
</item>
</layout>
</widget>
<widget class="QFrame" name="frame">
    <property name="frameShape">
        <enum>QFrame::StyledPanel</enum>
    </property>
    <property name="frameShadow">
        <enum>QFrame::Raised</enum>
    </property>
</widget>
</widget>
</item>
<item>
<layout class="QHBoxLayout" name="horizontalLayout">
    <item>
        <widget class="QProgressBar" name="progressBar">
            <property name="minimumSize">
                <size>
                    <width>0</width>
                    <height>40</height>
                </size>
            </property>
            <property name="maximumSize">
                <size>
                    <width>16777215</width>
                    <height>40</height>
                </size>
            </property>
            <property name="toolTip">
                <string>Ход выполнения построения расписания</string>
            </property>
            <property name="maximum">
                <number>10</number>
            </property>
            <property name="value">
                <number>0</number>
            </property>
            <property name="orientation">
                <enum>Qt::Horizontal</enum>
            </property>
            <property name="invertedAppearance">
                <bool>>false</bool>
            </property>
            <property name="format">
                <string/>
            </property>
        </widget>
    </item>
</item>

```

```

<widget class="QPushButton" name="createSchedule">
  <property name="minimumSize">
    <size>
      <width>40</width>
      <height>40</height>
    </size>
  </property>
  <property name="maximumSize">
    <size>
      <width>40</width>
      <height>40</height>
    </size>
  </property>
  <property name="toolTip">
    <string>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p&gt;Начать
создание расписания&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string>
  </property>
  <property name="text">
    <string/>
  </property>
  <property name="icon">
    <iconset resource="image.qrc">
      <normaloff>:/icon/glyphicons-424-git-
create.png</normaloff>:/icon/glyphicons-424-git-create.png</iconset>
    </property>
  <property name="iconSize">
    <size>
      <width>30</width>
      <height>30</height>
    </size>
  </property>
</widget>
</item>
<item>
  <widget class="QPushButton" name="pushButton">
    <property name="minimumSize">
      <size>
        <width>40</width>
        <height>40</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>40</width>
        <height>40</height>
      </size>
    </property>
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="image.qrc">
        <normaloff>:/icon/glyphicons-517-menu-
hamburger.png</normaloff>:/icon/glyphicons-517-menu-hamburger.png</iconset>
      </property>
    <property name="iconSize">
      <size>
        <width>30</width>
        <height>30</height>
      </size>
    </property>
  </widget>

```

```

        </item>
    </layout>
</item>
</layout>
</item>
</layout>
<zorder></zorder>
</widget>
<widget class="QMenuBar" name="menubar">
    <property name="geometry">
        <rect>
            <x>0</x>
            <y>0</y>
            <width>800</width>
            <height>18</height>
        </rect>
    </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources>
    <include location="image.qrc"/>
</resources>
<connections/>
</ui>

```

createschedule.h

```

#ifndef CREATESCCHEDULE_H
#define CREATESCCHEDULE_H

#include <QMainWindow>
#include <QSettings>
#include <QSqlDatabase>
#include <QDebug>
#include <QtConcurrent>
#include <QtCharts>
using namespace QtCharts;

namespace Ui {
class CreateSchedule;
}

class CreateSchedule : public QMainWindow
{
    Q_OBJECT

public:
    explicit CreateSchedule(QWidget *parent = 0);
    ~CreateSchedule();

private slots:
    void on_createSchedule_clicked();
    void valueChanged();
    void penaltyChanged(int *);
    void on_pushButton_clicked();

signals:
    void signalSettingToMainMenu();

```

```

private:
    Ui::CreateSchedule *ui;
    QSqlDatabase db;
    int * penaltyArray;
    int penaltyArraySize;
    QLineSeries * series;
    QBarSet * setbar;
    QBarSeries * barSeries;
    QChart *chart;
    QChartView *chartView;
};

#endif // CREATESCHEDULE_H

# createschedule.cpp

#include "createschedule.h"
#include "ui_createschedule.h"

#include "genetic.h"
#include "DB/ubd_faculty.h"
#include "DB/ubdc_ranks.h"

#include <QSqlTableModel>
#include <QSqlError>

CreateSchedule::CreateSchedule(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::CreateSchedule)
{
    ui->setupUi(this);
    db = QSqlDatabase::addDatabase("QPSQL");

    QSettings settings("GatesB", "MakeMeSchedule");
    db.setHostName(settings.value("SETTING/ADDRESS_DB","").toString());
    db.setDatabaseName(settings.value("SETTING/NAME_DB","").toString());
    db.setUserName(settings.value("SETTING/USER_DB","").toString());
    db.setPassword(settings.value("SETTING/PASSWORD_DB","").toString());
    db.setPort(settings.value("SETTING/PORT_DB","").toInt());
    db.open();

    qDebug() << "CreateSchedule";
    penaltyArraySize = this->ui->sizePopulation->text().toInt();
    penaltyArray = new int [penaltyArraySize];
    series = new QLineSeries();
    //setbar = new QBarSet("Schedules Penalty");
    QStringList categories;
    for (int i = 0; i < penaltyArraySize; i++)
    {
        penaltyArray[i] = 0;
        series->append(i, penaltyArray[i]);
        categories.append(QString::number(i));
        //setbar->insert(i, penaltyArray[i]);
    }
    //QBarCategoryAxis *axis = new QBarCategoryAxis();
    //axis->append(categories);
    //barSeries = new QBarSeries();
    //barSeries->append(setbar);

```

```

    chart = new QChart();
    //chart->legend()->hide();
    chart->addSeries(series);
    //chart->setAxisX(axis, series);
    chart->createDefaultAxes();
    //chart->addSeries(barSeries);
    chart->setAnimationOptions(QChart::SeriesAnimations);
    chart->setTitle("Simple line chart example");
    chartView = new QChartView(chart);
    chartView->setRenderHint(QPainter::Qt4CompatiblePainting);
    this->ui->splitter->addWidget(chartView);
    //this->ui->splitter->adjustSize();
    this->ui->splitter->repaint();
}

CreateSchedule::~CreateSchedule()
{
    delete ui;
}

void CreateSchedule::valueChanged()
{
    ui->progressBar->setValue(ui->progressBar->value() + 1);
}

void CreateSchedule::penaltyChanged(int * penaltyArr)
{
    int size = this->ui->sizePopulation->text().toInt();
    chart->removeSeries(series);
    for (int i = 0; i < size; i++){
        series->remove(i);
        series->insert(i, QPointF(i, penaltyArr[i]));
    }
    chart->addSeries(series);
}

void CreateSchedule::on_createSchedule_clicked()
{
    if (db.open())
    {
        Genetic gena;
        gena.setSizePopulation(ui->sizePopulation->text().toInt());
        gena.setQuantityStudyDays(ui->quantityDay->text().toInt());
        gena.setCountIteration(ui->countIteration->text().toInt());
        ui->progressBar->setMaximum(ui->countIteration->text().toInt());
        ui->progressBar->setValue(0);
        connect(&gena, SIGNAL(_ChangePenalty(int *)), this,
        SLOT(penaltyChanged(int *)));
        connect(&gena, SIGNAL(_ChangeValue()), this, SLOT(valueChanged()));

        res_Schedule * res = gena.StartGenetic();
        QSqlTableModel *model = new QSqlTableModel;
        model->setTable("res_schedule");
        model->setEditStrategy(QSqlTableModel::OnManualSubmit);
        if(!model->select())
        {
            QMessageBox q;
            q.setText("select - " + model->lastError().text());
            q.exec();
        }
        model->removeRows(0, model->rowCount());
        if(!model->submitAll())
        {

```

```

        QMessageBox q;
        q.setText(model->lastError().text());
        q.exec();
    }
    int newRow = 0;
    for (int i = 0; i < 313; i++, newRow++)
    {
        newRow = model->rowCount();
        if(!model->insertRow(newRow))
        {
            QMessageBox q;
            q.setText(model->lastError().text());
            q.exec();
        }
        model->setData(model->index(newRow,0), res[i].idSchedule);
        model->setData(model->index(newRow,1), res[i].codeSchedule);
        model->setData(model->index(newRow,2), res[i].idFaculty);
        model->setData(model->index(newRow,3), res[i].idDay);
        model->setData(model->index(newRow,4), 1);
        model->setData(model->index(newRow,5), res[i].idSubjectTeacher);
        model->setData(model->index(newRow,6), res[i].idHall);
        model->setData(model->index(newRow,7), res[i].idAlarm);
    }
    if(!model->submitAll())
    {
        QMessageBox q;
        q.setText(model->lastError().text());
        q.exec();
    }
}
else
{
    qDebug() << "DB is not open";
}
db.close();
}

void CreateSchedule::on_pushButton_clicked()
{
    this->close();
    emit signalSettingToMainMenu();
}

```

facultylistapplet.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>FacultyListApplet</class>
    <widget class="QFrame" name="FacultyListApplet">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>890</width>
                <height>516</height>
            </rect>
        </property>
        <property name="windowTitle">
            <string>Frame</string>
        </property>
        <property name="frameShape">

```

```

    <enum>QFrame::StyledPanel</enum>
</property>
<property name="frameShadow">
    <enum>QFrame::Raised</enum>
</property>
<layout class="QHBoxLayout" name="horizontalLayout_9">
<item>
    <layout class="QVBoxLayout" name="verticalLayout_4">
<item>
    <widget class="QGroupBox" name="FacultyListBox">
        <property name="title">
            <string>Факультет</string>
        </property>
        <layout class="QHBoxLayout" name="horizontalLayout">
<item>
            <widget class="QTableView" name="tableView">
                <property name="enabled">
                    <bool>true</bool>
                </property>
                <property name="gridStyle">
                    <enum>Qt::NoPen</enum>
                </property>
            </widget>
        </item>
    </layout>
</widget>
</item>
<item>
    <widget class="QGroupBox" name="FacultyFormBox">
        <property name="title">
            <string/>
        </property>
        <layout class="QHBoxLayout" name="horizontalLayout_8">
<item>
            <layout class="QVBoxLayout" name="verticalLayout_3">
<item>
                <layout class="QHBoxLayout" name="horizontalLayout_7">
<item>
                    <layout class="QHBoxLayout" name="horizontalLayout_3">
<item>
                        <layout class="QVBoxLayout" name="verticalLayout_2">
<item>
                            <layout class="QHBoxLayout" name="horizontalLayout_2">
<item>
                                <widget class="QLabel" name="label">
                                    <property name="text">
                                        <string>Факультет: </string>
                                    </property>
                                </widget>
                            </item>
                            <item>
                                <widget class="QComboBox" name="ComboFaculty"/>
                            </item>
                        </layout>
                    </item>
                    <item>
                        <widget class="QLineEdit" name="LineFaculty"/>
                    </item>
                </layout>
            </item>
        </item>
    </item>
</item>

```



```

<layout class="QVBoxLayout" name="verticalLayout">
  <item>
    <layout class="QHBoxLayout" name="horizontalLayout_4">
      <item>
        <widget class="QLabel" name="label_3">
          <property name="text">
            <string>Декан: </string>
          </property>
        </widget>
      </item>
      <item>
        <widget class="QLineEdit" name="LineDean"/>
      </item>
    </layout>
  </item>
  <item>
    <layout class="QHBoxLayout" name="horizontalLayout_6">
      <item>
        <widget class="QLabel" name="label_5">
          <property name="text">
            <string>Телефон: </string>
          </property>
        </widget>
      </item>
      <item>
        <widget class="QLineEdit" name="LinePhone">
          <property name="inputMask">
            <string/>
          </property>
          <property name="maxLength">
            <number>11</number>
          </property>
        </widget>
      </item>
    </layout>
  </item>
  </layout>
</item>
<item>
  <spacer name="horizontalSpacer">
    <property name="orientation">
      <enum>Qt::Horizontal</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>40</width>
        <height>20</height>
      </size>
    </property>
  </spacer>
</item>
</layout>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_5">
    <item>
      <widget class="QLabel" name="label_4">
        <property name="text">
          <string>Адрес: </string>
        </property>
      </widget>
    </item>
  </layout>
</item>

```

```

        </widget>
    </item>
</item>
    <widget class="QLineEdit" name="LineAdress"/>
</item>
</layout>
</item>
</layout>
</item>
</layout>
</widget>
</item>
</layout>
</widget>
</item>
</layout>
</item>
</layout>
</widget>
</item>
</layout>
</item>
</layout>
</widget>
</item>
</layout>
</item>
</layout>
</widget>
</resources/>
</connections/>
</ui>

```

facultylistapplet.h

```

#ifndef FACULTYLISTAPPLET_H
#define FACULTYLISTAPPLET_H

#include <QFrame>

#include <QSqlTableModel>

namespace Ui {
class FacultyListApplet;
}

class FacultyListApplet : public QFrame
{
    Q_OBJECT

public:
    explicit FacultyListApplet(QWidget *parent = 0);
    ~FacultyListApplet();
    void SetTableViewModel(QSqlTableModel*);
private slots:
    void on_tableView_clicked(const QModelIndex &index);

    void on_ComboFaculty_currentIndexChanged(const QString &arg1);

    void on_ComboFaculty_activated(const QString &arg1);

private:
    QSqlTableModel *model;
    Ui::FacultyListApplet *ui;
    bool isInterface;
};

#endif // FACULTYLISTAPPLET_H

```

facultylistapplet.cpp

```

#include "facultylistapplet.h"

```

```

#include "ui_facultylistapplet.h"
#include "lov.h"
#include "QDebug"

FacultyListApplet::FacultyListApplet(QWidget *parent) :
    QFrame(parent),
    ui(new Ui::FacultyListApplet)
{
    ui->setupUi(this);
    isInterface = false;
    LOV lov;
    lov.LookupValue("UDB_FACULTY", "", "name");
    ui->ComboFaculty->addItem(lov.getResult());
    ui->ComboFaculty->setCurrentIndex(-1);
}

FacultyListApplet::~FacultyListApplet()
{
    delete ui;
}

void FacultyListApplet::SetTableViewModel(QSqlTableModel * model)
{
    this->model = model;
    ui->tableView->setModel(this->model);
    ui->tableView->resizeColumnsToContents();
}

void FacultyListApplet::on_tableView_clicked(const QModelIndex &index)
{
    isInterface = true;
    int row = index.row();
    //ui->LineFaculty->setText(this->model->data(index.sibling(row,
1)).toString());
    ui->ComboFaculty->setCurrentText(this->model->data(index.sibling(row,
2)).toString());
    //ui->LineFacultyCode->setText(this->model->data(index.sibling(row,
2)).toString());
    ui->LineDean->setText(this->model->data(index.sibling(row,
3)).toString());
    ui->LineAdress->setText(this->model-
>data(index.sibling(row,4)).toString());
    ui->LinePhone->setText(this->model->data(index.sibling(row,
5)).toString());
    //ui->ComboFaculty->setItemData(index, 1, facId);
    //ui->ComboFaculty->setItemData(index, 10, subjId);
}

//const QString facId = Qt::UserRole + 1;
//const QString subjId = Qt::UserRole + 2;

/*void FacultyListApplet::on_ComboFaculty_currentIndexChanged(int index)
{
    ui->ComboFaculty->itemData(index, facId);
}*/

void FacultyListApplet::on_ComboFaculty_currentIndexChanged(const QString
&arg1)
{
    if (isInterface)
    {
        LOV lov;

```

```

        lov.LookupValue("UDB_FACULTY", " AND name = '" + arg1 + "'",
"value");
        ui->LineFaculty->setText(lov.getResult().takeFirst());
        isInterface = false;
    }
}

void FacultyListApplet::on_ComboFaculty_activated(const QString &arg1)
{
    isInterface = true;
}

```

TeacherView.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>TeacherView</class>
    <widget class="QFrame" name="TeacherView">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>903</width>
                <height>559</height>
            </rect>
        </property>
        <property name="windowTitle">
            <string>Frame</string>
        </property>
        <property name="frameShape">
            <enum>QFrame::StyledPanel</enum>
        </property>
        <property name="frameShadow">
            <enum>QFrame::Raised</enum>
        </property>
        <layout class="QHBoxLayout" name="horizontalLayout_10">
            <item>
                <widget class="QSplitter" name="splitter">
                    <property name="orientation">
                        <enum>Qt::Vertical</enum>
                    </property>
                    <widget class="QGroupBox" name="groupBox">
                        <property name="title">
                            <string>Преподаватели</string>
                        </property>
                        <layout class="QHBoxLayout" name="horizontalLayout">
                            <item>
                                <widget class="QTableView" name="tableView"/>
                            </item>
                        </layout>
                    </widget>
                    <widget class="QGroupBox" name="groupBox_2">
                        <property name="title">
                            <string/>
                        </property>
                        <layout class="QHBoxLayout" name="horizontalLayout_9">
                            <item>
                                <layout class="QVBoxLayout" name="verticalLayout">
                                    <item>
                                        <layout class="QHBoxLayout" name="horizontalLayout_5">

```

```

<item>
<layout class="QHBoxLayout" name="horizontalLayout_4">
  <item>
    <layout class="QHBoxLayout" name="horizontalLayout_2">
      <item>
        <widget class="QLabel" name="label">
          <property name="text">
            <string>Фамилия: </string>
          </property>
        </widget>
      </item>
      <item>
        <widget class="QLineEdit" name="LineLastName"/>
      </item>
    </layout>
  </item>
  <item>
    <layout class="QHBoxLayout" name="horizontalLayout_3">
      <item>
        <widget class="QLabel" name="label_2">
          <property name="text">
            <string>Имя: </string>
          </property>
        </widget>
      </item>
      <item>
        <widget class="QLineEdit" name="LineFirstName"/>
      </item>
    </layout>
  </item>
  <item>
    <widget class="QLabel" name="label_3">
      <property name="text">
        <string>Отчество: </string>
      </property>
    </widget>
  </item>
  <item>
    <widget class="QLineEdit" name="LineSecondName"/>
  </item>
</layout>
</item>
<item>
<layout class="QHBoxLayout" name="horizontalLayout_8">
  <item>
    <layout class="QHBoxLayout" name="horizontalLayout_6">
      <item>
        <widget class="QLabel" name="label_4">
          <property name="text">
            <string>Факультет: </string>
          </property>
        </widget>
      </item>
      <item>
        <widget class="QLineEdit" name="LineFaculty"/>
      </item>
    </layout>
  </item>
  <item>
    <layout class="QHBoxLayout" name="horizontalLayout_7">

```

```

        <item>
            <widget class="QLabel" name="label_5">
                <property name="text">
                    <string>Ранг: </string>
                </property>
            </widget>
        </item>
        <item>
            <widget class="QLineEdit" name="LineRank"/>
        </item>
    </layout>
</item>
<item>
    <spacer name="horizontalSpacer">
        <property name="orientation">
            <enum>Qt::Horizontal</enum>
        </property>
        <property name="sizeHint" stdset="0">
            <size>
                <width>40</width>
                <height>20</height>
            </size>
        </property>
    </spacer>
</item>
</layout>
</item>
</layout>
</item>
</layout>
</widget>
</widget>
</item>
</layout>
</widget>
<resources/>
<connections/>
</ui>

```

TeacherView.h

```

#ifndef TEACHERVIEW_H
#define TEACHERVIEW_H

#include <QFrame>
#include <QSqlTableModel>

namespace Ui {
class TeacherView;
}

class TeacherView : public QFrame
{
    Q_OBJECT

public:
    explicit TeacherView(QWidget *parent = 0);
    ~TeacherView();
    void SetTableViewModel(QSqlTableModel*);

private slots:

```

```

        void on_tableView_clicked(const QModelIndex &index);

private:
    QSqlTableModel *model;
    Ui::TeacherView *ui;
};

#endif // TEACHERVIEW_H

# TeacherView.cpp

#include "TeacherView.h"
#include "ui_TeacherView.h"

TeacherView::TeacherView(QWidget *parent) :
    QFrame(parent),
    ui(new Ui::TeacherView)
{
    ui->setupUi(this);
}

TeacherView::~TeacherView()
{
    delete ui;
}

void TeacherView::SetTableModel(QSqlTableModel * model)
{
    this->model = model;
    ui->tableView->setModel(this->model);
    ui->tableView->resizeColumnsToContents();
}

void TeacherView::on_tableView_clicked(const QModelIndex &index)
{
    int row = index.row();
    ui->LineLastName->setText(this->model->data(index.sibling(row,
4)).toString());
    ui->LineFirstName->setText(this->model->data(index.sibling(row,
3)).toString());
    ui->LineFaculty->setText(this->model-
>data(index.sibling(row,2)).toString());
    ui->LineRank->setText(this->model->data(index.sibling(row,
1)).toString());
}

# genetic.h

#ifndef GENETIC_H
#define GENETIC_H

#include <QString>
#include <QSqlQuery>
#include <QDebug>
#include <QThread>

#include "DB/res_schedule.h"
#include "DB/ubd_groups.h"
#include "DB/ubd_subjectteacher.h"

```

```

#include "DB/ubd_lecturehalls.h"
#include "DB/ubdc_schedulealarm.h"
#include "DB/ubd_subject.h"
#include "DB/ubdc_week.h"

#include "time.h"

#include "CriteriaThread.h"

struct critArr
{
    int code;
    int summ;
    critArr()
    {
        code = 0;
        summ = 0;
    }
};

struct criteriaArray
{
    int codeSchedule;
    //Критерий, что на учебной неделе, кол-во занятий будет равно учебному
    плану
    //CriteriaQuantityStudyHour - функция проверки данного критерия
    int criteria_1;
    //QStringList checkCr1;
    //Критерий, что в учебный день, одинаковые предметы будут идти друг за
    другом
    //CriteriaBetweenPair - функция проверки данного критерия
    int criteria_2;
    //QStringList checkCr2;
    //Критерий, что в учебный день, не будет больше двух одинаковых предметов
    //CriteriaTwoSubject - функция проверки данного критерия
    int criteria_3;
    //QStringList checkCr3;
    //Критерий, что нет пересечений по занятости аудитории
    //CriteriaFreeHall - функция проверки данного критерия
    int criteria_4;
    //QVariantList checkCr4;
    //Критерий, что физкультурный предмет проходит в физкультурной аудитории
    int criteria_5;
    bool cr1WasGood;

    criteriaArray()
    {
        codeSchedule = 0;
        criteria_1 = 0;
        cr1WasGood = false;
        criteria_2 = 0;
        criteria_3 = 0;
        criteria_4 = 0;
        criteria_5 = 0;
    }
    bool AllCriteriaGood()
    {
        if (criteria_1 == 0
            && criteria_2 == 0
            && criteria_3 == 0
            && criteria_4 == 0

```



```

        && criteria_5 == 0)
            return true;
        return false;
    }

    void ClearPenalty()
    {
        criteria_1 = 0;
        criteria_2 = 0;
        criteria_3 = 0;
        criteria_4 = 0;
        criteria_5 = 0;
    }

    int summCriteria()
    {
        return criteria_1 + criteria_2 + criteria_3 + criteria_4 +
criteria_5;
    }

    int resultCriteria()
    {
        return 1;
    }

    void PrintCriteria()
    {
        qDebug() << "criteria_1 = " << criteria_1 << endl;
        qDebug() << "criteria_2 = " << criteria_2 << endl;
        qDebug() << "criteria_3 = " << criteria_3 << endl;
        qDebug() << "criteria_4 = " << criteria_4 << endl;
        qDebug() << "criteria_5 = " << criteria_5 << endl;
        qDebug() << "_____";
    }

    /*criteriaArray & operator =(const criteriaArray & input)
    {
        this->codeSchedule = input.codeSchedule;
        this->criteria_1 = input.criteria_1;
        this->criteria_2 = input.criteria_2;
        this->criteria_3 = input.criteria_3;
        this->criteria_4 = input.criteria_4;
        return *this;
    }*/
    void setData(criteriaArray & in1, criteriaArray in2)
    {
        in1.codeSchedule = in2.codeSchedule;
        in1.criteria_1 = in2.criteria_1;
        in1.criteria_2 = in2.criteria_2;
        in1.criteria_3 = in2.criteria_3;
        in1.criteria_4 = in2.criteria_4;
        in1.criteria_5 = in2.criteria_5;
    }
};

class Genetic : public QObject
{
    Q_OBJECT

signals:
    void _ChangeValue();
    void _ChangePenalty(int *);

```

```

private:
    int sizePopulation; //Размер популяции
    int currentFaculty; //Факультет для которого составляется расписание
    int quatityStudyDays; //Кол-во учебных дней
    int codeCurrentFaculty;
    int sizeSchedule; //Формируется из размера популяции * кол-во учебных дней
    int sizeST;
    int countIteration;
    int goodSchedule;

    ubd_Groups * Groups; //Группы студентов
    int GroupsSize;
    ubd_SubjectTeacher * SubjectTeacher; //Назначения п-п
    int SubjectTeacherSize;
    ubd_LectureHalls * LectureHalls;
    int LectureHallsSize;
    ubdc_ScheduleAlarm * Alarm;
    int AlarmSize;
    ubd_Subject * Subject;
    int SubjectSize;
    ubdc_Week * Week;
    int WeekSize;

    ubd_LectureHalls * LectureHallsFixed;
    int LectureHallsFixedSize;
    ubd_LectureHalls * LectureHallsNotFixed;
    int LectureHallsNotFixedSize;

    ubd_LectureHalls * LectureHallsSport;
    int LectureHallsSportSize;

    void clearPenalty();
    int findGood();
public:
    res_Schedule * schedule;
    criteriaArray * schedulePenalty;

    Genetic()
    {
        countIteration = 100;
        sizePopulation = 50;
        currentFaculty = 17;
        quatityStudyDays = 6; //
    }

    ~Genetic()
    {
        delete [] SubjectTeacher;
        delete [] LectureHalls;
        delete [] LectureHallsFixed;
        delete [] LectureHallsNotFixed;
        delete [] LectureHallsSport;
        delete [] Alarm;
        delete [] schedule;
        delete [] schedulePenalty;
        delete [] Week;
    }

    void setSizePopulation(int);
    void setQuatityStudyDays(int);
    void setCurrentFaculty(int);
    void setCodeCurrentFaculty(int);

```

```

void setCountIteration(int);

bool CreateFirstPopulation();
bool CreateNewGen(int);
bool Mutation(int, int, int);
int Crossing();
bool Criteria();
res_Schedule * StartGenetic();

void CriteriaQuantityStudyHour();
void CriteriaBetweenPair();
void CriteriaTwoSubject();
void CriteriaFreeHall();

void ExportShedule(int);

void qsortCustom(int, int);
void bubbleSortCustom();
void shellSortCustom();
void shellSortCustomCritArr(critArr * arr, int _size);
void GetFromCodeMemory(res_Schedule *, int);
int ScheduleLessCriteria(critArr *);
};
#endif // GENETIC_H

```

genetic.cpp

```

#include "genetic.h"

#include <QDir>
#include <QMessageBox>
/*Genetic::Genetic()
{
    countIteration = 100;
    sizePopulation = 50;
    currentFaculty = 17;
    quatityStudyDays = 6;
}
*/
/*Genetic::~~Genetic()
{
    delete [] SubjectTeacher;
    delete [] LectureHalls;
    delete [] Alarm;
    delete [] schedule;
    delete [] schedulePenalty;
    delete [] Week;
}*/

void Genetic::setSizePopulation(int _input)
{
    this->sizePopulation = _input;
}

void Genetic::setQuatityStudyDays(int _input)
{
    this->quatityStudyDays = _input;
}

void Genetic::setCodeCurrentFaculty(int _input)
{

```

```

        this->codeCurrentFaculty = _input;
    }

void Genetic::setCountIteration(int _input)
{
    this->countIteration = _input;
}

res_Schedule * Genetic::StartGenetic()
{
    SubjectTeacherSize = SubjectTeacher->getQuantityRow(currentFaculty);
    SubjectTeacher = new ubd_SubjectTeacher [SubjectTeacherSize];

    LectureHallsSize = LectureHalls->getQuantityRow(currentFaculty);
    LectureHalls = new ubd_LectureHalls [LectureHallsSize];

    AlarmSize = Alarm->getQuantityRow(currentFaculty);
    Alarm = new ubdc_ScheduleAlarm [AlarmSize];

    WeekSize = Week->getQuantityRow(currentFaculty);
    Week = new ubdc_Week [WeekSize];

    sizeST = Subject->summSubjectHours(QString::number(currentFaculty));
    schedulePenalty = new criteriaArray [sizePopulation * sizeST];
    qDebug() << "SizeST: " << sizeST << " sizeS = " << sizePopulation *
sizeST;
    qDebug() << "Start Genetic";
    sizeSchedule = sizePopulation * sizeST;
    schedule = new res_Schedule [sizeSchedule];

    //Первая популяция
    CreateFirstPopulation();
    res_Schedule * tmp = new res_Schedule [sizeST];
    //tmp = schedule->GetFromCodeMemory(schedule, sizeSchedule, 0);
    //-----
    int wasGSC = 0;
    int greatScheduleCode = -1;
    int stepWithoutChange = 0;
    for(int i = 0; i < countIteration; i++)
    {

        //emit _ChangeValue();
        qDebug() << "begin";
        clearPenalty();
        qDebug() << "clearPenalty";
        Criteria();
        qDebug() << "Criteria";
        int findGoodCode = findGood();
        if (findGoodCode >= 0)
        {
            GetFromCodeMemory(tmp, findGoodCode);
            tmp->printSchedule(tmp, sizeST, findGoodCode);
            return tmp;
        }
        qDebug() << "findGood";
        wasGSC = greatScheduleCode;
        greatScheduleCode = Crossing();
        if (wasGSC == greatScheduleCode) stepWithoutChange ++;
        else stepWithoutChange = 0;
        qDebug() << "Crossing";
        //if (stepWithoutChange < 100)
            Mutation(0, sizeSchedule, greatScheduleCode);
        //else
    }
}

```

```

        //{
        //    Mutation(0, sizeSchedule, -1);
        //    stepWithoutChange = 0;
        //}
        qDebug() << "Mutation";
        qDebug() << "CreateNewGen = " << i;

    }
    GetFromCodeMemory(tmp, greatScheduleCode);
    tmp->printSchedule(tmp, sizeST, greatScheduleCode);
    qDebug() << "End Genetic";
    return tmp;
}

bool Genetic::CreateFirstPopulation()
{
    qDebug() << "CreateFirstPopulation";
    SubjectTeacher->downloadFromDB_All(SubjectTeacher, currentFaculty);
    LectureHalls->downloadFromDB_All(LectureHalls, currentFaculty);

    LectureHallsFixedSize = LectureHalls->getCountFixedHalls(LectureHalls,
    LectureHallsSize);
    LectureHallsFixed = new ubd_LectureHalls[LectureHallsFixedSize];
    LectureHallsFixed->getFixedHalls(LectureHallsFixed, LectureHalls,
    LectureHallsSize);

    LectureHallsNotFixedSize = LectureHalls-
>getCountNotFixedHalls(LectureHalls, LectureHallsSize);
    LectureHallsNotFixed = new ubd_LectureHalls[LectureHallsNotFixedSize];
    LectureHallsNotFixed->getNotFixedHalls(LectureHallsNotFixed,
    LectureHalls, LectureHallsSize);

    LectureHallsSportSize = LectureHalls->getCountSportHalls(LectureHalls,
    LectureHallsSize);
    LectureHallsSport = new ubd_LectureHalls[LectureHallsSportSize];
    LectureHallsSport->getSportHalls(LectureHallsSport, LectureHalls,
    LectureHallsSize);

    Alarm->downloadFromDB_All(Alarm, currentFaculty);
    Week->downloadFromDB_All(Week, currentFaculty);

    int idHall;
    for(int i = 0, j = 0; i < sizePopulation; i++)
    {
        for (int st = 0; st < SubjectTeacherSize; st++)
        {
            int countHour = Subject-
>countSubjectHoursFromST(QString::number(SubjectTeacher[st].idST));
            //qDebug() << "countHour=" << countHour;
            for(;countHour > 0; j++, countHour--)
            {
                schedule[j].idSubjectTeacher = SubjectTeacher[st].idST;
                if (Subject->fixedGetIdHall(idHall,
    QString::number(schedule[j].idSubjectTeacher))
                    schedule[j].idHall = idHall;
            }
        }
    }

    for(int i = 0, k = 0; i < sizePopulation; i++)
    {

```

```

        for(int j = 0; j < sizeST; j++, k++)
        {
            int indexST = SubjectTeacher->currFrom(SubjectTeacher,
SubjectTeacherSize, schedule[k].idSubjectTeacher);
            int randomize;
            schedule[k].idSchedule = k;
            schedule[k].codeSchedule = i;
            schedule[k].idFaculty = currentFaculty;
            //-----
            randomize = rand() % WeekSize;
            randomize = randomize < 0 ? 0 : randomize;
            schedule[k].idDay = Week[randomize].idDay;
            //-----
            //randomize = 0 + rand() % SubjectTeacherSize;
            //randomize = randomize < 0 ? 0 : randomize;
            //schedule[k].idSubjectTeacher = SubjectTeacher[randomize].idST;
            //-----
            if (!Subject->fixedGetIdHall(idHall,
QString::number(schedule[j].idSubjectTeacher)))
            {
                randomize = rand() % LectureHallsSize;
                randomize = randomize < 0 ? 0 : randomize;
                schedule[k].idHall = LectureHalls[randomize].idHall;
            }
            /*int idGroup = 0;
            if (!Subject->fixedHall(idGroup,
QString::number(schedule[i].idSubjectTeacher)))
            {
                randomize = rand() % LectureHallsNotFixedSize;
                randomize = randomize < 0 ? 0 : randomize;
                schedule[k].idHall = LectureHallsNotFixed[randomize].idHall;
            }
            else
            {
                randomize = rand() % LectureHallsFixedSize;
                randomize = randomize < 0 ? 0 : randomize;
                schedule[k].idHall = LectureHallsFixed[randomize].idHall;
            }*/
            //-----
            randomize = rand() % SubjectTeacher[indexST].maxPair;
            randomize = randomize < 0 ? 0 : randomize;
            schedule[k].idAlarm = Alarm[randomize].idAlarm;
        }
    }

    //schedule->printSchedule(schedule, sizeSchedule, 0);
    return true;
}

bool Genetic::CreateNewGen(int index)
{
    int randomize = rand() % WeekSize;
    randomize = randomize < 0 ? 0 : randomize;
    schedule[index].idDay = Week[randomize].idDay;
    //-----
    randomize = 0 + rand() % LectureHallsSize;
    randomize = randomize < 0 ? 0 : randomize;
    schedule[index].idHall = LectureHalls[randomize].idHall;
    //-----
    randomize = 0 + rand() % AlarmSize;
    randomize = randomize < 0 ? 0 : randomize;
    schedule[index].idAlarm = Alarm[randomize].idAlarm;
    return true;
}

```

```

}

bool Genetic::Mutation(int begin, int end, int exception)
{
    int randomize;
    int idGroup;
    int indexST;
    for(int i = begin; i < end; i++)
    {
        bool execMutation = false;
        if (exception == -1)
        {
            execMutation = true;
        }
        else
        {
            if (schedule[i].codeSchedule != exception &&
                schedule[i].summCriteria() != 0)
            {
                execMutation = true;
            }
        }
        if(execMutation)
        {
            /*randomize = rand() % 2;
            if (randomize == 0)
                schedule[i].idDay = Week->prevFrom(Week, WeekSize,
                schedule[i].idDay);
            if (randomize == 1)
                schedule[i].idDay = Week->nextFrom(Week, WeekSize,
                schedule[i].idDay);
            */
            indexST = SubjectTeacher->currFrom(SubjectTeacher,
                SubjectTeacherSize, schedule[i].idSubjectTeacher);
            schedule[i].idDay = Week->randomElement(Week, WeekSize);
            schedule[i].idAlarm = Alarm->randomElement(Alarm,
                SubjectTeacher[indexST].maxPair);

            indexST = SubjectTeacher->currFrom(SubjectTeacher,
                SubjectTeacherSize, schedule[i].idSubjectTeacher);
            if (!Subject->fixedHall(idGroup,
                QString::number(schedule[i].idSubjectTeacher)))
            {
                if (SubjectTeacher[indexST].X_SUBJECT_TYPE_CODE != "Sport")
                {
                    schedule[i].idHall = LectureHalls->
                    >randomElement(LectureHalls, LectureHallsSize);
                }
                else
                {
                    //qDebug()<<"sport1";
                    schedule[i].idHall = LectureHallsSport->
                    >randomElement(LectureHallsSport, LectureHallsSportSize);
                    //qDebug()<<"sport2";
                    //schedule[i].idAlarm = (rand() % 2) == 0 ?
                    Alarm[AlarmSize - 2].idAlarm : Alarm[AlarmSize - 1].idAlarm;
                    randomize = 1 + rand() % 3;
                    //if (randomize != 1 && randomize != 2) randomize = 1;
                    if (randomize == 1)
                        schedule[i].idAlarm =
                        Alarm[SubjectTeacher[indexST].maxPair - 1].idAlarm;
                    else

```

```

        schedule[i].idAlarm =
Alarm[SubjectTeacher[indexST].maxPair - 2].idAlarm;
        /*if (randomize == 3)
        schedule[i].idAlarm = Alarm[AlarmSize - 3].idAlarm;*/
    }
}
}
return true;
}
}

int Genetic::Crossing()
{
    critArr * crArr = new critArr [sizePopulation];
    int greatScheduleCode = ScheduleLessCriteria(crArr);
    shellSortCustomCritArr(crArr, sizePopulation);
    int size = sizePopulation / 2;
    for (int i = 0; i < size; i++)
    {
        int beginFrom = crArr[i].code * sizeST;
        int beginWhere = crArr[i + size].code * sizeST;
        for (int j = 0; j < sizeST; j++)
        {
            if (schedule[beginFrom + j].summCriteria() <= schedule[j +
beginWhere].summCriteria())
            {
                schedule[j + beginWhere].idDay = schedule[beginFrom +
j].idDay;
                schedule[j + beginWhere].idHall = schedule[beginFrom +
j].idHall;
                schedule[j + beginWhere].idAlarm = schedule[beginFrom +
j].idAlarm;
                schedule[j + beginWhere].criteria_5 = schedule[beginFrom +
j].criteria_5;
                schedule[j + beginWhere].criteria_10 = schedule[beginFrom +
j].criteria_10;

                schedule[j + beginWhere].criteria_1 = schedule[beginFrom +
j].criteria_1;
                schedule[j + beginWhere].criteria_2 = schedule[beginFrom +
j].criteria_2;
                schedule[j + beginWhere].criteria_3 = schedule[beginFrom +
j].criteria_3;
                schedule[j + beginWhere].criteria_4 = schedule[beginFrom +
j].criteria_4;
                schedule[j + beginWhere].criteria_6 = schedule[beginFrom +
j].criteria_6;
                schedule[j + beginWhere].criteria_7 = schedule[beginFrom +
j].criteria_7;
                schedule[j + beginWhere].criteria_8 = schedule[beginFrom +
j].criteria_8;
                schedule[j + beginWhere].criteria_9 = schedule[beginFrom +
j].criteria_9;
                schedule[j + beginWhere].criteria_11 = schedule[beginFrom +
j].criteria_11;
                schedule[j + beginWhere].criteria_12 = schedule[beginFrom +
j].criteria_12;
                schedule[j + beginWhere].criteria_13 = schedule[beginFrom +
j].criteria_13;
                schedule[j + beginWhere].criteria_14 = schedule[beginFrom +
j].criteria_14;
            }
        }
    }
}

```



```

    }
}
return greatScheduleCode;
}

bool Genetic::Criteria()
{
    CriteriaThread ths [sizePopulation];
    for (int i = 0; i < sizePopulation; i++)
    {
        ths[i].SetData(schedule, sizeSchedule
                      , Alarm, AlarmSize
                      , Subject, SubjectSize
                      , SubjectTeacher, SubjectTeacherSize
                      , LectureHalls, LectureHallsSize
                      , i, sizeST);

        ths[i].run();
    }
    for (int i = 0; i < sizePopulation; i++)
    {
        ths[i].wait();
    }
    return true;
}

```

CriteriaThread.h

```

#ifndef CRITERIATHREAD_H
#define CRITERIATHREAD_H

#include <QThread>
#include "DB/res_schedule.h"
#include "DB/ubd_groups.h"
#include "DB/ubd_subjectteacher.h"
#include "DB/ubd_lecturehalls.h"
#include "DB/ubdc_schedulealarm.h"
#include "DB/ubd_subject.h"
#include "DB/ubdc_week.h"

class CriteriaThread: public QThread
{
public:
    //public:
        //CriteriaThread();

public:
    explicit CriteriaThread(QString threadName);
    CriteriaThread(){};
    void SetData(res_Schedule * in, int size
                , ubdc_ScheduleAlarm * _alarm, int _alSize
                , ubd_Subject * _subject, int _subSize
                , ubd_SubjectTeacher * _st, int _stSize
                , ubd_LectureHalls * _lh, int _lhSize
                , int _codeSchedule, int _sizeST);

    // Переопределяем метод run(), в котором будет
    // располагаться выполняемый код
    void run();
    res_Schedule * schedule;
    int sizeSchedule;
    int codeSchedule;

    ubd_Groups * Groups; //Группы студентов

```

```

    int GroupsSize;
    ubd_SubjectTeacher * SubjectTeacher; //Назначения п-п
    int SubjectTeacherSize;
    ubd_LectureHalls * LectureHalls;
    int LectureHallsSize;
    ubdc_ScheduleAlarm * Alarm;
    int AlarmSize;
    ubd_Subject * Subject;
    int SubjectSize;
    ubdc_Week * Week;
    int WeekSize;
    int sizeST;
private:
    QString name;    // Имя потока
};

#endif // CRITERIATHREAD_H

```

CriteriaThread.cpp

```
#include "CriteriaThread.h"
```

```

CriteriaThread::CriteriaThread(QString threadName) :
    name(threadName)
{
}

void CriteriaThread::SetData( res_Schedule * in, int size
    , ubdc_ScheduleAlarm * _alarm, int _alSize
    , ubd_Subject * _subject, int _subSize
    , ubd_SubjectTeacher * _st, int _stSize
    , ubd_LectureHalls * _lh, int _lhSize
    , int _codeSchedule, int _sizeST)
{
    schedule = in;
    sizeSchedule = size;

    Alarm = _alarm;
    AlarmSize = _alSize;

    Subject = _subject;
    SubjectSize = _subSize;

    SubjectTeacher = _st;
    SubjectTeacherSize = _stSize;

    LectureHalls = _lh;
    LectureHallsSize = _lhSize;

    codeSchedule = _codeSchedule;
    sizeST = _sizeST;
}

void CriteriaThread::run()
{
    for (int j = sizeST * codeSchedule, n = 0; n < sizeST; n++, j++)
    {
        if (schedule[j].codeSchedule == codeSchedule)
        {

```

```

        int indexIdST = SubjectTeacher->currFrom(SubjectTeacher,
SubjectTeacherSize, schedule[j].idSubjectTeacher);
        int indexIdHall = LectureHalls->currFrom(LectureHalls,
LectureHallsSize, schedule[j].idHall);
        int quantHour = Subject-
>countSubjectHoursFromST(QString::number(schedule[j].idSubjectTeacher));
        int idGroup = 0;
        bool fixedHall = Subject->fixedHall(idGroup,
QString::number(schedule[j].idSubjectTeacher));

        int countCriteriaH1 = 1;
        int countCriteriaH2 = 0;
        int countCriteriaH3 = 0;
        int countCriteriaH4 = 0;
        int countCriteriaH11 = 0;
        int countCriteriaH13 = 0;
        for (int k = sizeST * codeSchedule, m = 0; m < sizeST; m++, k++)
        {
            if (schedule[k].codeSchedule == codeSchedule && k != j)
            {
                //Criteria
                if (schedule[k].idSubjectTeacher ==
schedule[j].idSubjectTeacher)
                {
                    countCriteriaH1 ++;
                }
                //Criteria2 + Criteria 3
                if (schedule[k].idSubjectTeacher ==
schedule[j].idSubjectTeacher
                    && schedule[k].idDay == schedule[j].idDay)
                {
                    countCriteriaH2 = abs(Alarm->getPairFormId(Alarm,
AlarmSize, schedule[j].idAlarm) - Alarm->getPairFormId(Alarm, AlarmSize,
schedule[k].idAlarm));
                    countCriteriaH3 ++;
                }
                //Criteria4
                if (schedule[k].idAlarm == schedule[j].idAlarm
                    && schedule[k].idDay == schedule[j].idDay
                    && (schedule[k].idHall == schedule[j].idHall))
                {
                    //|| SubjectTeacher[indexIdST].idGroup == SubjectTeacher[SubjectTeacher-
>currFrom(SubjectTeacher, SubjectTeacherSize,
schedule[k].idSubjectTeacher)].idGroup)
                    //&& !SubjectTeacher[indexIdST].sport &&
!SubjectTeacher[SubjectTeacher->currFrom(SubjectTeacher, SubjectTeacherSize,
schedule[k].idSubjectTeacher)].sport)
                    {
                        countCriteriaH4 ++;
                    }
                }
                //Criterial1
                if (schedule[k].idAlarm == schedule[j].idAlarm
                    && schedule[k].idDay == schedule[j].idDay
                    && SubjectTeacher[indexIdST].idTeacher ==
SubjectTeacher[SubjectTeacher->currFrom(SubjectTeacher, SubjectTeacherSize,
schedule[k].idSubjectTeacher)].idTeacher
                    && SubjectTeacher[indexIdST].X_SUBJECT_TYPE_CODE
!= "Sport" && SubjectTeacher[SubjectTeacher->currFrom(SubjectTeacher,
SubjectTeacherSize, schedule[k].idSubjectTeacher)].X_SUBJECT_TYPE_CODE !=
"Sport")
                {
                    countCriteriaH11 ++;
                }
            }
            //Criterial3

```

```

        /*if (schedule[k].idAlarm == schedule[j].idAlarm
            && schedule[k].idDay == schedule[j].idDay
            && SubjectTeacher[indexIdST].idGroup ==
SubjectTeacher[SubjectTeacher->currFrom(SubjectTeacher, SubjectTeacherSize,
schedule[k].idSubjectTeacher)].idGroup)
        {
            countCriteriaH13 ++;
        }*/
    }
    //else{qDebug() << "NOT EQUAL K";}
}
//int begin = j + (tmp[j].codeSchedule * sizeST);
//qDebug() << "begin = " << begin;
if (countCriteriaH1 != quantHour)
{
    schedule[j].criteria_1 ++;
}
if (countCriteriaH2 > 1)
{
    schedule[j].criteria_2 ++;
}
if (countCriteriaH3 > 2)
{
    schedule[j].criteria_3 ++;
}
if (countCriteriaH4 > 0)
{
    schedule[j].criteria_4 ++;
}
if (countCriteriaH11 > 0)
{
    schedule[j].criteria_11 ++;
}
//Criteria 13
if (countCriteriaH13 > 0)
{
    schedule[j].criteria_13 ++;
}
//Criteria 12
if (SubjectTeacher[indexIdST].X_SUBJECT_TYPE_CODE == "Sport" &&
LectureHalls[indexIdHall].X_SUBJECT_TYPE_CODE == "Sport")
{
    if (schedule[j].idAlarm !=
Alarm[SubjectTeacher[indexIdST].maxPair - 1].idAlarm
        && schedule[j].idAlarm !=
Alarm[SubjectTeacher[indexIdST].maxPair - 2].idAlarm)
        schedule[j].criteria_12 ++;
}

//-----
if (fixedHall)
{
    if (LectureHalls[indexIdHall].groupPriority != idGroup)
    {
        /*qDebug() << "ST = " << schedule[j].idSubjectTeacher;
qDebug() << "idh = " << schedule[j].idHall;
qDebug() << "gPriority = "<<
LectureHalls[indexIdHall].groupPriority;
qDebug() << "idGroup = " << idGroup;*/
        schedule[j].criteria_10 ++;
    }
}
else

```

```

        {
            /*if (SubjectTeacher[indexIdST].sport !=
LectureHalls[indexIdHall].sport)
            {
                //qDebug() << "ST" <<schedule[j].idSubjectTeacher <<
"bool = " << SubjectTeacher[indexIdST].sport;
                //qDebug() << "Hall" <<schedule[j].idHall << "bool = " <<
LectureHalls[indexIdHall].sport;
                schedule[j].criteria_5 ++;
            }
            if (SubjectTeacher[indexIdST].computer !=
LectureHalls[indexIdHall].computerHall)
            {
                schedule[j].criteria_6 ++;
            }
            if (SubjectTeacher[indexIdST].physic !=
LectureHalls[indexIdHall].physicHall)
            {
                schedule[j].criteria_7 ++;
            }
            if (SubjectTeacher[indexIdST].obj !=
LectureHalls[indexIdHall].objHall)
            {
                schedule[j].criteria_8 ++;
            }
            if (SubjectTeacher[indexIdST].technology !=
LectureHalls[indexIdHall].technologyHall)
            {
                schedule[j].criteria_9 ++;
            }
        }*/
        if (SubjectTeacher[indexIdST].X_SUBJECT_TYPE_CODE !=
LectureHalls[indexIdHall].X_SUBJECT_TYPE_CODE)
        {
            schedule[j].criteria_14 ++;
        }
    }
}
else{qDebug()<<"NOT EQUAL";}
}
}

```