

# Клиент к хранилищу фотографий

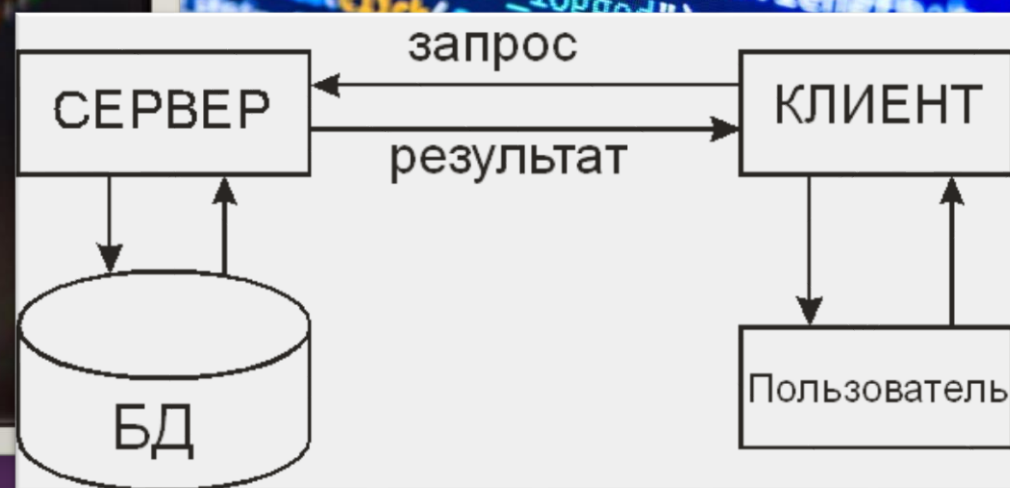
СИДОРЕНКО М. П.

РУКОВОДИТЕЛИ: ГОРШЕНИН А. К., МИРАБО Е. И.

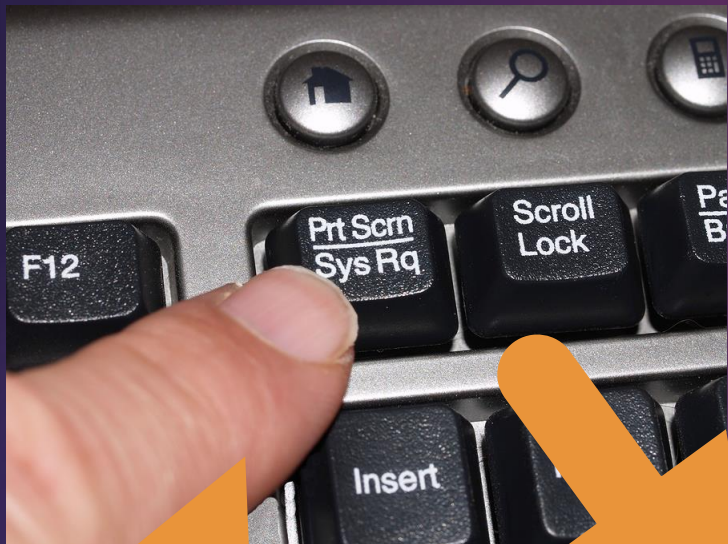
# Введение

```
private $password;  
private $database;  
private $charset;  
  
static private $link = null;  
  
static public function Connect()  
{  
    self::$link = mysql_connect(self::$host, self::$username, self::$password);  
    if (!$link) {  
        throw new MySQLException("Connection failed: " . mysql_error());  
    }  
    mysql_select_db(self::$database, $link);  
    mysql_query("SET CHARACTER SET " . self::$charset);  
    mysql_query("SET NAMES " . self::$charset);  
    return $link;  
}
```

```
function use_unique($a, $b) {  
    $a = array_unique($a);  
    $b = array_unique($b);  
    $c = array_merge($a, $b);  
    $c = array_unique($c);  
    return $c;  
}  
  
function use_unique(array $a, array $b) {  
    $c = array_merge($a, $b);  
    $c = array_unique($c);  
    return $c;  
}  
  
function use_unique($a, $b) {  
    $c = array_merge($a, $b);  
    $c = array_unique($c);  
    return $c;  
}
```



# Пример работы программы



[http://s.mtudev.ru/  
a733183fe38beecfd1c988a0e73cd0f9.png](http://s.mtudev.ru/a733183fe38beecfd1c988a0e73cd0f9.png)



# Задачи

- ▶ Организовать отправку изображения на сервер
- ▶ Реализовать спектр настроек программы
- ▶ Сделать программу ненавязчивой с графическим интерфейсом



# Организация отправки изображения

- ▶ Шаг 1: перехват клавиши Print Screen
- ▶ Шаг 2: копирование содержимого на экране в ОП
- ▶ Шаг 3: отправка на сервер изображения
- ▶ Шаг 4: полученный положительный результат отправить в буфер обмена

```
private static async void PrtHooked(object sender, KeyEventArgs e)
{
    PrtScr_Hook.StopHook(PrtHooked); // Во время обработки приостанавливаем перехват
    json_st.ResponsePost Resp = new json_st.ResponsePost(); // Хранилище ответа
    try
    {
        byte[] Scr = Combine.GetScreen(); // Шаг 2: получаем изображение
        if (Settings.Save) SaveScr.Save(Scr); // Сохраняем при надобности на диск
        Resp = await Combine.SendScreen(Scr); // Шаг 3: отправка изображения
        Scr = null;
    }
    catch
    {
        Resp.message = "Error"; // Если не удалось отправить изображение
    }
    if (Resp.filename.Length > 4 && Resp.code != 415 && Resp.code != 0)
    { // Шаг 4: Если результат положительный, то отправить его в буфер обмена
        Clipboard_s.ToClipboard(Settings.HTTP_ADDRESS + "/" + Resp.filename);
    }
    GC.Collect(); // Вызываем сборщик мусора дважды
    PrtScr_Hook.StartHook(PrtHooked); // Возобновление шага 1
    GC.Collect();
}
```

# Спектр настроек программы

```
/// <summary>
/// Адрес как DNS или IP сервера
/// </summary>
public static string ADDRESS
{
    get
    {
        return cont.ADDRESS;
    }
    set
    {
        cont.ADDRESS = value;
        update();
    }
}
```

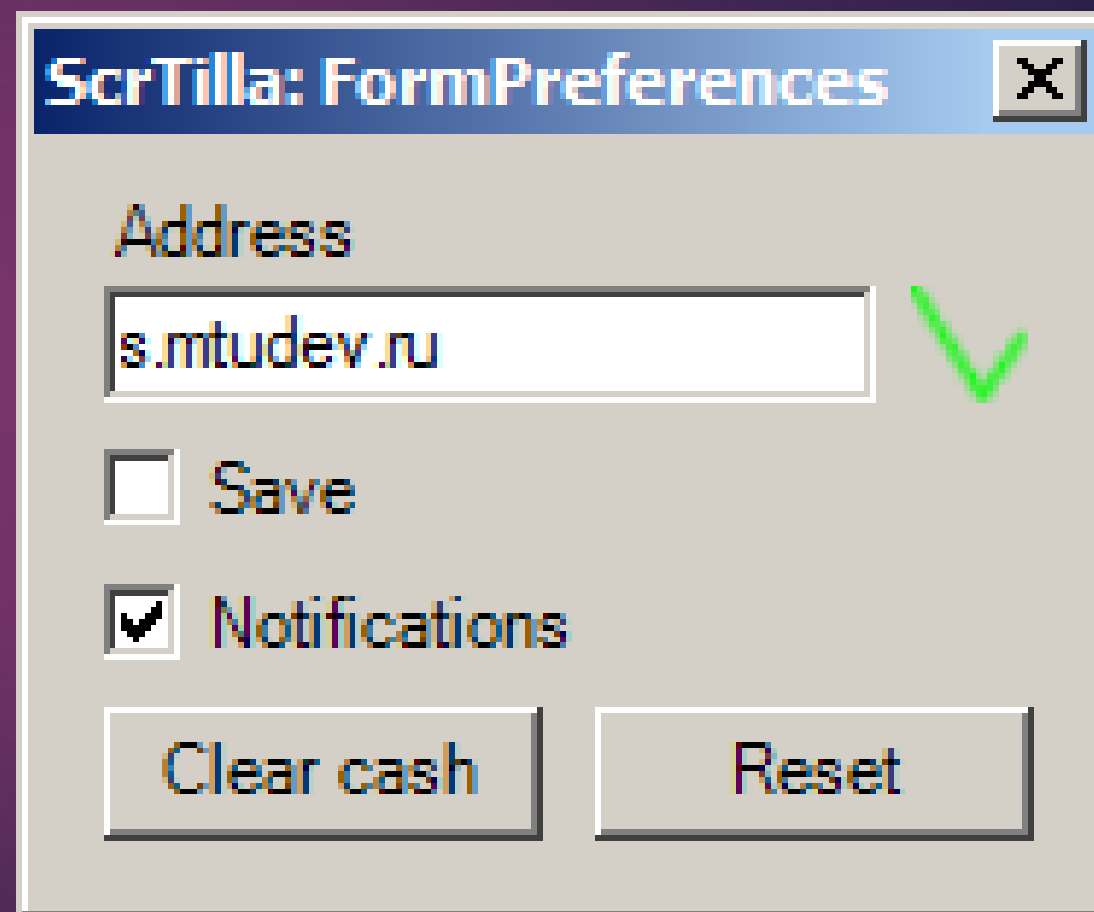
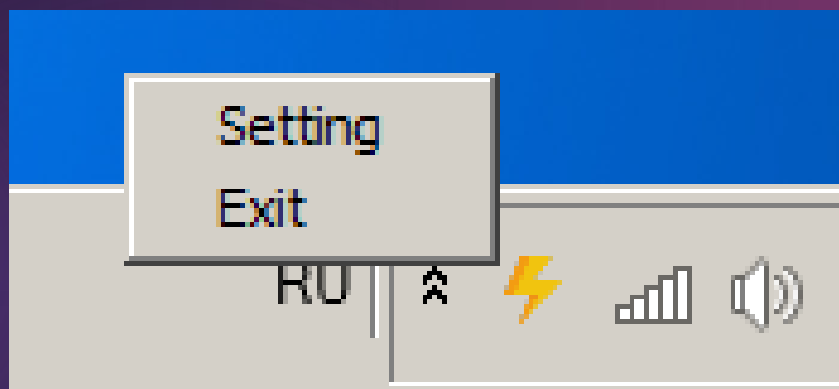


```
1 {
2     "UPLOAD": "http://s.mtudev.ru/upload",
3     "HTTP_ADDRESS": "http://s.mtudev.ru",
4     "PNGs": "http://s.mtudev.ru",
5     "ADDRESS": "s.mtudev.ru",
6     "Save": false,
7     "Notifications": true
8 }
```

# Спектр настроек программы

```
/// <summary>
/// Записать изменения в файл настроек
/// </summary>
private static void update()
{
    StreamWriter fp = null;
    try
    {
        using (fp = new StreamWriter(FILE_NAME, false, System.Text.Encoding.UTF8))
        { // Записать в файл настроек все настройки программы в JSON формате
            fp.Write(JsonConvert.SerializeObject(cont));
        }
    }
    catch { fp?.Close(); }
}
```

# Графический интерфейс программы





# Заключение

- ▶ В результате изучения стандартов сервера была написана соответствующая программа-клиент.
- ▶ Программа реализована на использовании асинхронных операций, http клиента, перехвате ввода/вывода клавиатуры, изображения экрана, буфера обмена.
- ▶ Был реализован следующий функционал (см. на сл. слайде)

# Заключение

- ▶ перехват клавиши Print Screen;
- ▶ получение сведений о сервере;
- ▶ отправка изображения экрана на сервер;
- ▶ отправка ссылки общего доступа в буфер обмена;
- ▶ резервное копирование отправляемых файлов;
- ▶ настройка адреса сервера, резервного копирования и уведомлений;
- ▶ оперативная работа с памятью: быстрое очищение ресурсов;
- ▶ обработка большого количества исключений;
- ▶ графический интерфейс.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- ▶ Орлов С.А., Программная инженерия. Технологии разработки программного обеспечения – 5-у изд., обновлённое и доп. – СПб.: Питер, 2016. – 640 с.
- ▶ Агуров П.В., С#. Сборник рецептов. – СПб.: БХВ-Петербург, 2007. – 432 с.: ил. ISBN 5-94157-969-1
- ▶ Алекс Дэвис, Асинхронное программирование в С# 5.0. / Пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2013. – 120 с.: ил. ISBN 978-5-94074-886-1
- ▶ <http://microsoft.com/>

# Последний слайд для заметок

- ▶ Программирование – это одна из интенсивно развивающихся областей в сфере информационных технологий и охватывает как теоретические вопросы, так и вопросы, непосредственно связанные с практикой. Изучаемая дисциплина основывается на использовании языка программирования C (Си).
- ▶ Для расширения, закрепления и систематизации теоретических знаний, полученных по дисциплине «Программирование» было сформировано задание для курсовой работы.
- ▶ Объектом исследования в данной работе является клиент сервера с базой данных изображений. Клиент состоит из перехвата средств ввода клавиатуры и вывода изображения, а также из отправителя, который отправляет изображение на сервер.
- ▶ Целью курсовой работы является разработка клиента-отправителя к серверу изображений на языке программирования C# (Си Шарп). Необходимо, чтобы программа умела перехватывать нажатие клавиши захвата изображения; считывать, кодировать и отправлять его; получать ссылку общего доступа к загруженному изображению через сервер. У программы должен быть простой, но удобный интерфейс взаимодействия с пользователем.