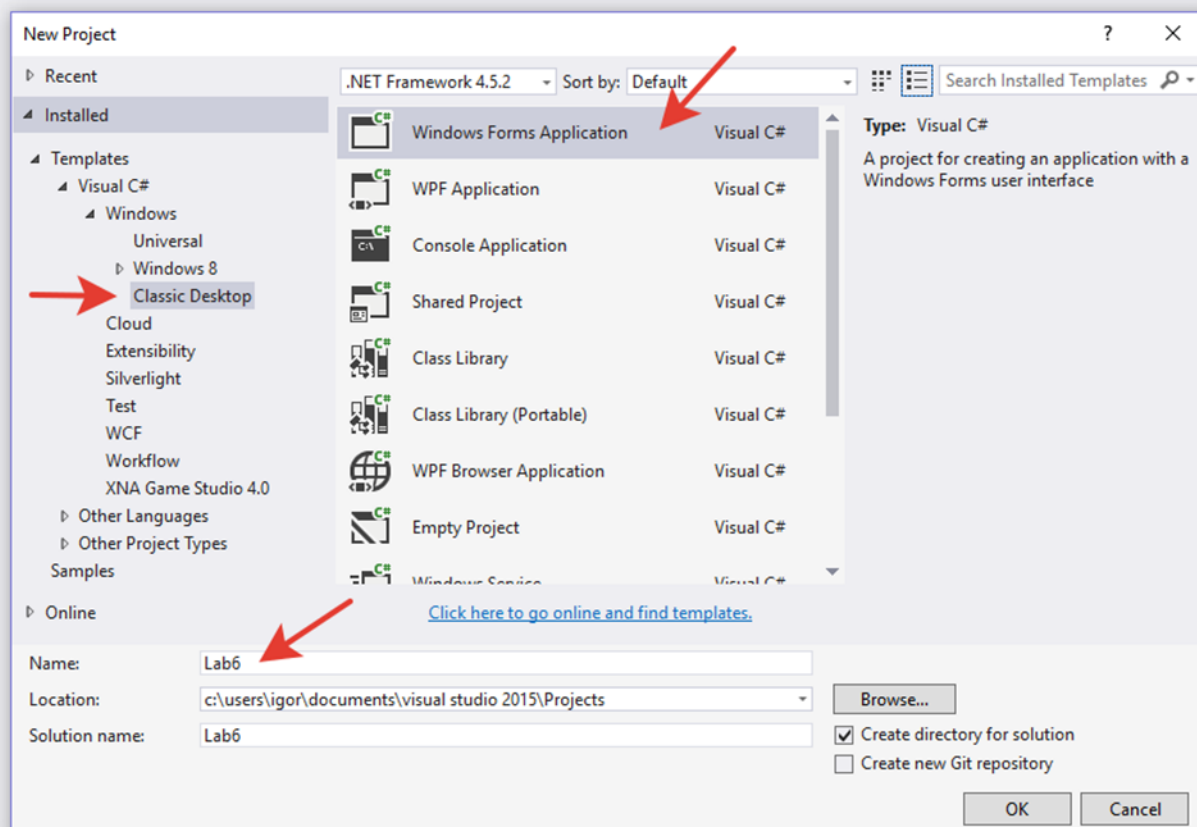


# ЛАБОРАТОРНАЯ РАБОТА №3

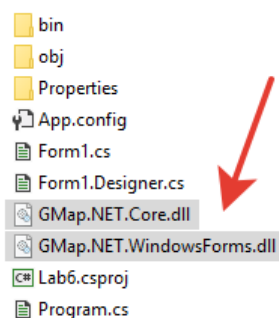
## ГИС СИСТЕМЫ

В лабораторной работе №6 нам потребуется специальный элемент управления – **GMap.NET**. Мы будем использовать его для добавления карты на форму, ее инициализации и для создания маркеров и полигонов.

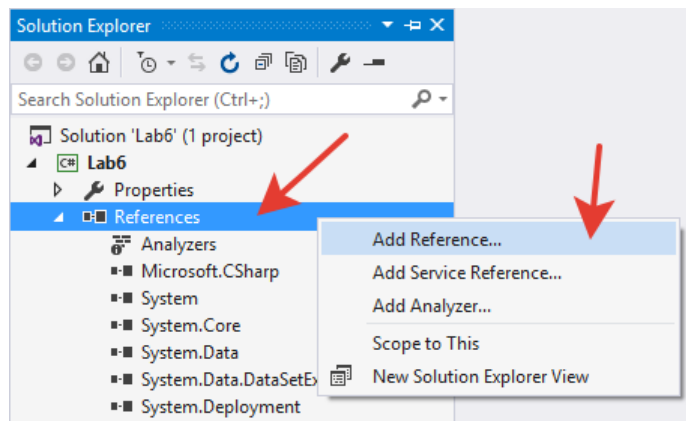
Для начала создайте новый проект **Windows Forms** с названием **Lab6**.



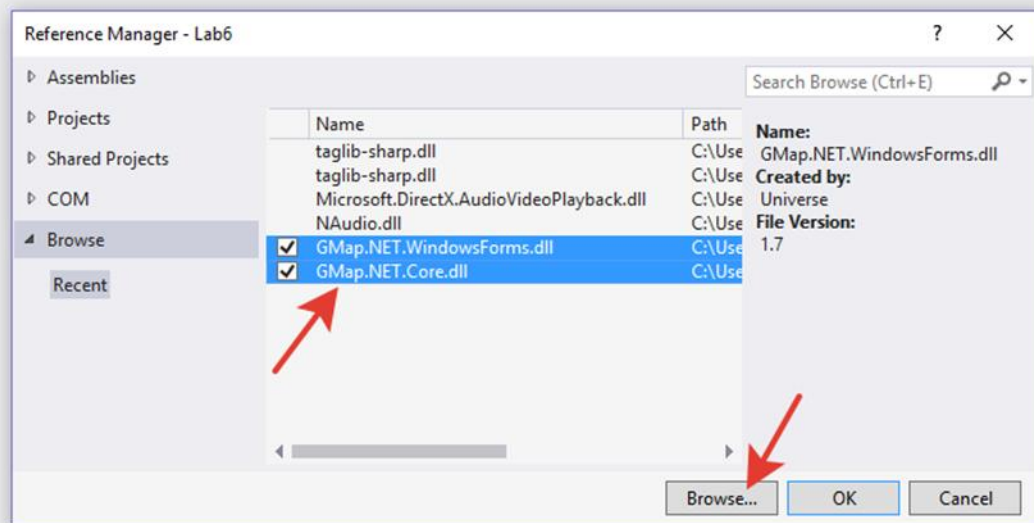
Компонент **GMap** размещается в двух **dll** библиотеках - **GMap.NET.Core.dll** и **GMap.NET.WindowsForms.dll**.



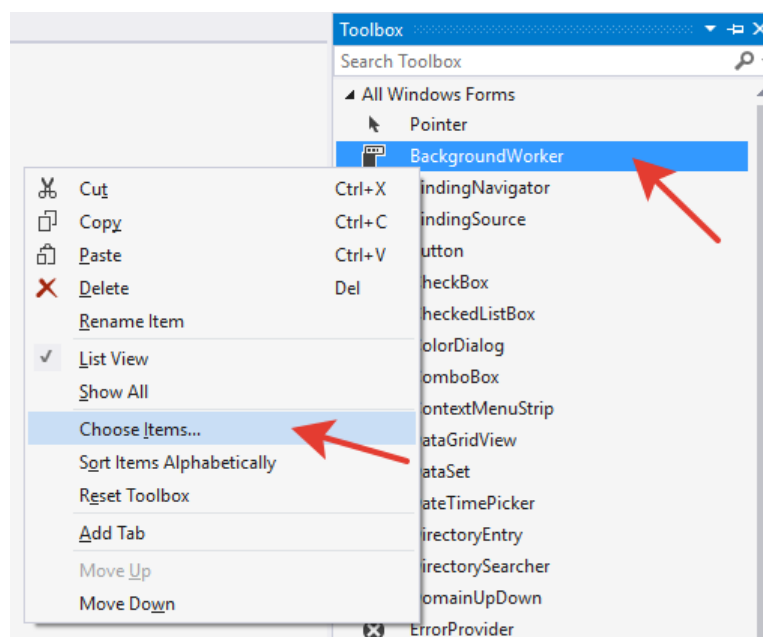
Скопируйте обе библиотеки в папку проекта и подключите их через узел ссылок проекта.



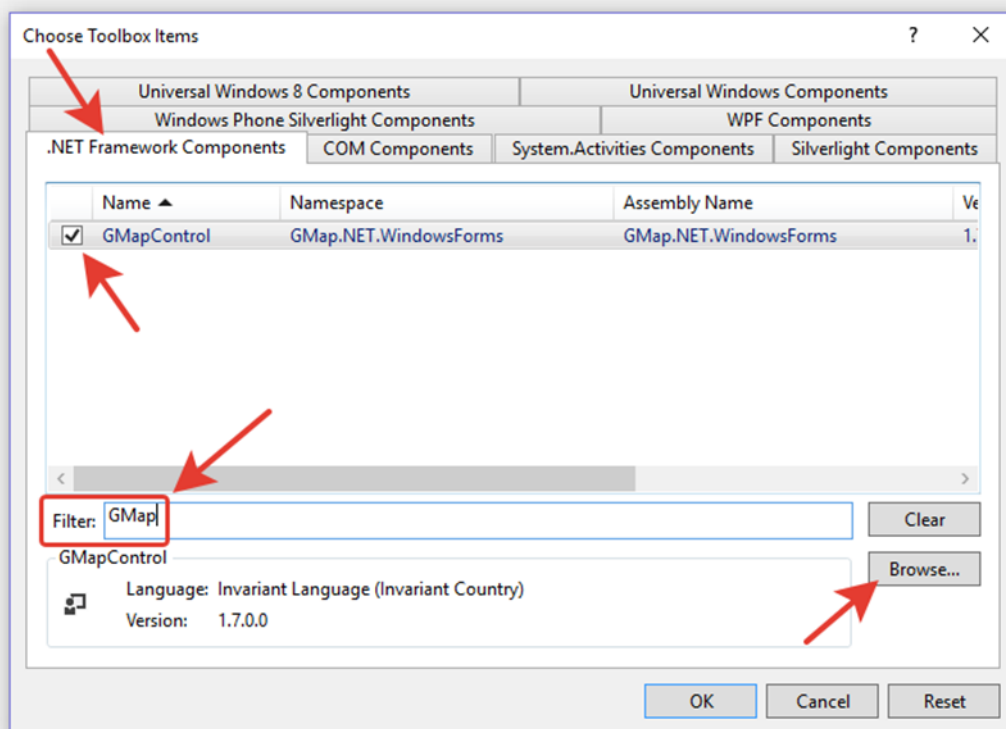
Используйте кнопку **Browse (Просмотр)** для поиска библиотек.



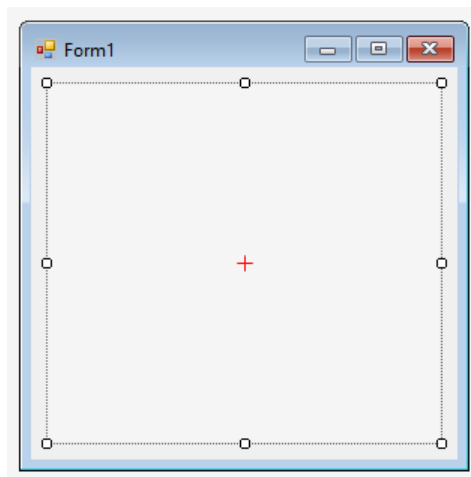
Теперь у нас появился доступ к классам карт. Мы можем добавить карту на нашу форму напрямую, создав объект нужного класса, но намного удобнее будет использовать элемент управления. Зайдите в конструктор форм и щелкните правой кнопкой мыши внутри **панели инструментов (Toolbox)**. Выберите пункт меню **Choose Items (Выбрать элементы)**.





В открывшемся диалоговом окне выберите закладку **.NET Framework Components (Компоненты .NET Framework)**. Нажмите кнопку **Browse (Просмотр)** и найдите библиотеку **GMap.NET.WindowsForms.dll**. После подключения библиотеки можно воспользоваться строкой фильтра для ее поиска в списке. В результате диалоговое окно должно выглядеть следующим образом:



Нажмите кнопку **OK**. На панели инструментов появился новый пункт – **GmapControl**. Перетащите его на форму и расположите по своему вкусу:



В свойствах этого элемента управления можно найти особый блок с названием **GMap.NET**:

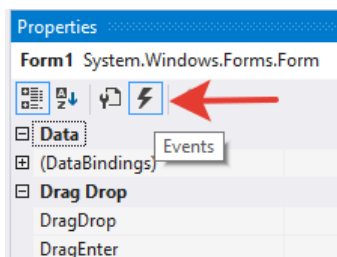
GMap.NET	
Bearing	0
CanDragMap	True
EmptyTileColor	 Navy
GrayScaleMode	False
MarkersEnabled	True
MaxZoom	2
MinZoom	2
MouseWheelZoomType	MousePositionAndCenter
NegativeMode	False
PolygonsEnabled	True
RoutesEnabled	True
ScaleMode	Integer
SelectedAreaFillColor	 33; 65; 105; 225
ShowTileGridLines	False
Zoom	0

Разберем некоторые из этих настроек более подробно.

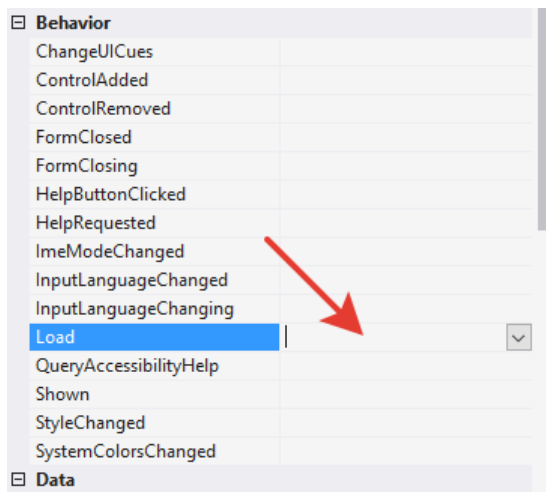
- **CanDragMap** – Если установлено значение True, то пользователь сможет передвигать карту при помощи правой кнопки мыши.
- **MarkersEnabled** – Если установлено значение True, то на карте будут отображаться все пользовательские маркеры. Установите это значение в True.
- **PolygonsEnabled** – Аналогично для полигонов. Установите в True.
- **ShowTileGridLines** – Если установлено значение True, то на карте будут показаны координаты тайлов. Может быть полезно для отладки.
- **Zoom, MinZoom, MaxZoom** – Параметры масштабирования. В случае карт Google максимальное приближение составляет 18 единиц, а минимальное будет 0. Установите текущее значение масштаба (**Zoom**) равным 5. Масштабирование осуществляется колесиком мыши.

Все эти настройки отвечают за визуальное представление карты, все остальное нам придется инициализировать при помощи кода.

В редакторе форм откройте свойства формы Form1. Переключитесь в режим отображения событий нажав на кнопку со значком молнии:



В разделе **Behavior (Поведение)** найдите событие **Load** и дважды щелкните в пустом поле справа от него. Конструктор создаст метод - обработчик события загрузки формы. Это событие вызывается после создания всех компонент на форме, поэтому внутри него мы можем спокойно прописать для них дополнительную инициализацию.



Новый метод называется **Form1\_Load**. Вставьте в него следующий код:

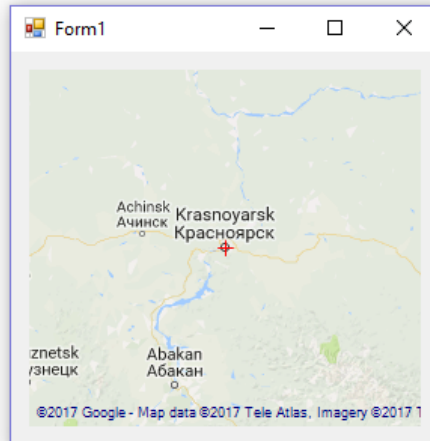
```
gMapControl1.MapProvider = GMap.NET.MapProviders.GoogleMapProvider.Instance;
GMap.NET.GMaps.Instance.Mode = GMap.NET.AccessMode.ServerOnly;
gMapControl1.SetPositionByKeywords("Krasnoyarsk, Russia");
```

Сначала мы указываем, какого поставщика карт мы будем использовать. **GMap** поддерживает множество различных видов карт, в лабораторной работе используются карты Google, но можно использовать, например, карты Bing.

Затем мы устанавливаем режим работы с картами. Возможных вариантов всего три: постоянно скачивать карты с сервера, скачивать карты и кэшировать их на диске и использовать только локальный кэш с сохраненной картой местности.

В конце мы указываем начальную точку, на которой будет центрированная наша карта.

Результат работы программы:



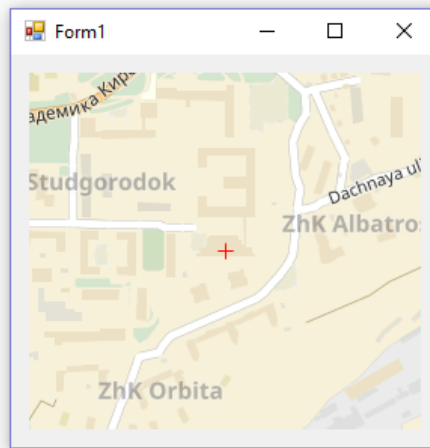
Вместо текстового представления начальной точки можно использовать широту и долготу:

```
gMapControl1.Position = new GMap.NET.PointLatLng(55.9943489195581, 92.7975721657276);
```

Заменяем поставщика карт на Яндекс и увеличим масштаб:

```
gMapControl1.MapProvider = GMap.NET.MapProviders.YandexMapProvider.Instance;
gMapControl1.Position = new GMap.NET.PointLatLng(55.9943489195581, 92.7975721657276);
gMapControl1.Zoom = 15;
```

Результат работы программы:



## ДОБАВЛЕНИЕ МАРКЕРОВ

Маркеры добавляются на карту при помощи **оверлеев (overlay)**. Оверлеи похожи на слои в графических редакторах. Один оверлей может содержать любое количество маркеров, включая и выключая оверлей можно одновременно включать и отключать расположенные в нем маркеры.

Для дальнейшего удобства подключим следующие пространства имен:

```
using GMap.NET;  
using GMap.NET.WindowsForms;  
using GMap.NET.WindowsForms.Markers;
```

Добавим в метод **Form1\_Load** следующий код:

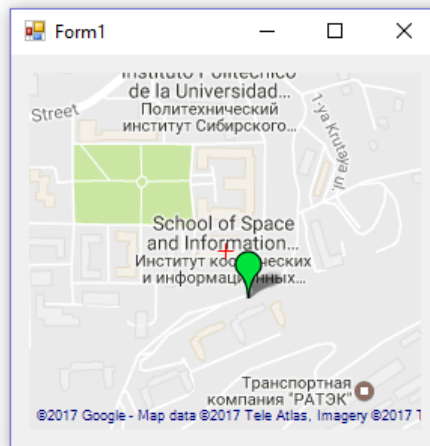
```
var markersOverlay = new GMapOverlay("markers");  
var marker = new GMarkerGoogle(  
    new PointLatLng(55.9943489195581, 92.7975721657276),  
    GMarkerGoogleType.green  
);  
markersOverlay.Markers.Add(marker);  
gMapControl1.Overlays.Add(markersOverlay);
```

Сначала мы создаем объект оверлея. Имя указывать не обязательно, но его наличие позволит в дальнейшем найти нужный оверлей через функцию поиска объекта карты.

Затем создается объект маркера. Конструктор маркера принимает два параметра – координаты маркера и его тип. При желании маркер можно инициализировать при помощи рисунка:

```
var marker = new GMarkerGoogle(  
    new PointLatLng(55.9943489195581, 92.7975721657276),  
    new Bitmap("c:\\images\\mymarker.png")  
);
```

Наконец, маркер добавляется в оверлей, а оверлей добавляется на карту. Результат работы программы:



## ДОБАВЛЕНИЕ ПОЛИГОНОВ

Полигон — многоугольник, минимальная поверхность для визуализации в трёхмерной графике.

Иногда маркеров бывает недостаточно для отображения нужной информации. Например, мы хотим отметить на карте место проведения строительных работ. Подобные задачи решаются при помощи полигонов.

Следующий код добавляет на карту новый полигон:

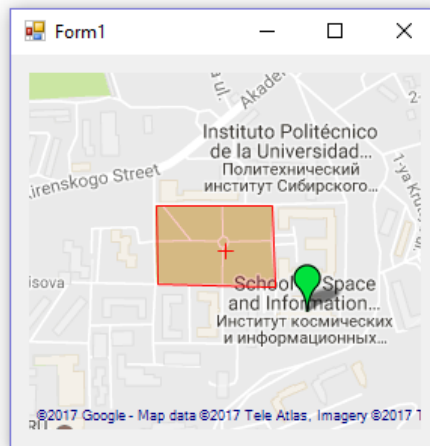
```
var polyOverlay = new GMapOverlay("polygons");
var points = new List<PointLatLng>();
points.Add(new PointLatLng(-25.969562, 32.585789));
points.Add(new PointLatLng(-25.966205, 32.588171));
points.Add(new PointLatLng(-25.968134, 32.591647));
points.Add(new PointLatLng(-25.971684, 32.589759));
var polygon = new GMapPolygon(points, "mypolygon");
polygon.Fill = new SolidBrush(Color.FromArgb(50, Color.Red));
polygon.Stroke = new Pen(Color.Red, 1);
polyOverlay.Polygons.Add(polygon);
gMapControl1.Overlays.Add(polyOverlay);
```

Правильные координаты полигона:

```
points.Add(new PointLatLng(55.996032, 92.793208));
points.Add(new PointLatLng(55.996050, 92.796673));
points.Add(new PointLatLng(55.994712, 92.796705));
points.Add(new PointLatLng(55.994754, 92.793261));
```

По аналогии с маркерами мы сначала создаем оверлей. Затем мы заполняем список точек, из которых будет состоять наш полигон. На основе этих точек мы создаем объект класса **GMapPolygon**. Чтобы полигон был хорошо различим, мы устанавливаем красный полупрозрачный цвет заполнения (**SolidBrush**), а цвет контура задаем тонкой красной линией (**Pen**). Наконец, мы добавляем полигон в оверлей, а оверлей на карту.

Результат работы программы:



Если полигон не отображается на карте, то попробуйте изменить масштаб при помощи колесика мыши.

#### ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ

1. В базе данных **aero** создайте таблицу для хранения координат маркеров. Вам понадобятся:
  - a. Индекс маркера. Не забудьте назначить этот столбец как первичный ключ с автоматическим приращением!
  - b. Два столбца для хранения координат маркера (широта и долгота). В качестве типа данных можно использовать float.
2. Заполните таблицу любыми корректными значениями.
3. Реализуйте в вашей программе отображение маркеров, содержащихся в базе данных.