

**Специальное программное обеспечение
«Набор координат объектов г. Москвы»**

Описание функциональных характеристик программного обеспечения и информация, необходимая для установки и эксплуатации программного обеспечения

Содержание

1. Общие сведения	3
2. Сопровождение СПО	3
3. Термины и определения	3
4. Функциональные возможности	4
5. Структура СПО	10
6. Требования к рабочему месту	11
7. Описание процесса подключения	13
8. Описание проверки работоспособности ПО	13
9. Описание процесса обновления ПО	13
10. Описание процесса удаления ПО	13

1. Общие сведения

Специальное программное обеспечение «Набор координат г. Москвы» (далее СПО) может быть использовано в навигационно-диспетчерских системах управления с целью решения задач по навигации и позиционированию объектов на мобильных рабочих местах.

СПО «Набор координат г. Москвы» используется в качестве отдельного модуля, встраиваемого в другое программное обеспечение.

СПО «Набор координат г. Москвы» может использоваться в составе автоматизированных мобильных рабочих мест, предназначенных для реализации задач, связанных с контролем местоположений удаленных или мобильных сотрудников, выбора мобильного сотрудника для выполнения задачи с учетом его местоположения.

СПО «Набор координат г. Москвы» также может применяться в системах, предназначенных для автоматизации деятельности как коммерческих, так и государственных организаций и учреждений, которым в силу своей деятельности приходится реализовывать функции диспетчерских служб или центров.

2. Сопровождение СПО

Для сопровождения пользователей и получения дополнительной информации следует обращаться по телефону +7 (495) 620-70-51) или электронной почте (info@digimap.ru).

3. Термины и определения

Определение/термин	Описание
ADT	Android Developer Tools- плагин для Eclipse
Eclipse	свободная интегрированная среда разработки модульных кроссплатформенных приложений
GPS	(англ. Global Positioning System — система глобального позиционирования, читается Джи Пи Эс) — спутниковая система навигации, обеспечивающая измерение расстояния, времени и определяющая местоположение во всемирной системе координат WGS 84

Определение/термин	Описание
ADT	Android Developer Tools- плагин для Eclipse
Eclipse	свободная интегрированная среда разработки модульных кроссплатформенных приложений
jar	(сокр. Java Archive)- представляет собой специальный формат для хранения компонентов Java-приложений
Java	язык программирования широкого назначения, используется в том числе для разработки программ под планшеты и смартфоны
POI	(англ. point of interest) –точки интереса пользователя
WGS	международная геодезическая система координат предназначена для ориентирования в мировом пространстве
ГЛОНАСС	Глобальная навигационная спутниковая система (ГЛОНАСС) — российская спутниковая система навигации
КЭШ	предварительно созданное картографические изображение
СПО	Специальное программное обеспечение
ПО	Программное обеспечение

4. Функциональные возможности

СПО «Набор координат г. Москвы» представляет собой программный интерфейс, в состав которого входят методы отображения экранной формы электронной карты, отображение текущего местоположения мобильного устройства (мобильного сотрудника), прокладка маршрута, а именно:

- отображение электронной карты г. Москвы, в том числе отображение:
 - ✓ точек адресных объектов;
 - ✓ границ административных районов;
 - ✓ границ муниципальных районов;

- ✓ границ населенных пунктов;
 - ✓ контуров зданий и сооружений;
 - ✓ мостов,
 - ✓ дорог,
 - ✓ границ топографических объектов (топооснова);
 - ✓ границ кварталов,
 - ✓ рек, водоемов,
 - ✓ островов,
 - ✓ железных дорог,
 - ✓ городских именованных площадей,
 - ✓ объектов транспорта,
 - ✓ точек интереса (POI).
- управление электронной картой местности: перемещение карты (4 направления), поворот карты (2 направления), изменения масштаба (2 направления).
 - поиск объектов карты по наименованию. При поиске объекта должны выполняться требования о том, что найденный объект должен содержать в себе информацию об иерархической подчиненности объектов.
 - поиск объектов карты по координатам. При поиске объекта должны выполняться требования о том, что найденный объект должен содержать в себе информацию об иерархической подчиненности.
 - построение и отображение на карте оптимального маршрута движения от текущего местоположения мобильного сотрудника (мобильного устройства) до заданной точки; При этом при прокладке маршрута должны быть соблюдены требования о том, что
 - ✓ в основе маршрута должен лежать граф дорожного движения;
 - ✓ граф должен быть представлен ребрами и узлами;
 - ✓ ребра и узлы должны быть связаны между собой;
 - ✓ маршрут должен прокладываться с учетом существующих запретов;
 - ✓ маршрут должен прокладываться с учетом вида транспорта, в том числе с учетом спецтранспорта.
 - получение данных текущего местоположения и параметров движения от ГЛОНАСС/GPS-приемника;
 - отображение данных текущего местоположения мобильного устройства (своего местоположения) и параметров движения на электронной карте;
 - отслеживание движения по маршруту (своего местоположения) и выполнение функций навигации;
 - предоставление пользователю информации с картографического слоя «POI», в случае если такой слой будет предоставлен Заказчиком;

- обеспечение возможности создания пользователем собственного картографического слоя.

4.1. Описание методов библиотеки

```
// Метод инициализации (загрузки ресурсных файлов)
bool __fastcall BindToFiles(TRouteOptimType rotRouteOptimType);

// Метод привязки библиотеке к экранной форме, на которой будет
отрисовываться электронная карта
void __fastcall BindToWnd(int iWidth, int iHeight);

// Преобразование координат: метры в экранные пиксели
void __fastcall MapToScr(int iMapX, int iMapY, int& iScrX, int& iScrY, bool&
bInScreen);

// Преобразование координат: метры в экранные пиксели (2)
void __fastcall MapToScr_(int iMapX, int iMapY, int& iScrX, int& iScrY, short
sAlphaAngle, int iZoom);

// Преобразование координат: экранные пиксели в метры
void __fastcall ScrToMap(int iScrX, int iScrY, int& iMapX, int& iMapY);

// Преобразование координат: экранные пиксели в метры (2)
void __fastcall ScrToMap_(int iScrX, int iScrY, int& iMapX, int& iMapY, short
sAlphaAngle, int iZoom);

// Преобразование координат: экранные пиксели в метры (на плоскости с учетом
наклона электронной карты)
void __fastcall ScrToMapFlat(unsigned int iZoomDescriptor, int iMapCenterX,
int iMapCenterY, int iScrX, int iScrY, int& iMapX, int& iMapY);

// Преобразование координат: WGS в метры
void __fastcall WGSToMeter(const double& dLongitude, const double& dLatitude,
TAngleMeasure amAngleMeasure, int& iX, int& iY);

// Преобразование координат: метры в WGS
void __fastcall MeterToWGS(int iX, int iY, TAngleMeasure amAngleMeasure,
double& dLongitude, double& dLatitude);

// Установка масштаба
bool __fastcall SetZoom(int iZoomDescriptor);

// Получение текущей установки масштаба
int __fastcall GetZoom();

// Получение текущей установки масштаба в соответствии с текущим номером
уровня
int __fastcall GetZoomValue();

// Получение масштаба в соответствии с указанным номером уровня
int __fastcall GetZoomValue(int iZoomDescriptor);

// Получение текстового наименования уровня
bool __fastcall GetZoomDescription(int iZoomDescriptor, int& iZoom, wchar_t*&
pwcZoomValue);

// Получение текстового наименования уровня (2)
void __fastcall GetZoomDescription(int& iZoom, wchar_t*& pwcZoomValue);
```

СПО «Набор координат объектов г. Москвы»,
ЗАО «Геоцентр-Консалтинг»

```
// Установка текущего угла поворота карты
bool __fastcall SetAlphaAngle(short sAlphaAngle);

// Получение текущей установки угла поворота карты
short __fastcall GetAlphaAngle();

// Установка текущего центра карты
void __fastcall SetCenter(const POINT& pCenter);

// Установка текущих центра карты и угла поворота карты
bool __fastcall SetCenterAndAngle(const POINT& pCenter, short sAlphaAngle);

// Установка текущих центра карты и масштаба
bool __fastcall SetCenterAndZoom(const POINT& pCenter, int iZoomDescriptor);

// Установка текущих центра карты, масштаба и режима отображения карты (3D
или 2D)
bool __fastcall SetCenterZoomMapMode(const POINT& pCenter, int
iZoomDescriptor, TMapMode mmMapMode);

// Загрузка объектов карты
void __fastcall ReloadObjects();

// Получение текущего центра карты
POINT __fastcall GetCenter();

// Получение габаритных размеров электронной карты
void __fastcall GetMapDimentions(int& iMinX, int& iMinY, int& iMaxX, int&
iMaxY, int& iZoomMin, int& iZoomMax);

// Получение дескриптора окна, на котором выполняется отрисовка электронной
карты
HDC __fastcall GetDC();

// Перерисовать карту
void __fastcall ReDraw();

// Перерисовать карту с максимальной производительностью (перерисовать только
контуры объектов)
void __fastcall ReDrawFast();

// Добавить выделенный объект
void __fastcall InsertSelObject(STK_HANDLE hObject);

// Очистить коллекцию выделенных объектов
void __fastcall ClearSelObjects();

// Поиск объекта по координатам на всех уровнях карты
TSTKError __fastcall FindObject(int iX, int iY, STK_HANDLE& hObject);

// Поиск объекта по координатам только на уровне, который соответствует
указанному масштабу
TSTKError __fastcall FindObject(int iX, int iY, int iZoom, STK_HANDLE&
hObject);

// Получение атрибута объекта карты
TSTKError __fastcall GetObjectInfoField(STK_HANDLE hObject, TObjectInfoField
oifInfoField, void* pvBuf, unsigned int iBufSize);

// Очистить маршрут
void __fastcall RouteClear();
```

СПО «Набор координат объектов г. Москвы»,
ЗАО «Геоцентр-Консалтинг»

```
// Проверка, идет ли в данный момент расчет маршрута (в асинхронном режиме)
bool __fastcall RouteTracingInProgress() const;

// Проверка, идет ли в данный момент подготовка данных, используемых для
расчета маршрутов
bool __fastcall CreateDeykstraInProgress() const;

// Построить маршрут
TSTKError __fastcall TraceRoute(const wchar_t* pwcUserPoint1, const POINT&
pP1, const wchar_t* pwcUserPoint2, const POINT& pP2, TTraceRouteCallback
pfCallback, void* pvUserPar);

// Перестроить маршрут
TSTKError __fastcall ReTraceRoute(const POINT& pBeg, TTraceRouteCallback
pfCallback, void* pvUserPar);

// Проверка, построен ли маршрут
bool __fastcall IsRouteTraced();

// Получить точки узлов маршрута
const std::deque<TRoutePoint>& __fastcall GetRoutePoints();

// Получить ближайшую к заданной точке секцию маршрута
TSTKError __fastcall GetNearestRouteSection(int iX, int iY, int iCourse, int
iSpeed, TRoutePoint*& pRoutePoint, TRoutePoint*& pNextPoint);

// Получить указатель на буфер экрана, на котором отрисована электронная
карта
BYTE* __fastcall GetScreenBuffer();

// Получить размер буфера экрана, на котором отрисована электронная карта
unsigned int __fastcall GetScreenBufferBMPSize();

// Сохранить в формат BMP экран, на котором отрисована электронная карта
void __fastcall SaveBMPScreenBuffer(void* pvBuffer);

// Изменить режим отображения рубрики объектов POI
void __fastcall SwitchPOI(unsigned int iRubric, bool bOnOff);

// Отменить отрисовку POI
void __fastcall ClearPOI();

// Получить список рубрик POI
void __fastcall GetPOIRubrics(unsigned int iParent, std::deque<TPOIRubric>&
dequePOIRubrics);

// Получить список подрубрик POI
void __fastcall GetChildRubrics(unsigned int iParent, std::deque<unsigned
int>& dequeRubrics);

// Получить наименование рубрики POI
const wchar_t* __fastcall GetPOIRubric(unsigned int iRubric);

// Добавить пользовательскую точку
TSTKError __fastcall SaveUserPoint(const wchar_t* pwcName, int iMapX, int
iMapY, int iAttr, const wchar_t* pwcId);

// Удалить пользовательскую точку
TSTKError __fastcall RemoveUserPoint(const wchar_t* pwcId);

// Очистить пользовательские точки
void __fastcall ClearUserPoints();
```


СПО «Набор координат объектов г. Москвы»,
ЗАО «Геоцентр-Консалтинг»

```
// Получить габариты объекта карты
TSTKError __fastcall GetObjectGabarits(STK_HANDLE hObject, RECT& rGabarit,
POINT& pCenter);

// Получить текущую установку режима оптимизации алгоритма построения
маршрута
void __fastcall SetRouteOptimType(TRouteOptimType rotRouteOptimType);

// Создать снимок экрана, на котором отрисована электронная карта
bool __fastcall CreateScreenShot(const wchar_t* pwcFileName);

// Получить текущую установку режима карты (3D или 2D)
TMapMode __fastcall GetMapMode();

// Установка текущего режима карты (3D или 2D)
void __fastcall SetMapMode(TMapMode mmMapMode);

// Очистить КЭШ
void __fastcall FlushMappedFiles();

// Установка текущего языка карты
void __fastcall SetMapLanguage(unsigned int iValue);

// Получить текущую установку языка карты
unsigned int __fastcall GetMapLanguage();

// Добавить точку трека
void __fastcall TrackInsertPoint(const POINT& pPoint);

// Очистить трек
void __fastcall TrackClear();

// Поиск рубрики POI
void __fastcall FindPOIRubric(wchar_t* pwcSubString,
std::deque<TFindPOIRubricResult>& dequeResult);

// Поиск объекта по наименованию
void __fastcall FindObject(TFindObjectType fotType, wchar_t* pwcSubString,
STK_HANDLE hObject, std::deque<TFindObjectResult>& dequeResult, unsigned int
iMaxResultCount);

// Поиск POI рядом с указанной точкой
TSTKError __fastcall FindPOIByPoint(wchar_t* pwcSubString, short sRubric, int
iX, int iY, int iRadius, std::deque<TFindObjectResult>& dequeResult, unsigned
int iMaxResultCount);

// Поиск POI вдоль маршрута
TSTKError __fastcall FindPOIByRoute(wchar_t* pwcSubString, short sRubric, int
iRadius, std::deque<TFindObjectResult>& dequeResult, unsigned int
iMaxResultCount);

// Поиск POI по наименованию
void __fastcall FindPOI(wchar_t* pwcSubString, short sRubric,
std::deque<TFindObjectResult>& dequeResult, unsigned int iMaxResultCount);

// Поиск объекта по наименованию (2)
void __fastcall FindGeneral(wchar_t* pwcSubString,
std::deque<TFindObjectResult>& dequeResult, unsigned int iMaxResultCount);

// Получение версии карты
void __fastcall GetMapVersion(WORD* pwBuffer);
```

```
// Установка данных трафика (данных пробок)
void __fastcall InstallTrafficData(TSTKTrafficData* pTData, unsigned int
iCount, TInstallTrafficDataCallback pfInstallTrafficDataCallback, void*
pvUserPar);

// Поиск POI рядом с указанным зданием
TSTKError __fastcall FindPOIByBuilding(STK_HANDLE hObject,
std::deque<STK_HANDLE>& dequePOI);

// Получение параметров расчета маршрута
void __fastcall GetRouteParams(TRouteParams& rpValue);

// Установка параметров расчета маршрута
void __fastcall SetRouteParams(const TRouteParams& rpValue);

// Установка угла наклона карты в 3D режиме
bool __fastcall FiAngleInc();

// Получение текущей установки угла наклона карты в 3D режиме
bool __fastcall FiAngleDec();
```

5. Структура СПО

СПО «Набор координат г. Москвы» представляет собой библиотеку программного кода, которая подключается в состав программного обеспечения мобильного рабочего места и является библиотекой программного кода языка программирования Java (формат jar), разработанной с помощью среды Eclipse с надстройкой ADT (Android Developer Tools).

Библиотека используется при сборке ПО для мобильных рабочих мест.

Для того чтобы работать с картографической информацией при помощи этой библиотеки, в составе мобильного рабочего места должна быть предусмотрена экранная форма для отображения карты и местоположения удаленных/мобильных сотрудников.

СПО «Набор координат г. Москвы» содержит в себе все необходимые для работы картографические данные и связанные с ними справочные таблицы в специальном формате, который обеспечивает возможность выполнения заявленного функционала и не требует подключения сторонних источников картографических данных и наличия интернета для отображения картографических данных.

СПО «Набор координат г. Москвы» передается заказчику посредством электронной связи.

Таблица 1. Состав СПО «Набор координат г. Москвы»

№ п/п	Наименование файла	Описание файла
-------	--------------------	----------------

№ п/п	Наименование файла	Описание файла
1	stk.bin	Данные атрибутов объектов электронной карты
2	stk.elc	Данные пространственного индекса
3	stk.hst	Данные дерева иерархии объектов электронной карты
4	stk.ils	Данные индекса дерева иерархии объектов электронной карты
5	stk.iop	Данные индекса списка наименований объектов электронной карты
6	Stk.jar - Executable Jar File	библиотекой программного кода языка программирования Java
7	stk.mop	Координаты точек контуров объектов объектов электронной карты
8	stk.tls	Данные наименований объектов электронной карты
9	stk_1.net	Данные узлов графа транспортной сети
10	stk_2.net	Данные ребер графа транспортной сети
11	stk_3.net	Таблицы связей ребер графа транспортной сети
12	license	Лицензия

6. Требования к рабочему месту

Специальных требований к рабочему месту не предъявляется, так как СПО «Набор координат г. Москвы» предназначено для использования в составе стороннего программного обеспечения.

Мобильное рабочее место, которое предназначено для решения задач по навигации и позиционированию, должно отвечать следующим требованиям:

Таблица 2. Технические характеристики мобильного устройства

№ п/п	Параметр	Значение

№ п/п	Параметр	Значение
1	Операционная система	Android 5.0 и выше
2	Частота работы процессора	не менее 1200 МГц
3	Количество ядер процессора	не менее 4
4	Оперативная память	не менее 1,5 Гб
5	Встроенная память	не менее 16Гб
6	Поддержка карт памяти	до 64Гб
7	Размер экрана по диагонали	8 дюймов
8	Разрешение экрана	1280x800 пикселей
9	Тип экрана	TFT, сенсорный
10	Поддержка Wi-Fi	Wi-Fi 802.11 a/b/g/n
11	Мобильная связь	LTE cat4, 3G, EDGE, HSDPA, GPRS, GSM900, GSM1800
12	Подключение к компьютеру	USB
13	Спутниковая навигация	GPS, ГЛОНАСС
14	Тыловая камера	разрешением не менее 3 млн. пикселей
15	Фронтальная камера	разрешением менее 1,3 млн. пикселей
16	Класс пылевлагозащищённости	IP-67, IP-68
17	Ёмкость аккумулятора	не менее 4000 мАч
18	Класс устойчивости к внешним воздействиям	MIL-STD 810G

7. Описание процесса подключения

Для подключения необходимо осуществить перенос (копирование) архивного файла jar в требуемую директорию и разархивировать его.

Подключение библиотеки (jar) в состав стороннего программного обеспечения производится стандартными способами.

8. Описание проверки работоспособности СПО

Проверка работоспособности производится путем обращения к функциям картографии с использованием функционала стороннего программного приложения.

9. Описание процесса обновления СПО

Обновление специального программного обеспечения выполняется путем обновления файлов, установленных на мобильных рабочих местах.

10. Описание процесса удаления СПО

Удаление специального программного обеспечения выполняется путем удаления файлов, установленных на мобильных рабочих местах.