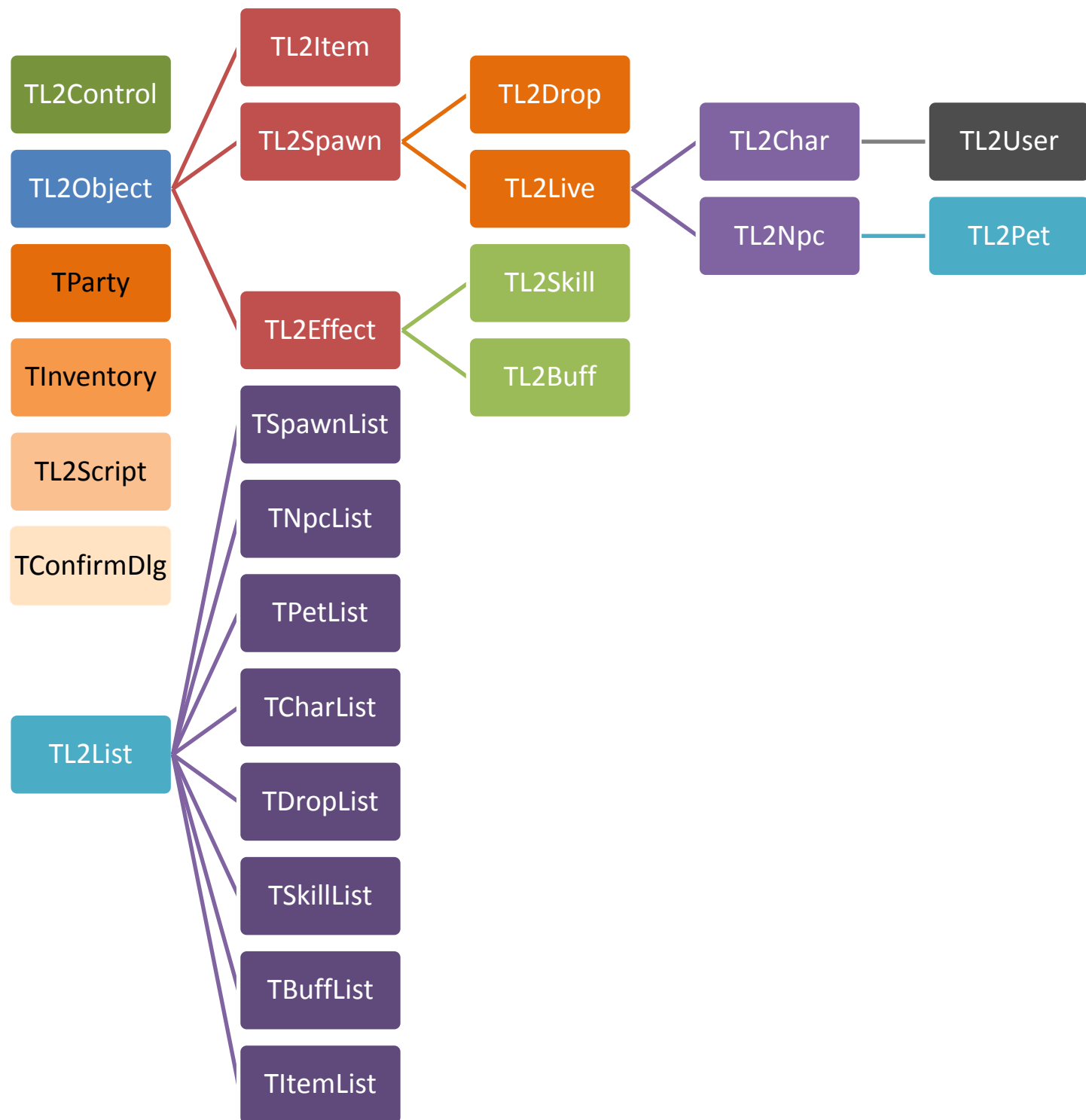


Иерархия классов



TL2Control = class

TL2Control = class	Основной класс управления ботом. В скриптах доступен по имени Engine. Можно получить для другого окна GetControl(const Name: string): TL2Control;
Assist(const Name: string): Boolean;	Делает своей целью, цель указанного игрока или NPC
Attack(TimeOut: Cardinal = 2000; Ctrl: Boolean = false): Boolean;	Атака. TimeOut - задержка (мс); Ctrl - принудительная атака
AutoSoulShot(const Name: string; Active: Boolean): Boolean;	Включает/выключает на шотах автоматический режим. Name - название шотов; Active - вкл/откл
AutoTarget(Range: Cardinal = 2000; ZRange: Cardinal = 300; NotBusy: Boolean = true): Boolean;	Авто-подбор цели в радиусе Range. NotBusy - выбирать только "свободную" цель. (учитывает зоны на карте и объекты добавленные в IgnorList)
BlinkWindow(GameWindow: Boolean): Boolean;	Мигает окном на панели задач. Если GameWindow - true то мигает окном игры, иначе окном бота
BotWindow: Cardinal;	Получить HWND окна с ботом.
ByPassToServer(const S : String) : Boolean;	
CancelTarget: Boolean;	Сбрасывает таргет
ClearIgnore;	Очищает список игнора
ClearZone;	Очистка всех зон на карте.
CrystalItem(ID : Cardinal) : Boolean;	Кристаллизировать итем.
ConfirmDialog(Accept: Boolean): Boolean;	Отвечает на запросы Да/Нет
ConfirmDlg : TConfirmDlg;	Получить класс TConfirmDlg;
DestroyItem(const Name: string; Count: Cardinal): Boolean;	Уничтожает предмет с названием Name в количестве Count
DismissParty(const Name: string): Boolean;	Исключает игрока с именем Name из группы
DismissPet: Boolean;	Отзывает пета (если есть)
DismissSum: Boolean;	Отзывает самона (если есть)
Dispel(const Name: string): Boolean;	Снимает с вашего персонажа баф с названием Name
DlgOpen: Boolean;	Начинает диалог с NPC
DlgSel(const Txt: string; const TimeOut : Integer = 1000): Boolean; overload;	Выбирает при диалоге строку Txt
DlgSel(Index: integer; const TimeOut : Integer = 1000): Boolean; overload;	Выбирает при диалоге строку с порядковым номером Index. TimeOut время в мс сколько ожидать диалога.
DlgText: string;	Содержит полный текст текущего диалога
DMoveTo(x, y, z : Integer) : Boolean;	Двигаться в указанную точку без ожидания завершения.
DUseSkill(id : Cardinal; ctrl, Shift : Boolean) : Boolean;	Использовать скил с указанным id без проверки на откат, количество мп.
Entry(var Param): Boolean;	Вызов функции в скрипте другого аккаунта. Вызываемая ф-ция должна иметь вид - function OnEntry(var Param): Boolean; Param - любой передаваемый параметр.
EnterText(const Txt: string): Boolean;	Нажать Enter, написать Txt, нажать Enter.
Equipped(const Name: string): Integer;	Проверка экипировки
FaceControl(ID: Integer; Active: Boolean): Boolean;	Вкл/откл клавишу интерфейса.
FindEnemy(var Enemy: TL2Live; Obj: TL2Live; Range: Cardinal = 2000; ZRange: Cardinal = 300): Boolean;	Поиск "врага" для объекта Obj в указанном радиусе (относительно объекта Obj). Если результат ф-ции - true, то найденный "враг" будет записан в переменную Enemy.
GameClose: Boolean;	Закрывает игру (клиент L2)
GameStart(CharIndex: Integer = -1): Boolean;	Заводит персонажа под номером CharIndex в игру (должны находиться на панели выбора персонажей), без параметров при -1, зайдет на последнего активного персонажа
GameWindow : Cardinal;	Получить HWND окна с игрой.
GetUser: TL2User;	Получение объекта User (TL2User) из другого аккаунта.
GetParty: TParty;	
GetSkillList: TSkillList;	
GetInventory: TInventory;	
GetDropList: TDropList;	
GetNpcList: TNpcList;	
GetCharList: TCharList;	
GetPetList: TPetList;	

FindPath(StartX, StartY, EndX, EndY: Integer; PathList: TList): Boolean;	Расчитывает путь из Startx,y До Endx,y и помещает точки в PathList подряд.
GetFaceState(ID: Integer): Boolean;	Узнать статус кнопки интерфейса. (FaceControl)
GoHome(ResType: TRestartType = rtTown) : Boolean;	Возвращает персонажа в город после смерти
Ignore(Obj: TL2Spawn);	Добавляет объект в список игнора. Методы AutoTarget и AutoPickup пропускают такие объекты
InviteParty(const Name: string; Loot: TLootType = ldLooter): Boolean;	Приглашает в группу игрока с именем Name. Loot - тип распределения дропа в группе
InZone(X, Y, Z: Integer): Boolean; overload;	Точка находится в зоне?
InZone(Obj: TL2Spawn): Boolean; overload;	Объект находится в зоне?
IsBusy(Obj: TL2Npc): Boolean;	Проверяет объект на "занятость" другими игроками
IsDay: Boolean;	День в игре?
JoinParty(Join: Boolean): Boolean;	Отвечает на приглашение в группу
LearnSkill(ID: Cardinal): Boolean;	Учит скил по ID. В HighFive и ниже должны находиться возле тренера
LeaveParty: Boolean;	Покидает группу
LoadZone(const Name: string): Boolean;	Загрузка зоны на карту из файла.
LoadConfig(const Name: string): Boolean;	Загрузка конфига с именем Name. По умолчанию из папки Settings, можно указать полный путь.
MakeItem(Index : Cardinal) : Boolean;	скрафтить предмет, Index из списка крафта берется.
MoveItem(const Name: string; Count: Cardinal; ToPet: Boolean): Boolean;	Передает/забирает предмет с названием Name у пета в количестве Count
MoveTo(ToX: integer; ToY: integer; ToZ: integer; const Timeout : Integer = 8000): Boolean; overload;	Двигаться в точку. Timeout время в мс сколько пытаться двигаться в точку
MoveTo(Obj: TL2Spawn; Dist: Integer): Boolean; overload;	Подойти к объекту Obj на дистанцию Dist
MoveToTarget(Dist: Integer = -100): Boolean;	Подойти к цели на дистанцию Dist
MSG(Who, What : String; Color : Integer);	Написать системное сообщение в окне бота, различного цвета.
NpcExchange(ID: Cardinal; Count: Cardinal): Boolean;	Обмен вещей у NPC. ID - id вещи которую хотим получить, Count - в каком количестве.
NpcTrade(Sell: Boolean; items: array of Cardinal): Boolean;	Торговать с Npc. Sell - покупка/продажа. Items - массив предметов для покупки/продажи. Массив должен состоять из пар ID, Count (кратен двум)
OpenQuestion: Boolean;	Открывает "знак вопроса" (требуется для некоторых квестов)
Pickup(Range: Cardinal = 250; ZRange: Cardinal = 150; OnlyMy: Boolean = false; Pet: Boolean = false): Integer; overload;	Авто-подбор всего дропа в радиусе Range. OnlyMy - Свой или весь дроп. Pet - Подбирает петом
Pickup(Obj: TL2Drop; Pet: Boolean = false): Boolean; overload;	Подбирает объект Obj. Pet - Подбирает петом
QuestStatus(QuestID: Cardinal; Step: Integer): Boolean;	Проверка выполнен шаг квеста или нет. Step - интересующий шаг квеста.
Restart: Boolean;	Выходит на панель выбора персонажей (чар не должен находиться в режиме боя)
Say(const Text: string; ChatType: Cardinal = 0; const Nick: string = ''): Boolean;	Написать в чат.
SendMail(const Recipient: string; const Theme: string; const Content: string; Items: array of Cardinal; Price: Cardinal = 0): Boolean;	Отправка почты. Recipient - адресат; Theme - тема; Content - содержание; Items - список (массив) прикрепленных вещей (ID/Кол-во); Price - цена (если указана считается "Безопасная сделка" иначе "Простая отправка"). Если вещь с отправляемым ID не найдена у персонажа, она будет автоматически исключена из отправки (отправка не срывается). Если при отправке какой либо вещи указанное кол-во превышает реальное, оно будет автоматически исправлено.
ServerTime: Cardinal;	
GameTime: Cardinal;	
SetPartyLeader(const Name: string): Boolean;	Передает лидерство в группе игроку с именем Name (ваш персонаж должен быть лидером группы)
SetTarget(ID: Cardinal): Boolean; overload;	Взятие цели по ID.

SetTarget(const Name: string): Boolean; overload;	Взятие цели по имени.
SetTarget(Obj: TL2Live): Boolean; overload;	Взятие объекта Obj в качестве цели.
Sit: Boolean;	Сесть
Stand: Boolean;	Встать
StopCasting : Boolean;	Прервать чтение заклинания.
Status: TL2Status;	Текущий статус аккаунта
Unstuck : Boolean;	Сделать Unstuck;
UpdateSkillList: Boolean;	Открыть скил лист, для IL серверов.
UseAction(ID: Cardinal; Ctrl: Boolean = false; Shift: Boolean = false): Boolean;	Использование игровых действий.
UseItem(ID: Cardinal; Pet: Boolean = false): Boolean; overload;	Использует предмет по ID. Pet - использует питом
UseItem(const Name: string; Pet: Boolean = false): Boolean; overload;	Использует предмет по имени. Pet - использует питом
UseItem(Obj: TL2Item; Pet: Boolean = false): Boolean; overload;	Использует предмет Obj. Pet - использует питом
UseKey(Key: Word; Ctrl: Boolean = False; Shift: Boolean = False): Boolean; overload;	Нажать кнопку. Зажимая Ctrl, Shift
UseKey(const Key: string; Ctrl: Boolean = False; Shift: Boolean = False): Boolean; overload;	
UseSkill(ID: Cardinal; Ctrl: Boolean = false; Shift: Boolean = false): Boolean; overload;	Использует скил по ID
UseSkill(const Name: string; Ctrl: Boolean = false; Shift: Boolean = false): Boolean; overload;	Использует скил по имени
WaitAction(Actions: TL2Actions; var P1; var P2; TimeOut: Cardinal = INFINITE): TL2Action;	Ожидание события или группы событий

TL2Object = class

TL2Object = class	Базовый класс всех игровых объектов
ID: Cardinal;	ID объекта
Name: String;	Имя объекта
OID: Cardinal;	Уникальный идентификатор для любого объекта в игре.
Valid: Boolean;	Проверка объекта на существование в игре (актуальность)
SetVar(Value: Cardinal);	назначить объекту переменную
GetVar: Cardinal;	получить значение переменной
L2Class: TL2Class;	Узнать класс к которому относится данный объект.

TL2Spawn = class(TL2Object);

TL2Spawn class(TL2Object);	Все объекты в не персонажа
DistTo(X: Integer; Y: Integer; Z: Integer): Cardinal; overload;	Возвращает дистанцию до заданной точки
DistTo(Obj: TL2Spawn): Cardinal; overload;	Возвращает дистанцию до объекта Obj
InRange(X: Integer; Y: Integer; Z: Integer; Range: Cardinal; ZRange: Cardinal = 250): Boolean;	Проверка вхождения точки (относительно объекта) в заданный радиус
InZone: Boolean;	Проверка на вхождение объекта в зону охоты
SpawnTime: Cardinal;	Время появление объекта.
X: Integer;	Координаты объекта
Y: Integer;	Координаты объекта
Z: Integer;	Координаты объекта

TL2Drop = class(TL2Spawn);

TL2Drop = class(TL2Spawn);	Дроп в игре
Count: int64;	Количество
IsMy: Boolean;	Дроп принадлежит нам или нет ("Нам" - если выбил наш чар, пет или члены пати)
Stackable: Boolean;	Стопковый предмет или не может стакаться

TL2Npc = class(TL2Live);

TL2Npc = class(TL2Live);	Базовый класс для всех NPC
IsPet: Boolean;	Пет или нет
PetType: Cardinal;	Тип пета (самон или пет)

TL2Pet = class(TL2Npc);

TL2Pet = class(TL2Npc);	Класс описывающий наших петов/самонов
Fed: Cardinal;	Еда (проценты)

TL2Char = class(TL2Live);

TL2Char = class(TL2Live);	Базовый класс для всех игроков
CP : Cardinal;	
CurCP : Cardinal;	
MaxCP : Cardinal;	
Hero : Boolean;	
Noble : Boolean;	
ClassID : Cardinal;	
MainClass: Cardinal;	
MountType: Byte;	Тип ездового животного
StoreType: Byte;	
Sex : Cardinal;	0 мужик 1 женщина
Race : Cardinal;	
CubicCount: Cardinal;	
Recom: Cardinal;	
Premium : Boolean;	

TL2User = class(TL2Char);

TL2User = class(TL2Char);	Класс описывающий нашего персонажа
CanCryst: Boolean;	Может кристаллизовать предметы наш герой или нет?
Charges: Cardinal;	для гладов зарядки
WeightPenalty: Cardinal;	
WeapPenalty: Cardinal;	
ArmorPenalty: Cardinal;	
DeathPenalty: Cardinal;	
Souls: Cardinal;	для камаэелей души

TL2Effect = class(TL2Object);

TL2Effect = class(TL2Object);	Базовый класс всех магических эффектов
Level: Cardinal;	Уровень скила
EndTime: Cardinal;	Время до окончания действия

TL2Buff = class(TL2Effect);

TL2Buff = class(TL2Effect);	Класс описывающий бафы
------------------------------------	-------------------------------

TL2Live = class(TL2Spawn);

<code>TL2Live = class(TL2Spawn);</code>	Базовый класс "живых" объектов в игре (игрок, нпс, пет и т.д.)
<code>AbnormalID : Cardinal;</code>	айди получившийся из наборов флагов. Примеры ниже.
<code>Abnormals : TBuffList;</code>	Для ГОД+ хроник.
<code>Ally: string;</code>	Имя альянса
<code>AllyID: Cardinal;</code>	ID альянса в который входит объект
<code>Attackable: Boolean;</code>	Свободно атакуемый (без ctrl)
<code>AtkOID: Cardinal;</code>	OID объекта который атакует
<code>AtkTime: Cardinal;</code>	время когда начал атаковать
<code>Bufs: TBuffList;</code>	Бафы объекта (доступны для нашего чара, пета и сопартийцев)
<code>Cast: TL2Cast;</code>	Скил который объект кастует в данный момент. Актуально если <code>Cast.EndTime > 0</code> , иначе объект в данный момент не кастует.
<code>Clan: string;</code>	Имя клана
<code>ClanID: Cardinal;</code>	ID клана в который входит объект
<code>CurHP: Cardinal;</code>	Точное количество жизней
<code>CurMP: Cardinal;</code>	Точное количество маны
<code>Dead: boolean;</code>	Жив или убит
<code>Dropped: Boolean;</code>	Объект выронил предмет или нет (<code>Dead</code> должен быть <code>True</code>)
<code>Exp: Int64;</code>	Опыт
<code>EXP2: Int64;</code>	
<code>Fishing: Integer;</code>	
<code>Fly: Boolean;</code>	This is Fly, a member of class TL2Live.
<code>HP: Cardinal;</code>	Текущее кол-во HP в процентах
<code>InCombat: Boolean;</code>	Объект находится в комбате или нет
<code>IsMember: Boolean;</code>	Является объект членом группы или нет
<code>Karma: Integer;</code>	Карма (начиная с GoD может быть как отрицательной (PK) так и положительной (репутация))
<code>Level: Byte;</code>	Уровень
<code>Load: Cardinal;</code>	Загруженность (проценты) (доступен для нашего чара или петов)
<code>MaxHP: Cardinal;</code>	Максимальное количество ХП
<code>MaxMP: Cardinal;</code>	
<code>MP: Cardinal;</code>	Текущее кол-во MP в процентах
<code>MyAtkTime: Cardinal;</code>	когда я его атаковал?
<code>PK: Boolean;</code>	Player Killer
<code>PvP: Boolean;</code>	Объект находится в режиме PvP
<code>Running: Boolean;</code>	Объект движется пешком или бегом
<code>Sitting: Boolean;</code>	Сидит?
<code>SP: Cardinal;</code>	Очки SP
<code>Speed: Double;</code>	
<code>Sweepable: Boolean;</code>	Можно свипать?
<code>Target: TL2Live;</code>	Цель объекта
<code>Team: Byte;</code>	для пvp серверов (красное синие подсвечивание), так же мобы "чемпионы"
<code>TeleportDist: Cardinal;</code>	Дистанция последней телепортации
<code>TeleportTime: Cardinal;</code>	Время последней телепортации
<code>Title: string;</code>	Титул объекта
<code>ToX: Integer;</code>	Координаты куда направился объект.
<code>ToY: Integer;</code>	
<code>ToZ: Integer;</code>	

TL2Skill = class(TL2Effect);

<code>TL2Skill = class(TL2Effect);</code>	Класс описывающий скилы
<code>Disabled: Boolean;</code>	Скил не доступен
<code>Enchanted: Boolean;</code>	This is Enchanted, a member of class TL2Skill.
<code>Passive: Boolean;</code>	Скил пассивный

TL2Item = class(TL2Object);

<code>TL2Item = class(TL2Object);</code>	Класс описывающий итемы в инвентаре
<code>Count: Int64;</code>	Количество (если стопка)
<code>Equipped: Boolean;</code>	Вещь надета или нет
<code>EnchantLevel: Word;</code>	This is EnchantLevel, a member of class TL2Item.
<code>ItemType : Cardinal;</code>	0 оружие; 1 броня; 2 бижа; 5 ресурсы и все остальное
<code>Grade: Cardinal;</code>	
<code>GradeName: string;</code>	('NG', 'D', 'C', 'B', 'A', 'S', 'S80', 'S84', 'R', 'R95', 'R99');

TL2List = class;

<code>TL2List = class;</code>	Все листы объектов в боте наследуются от этого класса, списки: бафов, нпц, чаров, дропа, предметов.
<code>ByID(ID: Cardinal; var Obj): Boolean;</code>	Поиск объекта в списке по ID. Если объект найден, он помещается в переменную Obj.
<code>ByName(const Name: string; var Obj): Boolean;</code>	Поиск объекта в списке по имени. Если объект найден, он помещается в переменную Obj.
<code>Count: integer;</code>	Количество объектов в списке
<code>Items(Index: integer): TL2Object;</code>	Позволяет обратиться к объекту в списке по индексу

TSpawnList = class(TL2List);

<code>TSpawnList = class(TL2List);</code>	Список объектов все что имеют координаты.
<code>Items(Index: integer): TL2Spawn;</code>	Позволяет обратиться к объекту в списке по индексу

TNpcList = class(TL2List);

<code>TNpcList = class(TL2List);</code>	Список окружающих нас NPC. В скриптах доступен по имени NpcList
<code>Items(Index: integer): TL2Npc;</code>	Позволяет обратиться к объекту в списке по индексу

TPetList = class(TL2List);

<code>TPetList = class(TL2List);</code>	Список наших питомцев. В скриптах доступен по имени PetList
<code>Items(Index: integer): TL2Pet;</code>	Позволяет обратиться к объекту в списке по индексу

TCharList = class(TL2List);

<code>TCharList = class(TL2List);</code>	Список окружающих нас игроков. В скриптах доступен по имени CharList
<code>Items(Index: integer): TL2Char;</code>	Позволяет обратиться к объекту в списке по индексу

TDropList = class(TL2List);

<code>TDropList = class(TL2List);</code>	Список окружающего нас дропа. В скриптах доступен по имени DropList
<code>Items(Index: integer): TL2Drop;</code>	Позволяет обратиться к объекту в списке по индексу

TSkillList = class(TL2List);

<code>TSkillList = class(TL2List);</code>	Список содержащий скилы нашего персонажа. В скриптах доступен по имени SkillList
<code>Items(Index: integer): TL2Skill;</code>	Позволяет обратиться к объекту в списке по индексу

TBuffList = class(TL2List);

<code>TBuffList = class(TL2List);</code>	Список содержащий бафы объекта.
<code>Items(Index: integer): TL2Buff;</code>	Позволяет обратиться к объекту в списке по индексу

TItemList = class(TL2List);

<code>TItemList = class(TL2List);</code>	Список инвентаря.
<code>Items(Index: integer): TL2Item;</code>	Позволяет обратиться к объекту в списке по индексу

TParty = class;

TParty = class;	Класс описывающий нашу группу. В скриптах доступен по имени Party.
Pets: TNpcList;	Список питомцев в группе.
Chars: TCharList;	Список чаров в группе.
LootType: TLootType;	Тип распределения лута в группе.
Leader: TL2Char;	Лидер группы

TInventory = class;

TInventory = class;	Класс содержащий инвентари. В скриптах доступен по имени Inventory.
Pet: TItemList;	Инвентарь нашего пета
User: TItemList;	Инвентарь нашего персонажа.
Quest: TItemList;	Инвентарь нашего персонажа (квестовый)

TL2Script = class;

TL2Script = class;	Класс доступный по имени Script
MainProc(Proc: Pointer);	Вызов процедуры в основном потоке программы. Proc - указатель на процедуру потока; Parameter - любой передаваемый параметр (не обязательный)
NewThread(Proc: TThreadFunc; Parameter: Pointer = nil);	Создает новый поток скрипта. Proc - указатель на процедуру потока (procedure MyThread(Prm: Pointer)) Parameter - любой передаваемый параметр (не обязательный)
Path: String;	Полный путь к файлу скрипта.
Resume;	Возобновляет работу скрипта
Suspend;	Ставит скрипт на паузу

TConfirmDlg = class

TConfirmDlg = class	Класс подробно описывающий диалоги
MsgID : Cardinal;	
ReqID : Cardinal;	
Sender: string;	
EndTime: Cardinal;	
Valid : Boolean;	

Функции:

Функции API бота:

Функции API бота:	Неочевидные
PlaySound(const FileName: string; Loop: Boolean);	Проигрывает звуковой файл в формате wave (.wav). Loop - зациклить воспроизведение.
StopSound;	Останавливает воспроизведение звукового файла
Delay(ms: Cardinal): Boolean;	Задержка скрипта на указанное число миллисекунд.
ExePath: string;	Возвращает путь к папке с Adrenaline
GetControl(const Name: string): TL2Control;	Получает объект Engine другого персонажа. Name - имя персонажа.
BotLoginID: Cardinal;	Получает уникальный ID (хэш) основанный на бот-логине.

Функции преобразования данных

Функции преобразования данных
MemToHex(const dt; size: Word; sep: char = #0): String; overload;
MemToHex(const Mem: AnsiString): String; overload;
HexToMem(const Hex: string; var Buf): Cardinal; overload;
HexToMem(const Hex: string): AnsiString; overload;

API для каптчи

Captcha API	
<pre>function GetCaptcha(ImageData: TMemoryStream; APIKey: String; var CaptchaRes: String; MinLen: integer=0; MaxLen: integer=0; Numeric: integer=0; Phrase: integer=0; RegSense: integer=0; Calc: integer=0; Russian: integer=0): Integer; overload;</pre>	<p>Функция GetCaptcha - распознает картинку используя сервис AntiGate.com</p> <p>Параметры функции:</p> <ul style="list-style-type: none">ImageData - стрим, содержащий картинку с каптчейImageFile - путь к файлу каптчиAPIKey - ключ сервиса AntiGate.com для распознавания <p>CaptchaRes - буфер, в который попадает текст каптчи, либо сообщение об ошибке</p> <ul style="list-style-type: none">MinLen - минимальная длина текста каптчиMaxLen - максимальная длина каптчиPhrase - если 1, помечает что каптча состоит из нескольких словRegsense - если 1, помечает что текст каптчи чувствителен к региструNumeric - если 1, помечает что текст каптчи состоит только из цифр, 2 помечает что на каптче нет цифрCalc - если 1, помечает что цифры на каптче должны быть высчитаныRussian - если 1, помечает что вводить нужно только русский текст, 2 - русский или английский <p>Функция возвращает: Номер каптчи CaptchaID, при ошибке 0</p>
<pre>function CaptchaBalance(APIKey: String): String;</pre>	<p>Функция CaptchaBalance - выводит текущий баланс</p> <p>Параметры функции:</p> <ul style="list-style-type: none">APIKey - ключ сервиса AntiGate.com для распознавания <p>Функция возвращает: Строку с содержанием баланса, либо 'N/A' при ошибке</p>
<pre>function GetCaptchaReportBad(APIKey: String; CaptchaID: Integer): String;</pre>	<p>Функция CaptchaReportBad - отправляет жалобу о неверно распознанной каптче</p> <p>Параметры функции:</p> <ul style="list-style-type: none">APIKey - ключ сервиса AntiGate.com для распознавания <p>CaptchaID - номер каптчи, которая была неверно распознана</p> <p>Функция возвращает: Строку с результатом ответа от сервиса (OK_REPORT_RECORDED - в случае успеха), либо 'N/A' при ошибке</p>
CaptchaServer	по умолчанию равен AntiGate.com, при необходимости его можно изменить

ChatMessage

ChatMessage	Объект для работы с чатом
ChatMessage.unread: boolean;	Не прочитано нами?
ChatMessage.sender: string;	Кто отправил.
ChatMessage.text: string;	Текст что напечатал отправивший
ChatMessage.chattype: TMessageType;	Тип отправленного сообщения.

Прочее

Прочее	Для опытных скриптеров.
<code>Procedure OnFree;</code>	Вызывается при завершении скрипта. Тело процедуры надо писать самому.
<code>function OnEntry(var Param)</code>	Функция, тело которой должно быть написано в вашем коде. Для вызова с другого TL2Control, через Entry
<code>ShMem: array[0..1000] of integer;</code>	Массив, который доступен из любого TL2Control (любого персонажа)
<code>procedure OnPacket(ID, ID2: Cardinal; Data: Pointer; Size: Word);</code>	Пакетный обработчик, Пример: <code>procedure OnPacket(ID, ID2: Cardinal; Data: Pointer; Size: Word); begin print(ID); print(MemToHex(Data^, Size)); end;</code>

Перечисляемые типы.

Enum values

```
TL2Status = (  
    lsOff,  
    lsOnline,  
    lsOffline  
);
```

Отключен
Онлайн
Оффлайн

```
TL2Race = (  
    rtHuman,  
    rtElf,  
    rtDarkelf,  
    rtOrc,  
    rtDwarf,  
    rtKamael  
);
```

```
TLootType = (  
    ldLooter  
    ldRandom  
    ldRandomSpoil  
    ldOrder  
    ldOrderSpoil  
);
```

Нашедшому
Случайно
Случайно + присвоить
По очереди
По очереди + присвоить

```
TStoreType = (  
    stNone,  
    stSell,  
    stPrepare,  
    stBuy,  
    stUnknown2,  
    stManufacture,  
    stUnknown6,  
    stObservingGames,  
    stSellPackage  
);
```

Тип торговой лавки

```
TL2Action = (  
    laSpawn,  
    laDelete,  
    laPetSpawn,  
    laPetDelete,  
    laInvite,  
    laDie,  
    laRevive,  
    laTarget,  
    laUnTarget,  
    laInGame  
    laStatus  
    laBuffs,  
    laSkills,  
    laDlg,  
    laConfirmDlg,  
    laStop,  
    laStartAttack,  
    laStopAttack,  
    laCast,  
    laCancelCast,  
    laTeleport,  
    laAutoSoulShot,  
    laNpcTrade,  
    laSysMsg,  
    laChat,  
    laKey  
);
```

Перечисляемый тип для WaitAction
отреспился предмет, любой TL2Spwan

Любой TL2Live взял кого то в таргет

в игре
поменялся статус

p1 : TConfirmDlg

Любой TL2Live кастует скил.

TL2Live телепортировался

p1 = id системного сообщения
не юзать, вместо него ChatMessage
p1 = id клавиши

```
TRestartType = (
```

для функции TL2Control.GoHome

```
    rtTown,  
    rtClanHoll,  
    rtCastle,  
    rtFort,  
    rtFlags  
);
```

```
TMessageType = (
```

для ChatMessage.Type

```
    mtSystem,  
    mtAll,  
    mtPrivate,  
    mtParty,  
    mtClan,  
    mtFriend,  
    mtShout  
);
```

```
TL2Class = (
```

для поля у TL2Object

```
    lcError,  
    lcDrop,  
    lcNpc,  
    lcPet,  
    lcChar,  
    lcUser,  
    lcBuff,  
    lcSkill,  
    lcItem  
);
```

TTCPSocket = class

Класс доступен в модуле TCP.

TTCPSocket = class	Класс для работы с TCP соединением
<code>constructor Create;</code>	
<code>destructor Destroy; override;</code>	
<code>procedure CloseSocket; override;</code>	
<code>function WaitingData: integer; override;</code>	
<code>procedure Listen; override;</code>	
<code>function Accept: integer; override;</code>	
<code>procedure Connect(IP, Port: string); override;</code>	
<code>function SendBuffer(Buffer : pointer; Len : integer) : integer; override;</code>	
<code>Function RecvBuffer(Buffer : pointer; Len : integer; Timeout : integer) : integer; virtual;</code>	
<code>Procedure SendByte(Data : byte); virtual;</code>	
<code>Function RecvByte(Timeout : integer) : byte; virtual;</code>	
<code>Procedure SendString(Data : ansistring); virtual;</code>	
<code>Function RecvString(Timeout : integer) : ansistring; virtual;</code>	
<code>Procedure SendInteger(Data : integer); virtual;</code>	
<code>Function RecvInteger(Timeout : integer) : integer;</code>	
<code>Property LastError : integer;</code>	

Класс доступен в модуле ICQ.

TICQ = class	Класс для работы с ICQ.
constructor Create;	
destructor Destroy; override ;	
property Status: Cardinal;	Текущий статус: ONLINE = \$00000000; INVISIBLE = \$00000100; AWAY = \$00000001; NA = \$00000005; OFFLINE = \$FFFFFFFF;
function Connected: Boolean;	Подключены или нет в текущий момент к серверу ICQ.
function Connect(UIN : Cardinal; const Password : string ; const Server : string = 'login.icq.com'; Port : Word = 5190; TimeOut : Byte = 5) : Boolean;	Осуществляет подключение к серверу ICQ. UIN - номер ICQ под которым требуется войти. Password - пароль от ICQ. Server - адрес сервера ICQ. Port - порт сервера. TimeOut - время на попытку подключения (в секундах).
procedure Disconnect;	Отключение от сервера.
procedure SendMessage(UIN : Cardinal; const Msg : string);	Отправка сообщения. UIN - номер получателя. Msg - текст сообщения.
procedure OnError(Sender : TObject; ErrorType : TErrorType; const ErrorMsg : string); virtual ;	Вызывается при ошибке. ErrorType - тип ошибки ErrorMsg - текст ошибки (Используется в наследующих классах для перекрытия - override)
procedure OnMessageRecv(Sender : TObject; Msg, UIN : string); virtual ;	Получено сообщение. Msg - текст сообщения. UIN - номер отправителя. (Используется в наследующих классах для перекрытия - override)
procedure OnUserOffline(Sender : TObject; UIN : string); virtual ;	Контакт из вашего списка ICQ отключился от сервера. UIN - номер отключившегося (Используется в наследующих классах для перекрытия - override)
procedure OnServerDisconnect(Sender : TObject; Reason : LongInt; Description : string); virtual ;	Вы были отключены сервером. Reason - причина (код) Description - текстовое описание причины. (Используется в наследующих классах для перекрытия - override)

ChatMessage

```

var
  Obj: TL2Live;
begin
  while True do
  begin
    if ChatMessage.Unread and (ChatMessage.Time < 3000) then
    begin
      if CharList.ByName(ChatMessage.sender, Obj) and not Obj.IsMember then
      if Obj.InZone then
      begin
        print(ChatMessage.sender);
        print(ChatMessage.text);
        print(ChatMessage.Time);
        print(ChatMessage.ChatType);
      end;
    end;
    Delay(111);
  end;
end.

```

Печатать в системном окне бота информацию о сообщении, которую написал игрок, находящийся в зоне и не ваш сопартнец.

Function TL2Live.AbnormalID

```

Function IsUD(actor : TL2Live) : Boolean;
begin
  Result := actor.AbnormalId and $8000000 = $8000000;
end;

//Цель в УД стоит?
begin
  if isUD(User.target) then
    print('Да в УД');
end.

```

Проверить находится цель в УД или нет. Значение \$8000000 это один из битов переменной поля AbnormalID, обозначающий есть ли эффект UD. Некоторые другие эффекты:

```

bleeding = $1;
poison = $2;
redcircle = $4;
ice = $8;

```

function TL2Control.GetSkillList

Для текущего скрипта **TL2Control** доступен по имени Engine так же список скилов доступен по переменной **SkillList**. С другого **TL2Control** можно получить список скилов с помощью **GetSkillList**

```

Function GetSkill(const Control : String; const ID : Cardinal) : TL2Skill;
var
  NovObj : TL2Skill;
  NovEngine : TL2Control;
begin
  NovEngine := GetControl(control);
  if Assigned(NovEngine) and NovEngine.GetSkillList.ByID(ID, NovObj) then
    Result := NovObj;
end;

begin
  if Assigned(GetSkill('MoyaEEshka', 1255)) then
    GetControl('MoyaEEshka').UseSkill(1255) //Recall
  Else
    Engine.UseItem(736); //Юзаем сое
end.

```

Если есть у нашей EE с ником 'MoyaEEshka' скилл Party Recall, то используем его на ней, если нету то используем сое.

function TL2Control.GameWindow

Зачем он нужен этот HWND игрового окна?
Например в скрипте можно использовать WinApi функции.

```
function SetForegroundWindow(hwnd: integer) : Boolean; stdcall;
    external 'user32.dll';

begin
    While Engine.Delay(5000) then
        if User.dead then
            SetForegroundWindow(Engine.GameWindow);
end.
```

развернуть окно с игрой на передний план, если мы померли.

procedure TL2Control.MSG(Who, What : String; Color : Integer);

Чтобы раскрасить системный чат в Боте можно использовать данный метод.
Цвет задается целочисленным значением. Некоторые из возможных:

```
clBlack = 0; //черный
clMaroon = 128; //Тёмно-красный
clGreen = 32768; //Зелёный
clOlive = 32896; //Оливковый
clNavy = 8388608; //Тёмно-синий
clPurple = 8388736; //Пурпурный
clTeal = 8421376; //Стальной
clGray = 8421504; //Серый
clSilver = 12632256; //Серебряный
clRed = 255; //Красный
clLime = 65280; //Ярко-зелёный
clYellow = 65535; //Жёлтый
clBlue = 16711680; //Синий
clFuchsia = 16711935; //Фиолетовый
clAqua = 16776960; //Бирюзовый
clWhite = 16777215; //Белый
```

TL2Live.Teleport...

Знать информацию о каждом объекте, когда телепортировался и насколько далеко. В том числе и о себе.

```
begin
    While Engine.Delay(1000) do
        if (GettickCount - User.TeleportTime < 5000) and (User.TeleportDist < 2000) and
(User.TeleportDist > 10) then //5 сек назад был телепорт
            begin
                print('Teleported!');
                PlaySound(exepath + '\sounds\' + 'dc' + '.wav', False);
            end;
end.
```

Проиграть звук при телепорте из папки с ботом, в папке Sounds звук dc.wav.
