# Модуль описания типов и констант GearTypesUnit

## Типы передач

// Тип ремённой передачи

TBeltType =

(

btFlat {Плоский},

btVee {Клиновой},

btMultiGroove {Поликлиновой},

btFilm {Плёночнный},

btNarrowVee {Узкоклиновой},

btTiming {Зубчатый}

);

// Тип цепной передачи

TChainType =

(

ctTooth {Зубчатая},

ctRoller {Роликовая}

);

// Тип зубчатой передачи

TGearType =

(

gtSpur {Прямозубая},

gtHelical {Косозубая},

gtDoubleHelical {Шевронная},

gtWorm {Червячная},

gtInternal {Внутреннего зацепления},

gtPlanet {Планетарная}

);

## Описание входного и выходного потока данных

### Было

type

IslRec=record {Общая для всех передач}

i: byte;// Идентификатор

Itype:byte;

P1,n1,

d1,d2,Cp,Fdelta,alfa\_min,nyu\_max,V\_max,F\_pr,a,b:real;

Tip\_pl: word;

Tip\_nat: boolean;

end;

DatRec=record {Общая для всех передач}

i: byte;// Идентификатор

P1 {Мощность на ведущем валу},n1 {Частота вращения ведущего шкива},

up,d1,Cp {Коэффциент режима работы},Fdelta,alfa\_min,

nyu\_max,V\_max,F\_pr,a {Межосевое расстояние},Sdelta:real;

dud,m,Cpo:real; //Зуб

Q,Tip\_pl {Тип плоского ремня},Tip\_kl{Зуб},Lh\_0,Change: word;

z1:byte;

Tip\_nat{Тип натяжения},Tip\_p: boolean;

Mark,bet: byte;

Ka: word;

x, y, z, ii, j, kk: single;

Kd,Ke: real;

end;

type

//Плоскоременная

TransRes=record

d1,d2,a,l,Fdelta,bs,b,F0,Fb,Lh,Dmin:real;

end;

### Стало

// Входные параметры для узкоклиновой ремённой передачи

TNarrowVeeInput = record

P1, {Мощность на ведущем валу, кВт}

n1, {Частота вращения ведущего шкива, об/мин}

up, {Передаточное число}

d1, {Диаметр меньшего шкива, мм; 0 - Любой}

a: extended; {Межосевое расстояние, мм; 0 - Любое}

Q: integer; {Угол наклона передачи} {Судя по всему в расчёте не используется}

Cp: extended; {Коэффициент режима работы}

{1 - спокойная нагрузка, 1.15 - умеренные колебания}

{1.25 - значительные колебания, 1.55 - ударная нагрузка}

Tension: TTensionType;

{Тип натяжения ремня; автоматическое-периодическое}

SectionIndex: integer;

{Номер типоразмера ремня; 0 - Любой, 1 - УО, 2 - УА, 3 - УБ, 4 - УВ}

end;

// Выходные параметры для одного экземпляра узкоклиновой ремённой передачи

TNarrowVeeOutput = record

d1, {Диаметр меньшего (ведущего) шкива, мм}

l, {Неизвестный выходной параметр, мм}

d2, {Диаметр большего (ведомого) шкива, мм}

a: extended; {Межосевое расстояние, мм}

SectionIndex: integer;

{Номер типоразмера ремня, 1 - УО, 2 - УА, 3 - УБ, 4 - УВ}

z: integer; {Число ремней}

d\_e1, {Наружный диаметр ведущего шкива, мм}

d\_e2, {Наружный диаметр ведомого шкива, мм}

d\_f1, {Диаметр по дну впадин ведущего шкива, мм}

d\_f2: extended; {Диаметр по дну впадин ведомого шкива, мм}

Bs: extended; {Ширина шкива, мм}

F0, {Усилие предварительного натяжения, Н}

Fb, {Давление на валы от натяжения ремня, Н}

Lh, {Долговечность ремня, часы}

Vol, {Объём передачи}

alfa, {Угол обхвата ремнем шкива, градусы}

nyu, {Число проходов ремня, 1/с}

PMax, {Максимально допустимая мощность, кВт}

V: extended; {Линейная скорость ремня, мм/с}

end;

Поменялись типы с real, word, boolean, byte и т.д. на унифицированные – extended, integer.

Натяжение ремня теперь перечисляемый тип:

// Тип натяжения ремня

TTensionType =

(

ttAutomatic, {Автоматическое}

ttTiming {Периодическое} );

Тип данных «коллекция передач» образуется добавлением окончания «s»

// Коллекция выходных параметров узкоклиновой ремённой передачи

TNarrowVeeOutputs = array of TNarrowVeeOutput;

## Константы расчёта

const

//Максимально допустимое значение мощности на ведущем валу, кВт

P1Max: extended = 200;

//Минимально допустимое значение мощности на ведущем валу, кВт

P1Min: extended = 0.1;

//Максимально допустимое значение передаточного числа

UpMax: extended = 10;

//Минимально допустимое значение передаточного числа

UpMin: extended = 0.1;

//Максимально допустимое значение частоты вращения ведущего шкива, об/мин

N1Max: extended = 10000;

//Минимально допустимое значение частоты вращения ведущего шкива, об/мин

N1Min: extended = 0.1;

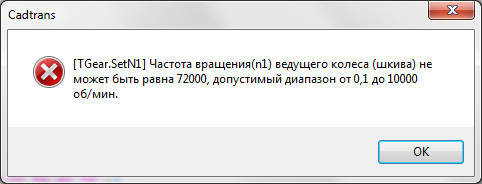
# Синтаксис сообщений об ошибке

if (Value < N1Min) or (Value > N1Max) then

raise ERangeError.CreateFmt

('[TGear.SetN1] Частота вращения(n1) ведущего колеса (шкива) не может быть равна %g, допустимый диапазон от %g до %g об/мин',

[Value, N1Min, N1Max]);



### Расшифровка:

[TGear.SetN1] – метод, в котором произошла ошибка

(n1) – имя свойства

Обязательно указываются:

введенное значение – 72000

допустимый диапазон – от 0,1 до 10000

единицы измерения – об/мин

В данном случае допустимый диапазон берется из констант, описанных в модуле GearTypesUnit однако, это не всегда так:

if (Value < MinDistance) and (Value <> 0) then

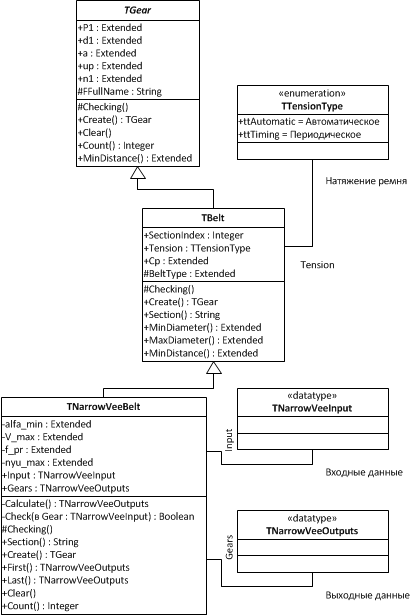
raise ERangeError.CreateFmt

('[TBelt.SetA] Межосевое расстояние (a) не может быть равно %g, значение должно быть больше %g мм',

[Value, MinDistance]);

MinDistance здесь – это метод, возвращающий минимально допустимое межосевое расстояние.

# Иерархия классов CADTrans



# Иерархия классов CADTrans

