Sketchy Text Model

Metasymbols: [ ] -- 0 or 1
{ } -- 0 or more
| -- or

Sketch :
sketchType Head Branch{Branch} Comment

sketchType:
SIMPLE | LOOP | SWITCH | COMMENT | ...

Head:
{ElementaryHead}

ElementaryHead:
headType Text

headType:
LOOPHEAD | SWITCHHEAD | ...

Text:
{Row} Comment

Row:
{RowElement}

RowElement:
Symbol
| Image

Symbol:
char Font Color

Comment:
Row

Branch:
branchType Head {Member} Comment

branchType:
CASE | DEFAULT | TRY | CATCH

Member:
Sketch
| NonSketch

NonSketch:
nonSketchType Text

nonSketchType:
PRIMITIVE | CONDITION | ARROW

Arrow:
extends NonSketch
arrowType endpointLevel

arrowType:
STRONG | WEAK

endpointLevel:
short

```
class AmSketch implements Serializable{
  //skeemityybid (pa"iseta == pa"is on tyhi):
    static final byte SIMPLE = 10;     // yhe pa"iseta haruga
    static final byte LOOP = 11;  // yhe pa"iseta haruga
    static final byte SWITCH = 12;   // pa"isega, harudega CASE ja DEFAULT(0 voi enam)
    static final byte TRY = 13;     // pa"isega, harudega CATCH(0 voi enam)
    static final byte PARALLEL = 14; // harudega
  // s.h. harutyybid:
      static final byte BRANCH = 0; // suvaline
      static final byte CASE = 120; // lylitis
      static final byte DEFAULT = 121; // lylitis
  // väljad:
      byte type;
      Vector head; // ainult tyypi AmHead
      Vector subs;  // alamskeemid, mitteharu korral ainult  tyypi AmSketch
      AmRow comment;
      AmSketch parent; // ylemus-skeem
      AmView view; // vaade
      boolean  icon; // kas on ikoniseeritud
  // meetodid:
. . .
}// class AmSketch
```

```
class AmHead extends Rectangle implements Serializable{
// elementaarpais

   // päisetüübid:
    static final byte SIMPLE = 0;
    static final byte LOOP = 1;
    static final byte SWITCH = 2;
    static final byte CASE = 3;
    static final byte TRY = 4;
    static final byte CATCH = 5;
    static final short d = 10; // paise tyybi kujutise laius
  // väljad:
    byte type;
    AmText text;
  // meetodid:
. . .
}// AmHead
```

```
class AmNonSketch extends Rectangle implements Serializable{
   // tüübid:
      static final  byte PRIMITIVE = 0;
      static final  byte CONDITION = 1;
       static final  byte ARROW = 2;
   // väljad:
      byte type;
      AmText text;
   // meetodid:
. . .
}// AmNonSketch
```

```
class AmArrow extends AmNonSketch{

   // tüübid:
      static final  byte STRONG = 0;
      static final  byte WEAK = 1;
   // väljad:
      byte type;
      short endlevel;
  // meetodid:
. . .
}// AmArrow
```

```
class AmText extends Rectangle implements Serializable{
// tekst on ristkylik igas vaates

   // konstant:
      static final int d = 0; // intervall iga teksti loppu
   // väljad:
      Vector rows; // koik p.o. tyypi AmRow
      AmRow comment; // teksti kommentaar
                                 // (pannakse teksti viimase rea loppu)
// meetodid:
. . .
}// AmText
```

```
class AmRow extends Rectangle implements Serializable{
// rida on ristkylik igas vaates

   // konstant:
      static final int ROW_ADVANCE = 0;
   // väljad:
      Vector rowelem; // liige p.o. tyypi AmSymbol voi AmImage
       int maxAscent, maxDescent;

   // meetodid:
. . .
}// AmRow
```

```
abstract class AmView extends Rectangle{
// ylemklass ko~ikidele vaadetele
   static final Font font = new Font("TimesRoman", Font.PLAIN, 12);// paise tunnusele
   abstract void plan(AmSketch s, int x, int y); // planeerida skeem s alates kohast (x, y)
   abstract void draw(Graphics g, AmSketch s); // joonistada graafikakonteksti g
                                                          // planeeritud skeem s
   abstract void unpaintSelection(); // kustutada valitu tunnused
   abstract void repaintSelection(); // panna valitu tunnused
   abstract AmSketch pointLocation(Point p, AmSketch s); // leida va"him alamskeem,
                                                                  // kus asub antud punkt
            // tagastab selle alamskeemi viida, kuhu antud punkt kuulub;
            // null, kui punkt ei asu skeemis
            // pyytakse leida va"him skeem, kuhu punkt kuulub:  leht vo~i skeemijoone osa
            // globaalne: finish (p.o. eelnevalt false)
   static boolean finish; // meetodi pointLocation jaoks globaalne
   static AmCaret result; // meetodi pointLocation jaoks globaalne
    boolean pointLocation(Point p, AmRow r){// reas (kommentaaris) otsimine   . . .
    }// pointLocation(Point p, AmRow r)
     boolean pointLocation(Point p, AmText t){// tekstis otsimine         . . .
    }// pointLocation(Point p, AmText t)
     boolean pointLocation(Point p, AmNonSketch ns){// mitteskeemis otsimine    . . .
     }// pointLocation(Point p, AmNonSketch ns)
}// AmView
```

```
class AmCaret {
 // tekstikursor
  AmSymbol sy; // symboli jooksvate atribuutide jaoks;
                       // ei pruugi olla symb, reas
  AmSketch parent; // kursorit holmav la"him skeem (vo~i haru)
  AmText parentText; // tekst, millel asub; null, kui on kommentaaril
  AmRow  row;  // rida, millel asub
  int syNo;// symboli nr., mille ja"rel reas asub (-1, kui rea alguses)
  int shift;    // kaugus pikselites rea algusest
  AmHead parentHead; // el-pais, millel asub
  AmNonSketch parentPrimitive; // primitiiv (nonSketch), millel asub
  // parentHead == null && parentPrimitive == null :
  //                                                  on skeemikommentaaril
  Color color;  // kursori varv,
                     // kustutamiseks - AmDefault.backgroundColor
  // meetodid:
. . .
}// AmCaret
```