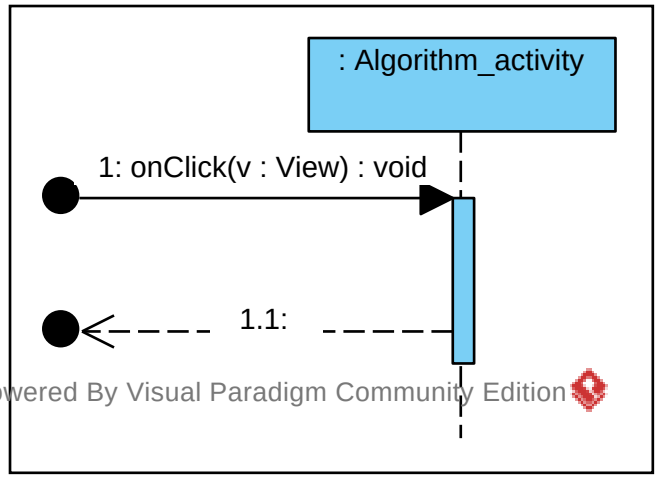
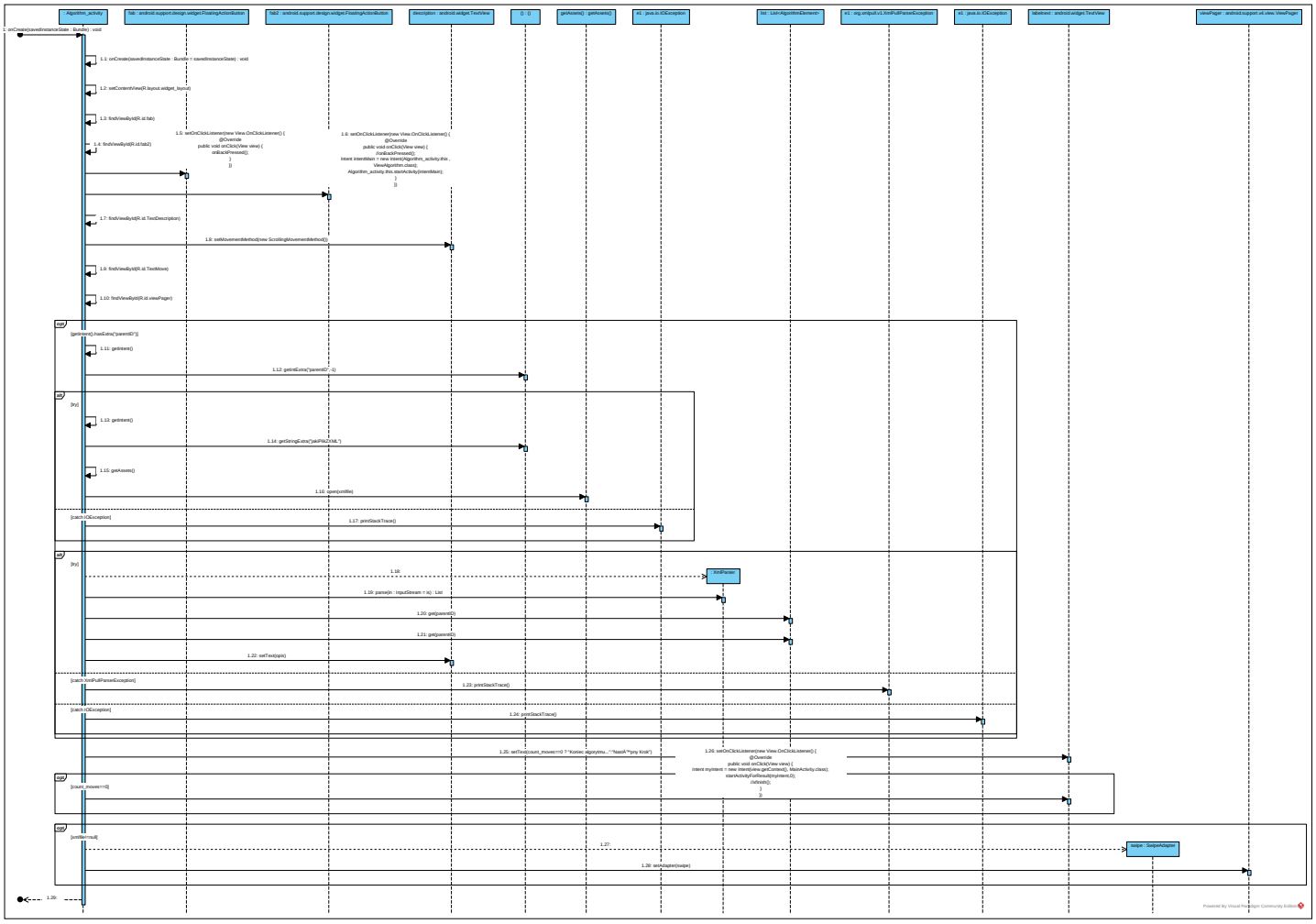
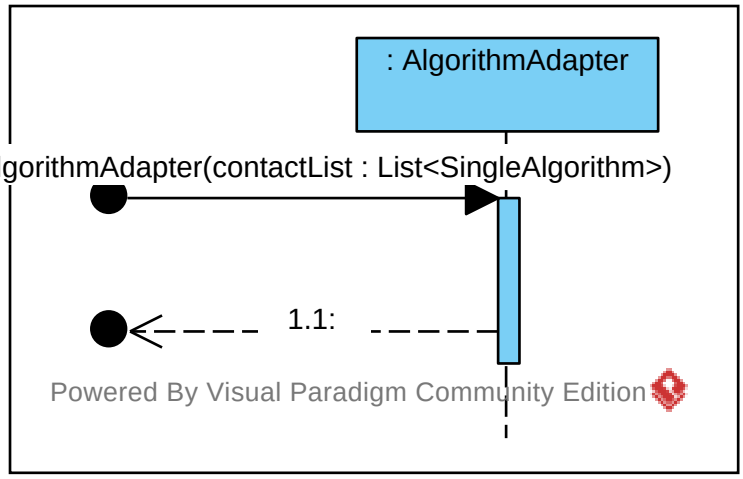
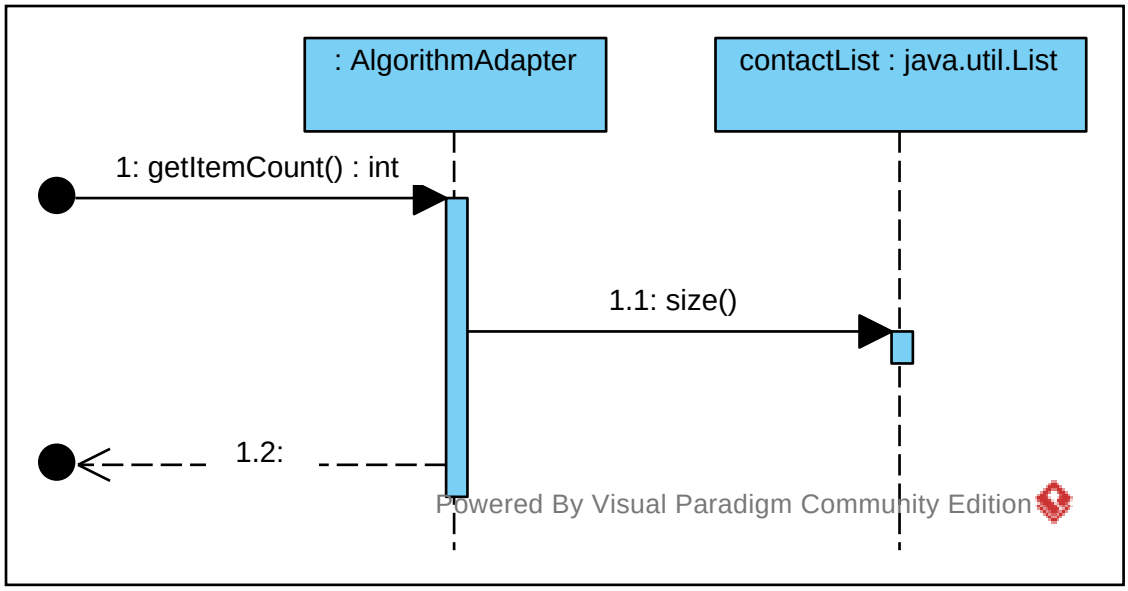
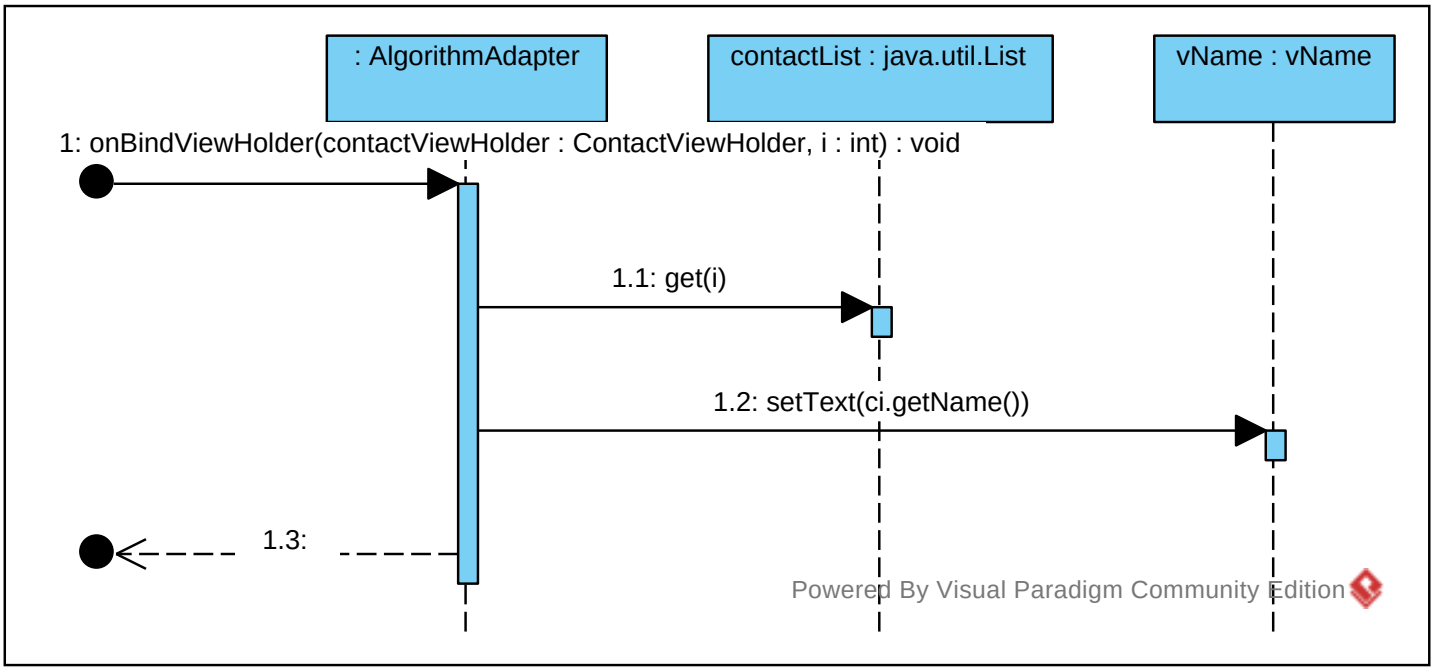

: Algorithm_activity
1: onClick(v : View) : void
1.1:
Powered By Visual Paradigm Community Edition

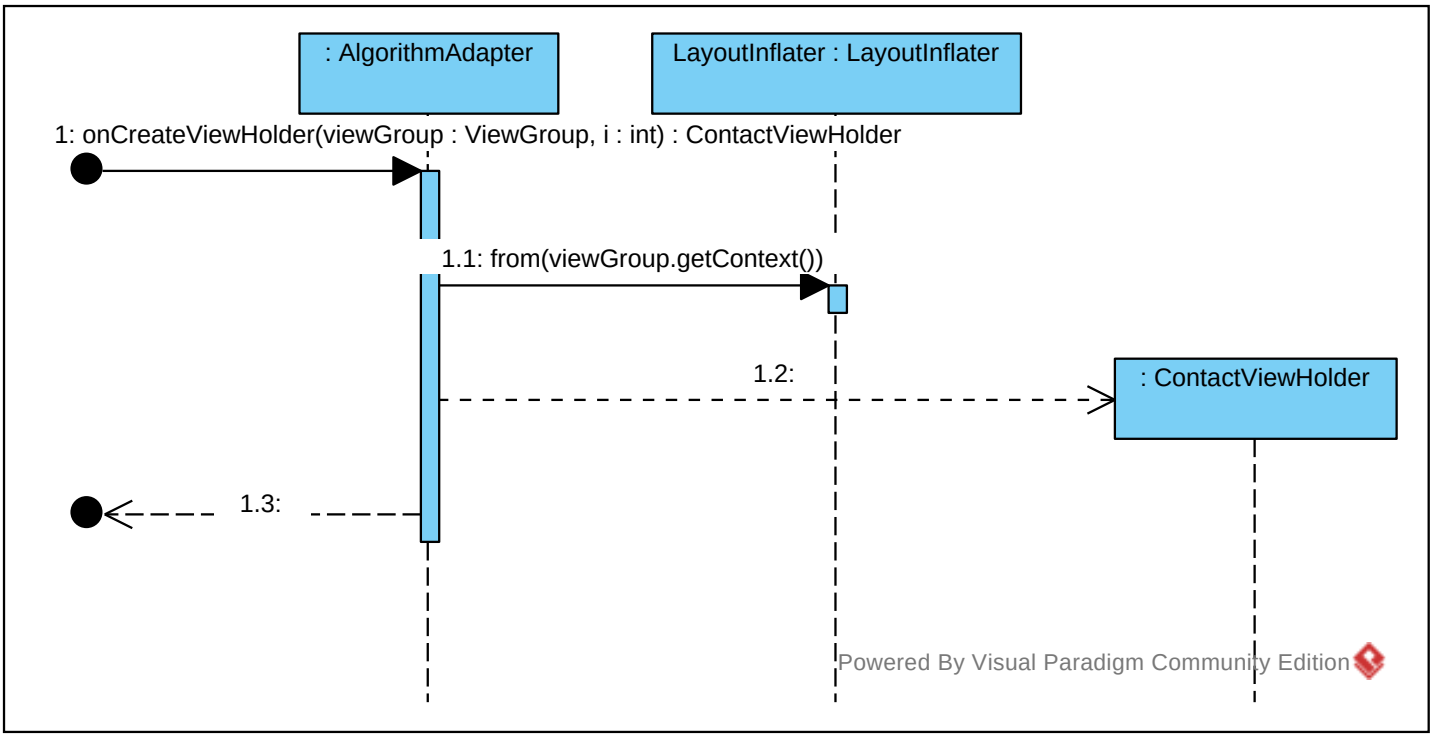


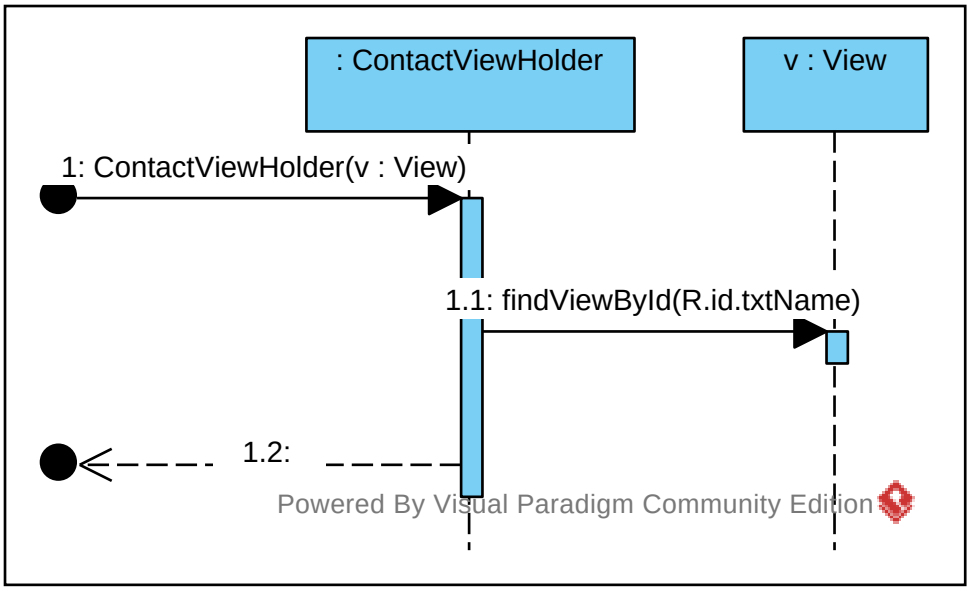
Algorithm_activity
fab : android.support.design.widget.FloatingActionButton
fab2 : android.support.design.widget.FloatingActionButton
description : android.widget.TextView
0 : 0
getAssets() : getAssets()
e1 : java.io.IOException
list : List<AlgorithmElement>
e1 : org.xmlpull.v1.XmlPullParserException
e1 : java.io.IOException
labelnext : android.widget.TextView
viewPager : android.support.v4.view.ViewPager
1 : onCreate(savedInstanceState : Bundle) : void
1.1: onCreate(savedInstanceState : Bundle = savedInstanceState) : void
1.2: setContentView(R.layout.widget_layout)
1.3: findViewById(R.id.fab)
1.4: findViewById(R.id.fab2)
1.5: setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
onBackPressed();
}
})
1.6: setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
//onBackPressed();
Intent intentMain = new Intent(Algorithm_activity.this ,
ViewAlgorithm.class);
Algorithm_activity.this.startActivity(intentMain);
}
})
1.7: findViewById(R.id.TextDescription)
1.8: setMovementMethod(new ScrollingMovementMethod())
1.9: findViewById(R.id.TextMove)
1.10: findViewById(R.id.viewPager)
opt
[getIntent().hasExtra("parentID")]
1.11: getIntent()
1.12: getIntExtra("parentID" -1)
alt
[try]
1.13: getIntent()
1.14: getStringExtra("jakPikZXML")
1.15: getAssets()
1.16: open(xmlFile)
[catch IOException]
1.17: printStackTrace()
alt
[try]
1.18:
XmlParser
1.19: parse(in : InputStream = is) : List
1.20: get(parent0)
1.21: get(parent0)
1.22: setText(jpis)
[catch XmlPullParserException]
1.23: printStackTrace()
[catch IOException]
1.24: printStackTrace()
1.25: setText(count_moves==0 ? "Koniec algorytmu..." "NastA"*pny Krok")
1.26: setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent myIntent = new Intent(view.getContext(), MainActivity.class);
startActivityForResult(myIntent,0);
//ofinish();
}
})
opt
[count_moves==0]
opt
[xmlFile==null]
1.27:
swipe : SwipeAdapter
1.28: setAdapter(swipe)
1.29:
Powered By Visual Paradigm Community Edition

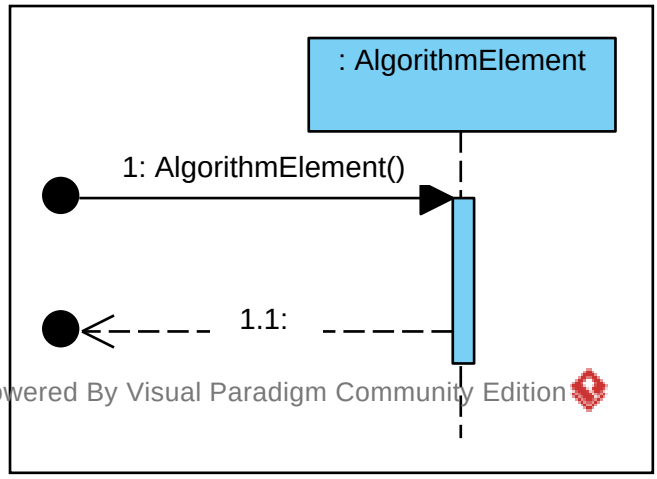

: AlgorithmAdapter
1: AlgorithmAdapter(contactList : List<SingleAlgorithm>)
1.1:
Powered By Visual Paradigm Community Edition

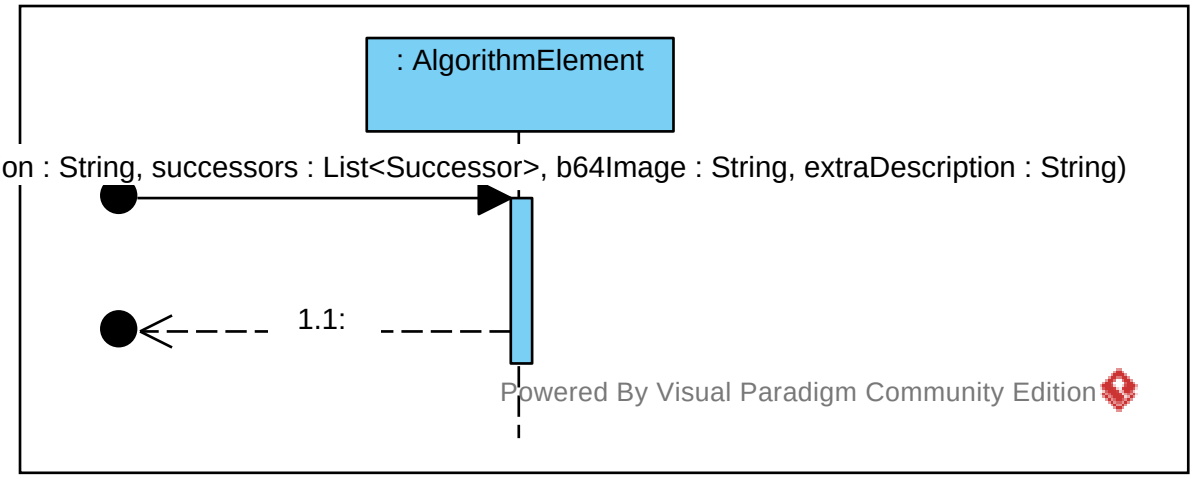

: AlgorithmAdapter
contactList : java.util.List
1: getItemCount() : int
1.1: size()
1.2:
Powered By Visual Paradigm Community Edition

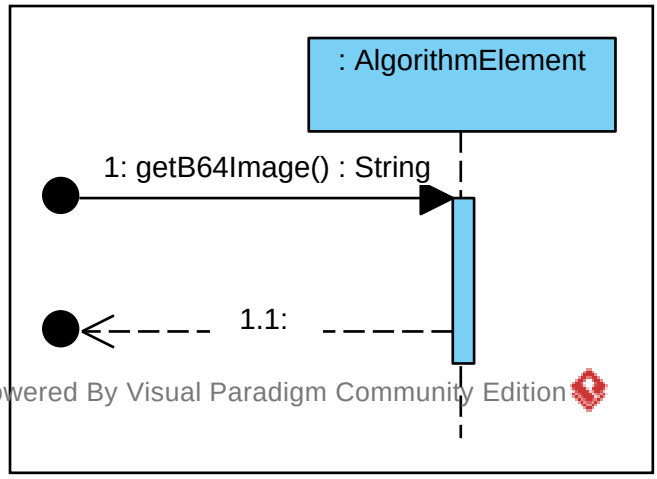

: AlgorithmAdapter
contactList : java.util.List
vName : vName
1: onBindViewHolder(contactViewHolder : ContactViewHolder, i : int) : void
1.1: get(i)
1.2: setText(ci.getName())
1.3:
Powered By Visual Paradigm Community Edition

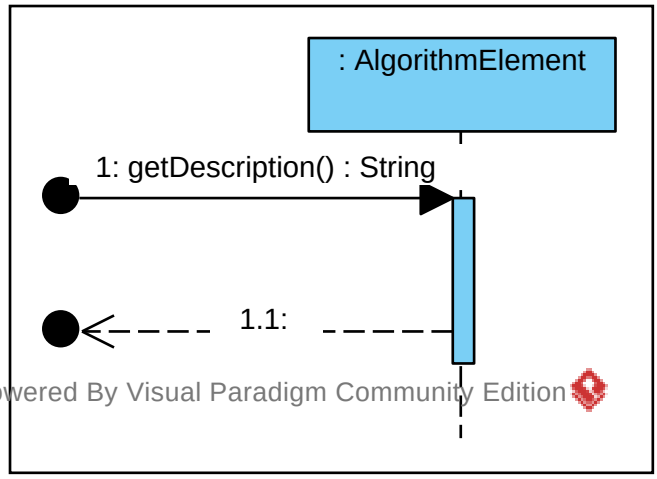

: AlgorithmAdapter
LayoutInflater : LayoutInflater
1: onCreateViewHolder(viewGroup : ViewGroup, i : int) : ContactViewHolder
1.1: from(viewGroup.getContext())
1.2:
: ContactViewHolder
1.3:
Powered By Visual Paradigm Community Edition

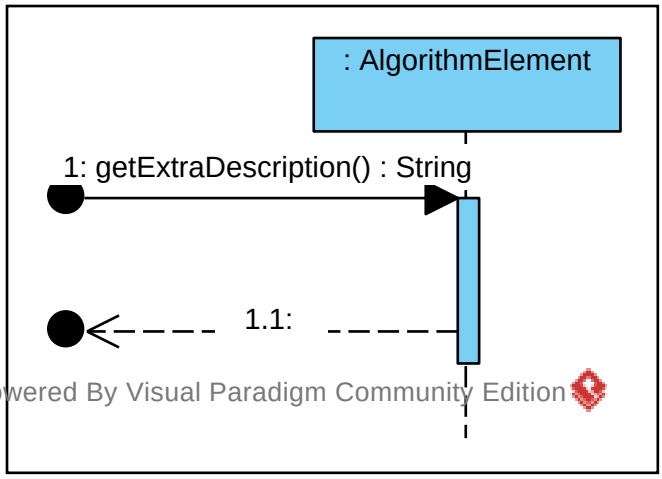

: ContactViewHolder
v : View
1: ContactViewHolder(v : View)
1.1: findViewById(R.id.txtName)
1.2:
Powered By Visual Paradigm Community Edition

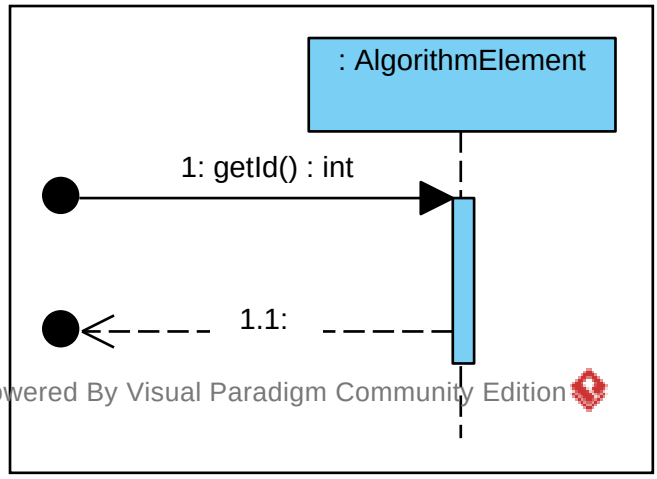

: AlgorithmElement
1: AlgorithmElement()
1.1:
Powered By Visual Paradigm Community Edition

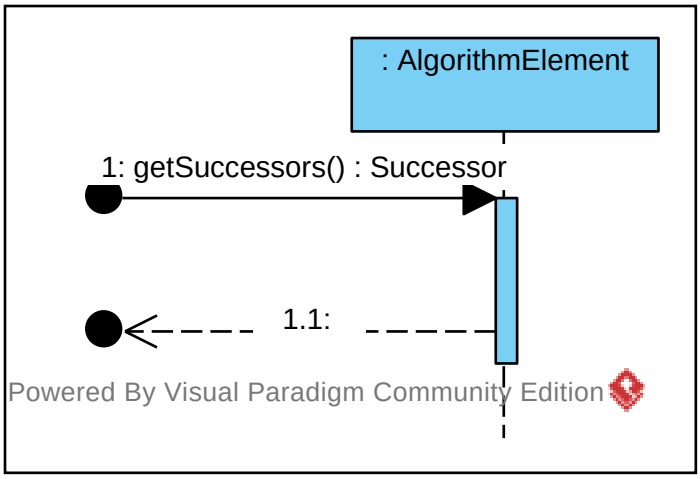

: AlgorithmElement
cription : String, successors : List<Successor>, b64Image : String, extraDescription : String)
1.1:
Powered By Visual Paradigm Community Edition

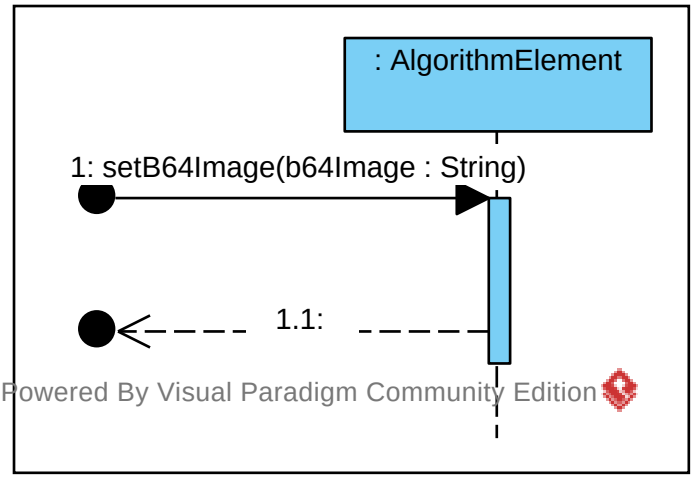

: AlgorithmElement
1: getB64Image() : String
1.1:
Powered By Visual Paradigm Community Edition

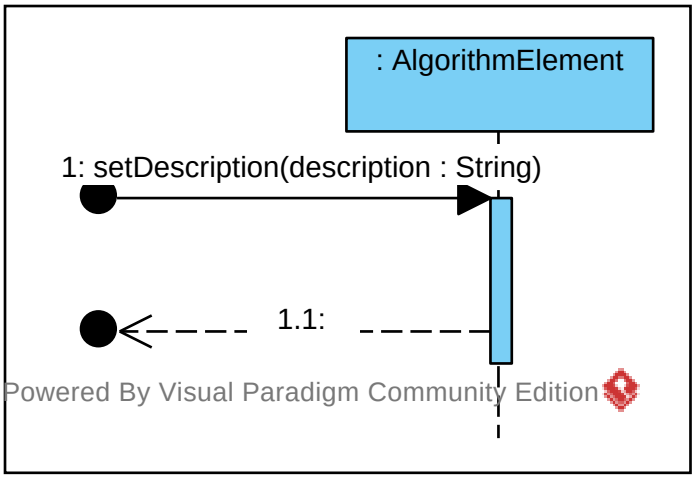

: AlgorithmElement
1: getDescription() : String
1.1:
Powered By Visual Paradigm Community Edition

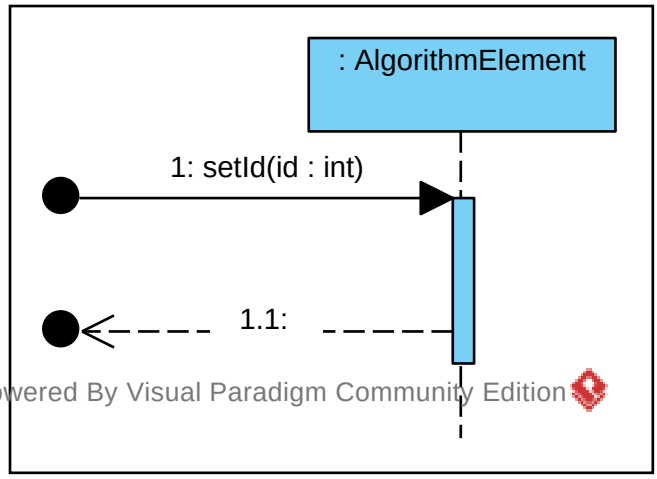

: AlgorithmElement
1: getExtraDescription() : String
1.1:
Powered By Visual Paradigm Community Edition

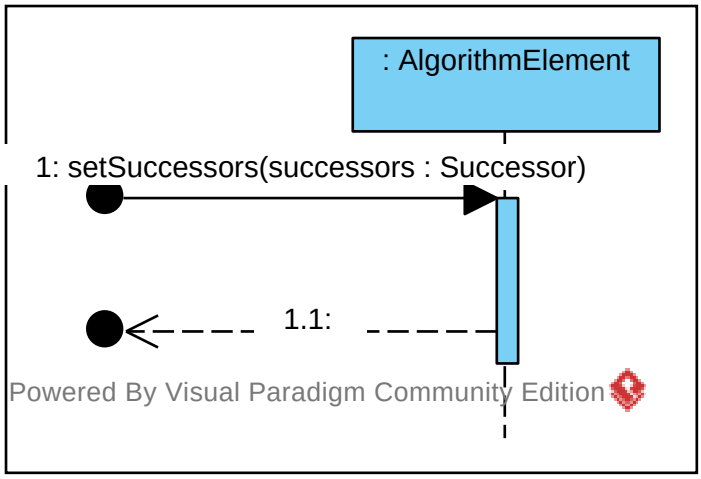

: AlgorithmElement
1: getId() : int
1.1:
Powered By Visual Paradigm Community Edition

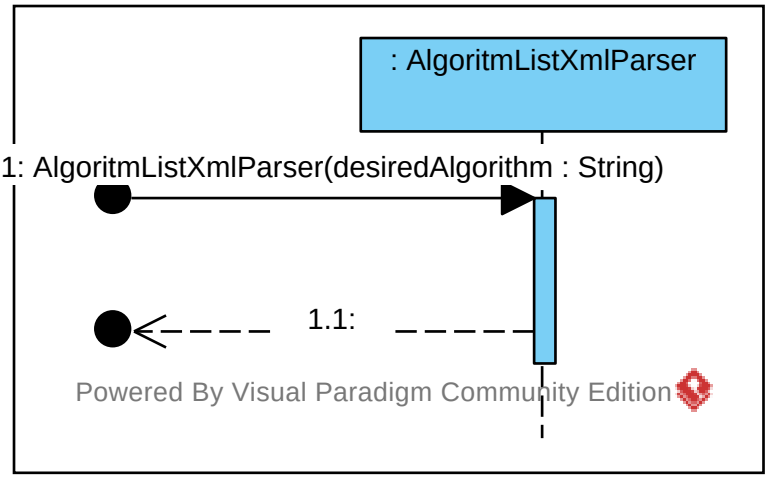

: AlgorithmElement
1: getSuccessors() : Successor
1.1:
Powered By Visual Paradigm Community Edition

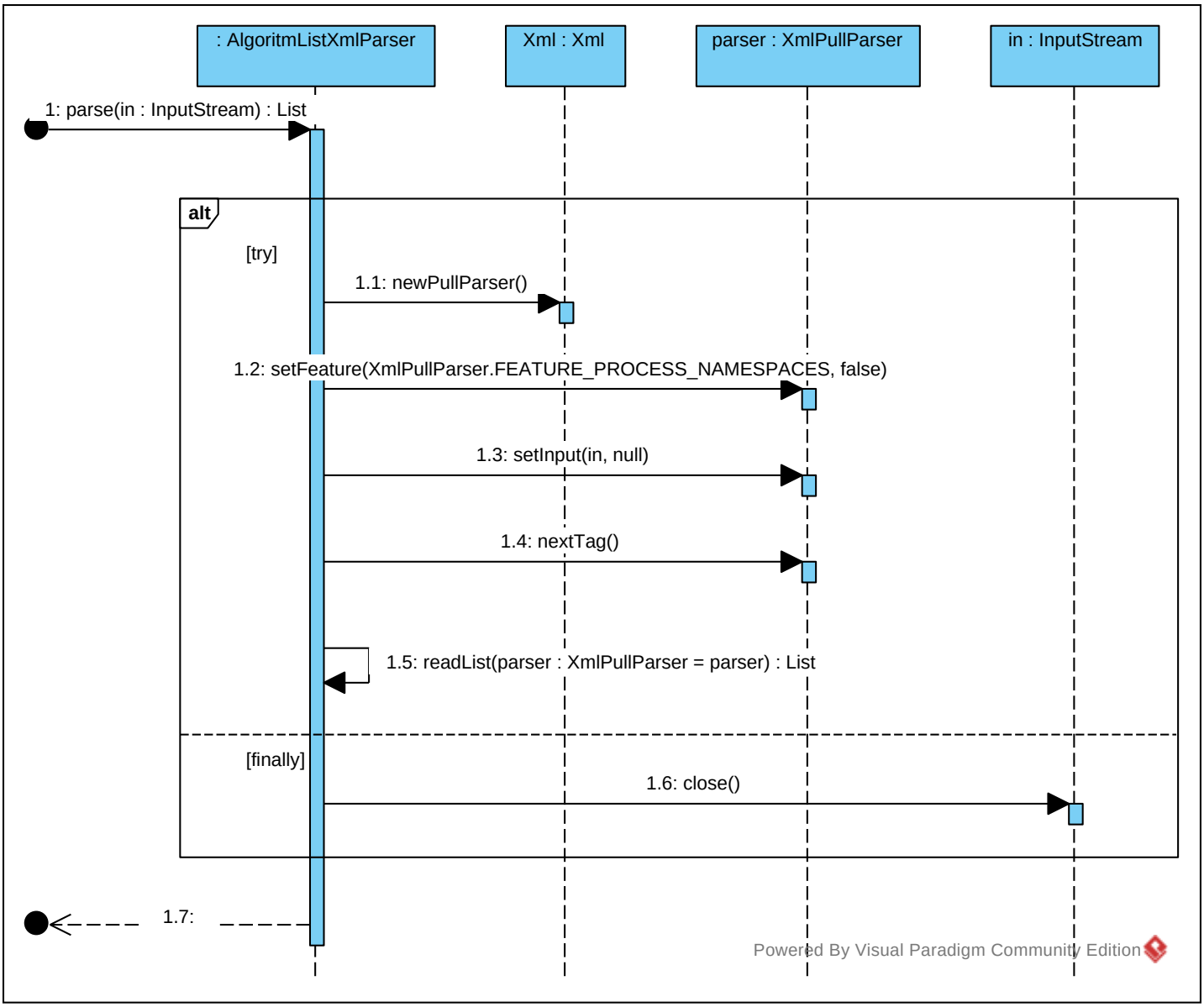

: AlgorithmElement
1: setB64Image(b64Image : String)
1.1:
Powered By Visual Paradigm Community Edition

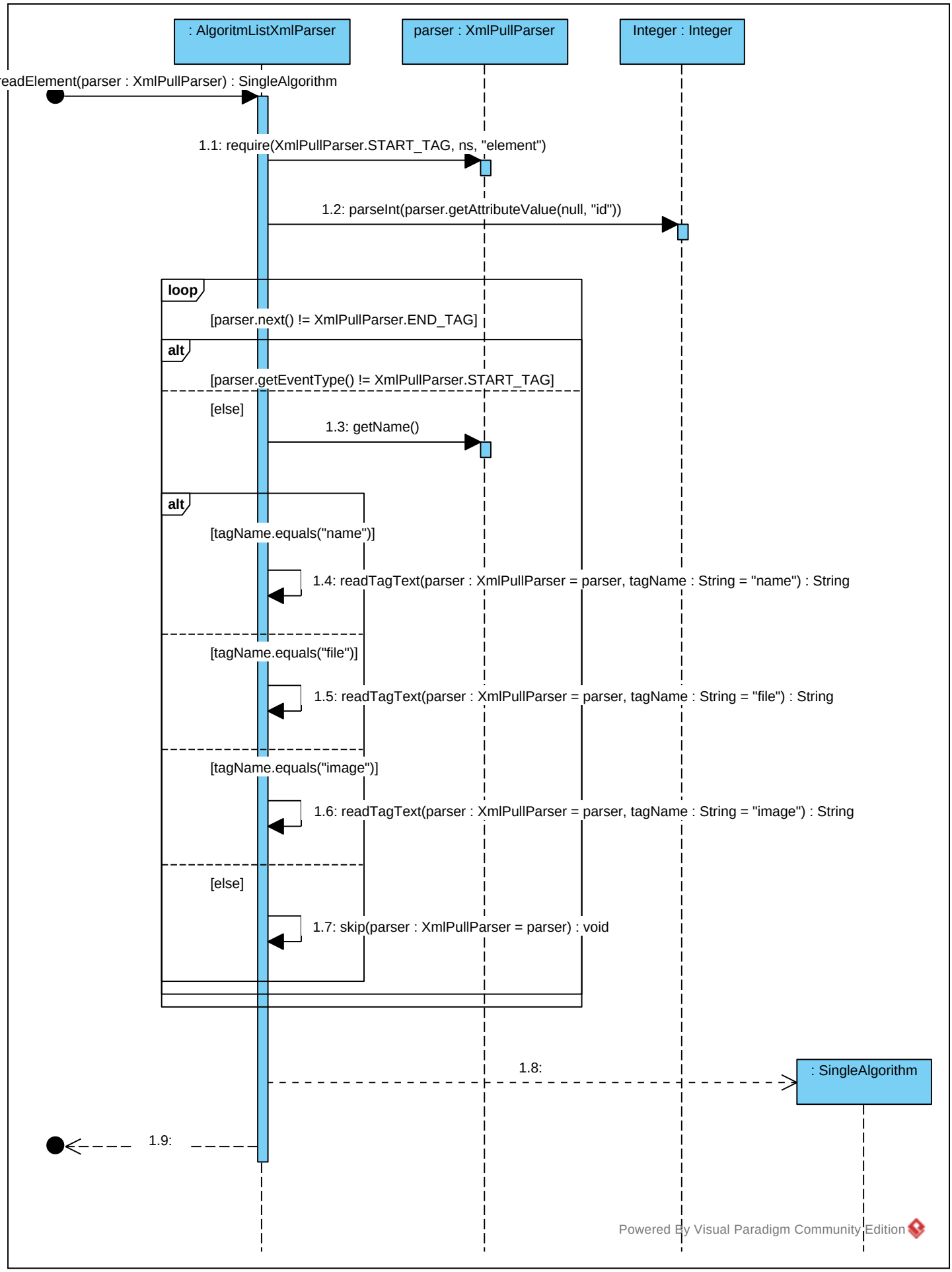

: AlgorithmElement
1: setDescription(description : String)
1.1:
Powered By Visual Paradigm Community Edition

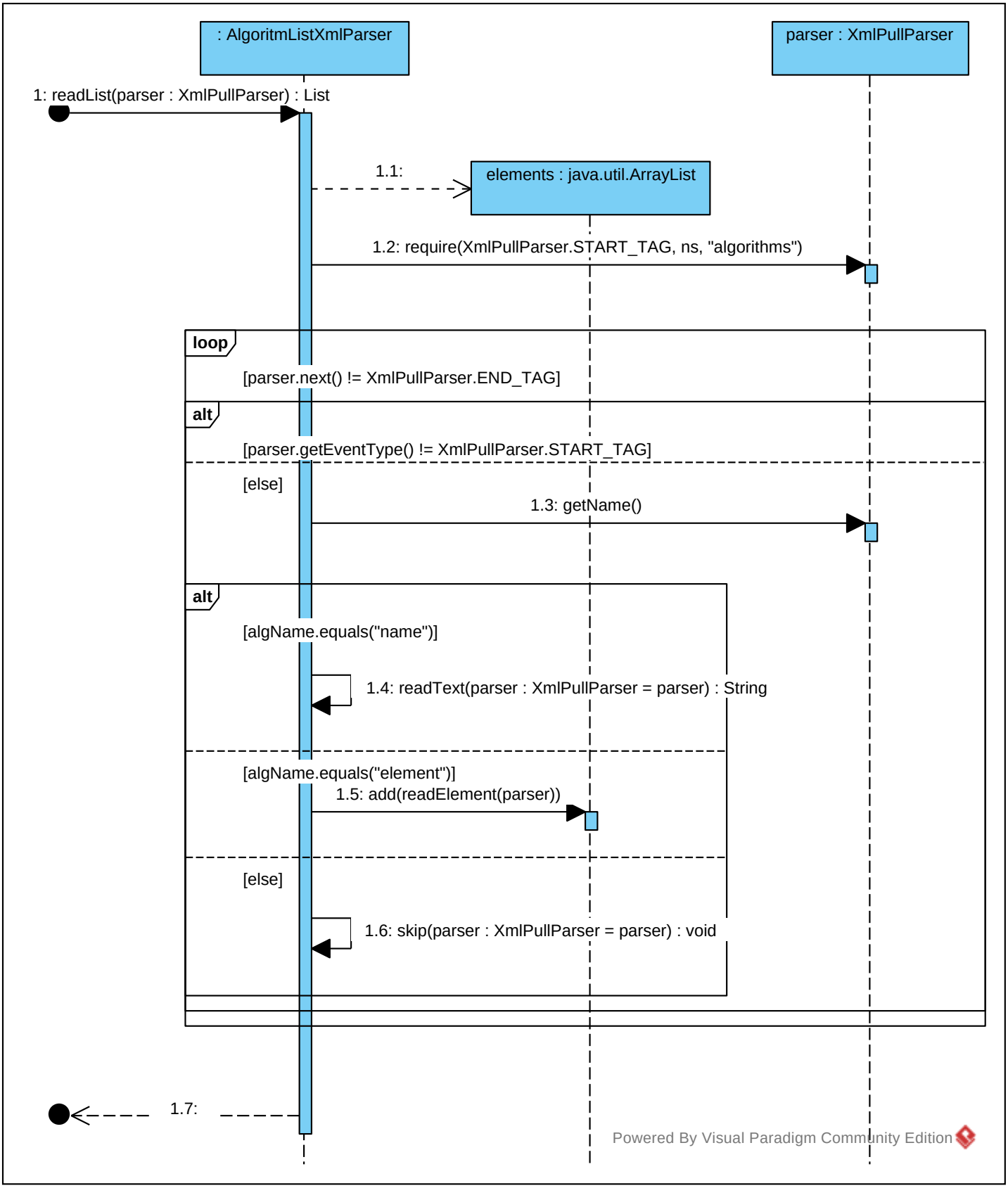

: AlgorithmElement
1: setId(id : int)
1.1:
Powered By Visual Paradigm Community Edition

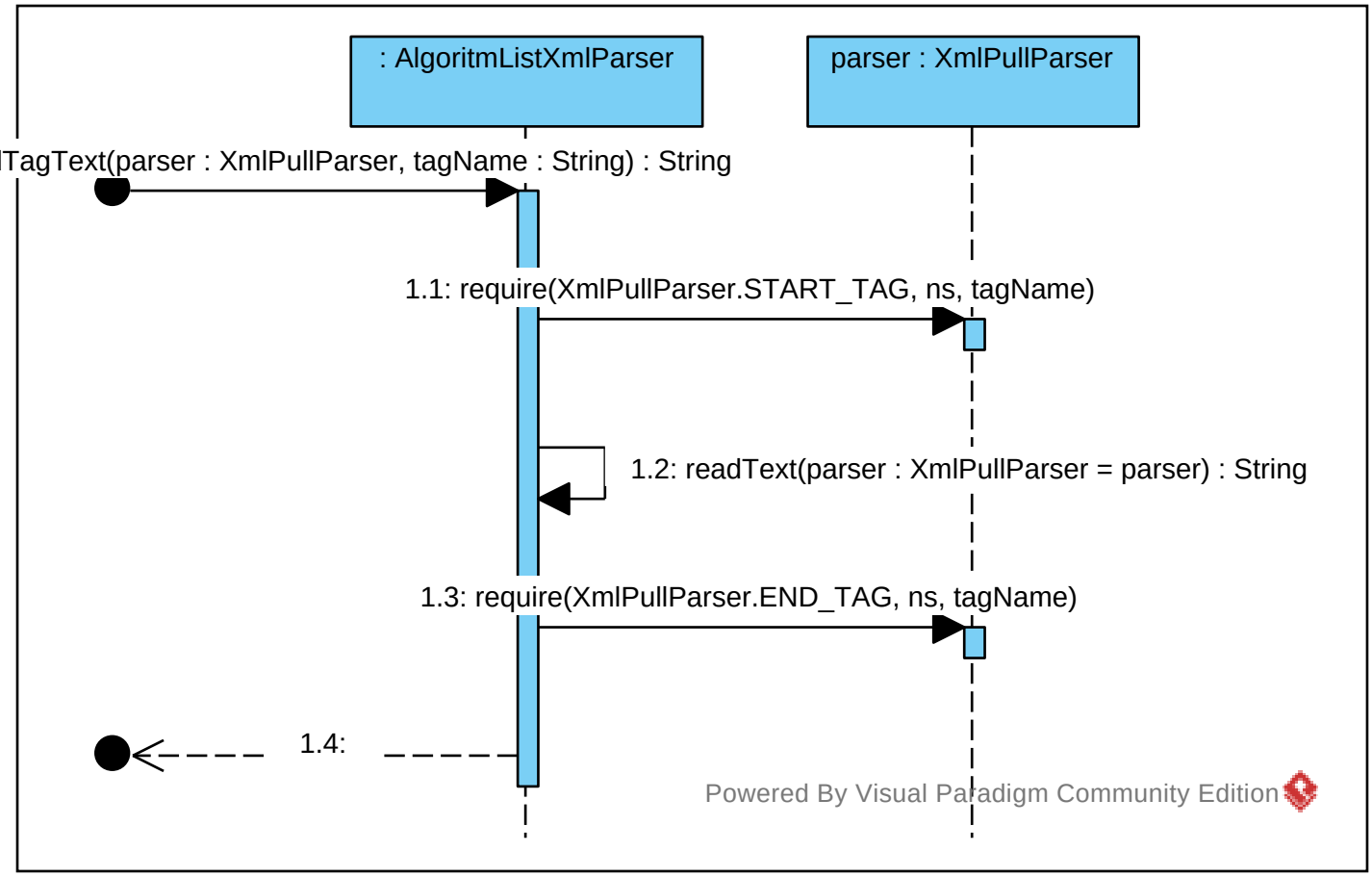

: AlgorithmElement
1: setSuccessors(successors : Successor)
1.1:
Powered By Visual Paradigm Community Edition

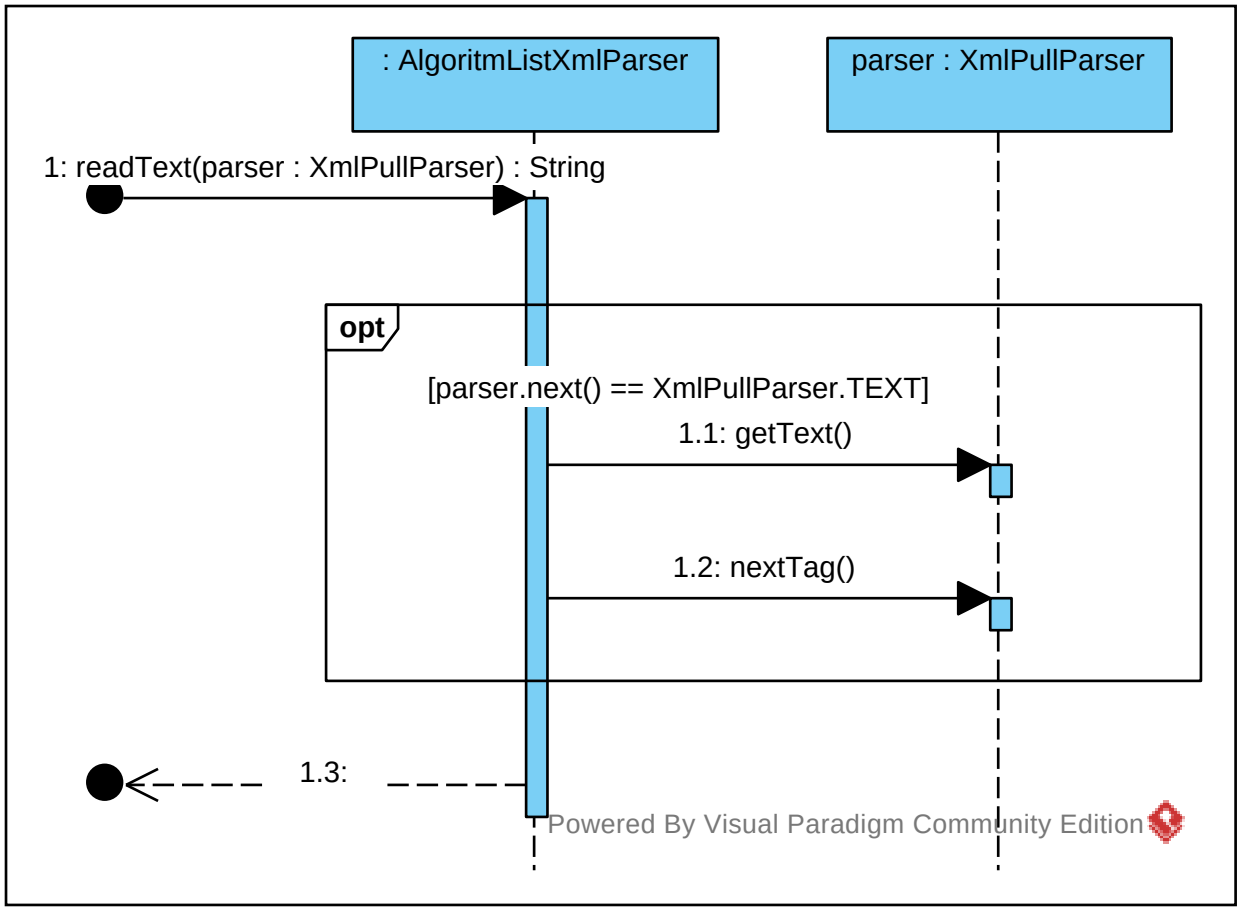

: AlgoritmListXmlParser
1: AlgoritmListXmlParser(desiredAlgorithm : String)
1.1:
Powered By Visual Paradigm Community Edition

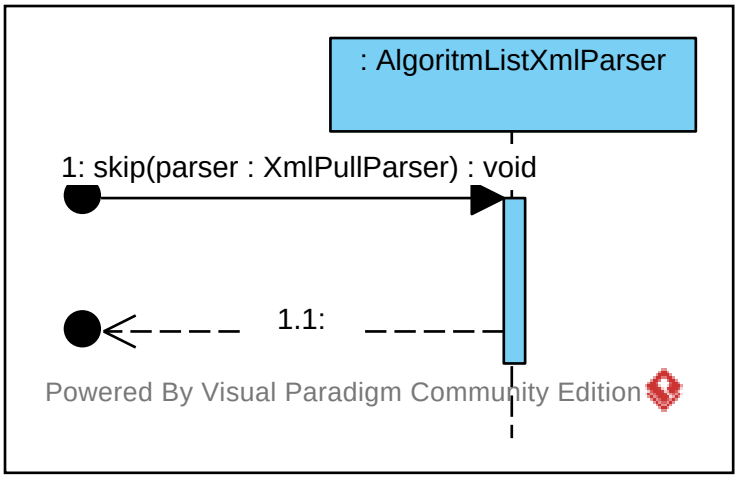

: AlgoritmListXmlParser
Xml : Xml
parser : XmlPullParser
in : InputStream
1: parse(in : InputStream) : List
alt
[try]
1.1: newPullParser()
1.2: setFeature(XmlPullParser.FEATURE_PROCESS_NAMESPACES, false)
1.3: setInput(in, null)
1.4: nextTag()
1.5: readList(parser : XmlPullParser = parser) : List
[finally]
1.6: close()
1.7:
Powered By Visual Paradigm Community Edition

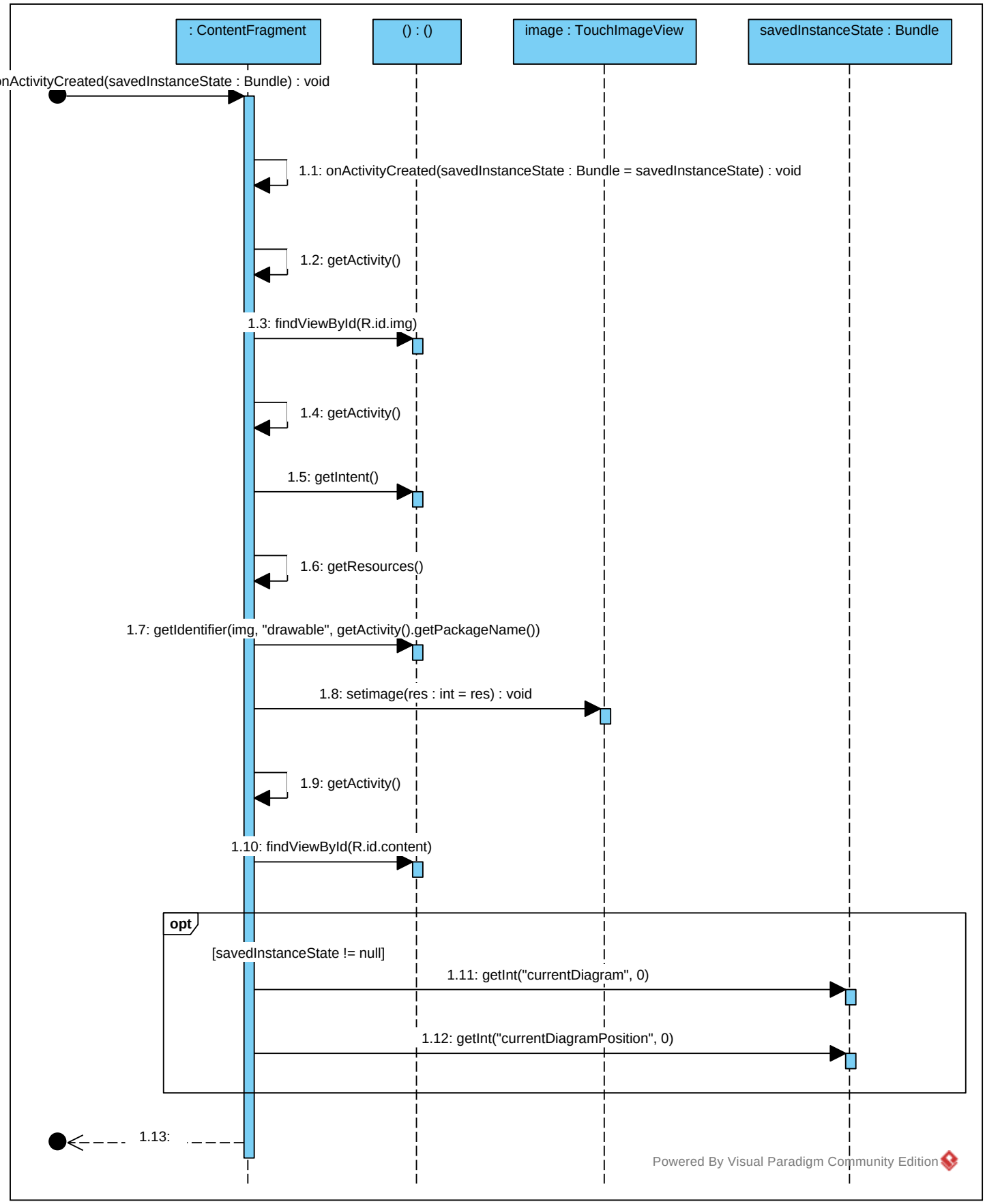

: AlgoritmListXmlParser
parser : XmlPullParser
Integer : Integer
1: readElement(parser : XmlPullParser) : SingleAlgorithm
1.1: require(XmlPullParser.START_TAG, ns, "element")
1.2: parseInt(parser.getAttributeValue(null, "id"))
loop
[parser.next() != XmlPullParser.END_TAG]
alt
[parser.getEventType() != XmlPullParser.START_TAG]
[else]
1.3: getName()
alt
[tagName.equals("name")]
1.4: readTagText(parser : XmlPullParser = parser, tagName : String = "name") : String
[tagName.equals("file")]
1.5: readTagText(parser : XmlPullParser = parser, tagName : String = "file") : String
[tagName.equals("image")]
1.6: readTagText(parser : XmlPullParser = parser, tagName : String = "image") : String
[else]
1.7: skip(parser : XmlPullParser = parser) : void
1.8:
: SingleAlgorithm
1.9:
Powered By Visual Paradigm Community Edition

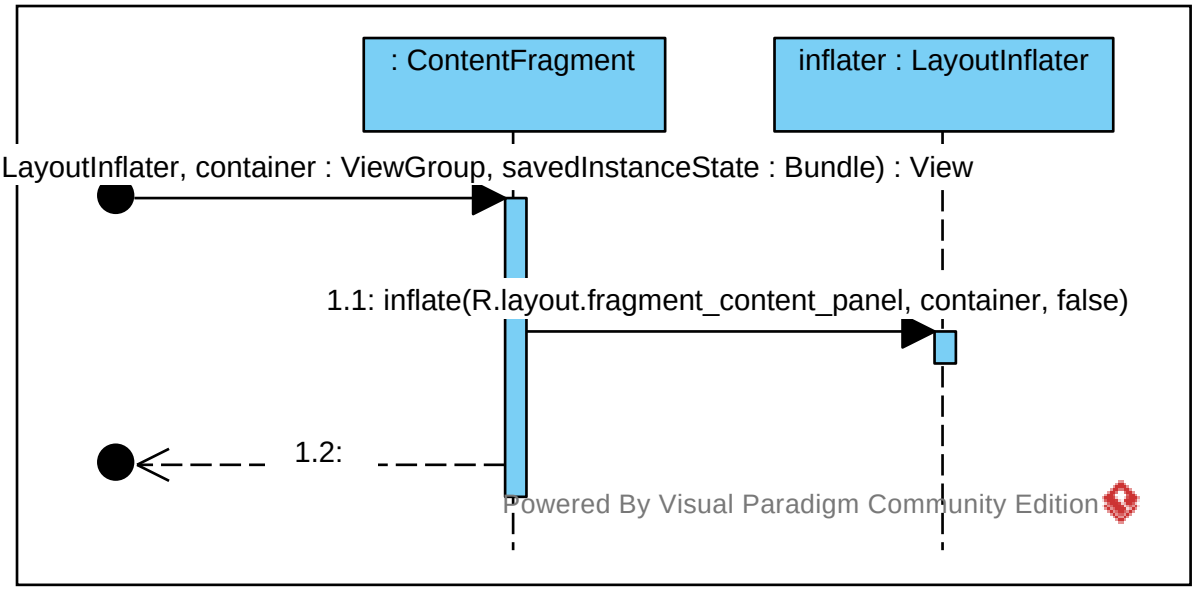

: AlgoritmListXmlParser
parser : XmlPullParser
1: readList(parser : XmlPullParser) : List
1.1:
elements : java.util.ArrayList
1.2: require(XmlPullParser.START_TAG, ns, "algorithms")
loop
[parser.next() != XmlPullParser.END_TAG]
alt
[parser.getEventType() != XmlPullParser.START_TAG]
[else]
1.3: getName()
alt
[algName.equals("name")]
1.4: readText(parser : XmlPullParser = parser) : String
[algName.equals("element")]
1.5: add(readElement(parser))
[else]
1.6: skip(parser : XmlPullParser = parser) : void
1.7:
Powered By Visual Paradigm Community Edition

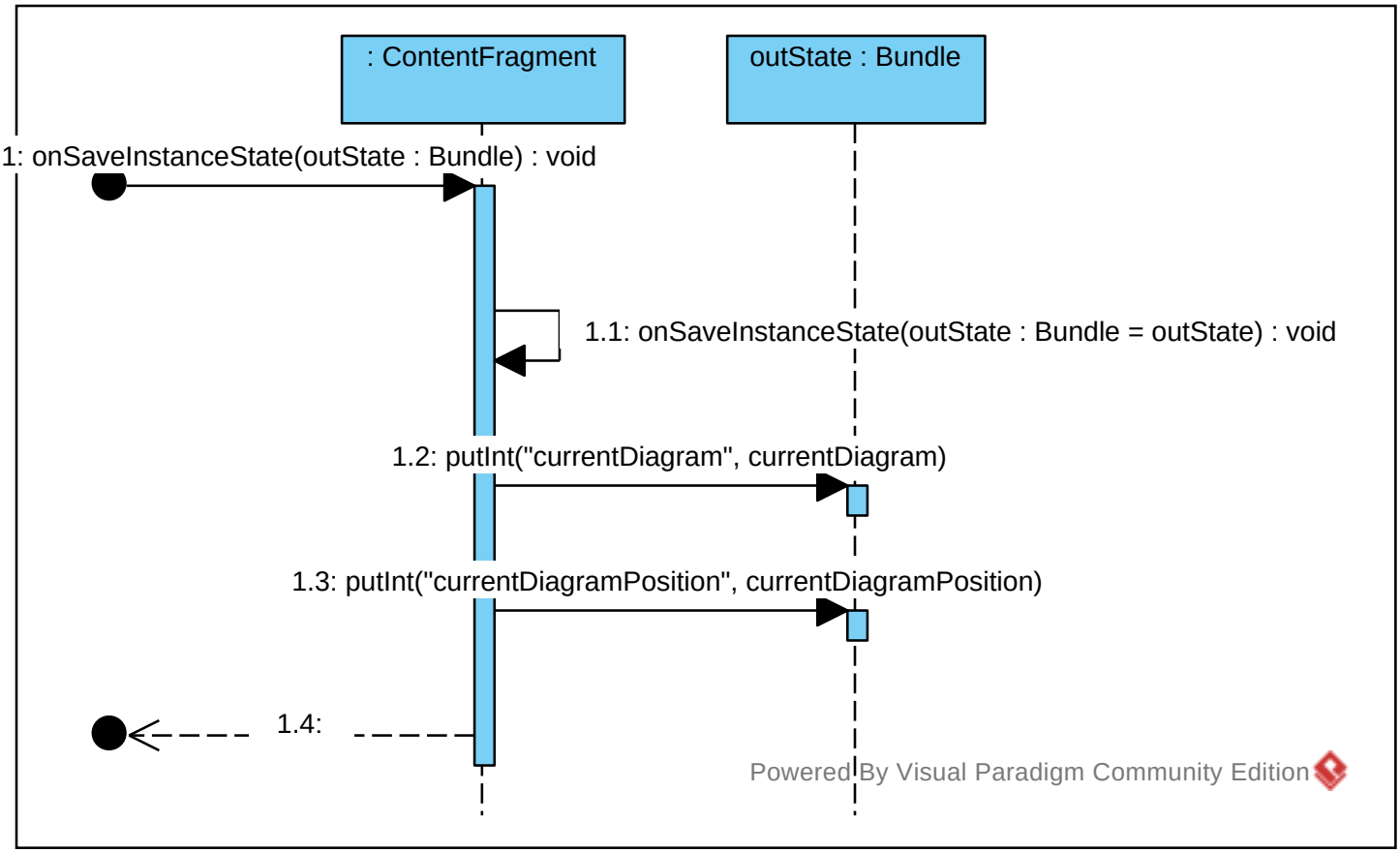

: AlgoritmListXmlParser
parser : XmlPullParser
readTagText(parser : XmlPullParser, tagName : String) : String
1.1: require(XmlPullParser.START_TAG, ns, tagName)
1.2: readText(parser : XmlPullParser = parser) : String
1.3: require(XmlPullParser.END_TAG, ns, tagName)
1.4:
Powered By Visual Paradigm Community Edition

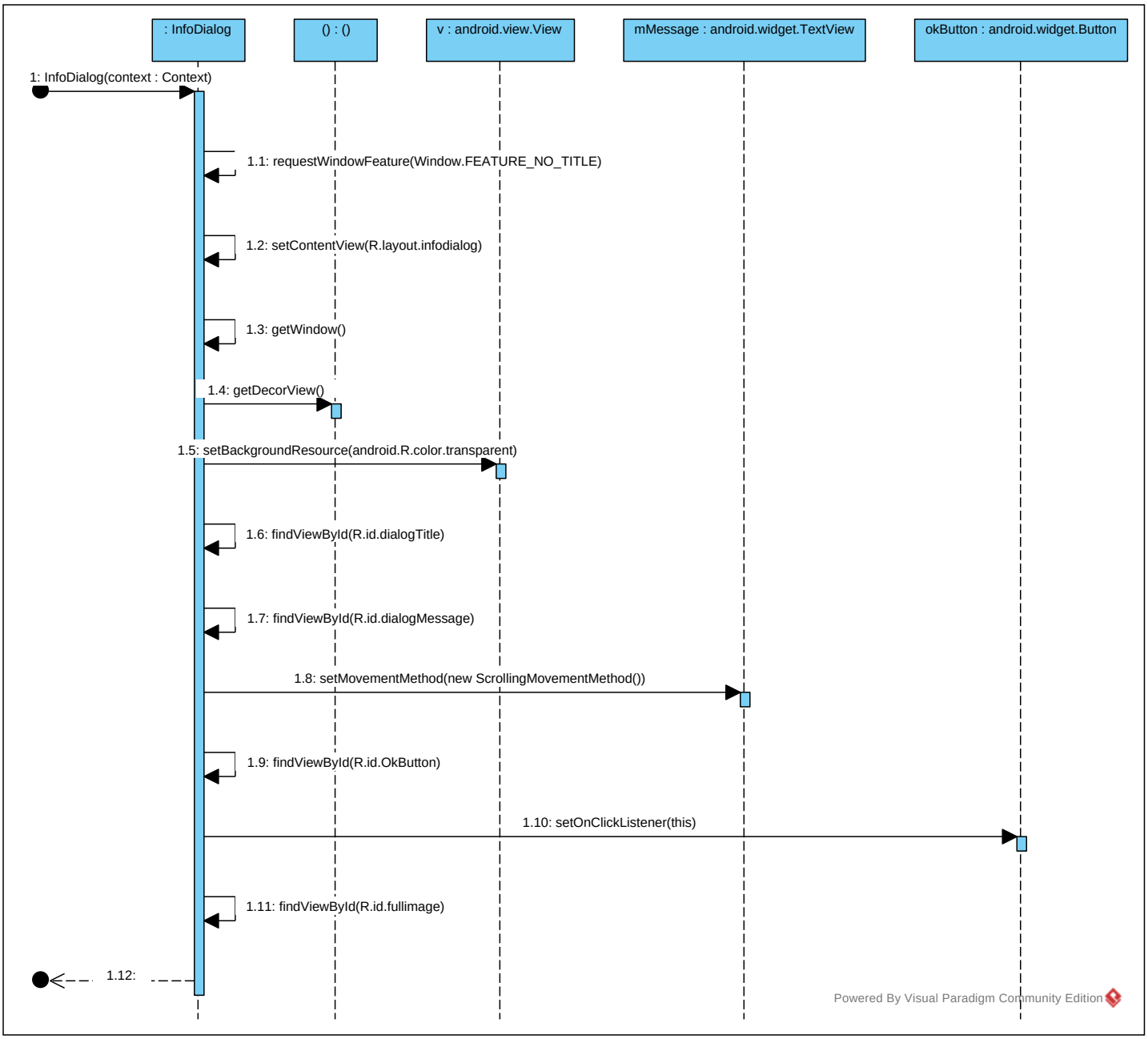

: AlgoritmListXmlParser
parser : XmlPullParser
1: readText(parser : XmlPullParser) : String
opt
[parser.next() == XmlPullParser.TEXT]
1.1: getText()
1.2: nextTag()
1.3:
Powered By Visual Paradigm Community Edition

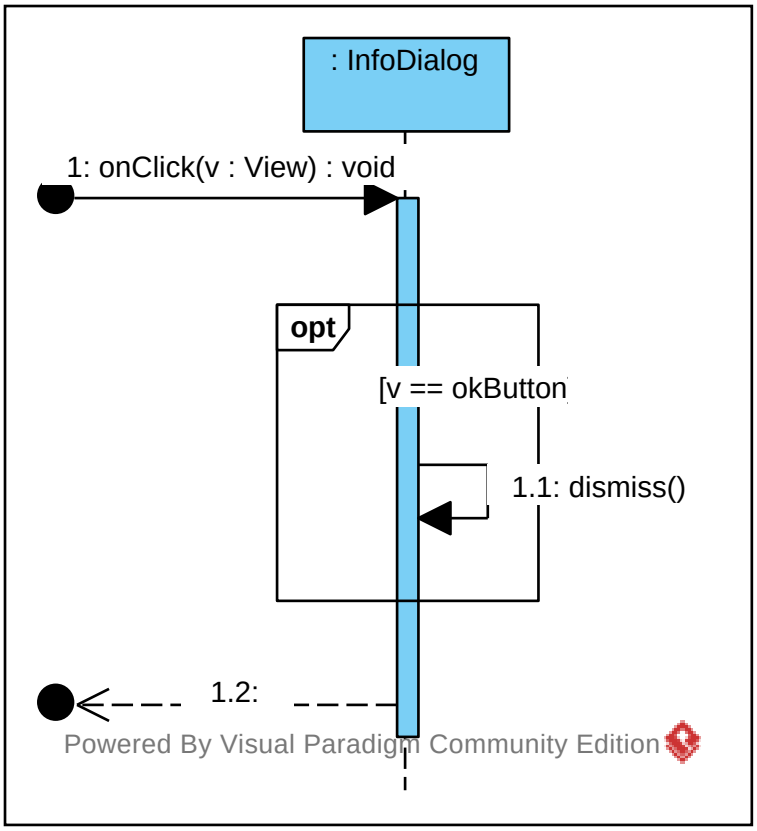

: AlgoritmListXmlParser
1: skip(parser : XmlPullParser) : void
1.1:
Powered By Visual Paradigm Community Edition

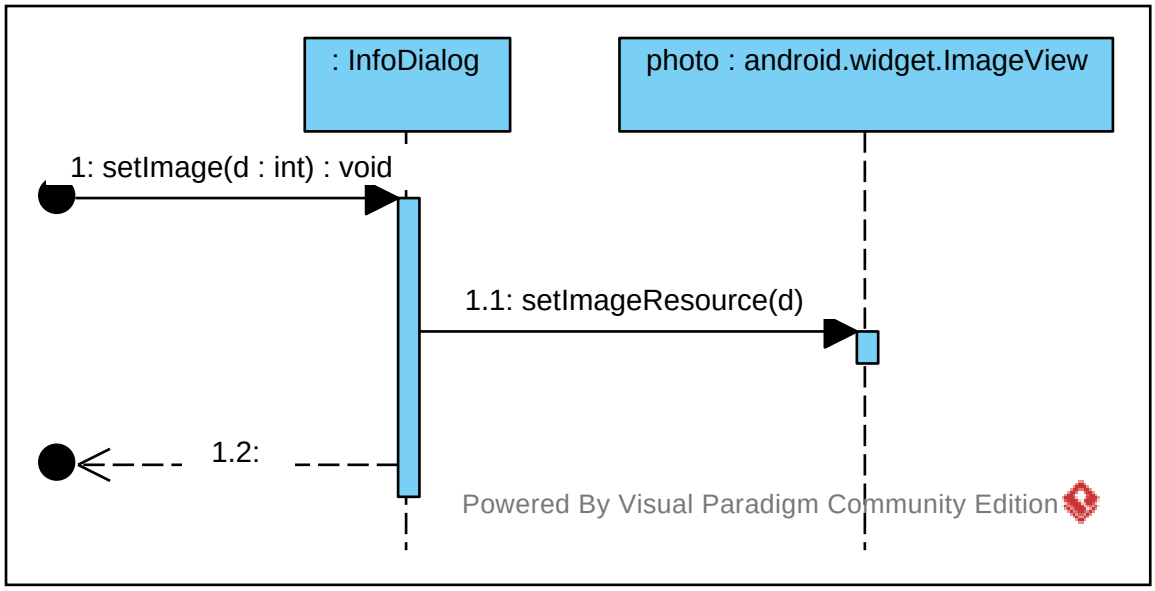

: ContentFragment
() : ()
image : TouchImageView
savedInstanceState : Bundle
1: onActivityCreated(savedInstanceState : Bundle) : void
1.1: onActivityCreated(savedInstanceState : Bundle = savedInstanceState) : void
1.2: getActivity()
1.3: findViewById(R.id.img)
1.4: getActivity()
1.5: getIntent()
1.6: getResources()
1.7: getIdentifier(img, "drawable", getActivity().getPackageName())
1.8: setimage(res : int = res) : void
1.9: getActivity()
1.10: findViewById(R.id.content)
opt
[savedInstanceState != null]
1.11: getInt("currentDiagram", 0)
1.12: getInt("currentDiagramPosition", 0)
1.13:
Powered By Visual Paradigm Community Edition

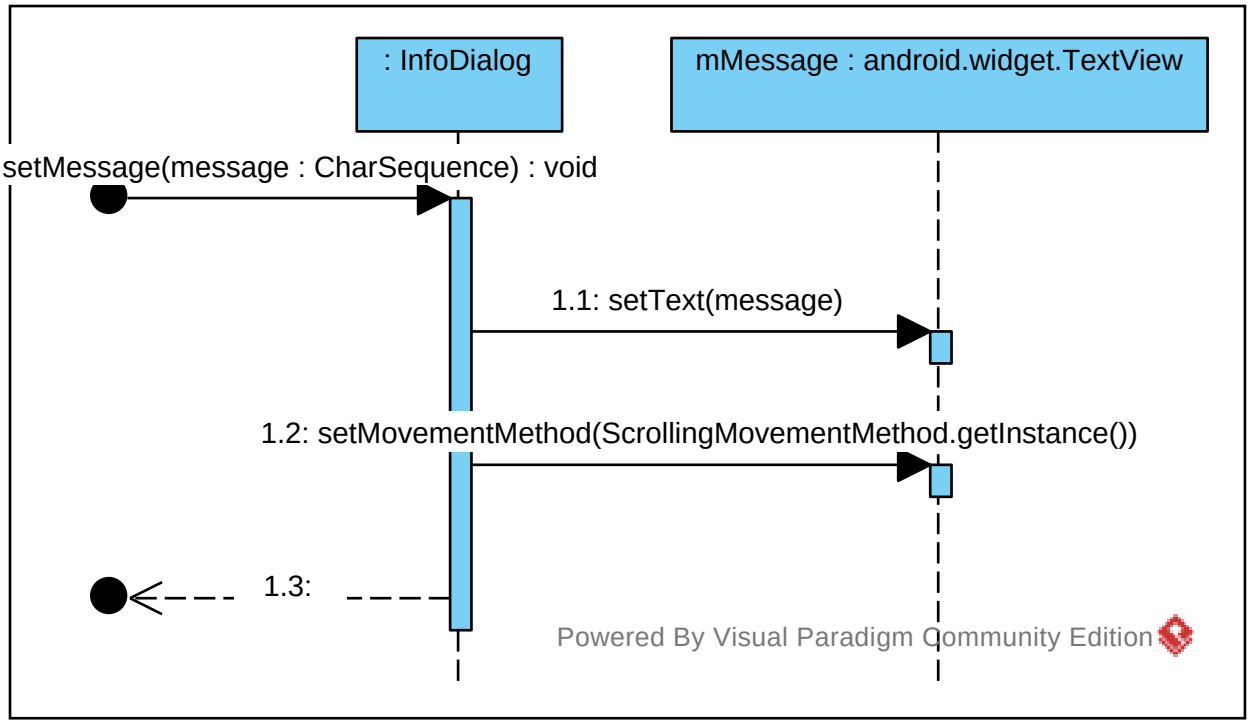

: ContentFragment
inflater : LayoutInflater
ater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View
1.1: inflate(R.layout.fragment_content_panel, container, false)
1.2:
Powered By Visual Paradigm Community Edition

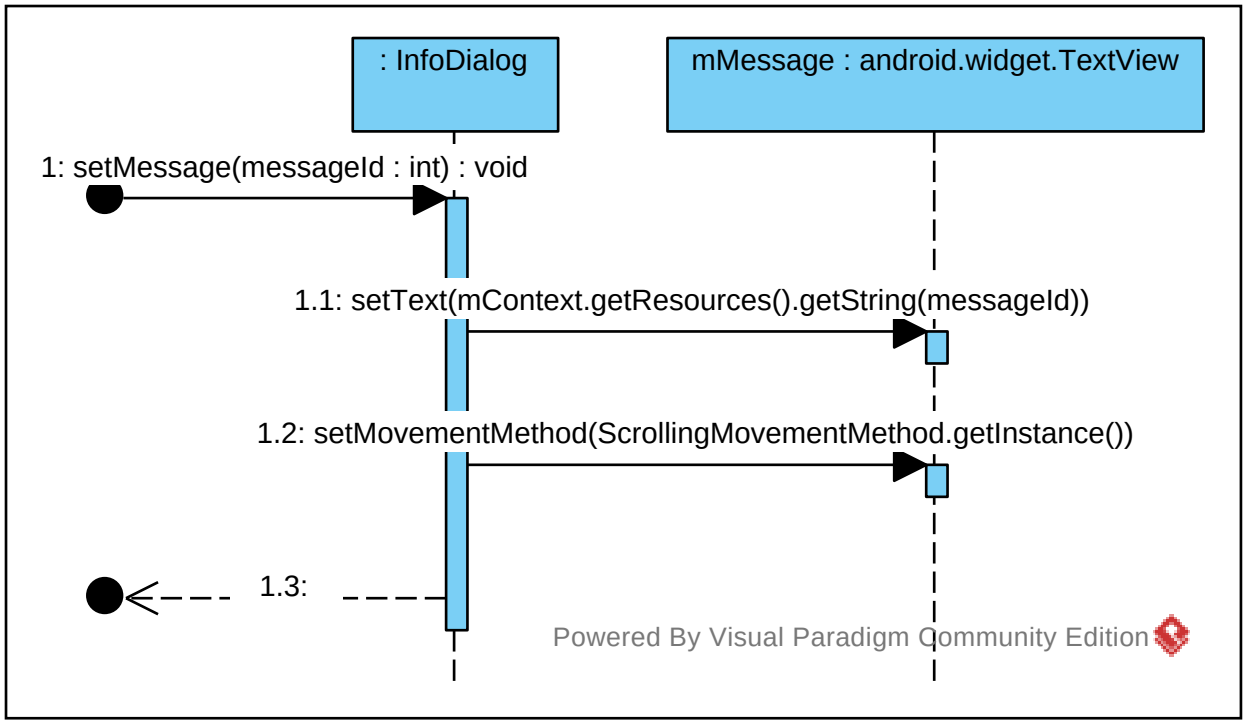

: ContentFragment
outState : Bundle
1: onSaveInstanceState(outState : Bundle) : void
1.1: onSaveInstanceState(outState : Bundle = outState) : void
1.2: putInt("currentDiagram", currentDiagram)
1.3: putInt("currentDiagramPosition", currentDiagramPosition)
1.4:
Powered By Visual Paradigm Community Edition

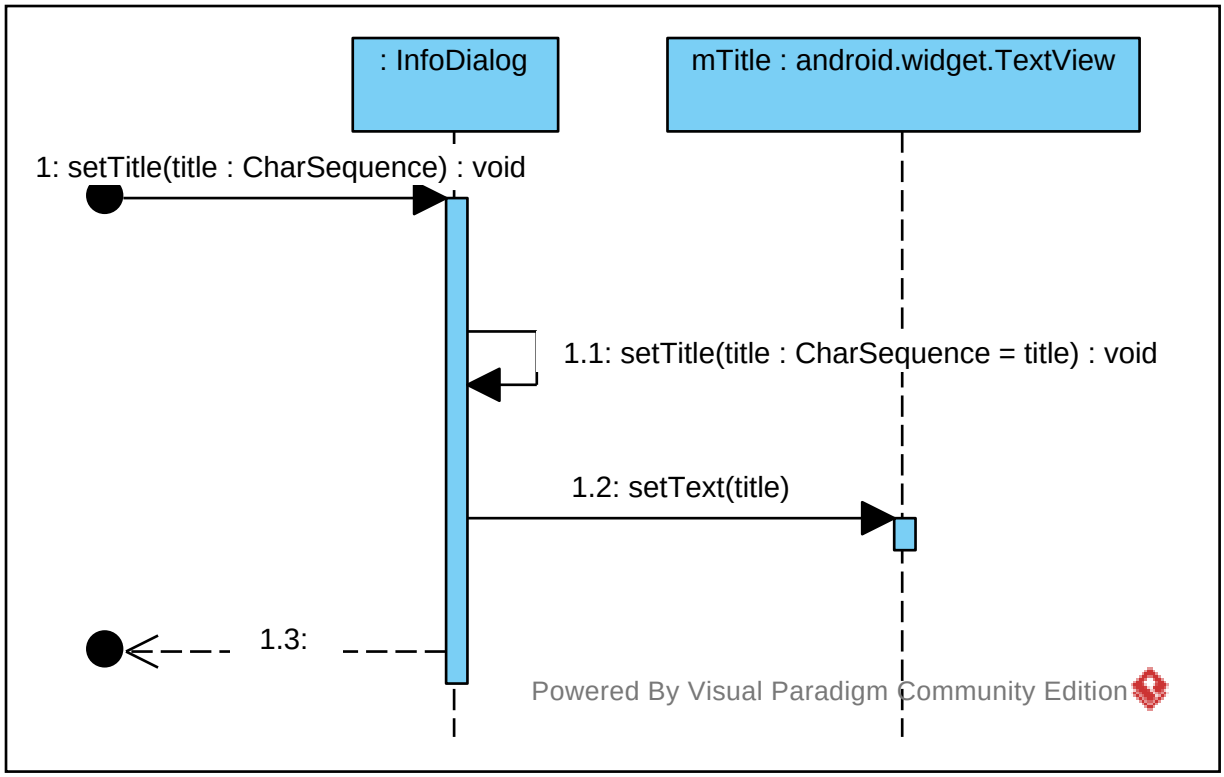

: InfoDialog
() : ()
v : android.view.View
mMessage : android.widget.TextView
okButton : android.widget.Button
1: InfoDialog(context : Context)
1.1: requestWindowFeature(Window.FEATURE_NO_TITLE)
1.2: setContentView(R.layout.infodialog)
1.3: getWindow()
1.4: getDecorView()
1.5: setBackgroundResource(android.R.color.transparent)
1.6: findViewById(R.id.dialogTitle)
1.7: findViewById(R.id.dialogMessage)
1.8: setMovementMethod(new ScrollingMovementMethod())
1.9: findViewById(R.id.OkButton)
1.10: setOnClickListener(this)
1.11: findViewById(R.id.fullimage)
1.12:
Powered By Visual Paradigm Community Edition

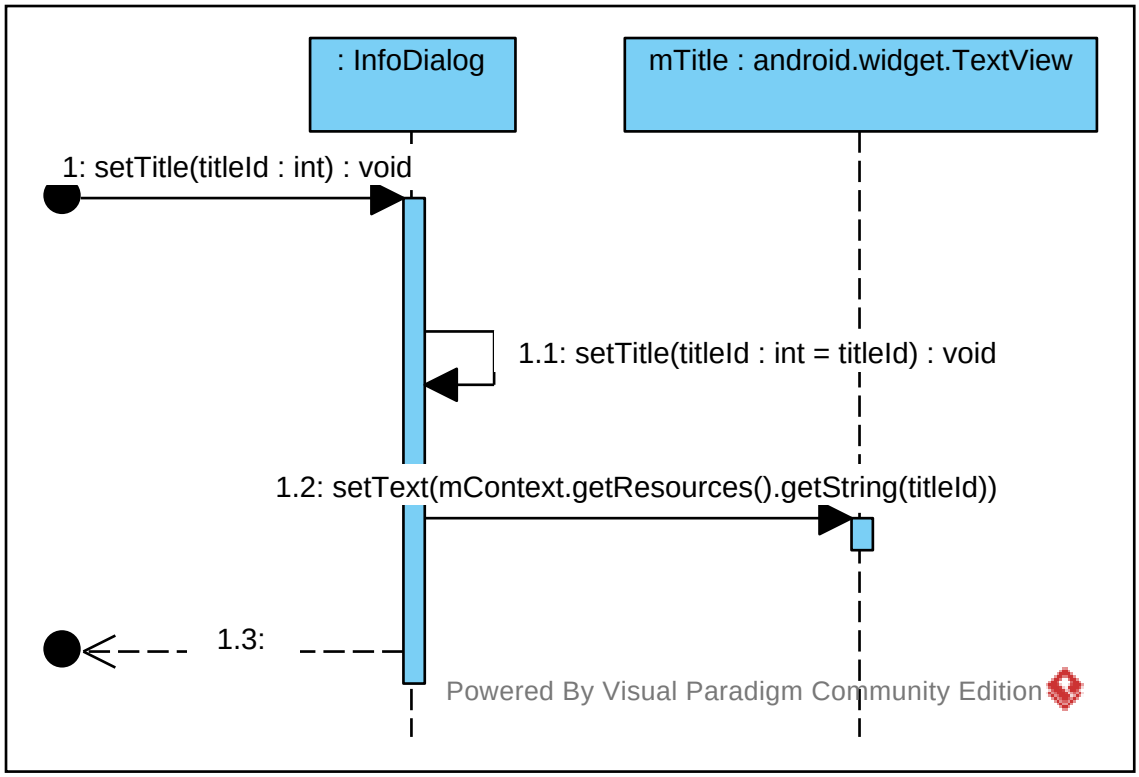

: InfoDialog
1: onClick(v : View) : void
opt
[v == okButton
1.1: dismiss()
1.2:
Powered By Visual Paradigm Community Edition

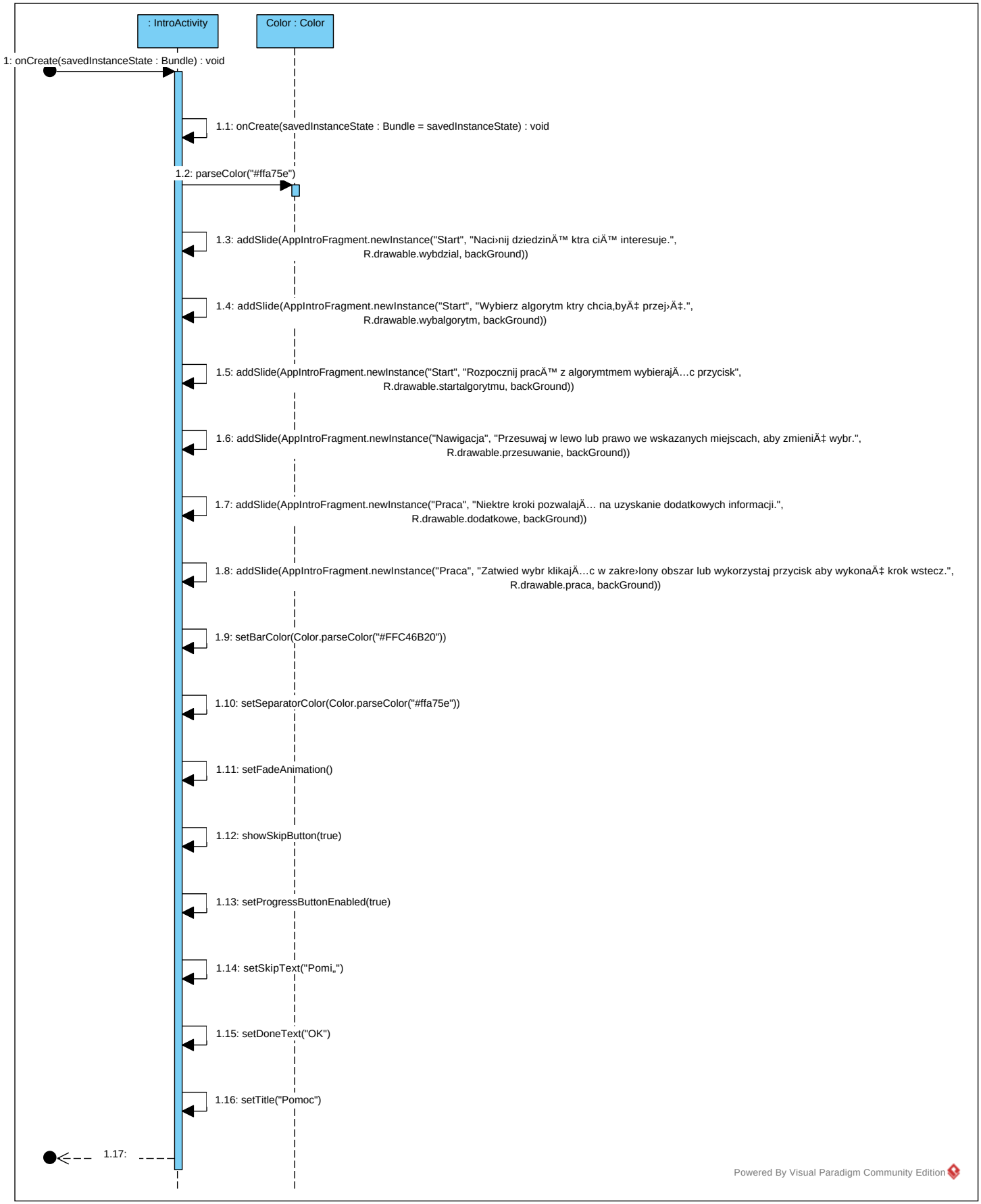

: InfoDialog
photo : android.widget.ImageView
1: setImage(d : int) : void
1.1: setImageResource(d)
1.2:
Powered By Visual Paradigm Community Edition

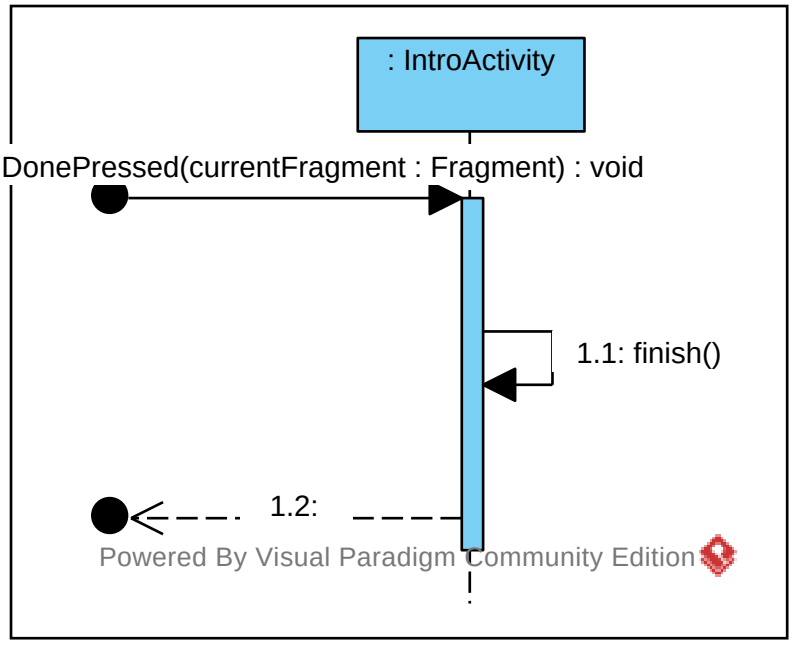

: InfoDialog
mMessage : android.widget.TextView
1: setMessage(message : CharSequence) : void
1.1: setText(message)
1.2: setMovementMethod(ScrollingMovementMethod.getInstance())
1.3:
Powered By Visual Paradigm Community Edition

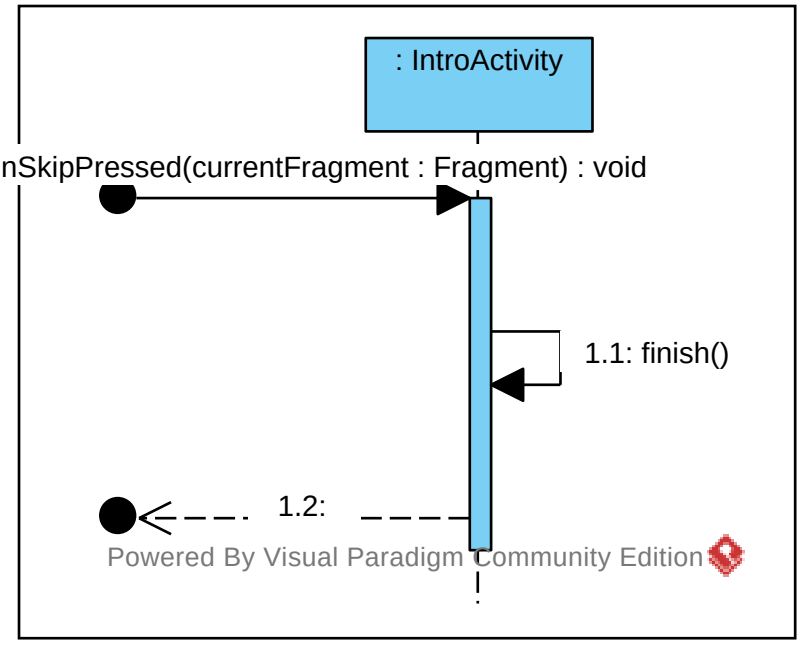

: InfoDialog
mMessage : android.widget.TextView
1: setMessage(messageId : int) : void
1.1: setText(mContext.getResources().getString(messageId))
1.2: setMovementMethod(ScrollingMovementMethod.getInstance())
1.3:
Powered By Visual Paradigm Community Edition

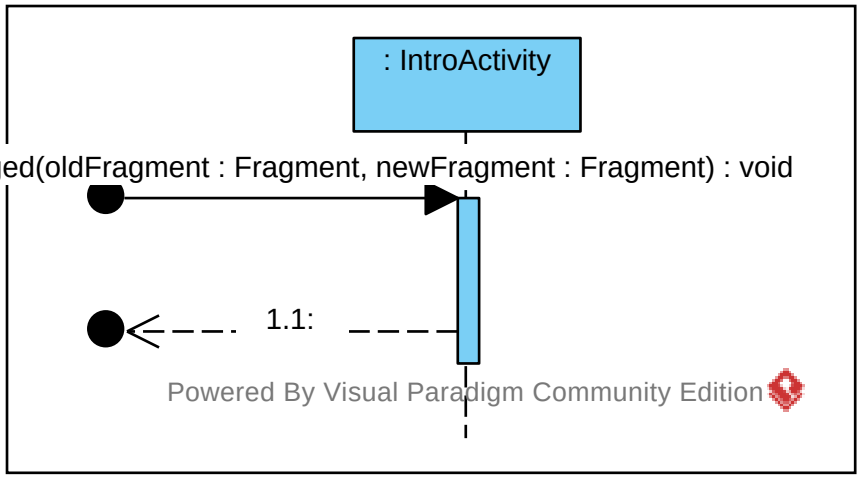

: InfoDialog
mTitle : android.widget.TextView
1: setTitle(title : CharSequence) : void
1.1: setTitle(title : CharSequence = title) : void
1.2: setText(title)
1.3:
Powered By Visual Paradigm Community Edition

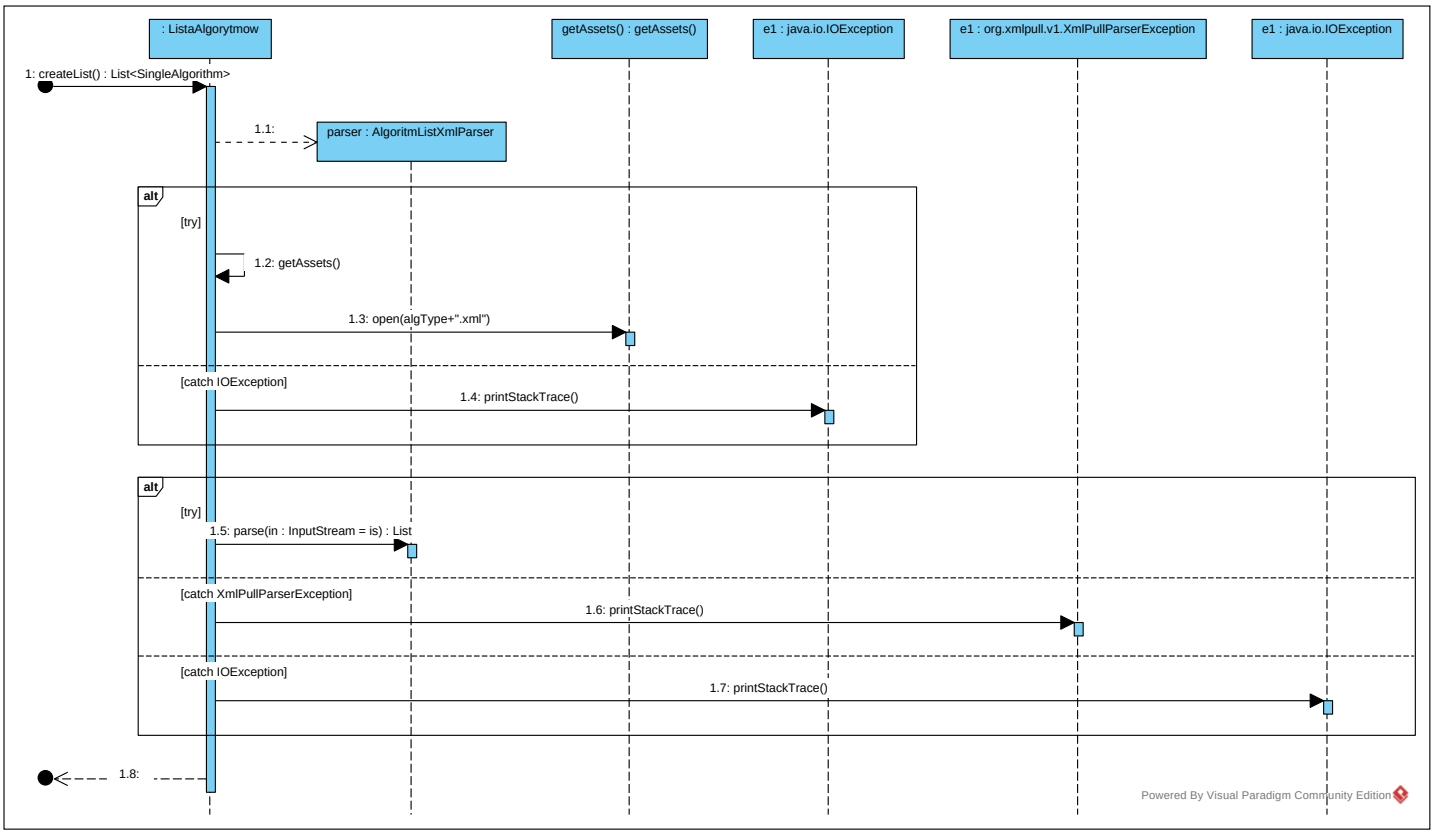

: InfoDialog
mTitle : android.widget.TextView
1: setTitle(titleId : int) : void
1.1: setTitle(titleId : int = titleId) : void
1.2: setText(mContext.getResources().getString(titleId))
1.3:
Powered By Visual Paradigm Community Edition

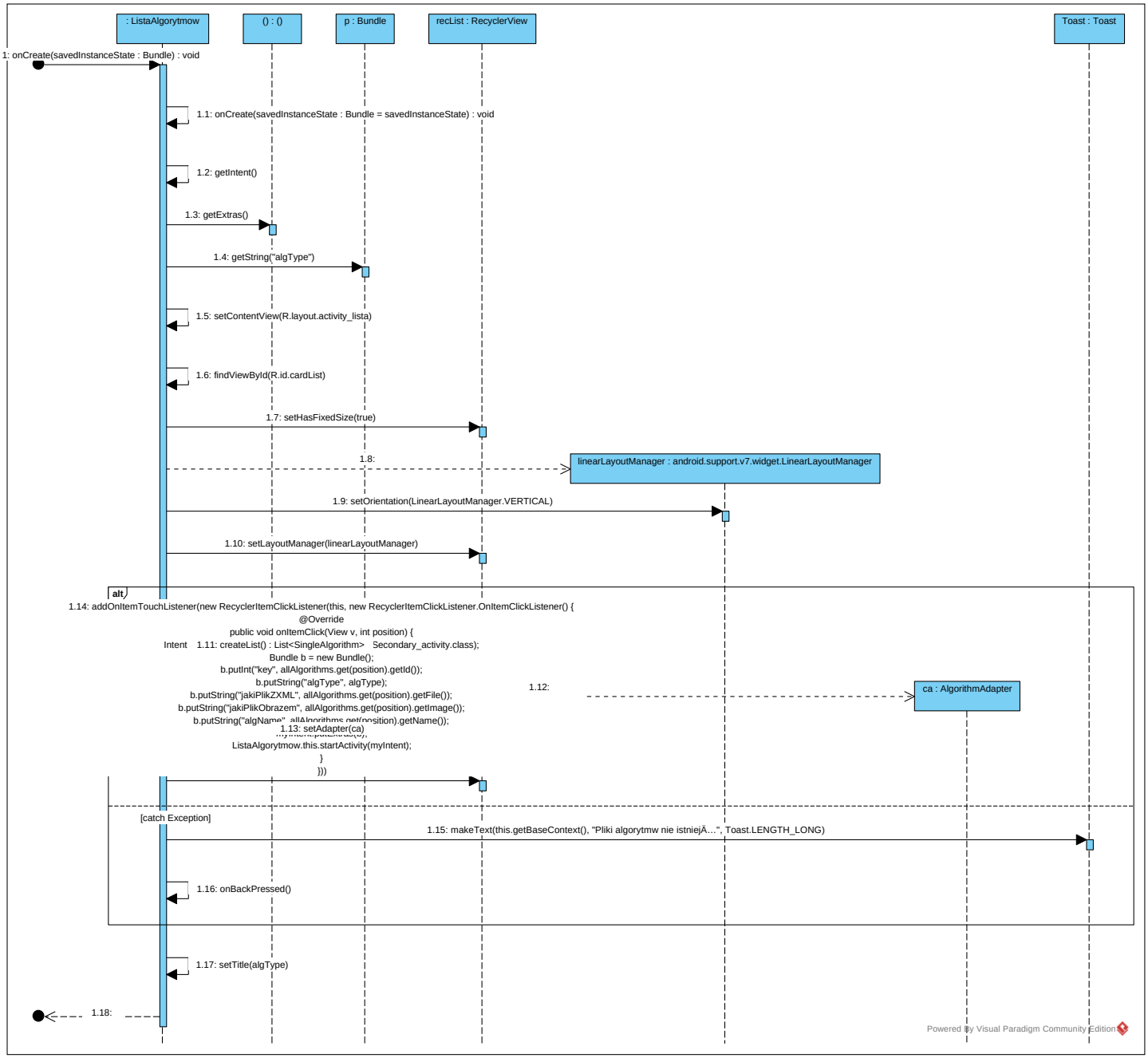

: IntroActivity
Color : Color
1: onCreate(savedInstanceState : Bundle) : void
1.1: onCreate(savedInstanceState : Bundle = savedInstanceState) : void
1.2: parseColor("#ffa75e")
1.3: addSlide(AppIntroFragment.newInstance("Start", "Naci›nij dziedzinÄ™ ktra ciÄ™ interesuje.",
R.drawable.wybdzial, backGround))
1.4: addSlide(AppIntroFragment.newInstance("Start", "Wybierz algorytm ktry chcia,byÄ‡ przej›Ä‡.",
R.drawable.wybalgorytm, backGround))
1.5: addSlide(AppIntroFragment.newInstance("Start", "Rozpocznij pracÄ™ z algorymtmem wybierajÄ…c przycisk",
R.drawable.startalgorytmu, backGround))
1.6: addSlide(AppIntroFragment.newInstance("Nawigacja", "Przesuwaj w lewo lub prawo we wskazanych miejscach, aby zmieniÄ‡ wybr.",
R.drawable.przesuwanie, backGround))
1.7: addSlide(AppIntroFragment.newInstance("Praca", "Niektre kroki pozwalajÄ… na uzyskanie dodatkowych informacji.",
R.drawable.dodatkowe, backGround))
1.8: addSlide(AppIntroFragment.newInstance("Praca", "Zatwied wybr klikajÄ…c w zakre›lony obszar lub wykorzystaj przycisk aby wykonaÄ‡ krok wstecz.",
R.drawable.praca, backGround))
1.9: setBarColor(Color.parseColor("#FFC46B20"))
1.10: setSeparatorColor(Color.parseColor("#ffa75e"))
1.11: setFadeAnimation()
1.12: showSkipButton(true)
1.13: setProgressButtonEnabled(true)
1.14: setSkipText("Pomi„")
1.15: setDoneText("OK")
1.16: setTitle("Pomoc")
1.17:
Powered By Visual Paradigm Community Edition

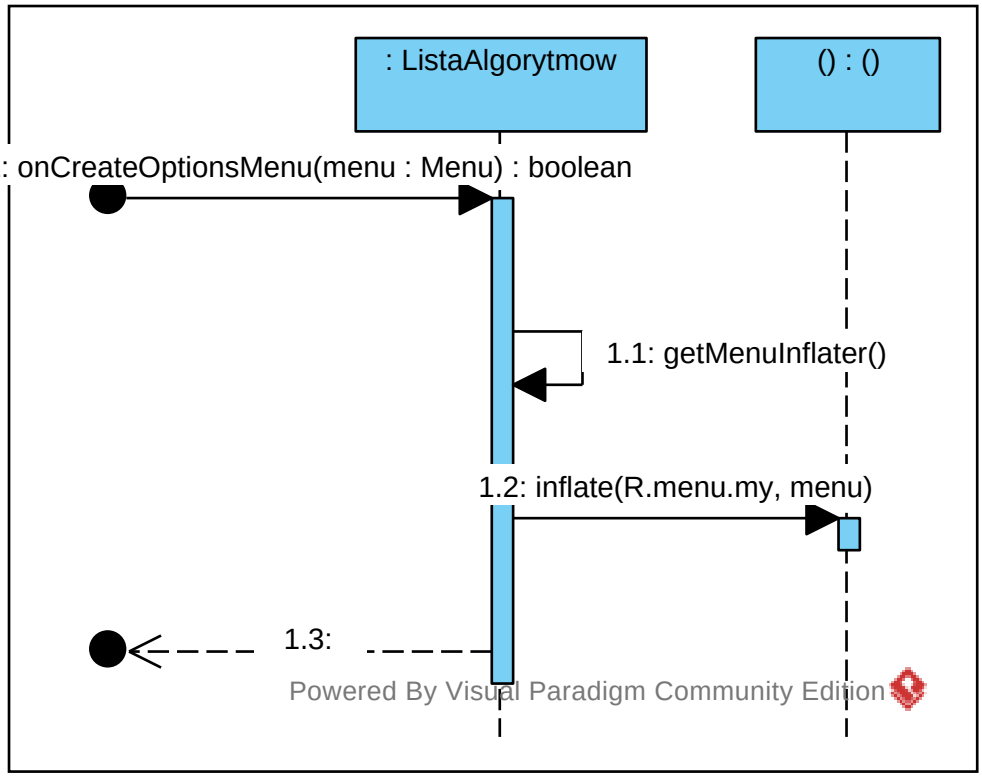

: IntroActivity
1: onDonePressed(currentFragment : Fragment) : void
1.1: finish()
1.2:
Powered By Visual Paradigm Community Edition

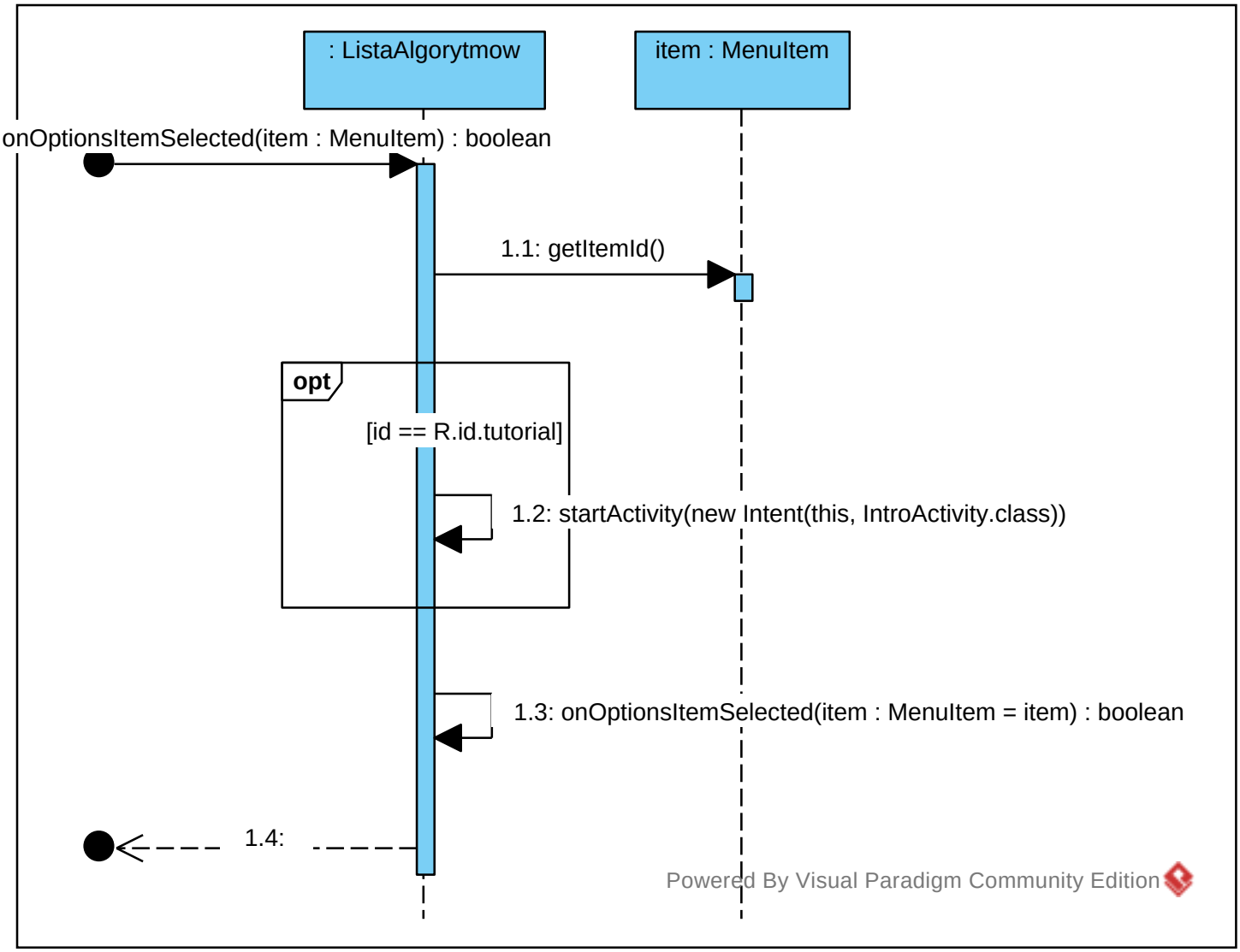

: IntroActivity
1: onSkipPressed(currentFragment : Fragment) : void
1.1: finish()
1.2:
Powered By Visual Paradigm Community Edition

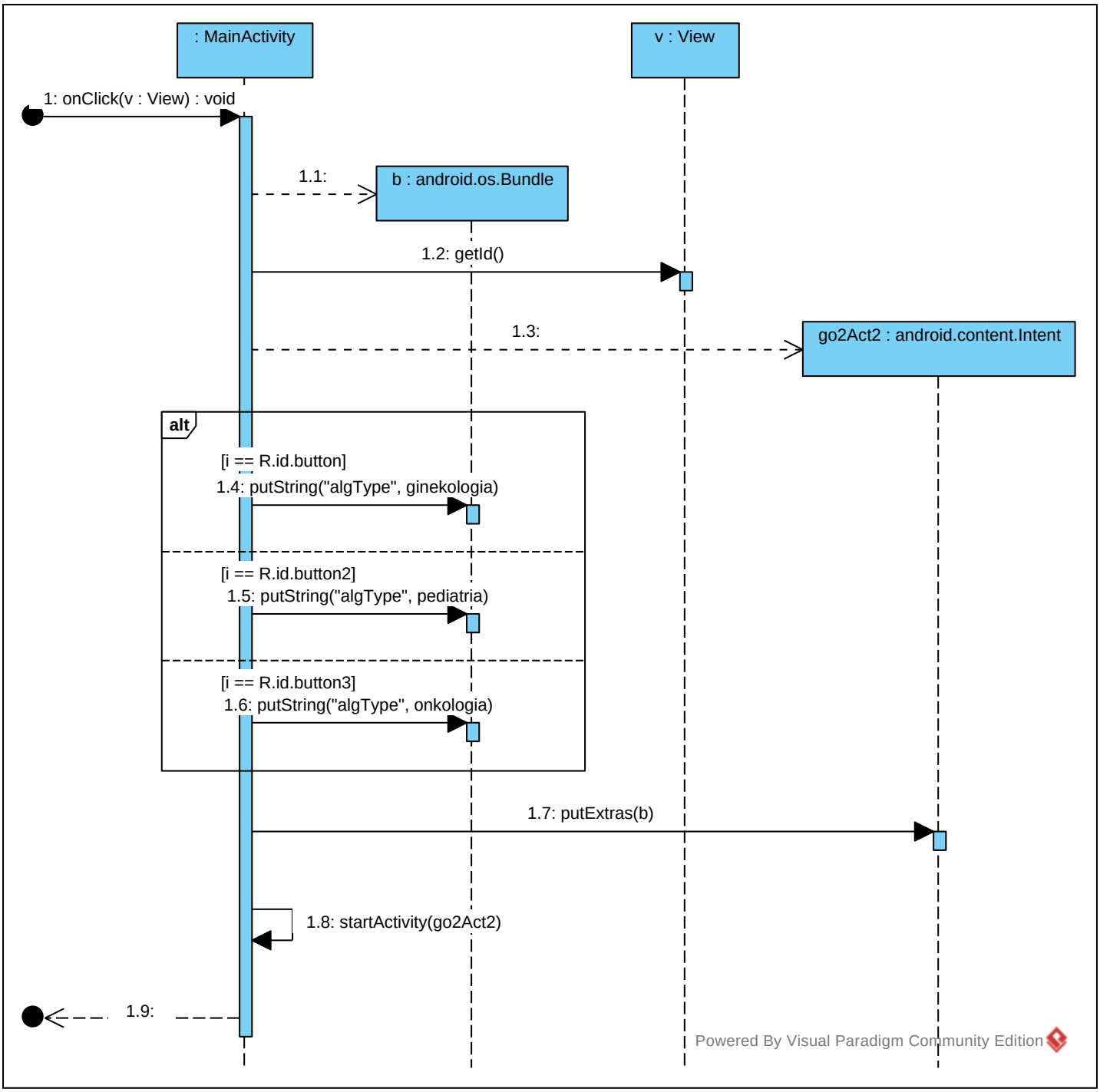

: IntroActivity
Changed(oldFragment : Fragment, newFragment : Fragment) : void
1.1:
Powered By Visual Paradigm Community Edition

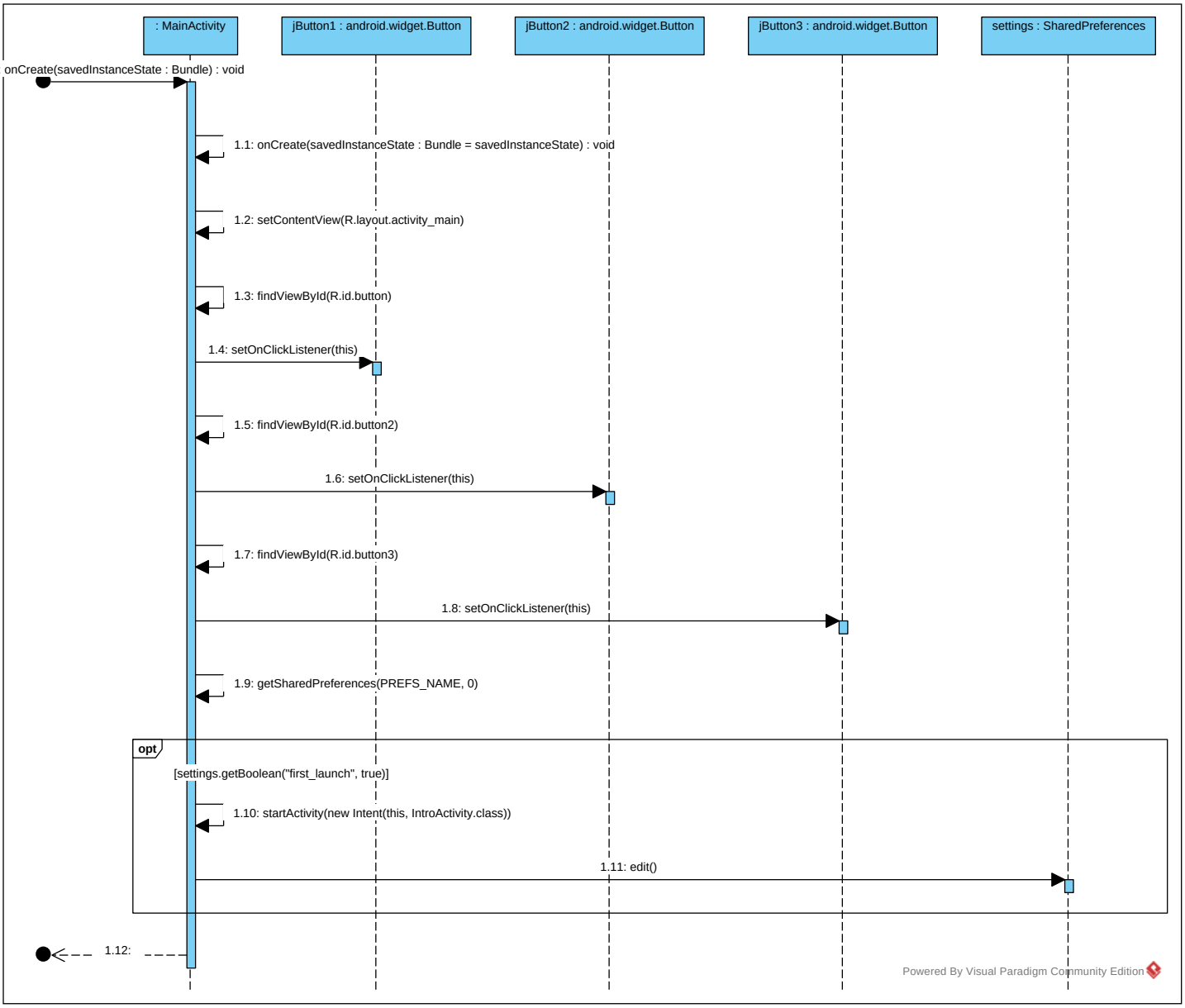

: ListaAlgorytmow
getAssets() : getAssets()
e1 : java.io.IOException
e1 : org.xmlpull.v1.XmlPullParserException
e1 : java.io.IOException
1: createList() : List<SingleAlgorithm>
1.1:
parser : AlgoritmListXmlParser
alt
[try]
1.2: getAssets()
1.3: open(algType+".xml")
[catch IOException]
1.4: printStackTrace()
alt
[try]
1.5: parse(in : InputStream = is) : List
[catch XmlPullParserException]
1.6: printStackTrace()
[catch IOException]
1.7: printStackTrace()
1.8:
Powered By Visual Paradigm Community Edition

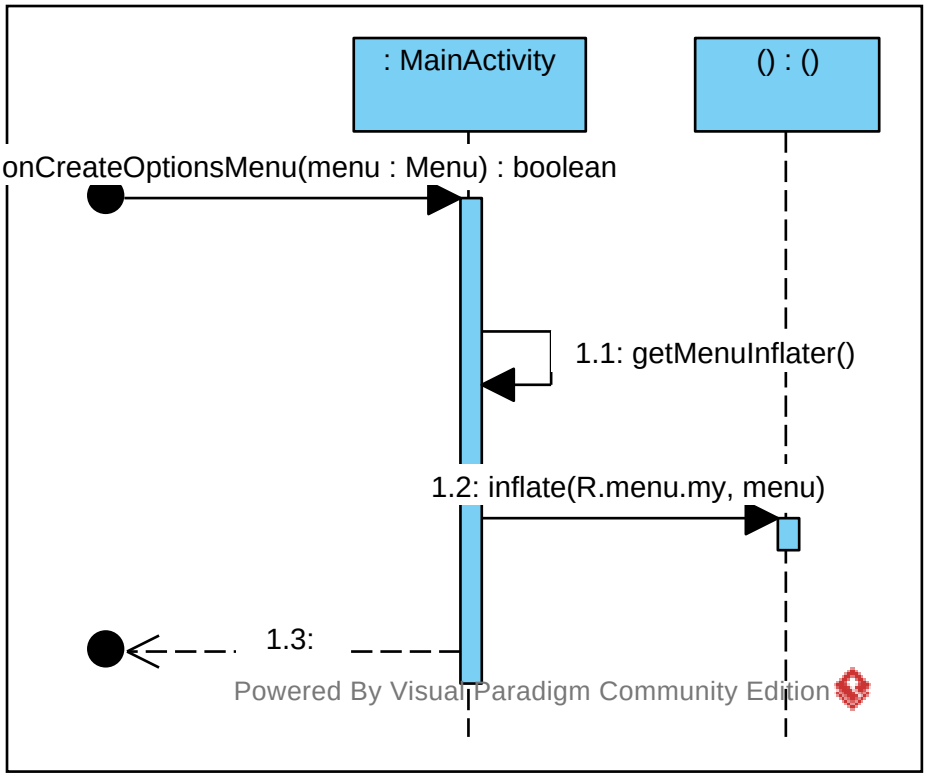

: ListaAlgorytmow
() : ()
p : Bundle
recList : RecyclerView
Toast : Toast
1: onCreate(savedInstanceState : Bundle) : void
1.1: onCreate(savedInstanceState : Bundle = savedInstanceState) : void
1.2: getIntent()
1.3: getExtras()
1.4: getString("algType")
1.5: setContentView(R.layout.activity_lista)
1.6: findViewById(R.id.cardList)
1.7: setHasFixedSize(true)
1.8:
linearLayoutManager : android.support.v7.widget.LinearLayoutManager
1.9: setOrientation(LinearLayoutManager.VERTICAL)
1.10: setLayoutManager(linearLayoutManager)
alt
1.14: addOnItemTouchListener(new RecyclerItemClickListener(this, new RecyclerItemClickListener.OnItemClickListener() {
@Override
public void onItemClick(View v, int position) {
Intent 1.11: createList() : List<SingleAlgorithm> Secondary_activity.class);
Bundle b = new Bundle();
b.putInt("key", allAlgorithms.get(position).getId());
b.putString("algType", algType);
b.putString("jakiPlikZXML", allAlgorithms.get(position).getFile());
b.putString("jakiPlikObrazem", allAlgorithms.get(position).getImage());
b.putString("algName", allAlgorithms.get(position).getName());
1.12:
ca : AlgorithmAdapter
1.13: setAdapter(ca)
ListaAlgorytmow.this.startActivity(myIntent);
}
)))
[catch Exception]
1.15: makeText(this.getBaseContext(), "Pliki algorytmw nie istniejÄ…", Toast.LENGTH_LONG)
1.16: onBackPressed()
1.17: setTitle(algType)
1.18:
Powered By Visual Paradigm Community Edition

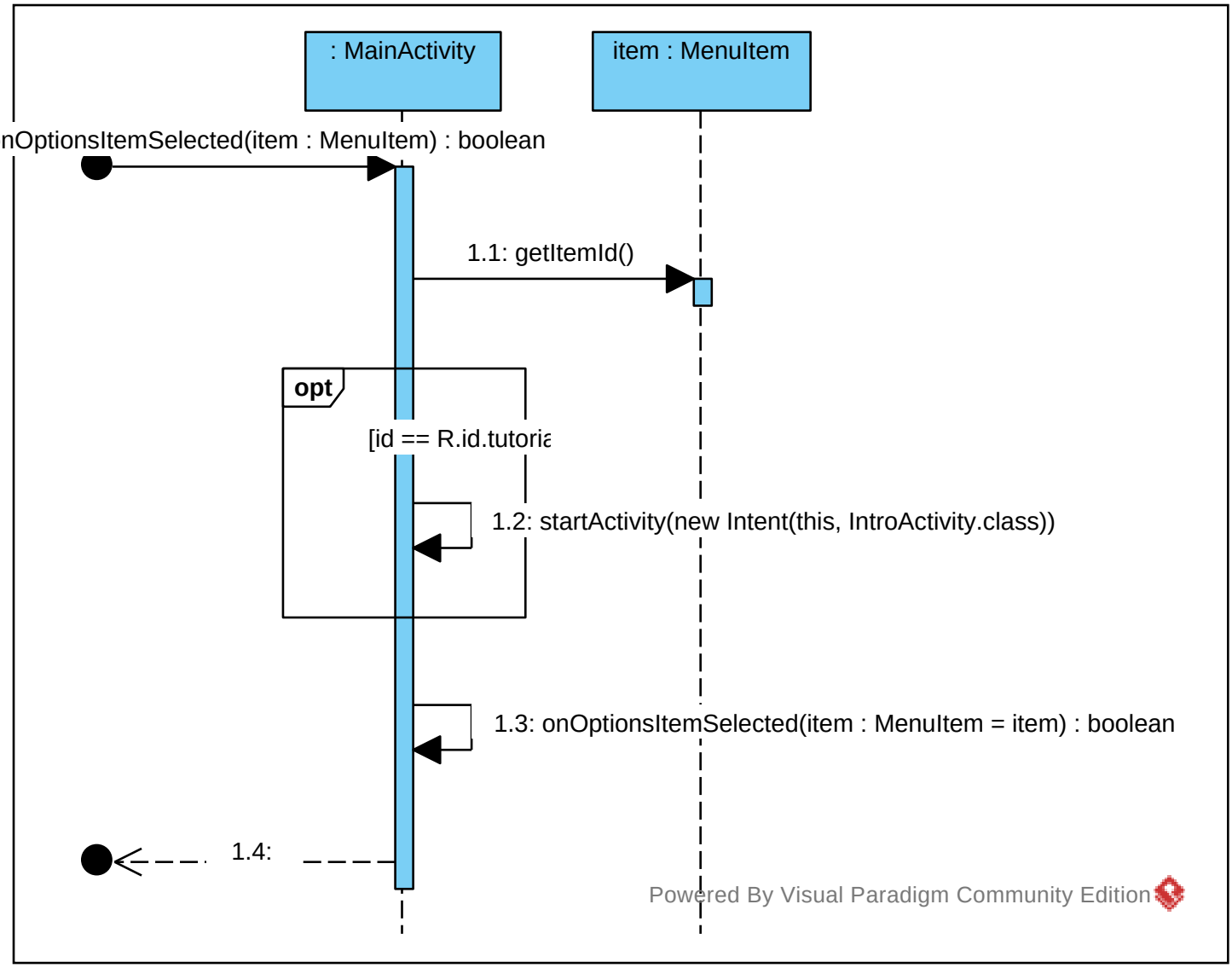

: ListaAlgorytmow
() : ()
1: onCreateOptionsMenu(menu : Menu) : boolean
1.1: getMenuInflater()
1.2: inflate(R.menu.my, menu)
1.3:
Powered By Visual Paradigm Community Edition

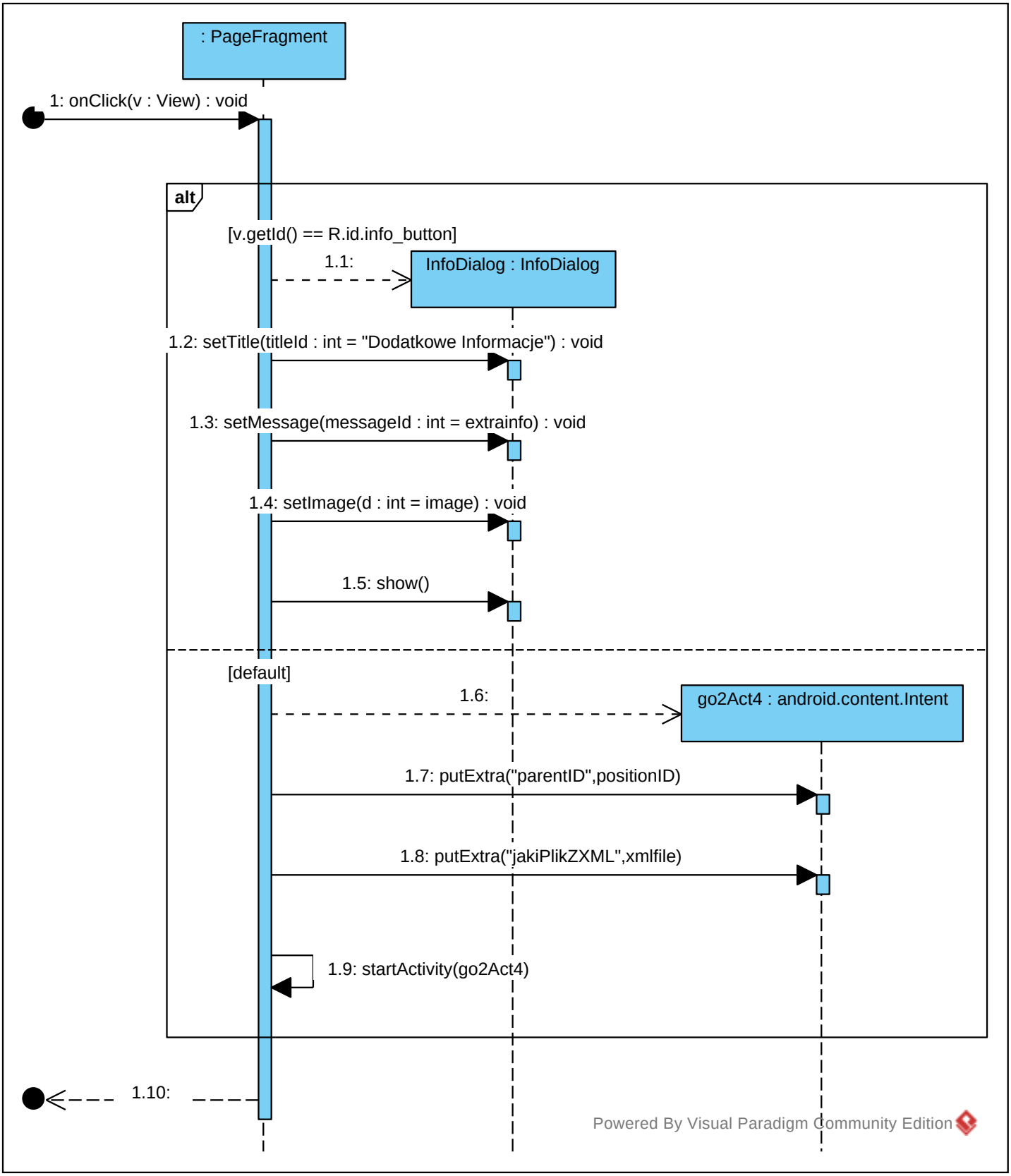

: ListaAlgorytmow
item : MenuItem
1: onOptionsItemSelected(item : MenuItem) : boolean
1.1: getItemId()
opt
[id == R.id.tutorial]
1.2: startActivity(new Intent(this, IntroActivity.class))
1.3: onOptionsItemSelected(item : MenuItem = item) : boolean
1.4:
Powered By Visual Paradigm Community Edition

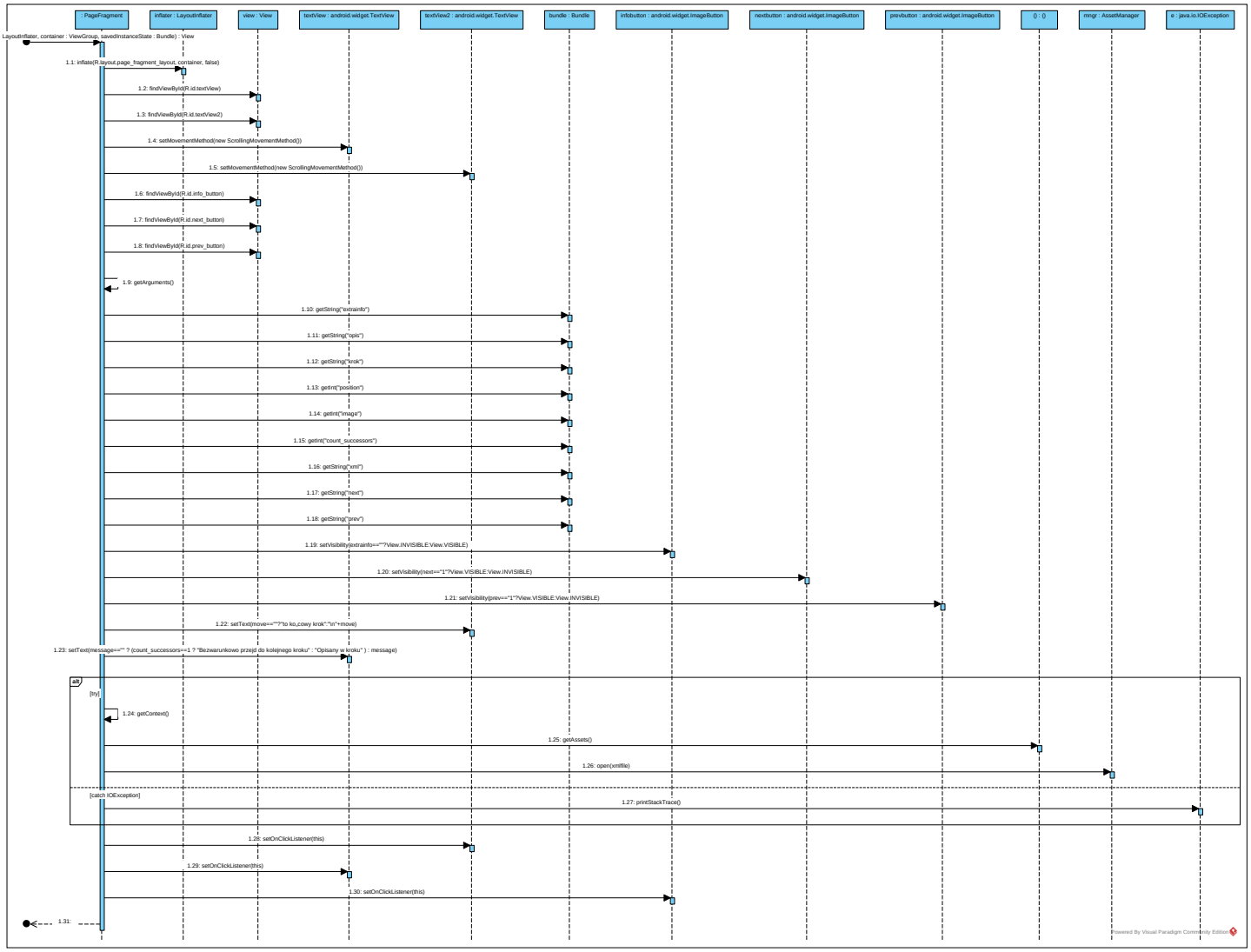

: MainActivity
v : View
1: onClick(v : View) : void
1.1:
b : android.os.Bundle
1.2: getId()
1.3:
go2Act2 : android.content.Intent
alt
[i == R.id.button]
1.4: putString("algType", ginekologia)
[i == R.id.button2]
1.5: putString("algType", pediatria)
[i == R.id.button3]
1.6: putString("algType", onkologia)
1.7: putExtras(b)
1.8: startActivity(go2Act2)
1.9:
Powered By Visual Paradigm Community Edition

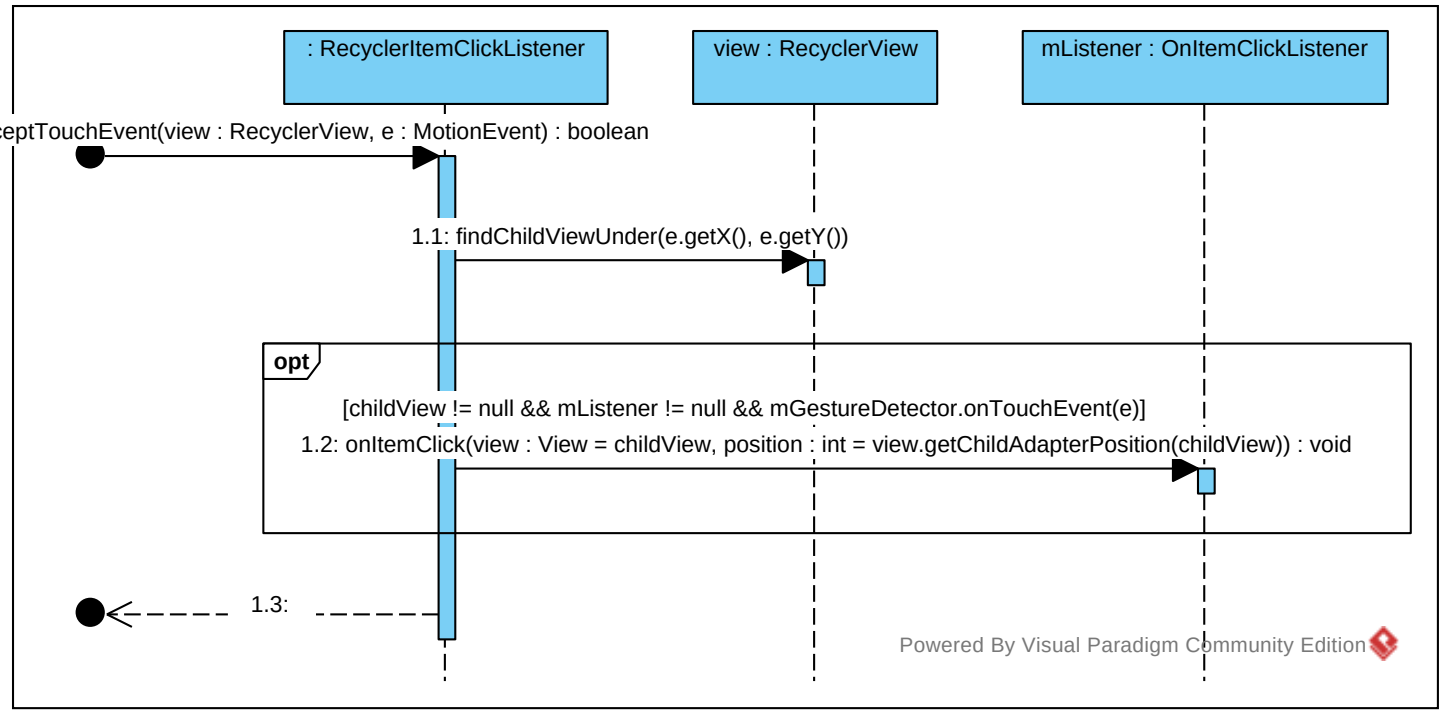

: MainActivity
jButton1 : android.widget.Button
jButton2 : android.widget.Button
jButton3 : android.widget.Button
settings : SharedPreferences
1: onCreate(savedInstanceState : Bundle) : void
1.1: onCreate(savedInstanceState : Bundle = savedInstanceState) : void
1.2: setContentView(R.layout.activity_main)
1.3: findViewById(R.id.button)
1.4: setOnClickListener(this)
1.5: findViewById(R.id.button2)
1.6: setOnClickListener(this)
1.7: findViewById(R.id.button3)
1.8: setOnClickListener(this)
1.9: getSharedPreferences(PREFS_NAME, 0)
opt
[settings.getBoolean("first_launch", true)]
1.10: startActivity(new Intent(this, IntroActivity.class))
1.11: edit()
1.12:
Powered By Visual Paradigm Community Edition

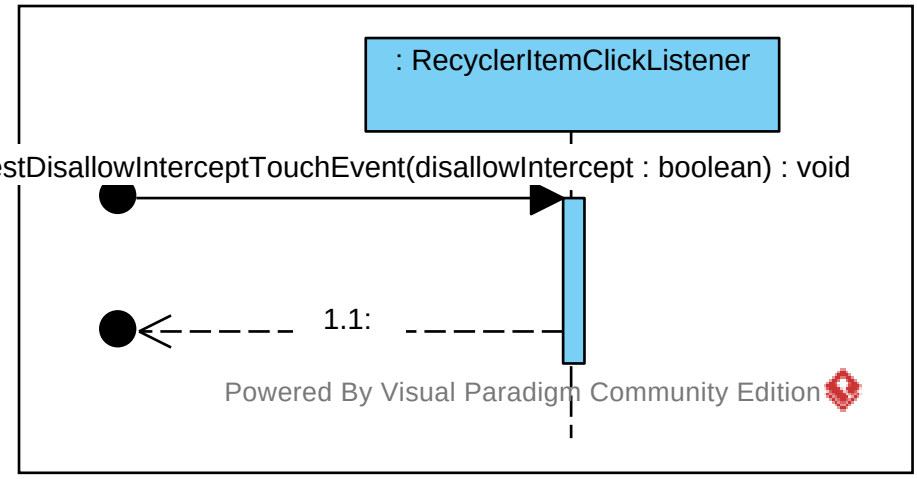

: MainActivity
() : ()
1: onCreateOptionsMenu(menu : Menu) : boolean
1.1: getMenuInflater()
1.2: inflate(R.menu.my, menu)
1.3:
Powered By Visual Paradigm Community Edition

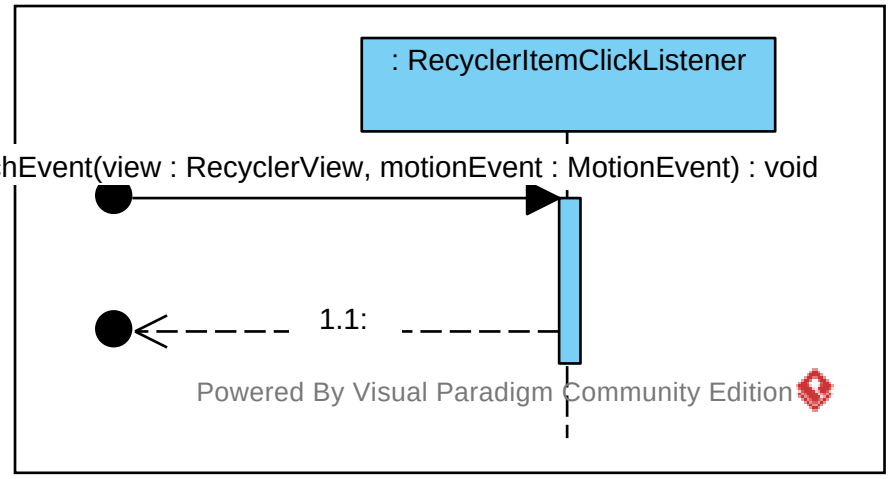

: MainActivity
item : MenuItem
1: onOptionsItemSelected(item : MenuItem) : boolean
1.1: getItemId()
opt
[id == R.id.tutoria
1.2: startActivity(new Intent(this, IntroActivity.class))
1.3: onOptionsItemSelected(item : MenuItem = item) : boolean
1.4:
Powered By Visual Paradigm Community Edition

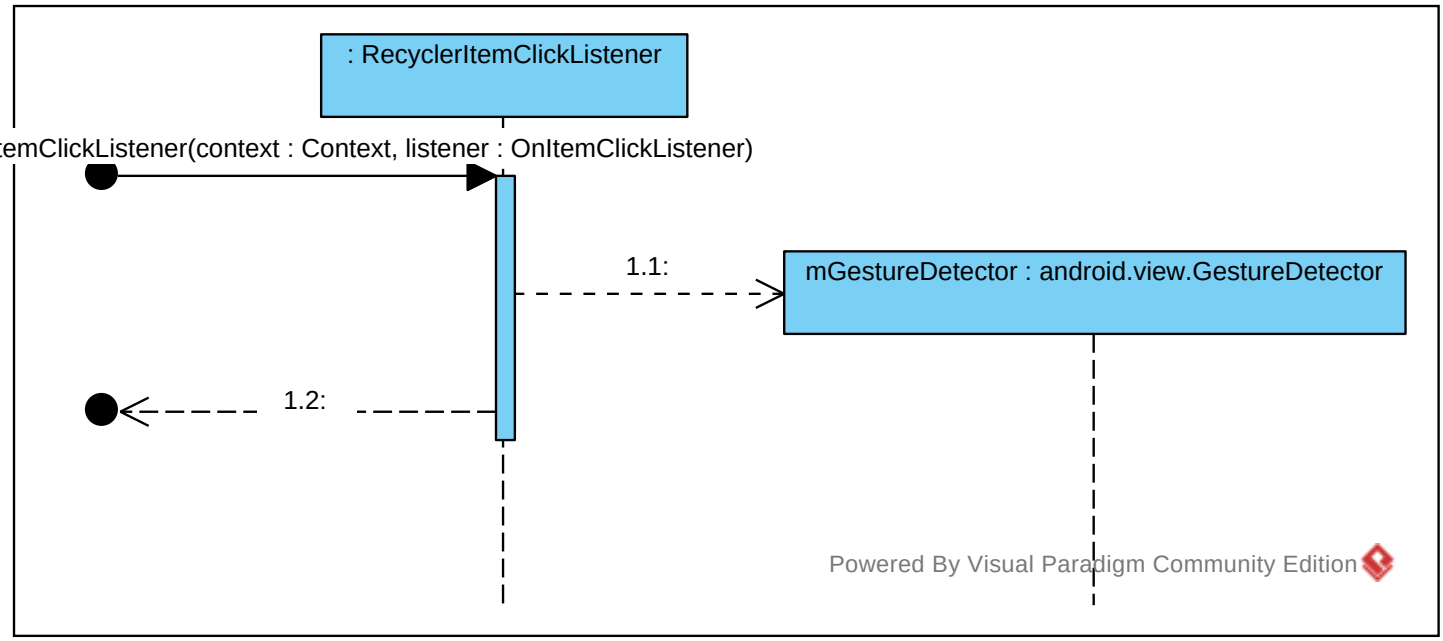

: PageFragment
1: onClick(v : View) : void
alt
[v.getId() == R.id.info_button]
1.1:
InfoDialog : InfoDialog
1.2: setTitle(titleId : int = "Dodatkowe Informacje") : void
1.3: setMessage(messageId : int = extrainfo) : void
1.4: setImage(d : int = image) : void
1.5: show()
[default]
1.6:
go2Act4 : android.content.Intent
1.7: putExtra("parentID",positionID)
1.8: putExtra("jakiPlikZXML",xmlfile)
1.9: startActivity(go2Act4)
1.10:
Powered By Visual Paradigm Community Edition

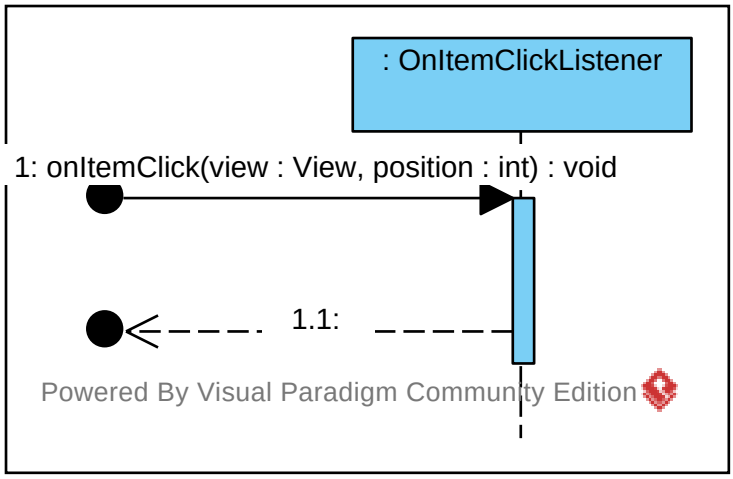

: PageFragment
inflater : LayoutInflater
view : View
textView : android.widget.TextView
textView2 : android.widget.TextView
bundle : Bundle
infobutton : android.widget.ImageButton
nextbutton : android.widget.ImageButton
prevbutton : android.widget.ImageButton
() : ()
mngr : AssetManager
e : java.io.IOException
: onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View
1.1: inflate(R.layout.page_fragment_layout, container, false)
1.2: findViewById(R.id.textView)
1.3: findViewById(R.id.textView2)
1.4: setMovementMethod(new ScrollingMovementMethod())
1.5: setMovementMethod(new ScrollingMovementMethod())
1.6: findViewById(R.id.info_button)
1.7: findViewById(R.id.next_button)
1.8: findViewById(R.id.prev_button)
1.9: getArguments()
1.10: getString("extrainfo")
1.11: getString("opis")
1.12: getString("krok")
1.13: getInt("position")
1.14: getInt("image")
1.15: getInt("count_successors")
1.16: getString("xml")
1.17: getString("next")
1.18: getString("prev")
1.19: setVisibility(extrainfo==""?View.INVISIBLE:View.VISIBLE)
1.20: setVisibility(next=="1"?View.VISIBLE:View.INVISIBLE)
1.21: setVisibility(prev=="1"?View.VISIBLE:View.INVISIBLE)
1.22: setText(move+""?to ko,cowy krok":"\n"+move)
1.23: setText(message+"" ? (count_successors==1 ? "Bezwarunkowo przejd do kolejnego kroku" : "Opisany w kroku") : message)
alt
[try]
1.24: getContext()
1.25: getAssets()
1.26: open(xmlfile)
[catch IOException]
1.27: printStackTrace()
1.28: setOnClickListener(this)
1.29: setOnClickListener(this)
1.30: setOnClickListener(this)
1.31:
Powered By Visual Paradigm Community Edition

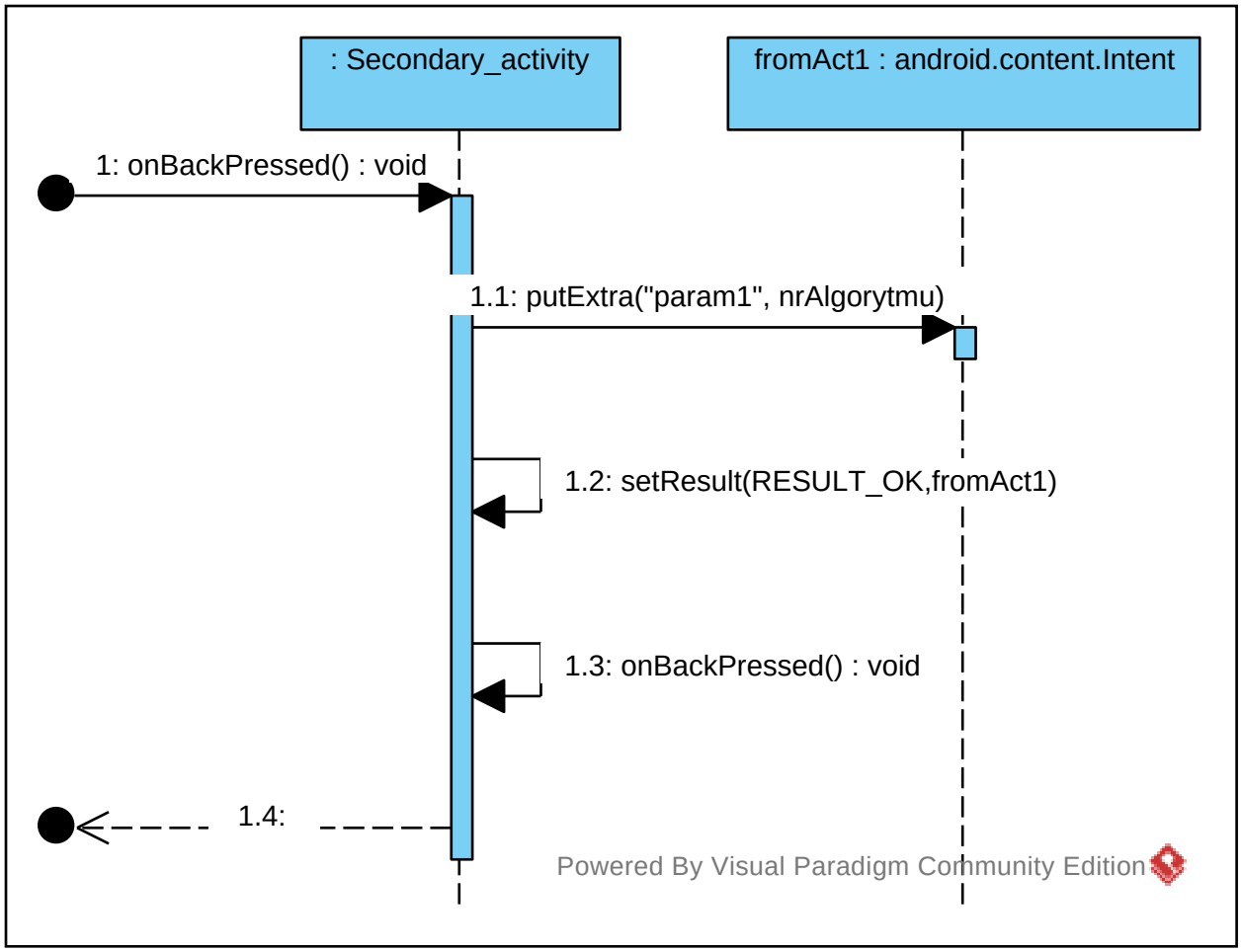

: RecyclerItemClickListener
view : RecyclerView
mListener : OnItemClickListener
nInterceptTouchEvent(view : RecyclerView, e : MotionEvent) : boolean
1.1: findChildViewUnder(e.getX(), e.getY())
opt
[childView != null && mListener != null && mGestureDetector.onTouchEvent(e)]
1.2: onItemClick(view : View = childView, position : int = view.getChildAdapterPosition(childView)) : void
1.3:
Powered By Visual Paradigm Community Edition

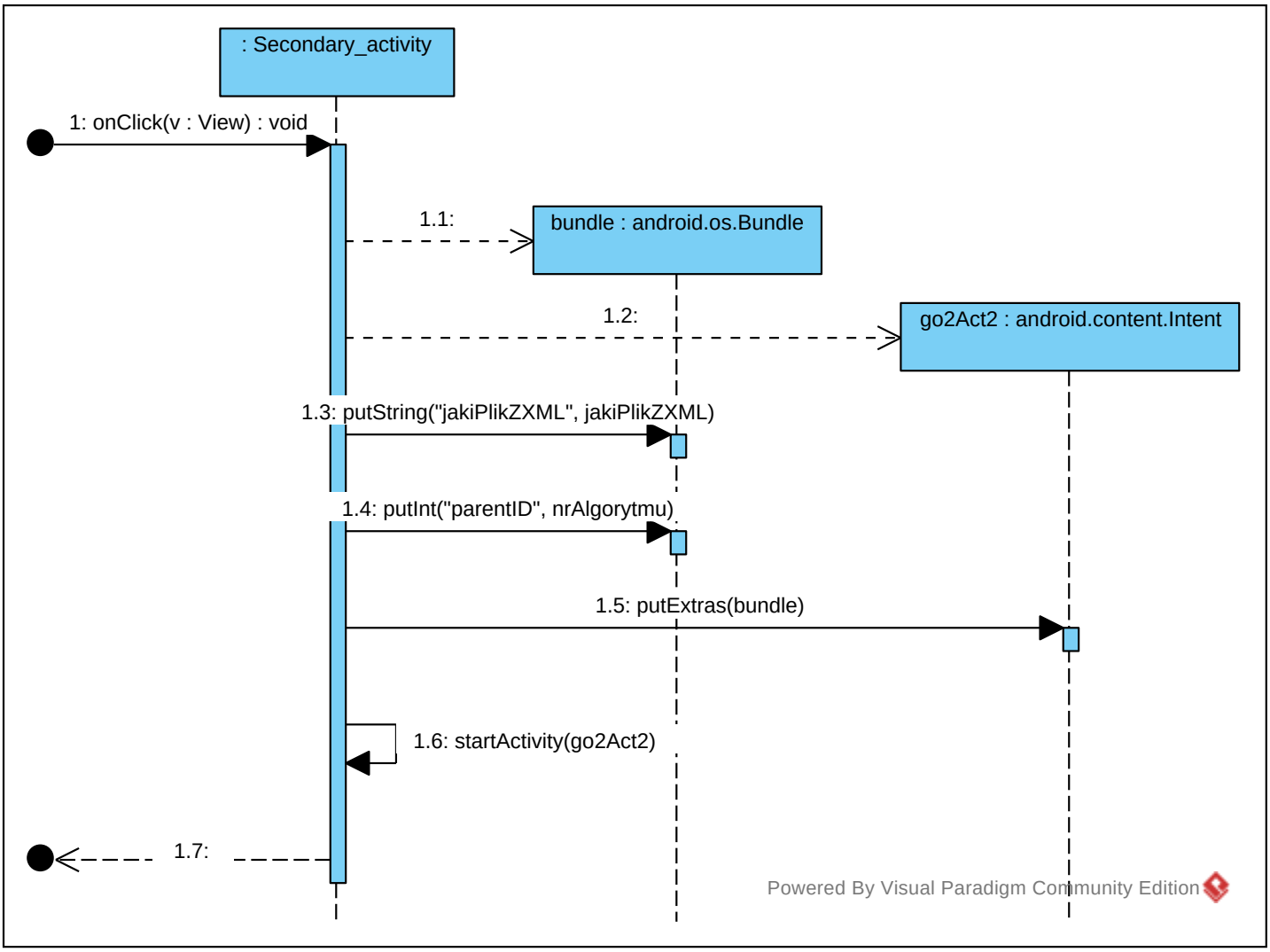

: RecyclerItemClickListener
equestDisallowInterceptTouchEvent(disallowIntercept : boolean) : void
1.1:
Powered By Visual Paradigm Community Edition

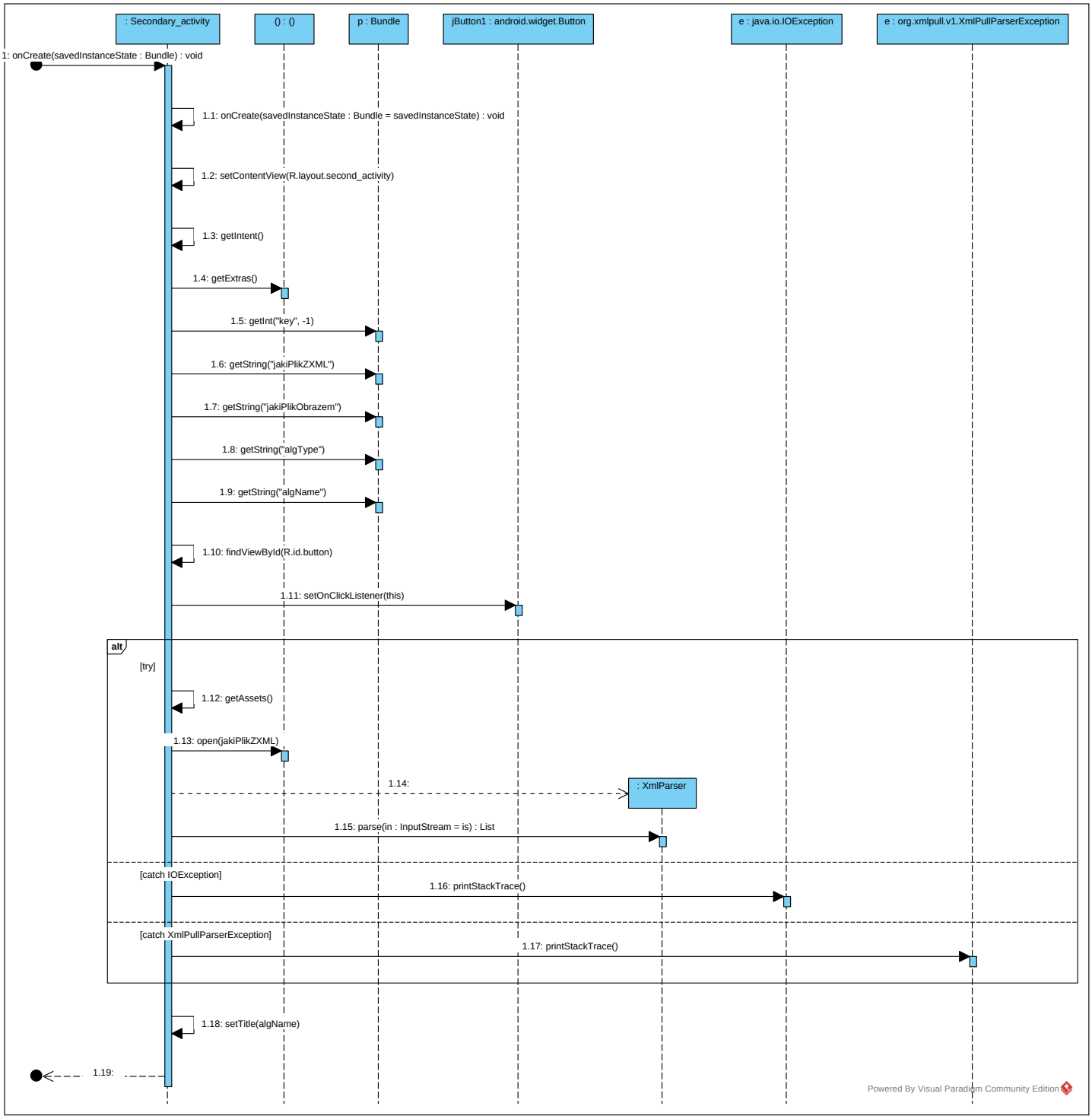

: RecyclerItemClickListener
TouchEvent(view : RecyclerView, motionEvent : MotionEvent) : void
1.1:
Powered By Visual Paradigm Community Edition

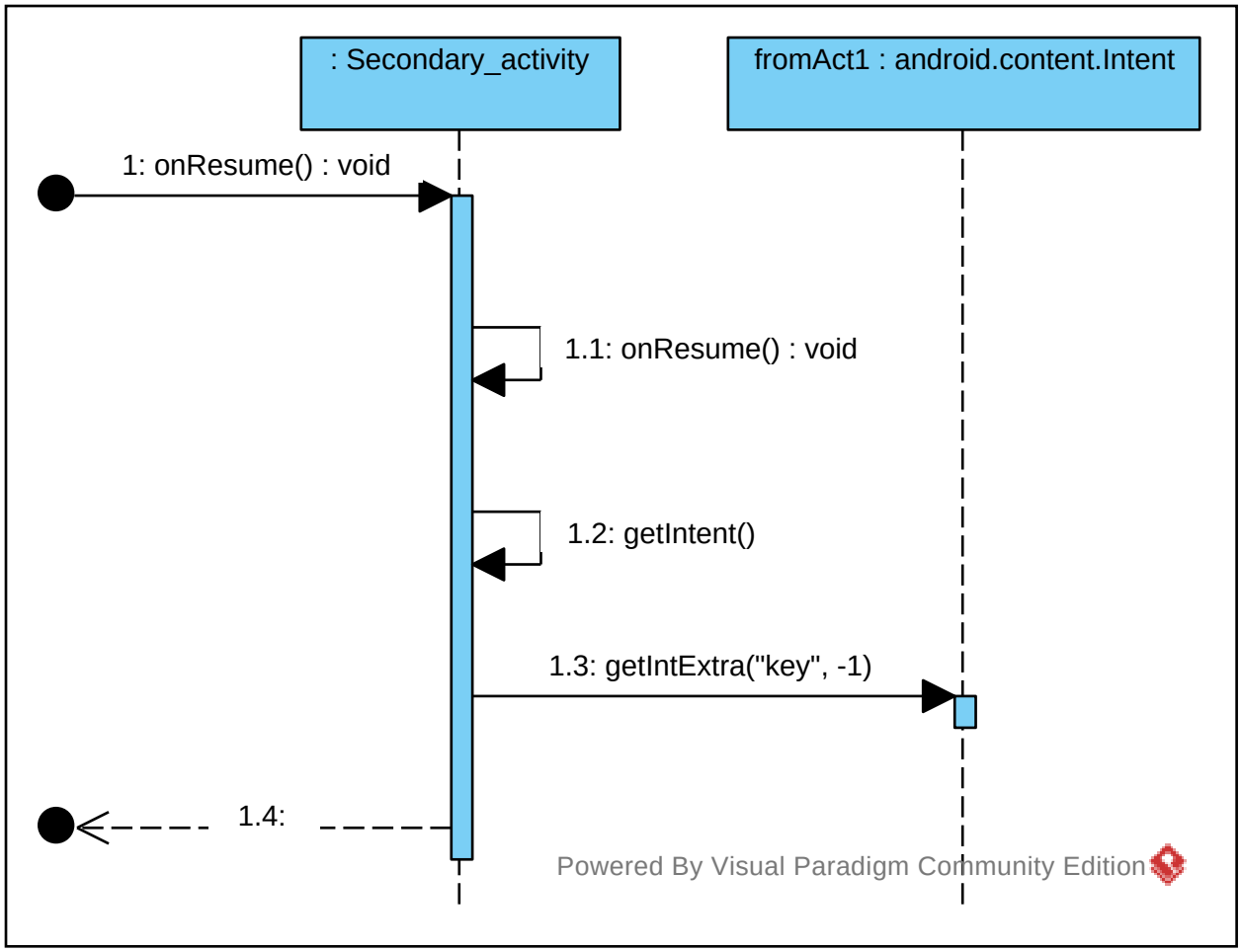

: RecyclerItemClickListener
yclerItemClickListener(context : Context, listener : OnItemClickListener)
1.1:
mGestureDetector : android.view.GestureDetector
1.2:
Powered By Visual Paradigm Community Edition

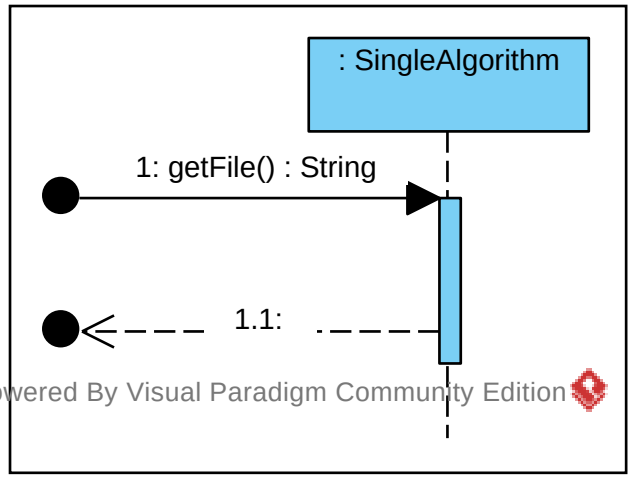

: OnItemClickListener
1: onItemClick(view : View, position : int) : void
1.1:
Powered By Visual Paradigm Community Edition

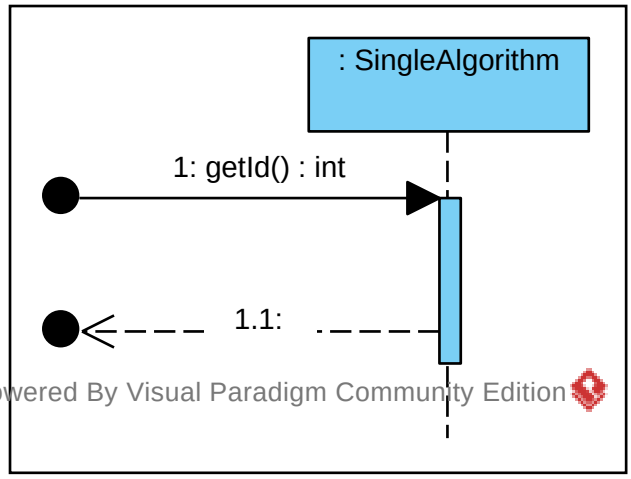

: Secondary_activity
fromAct1 : android.content.Intent
1: onBackPressed() : void
1.1: putExtra("param1", nrAlgorytmu)
1.2: setResult(RESULT_OK,fromAct1)
1.3: onBackPressed() : void
1.4:
Powered By Visual Paradigm Community Edition

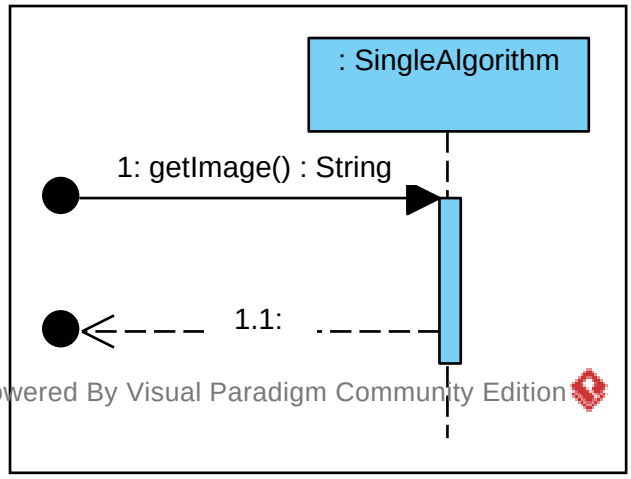

: Secondary_activity
1: onClick(v : View) : void
1.1:
bundle : android.os.Bundle
1.2:
go2Act2 : android.content.Intent
1.3: putString("jakiPlikZXML", jakiPlikZXML)
1.4: putInt("parentID", nrAlgorytmu)
1.5: putExtras(bundle)
1.6: startActivity(go2Act2)
1.7:
Powered By Visual Paradigm Community Edition

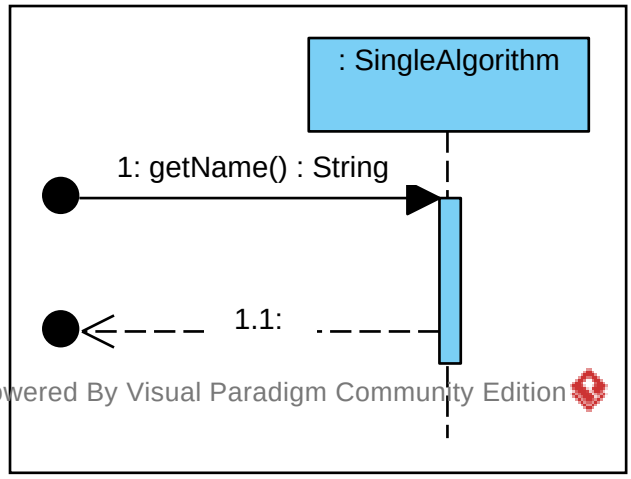

: Secondary_activity
() : ()
p : Bundle
jButton1 : android.widget.Button
e : java.io.IOException
e : org.xmlpull.v1.XmlPullParserException
1: onCreate(savedInstanceState : Bundle) : void
1.1: onCreate(savedInstanceState : Bundle = savedInstanceState) : void
1.2: setContentView(R.layout.second_activity)
1.3: getIntent()
1.4: getExtras()
1.5: getInt("key", -1)
1.6: getString("jakiPlikZXML")
1.7: getString("jakiPlikObrazem")
1.8: getString("algType")
1.9: getString("algName")
1.10: findViewById(R.id.button)
1.11: setOnClickListener(this)
alt
[try]
1.12: getAssets()
1.13: open(jakiPlikZXML)
1.14:
: XmlParser
1.15: parse(in : InputStream = is) : List
[catch IOException]
1.16: printStackTrace()
[catch XmlPullParserException]
1.17: printStackTrace()
1.18: setTitle(algName)
1.19:
Powered By Visual Paradigm Community Edition

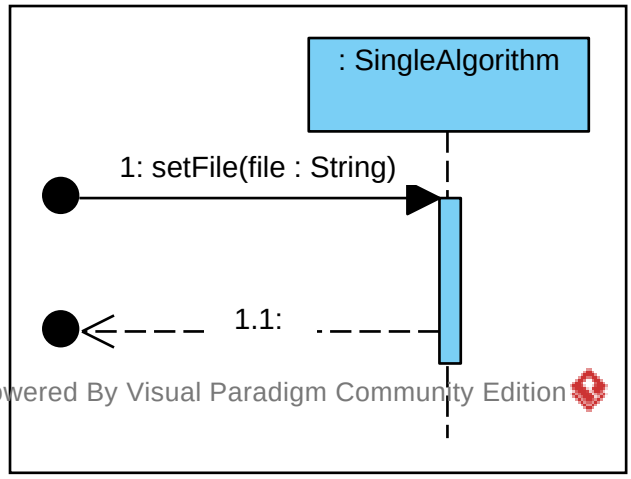

: Secondary_activity
fromAct1 : android.content.Intent
1: onResume() : void
1.1: onResume() : void
1.2: getIntent()
1.3: getIntExtra("key", -1)
1.4:
Powered By Visual Paradigm Community Edition

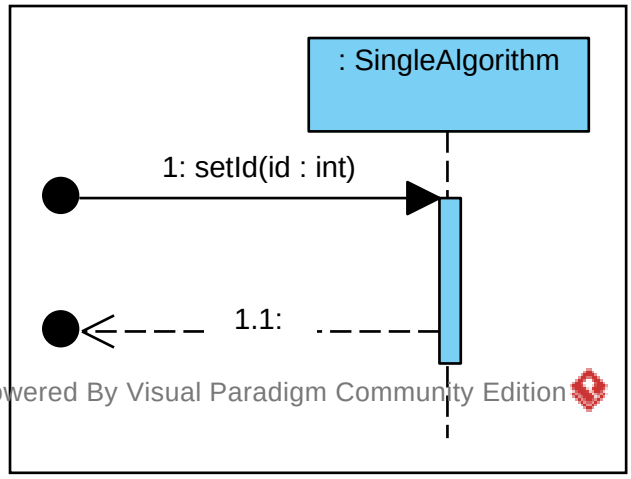

: SingleAlgorithm
1: getFile() : String
1.1:
Powered By Visual Paradigm Community Edition

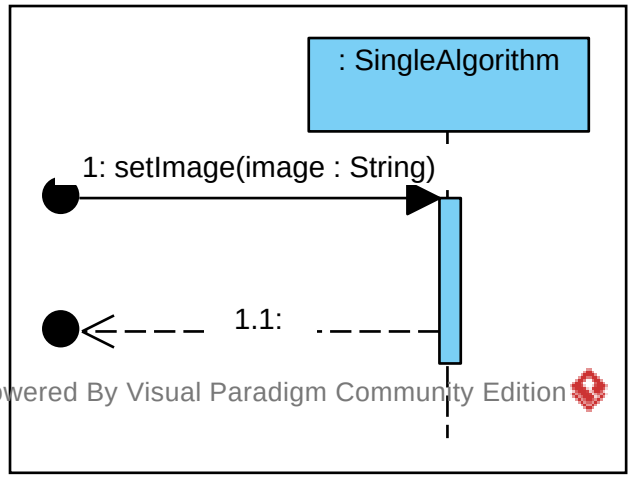

: SingleAlgorithm
1: getId() : int
1.1:
Powered By Visual Paradigm Community Edition

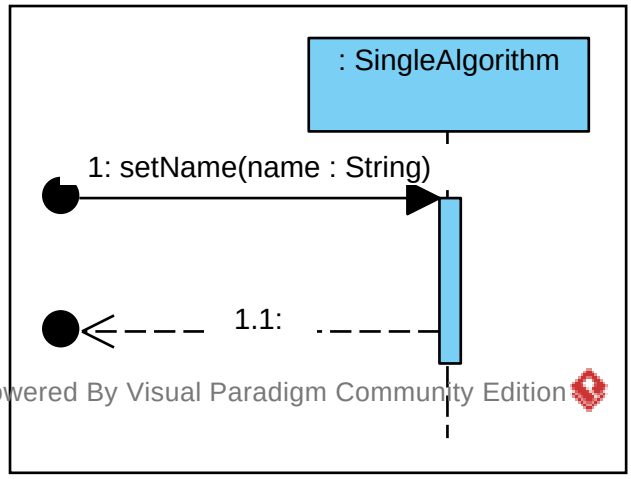

: SingleAlgorithm
1: getImage() : String
1.1:
Powered By Visual Paradigm Community Edition

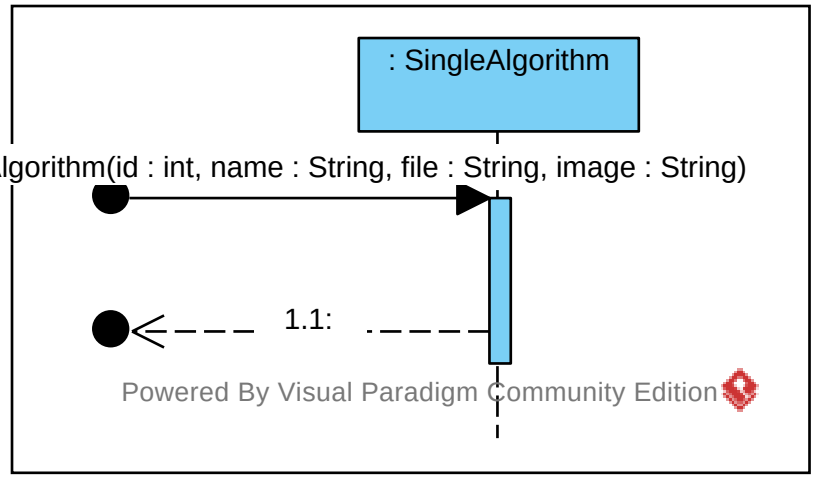

: SingleAlgorithm
1: getName() : String
1.1:
Powered By Visual Paradigm Community Edition

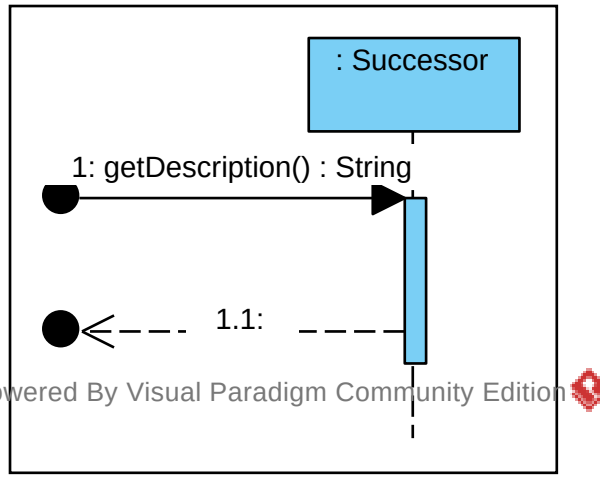

: SingleAlgorithm
1: setFile(file : String)
1.1:
Powered By Visual Paradigm Community Edition

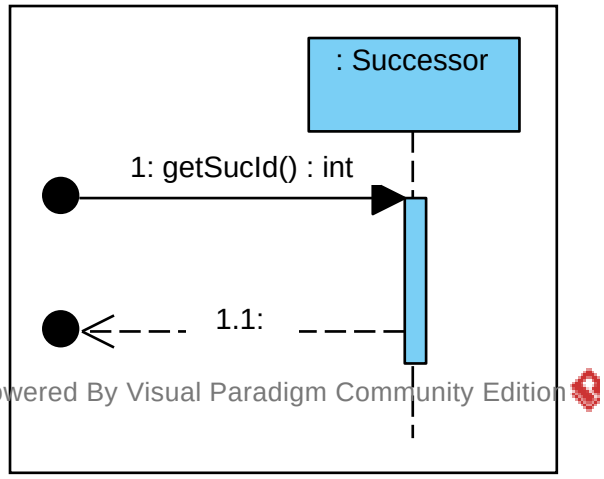

: SingleAlgorithm
1: setId(id : int)
1.1:
Powered By Visual Paradigm Community Edition

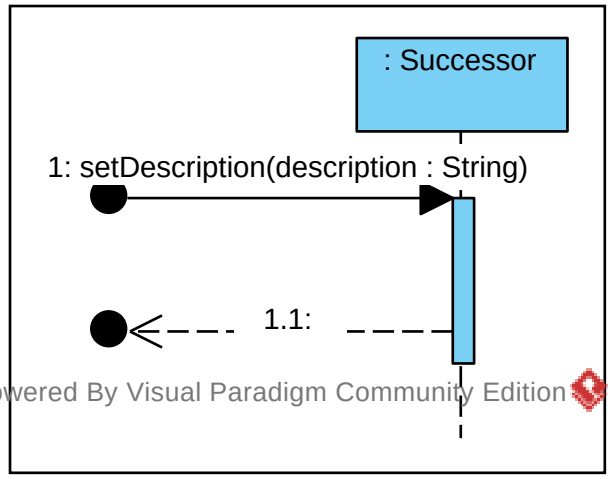

: SingleAlgorithm
1: setImage(image : String)
1.1:
Powered By Visual Paradigm Community Edition

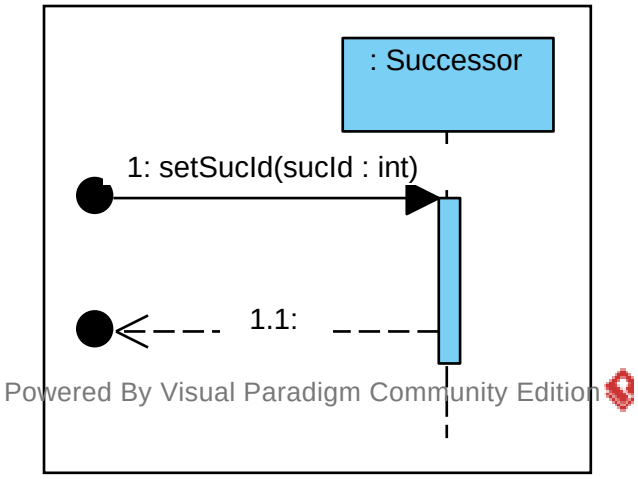

: SingleAlgorithm
1: setName(name : String)
1.1:
Powered By Visual Paradigm Community Edition

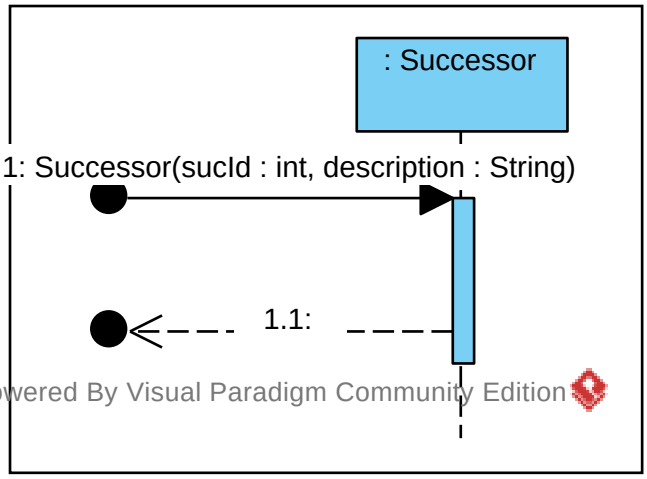

: SingleAlgorithm
ngleAlgorithm(id : int, name : String, file : String, image : String)
1.1:
Powered By Visual Paradigm Community Edition

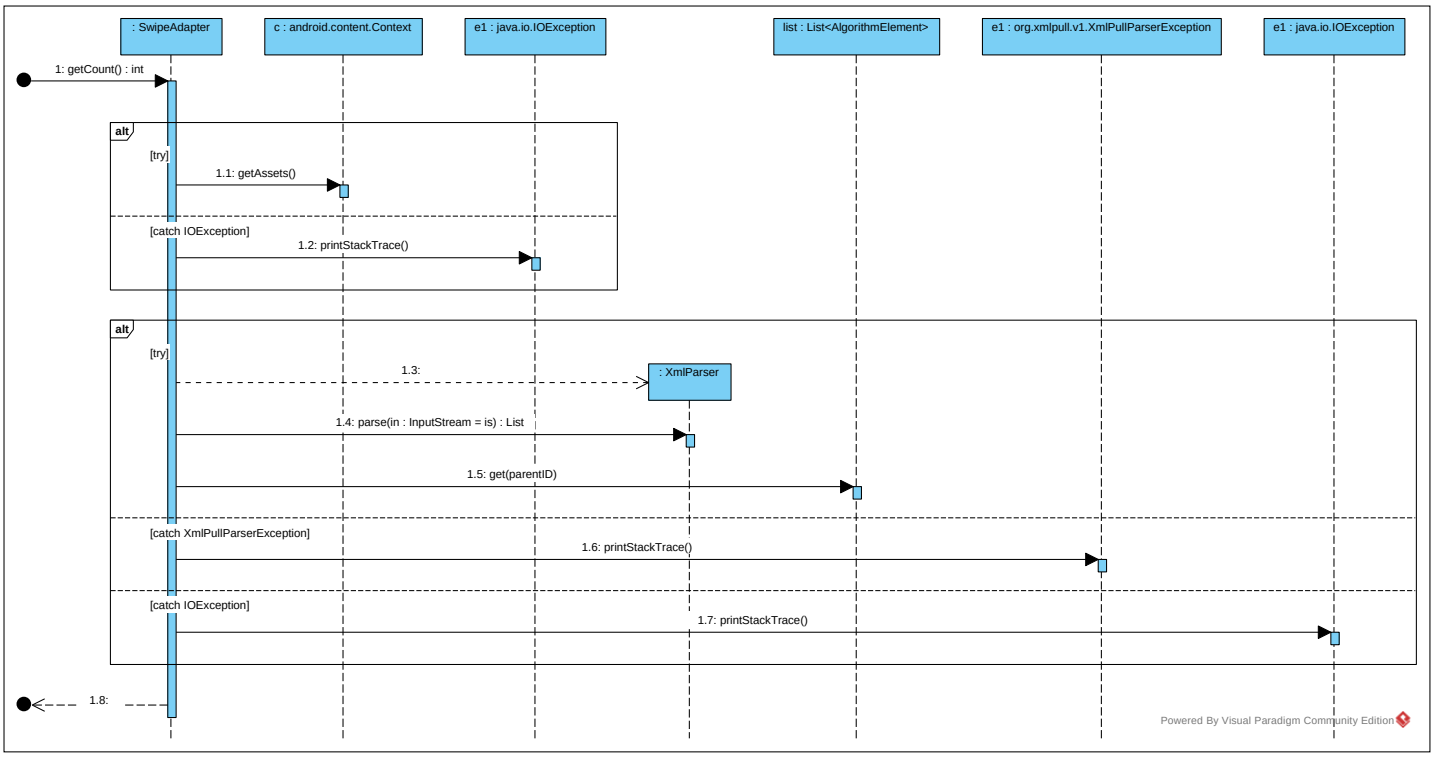

: Successor
1: getDescription() : String
1.1:
Powered By Visual Paradigm Community Edition

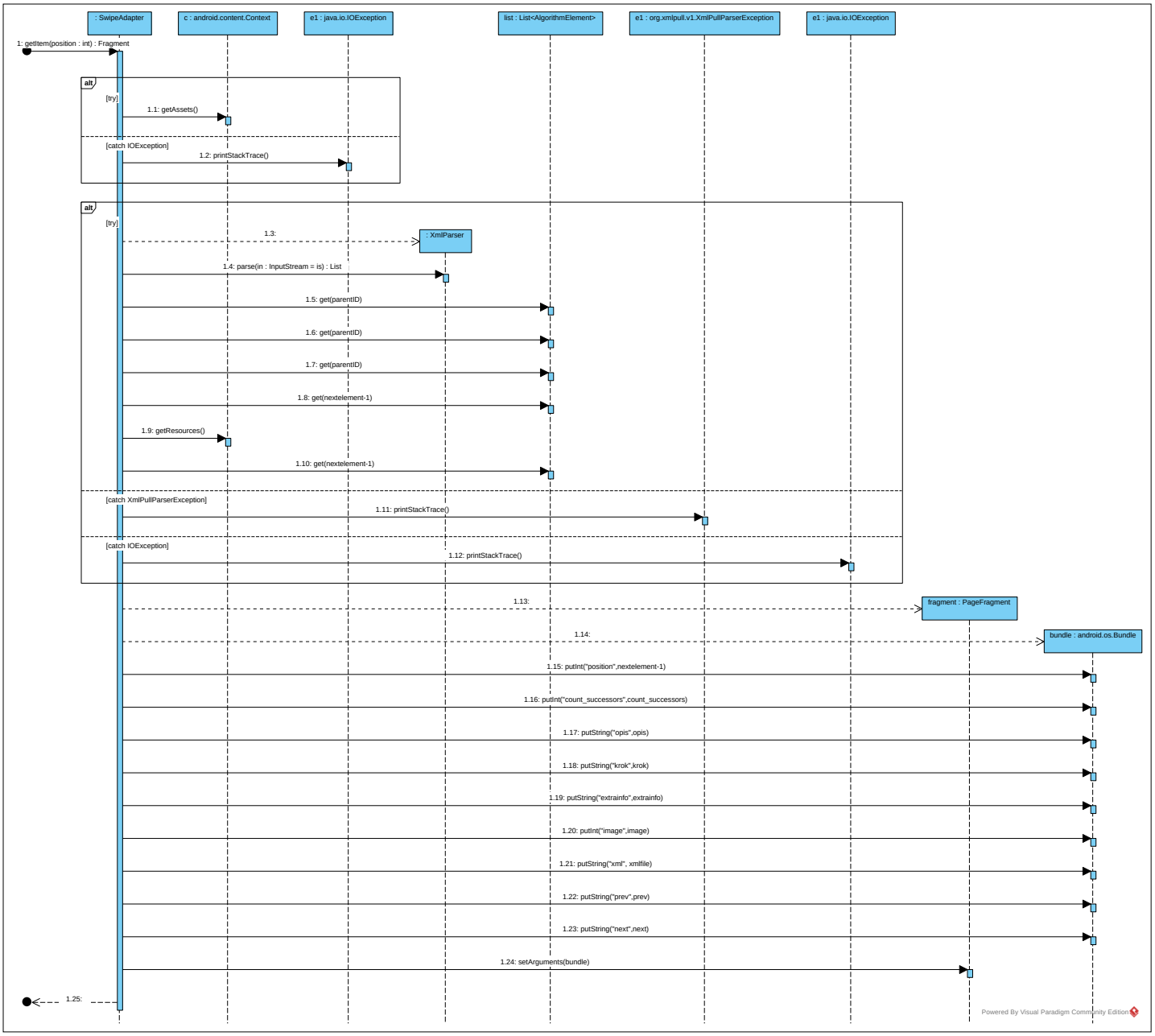

: Successor
1: getSucId() : int
1.1:
Powered By Visual Paradigm Community Edition

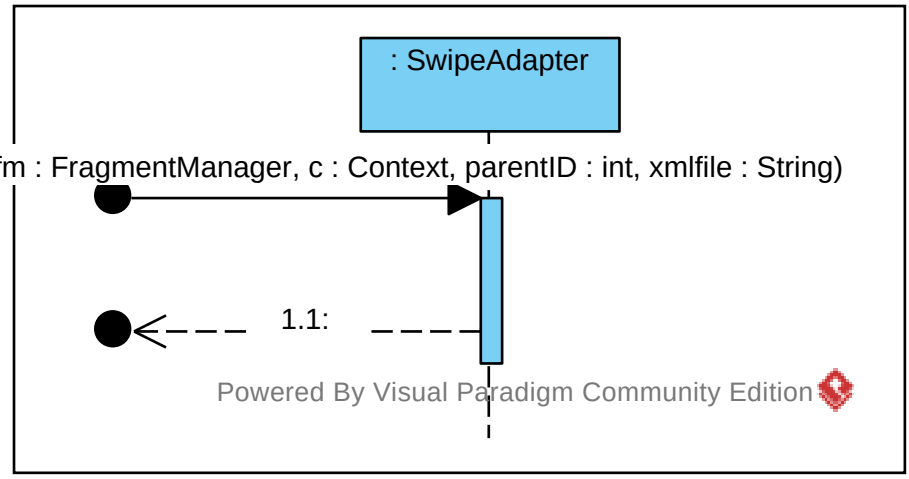

: Successor
1: setDescription(description : String)
1.1:
Powered By Visual Paradigm Community Edition

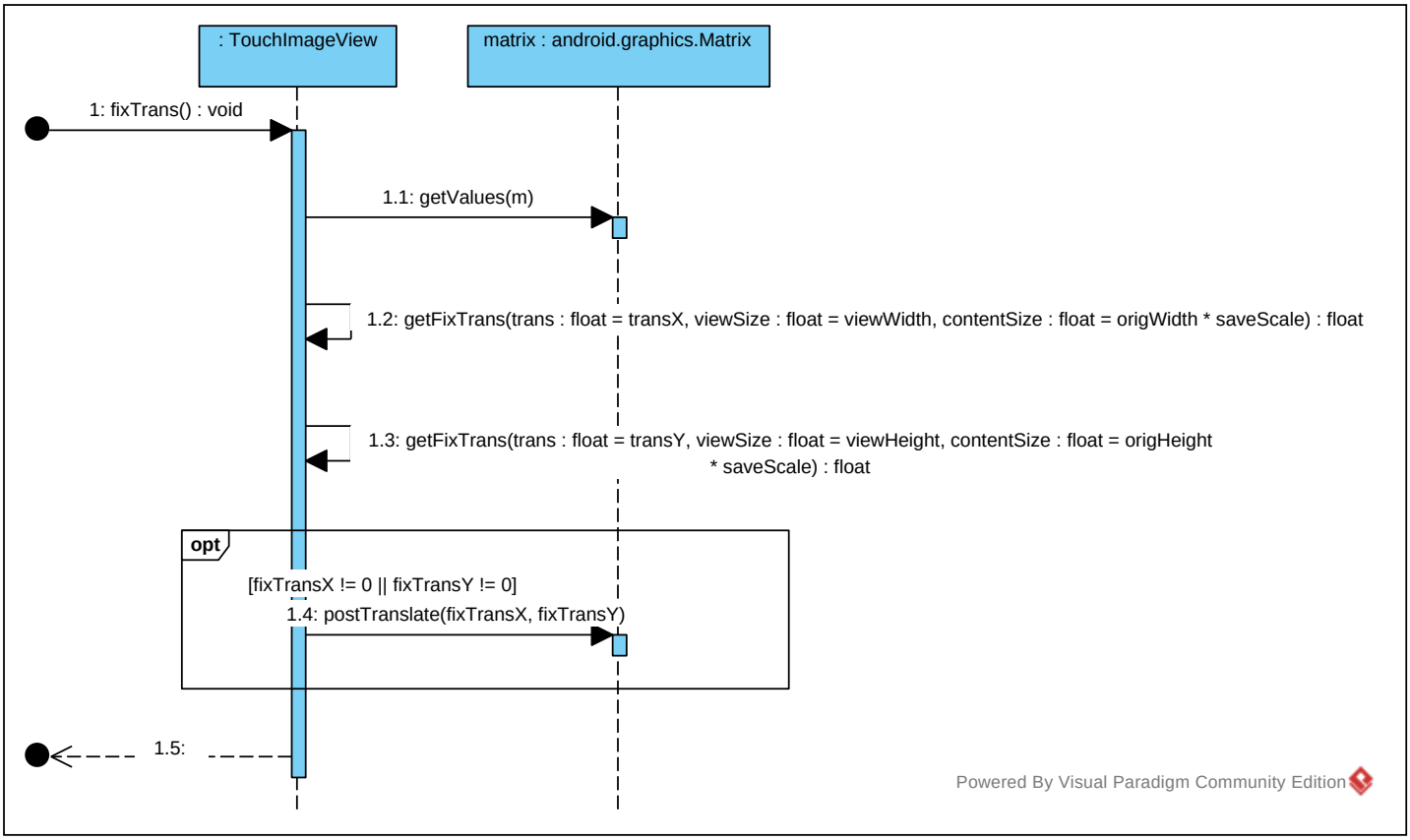

: Successor
1: setSucId(sucId : int)
1.1:
Powered By Visual Paradigm Community Edition

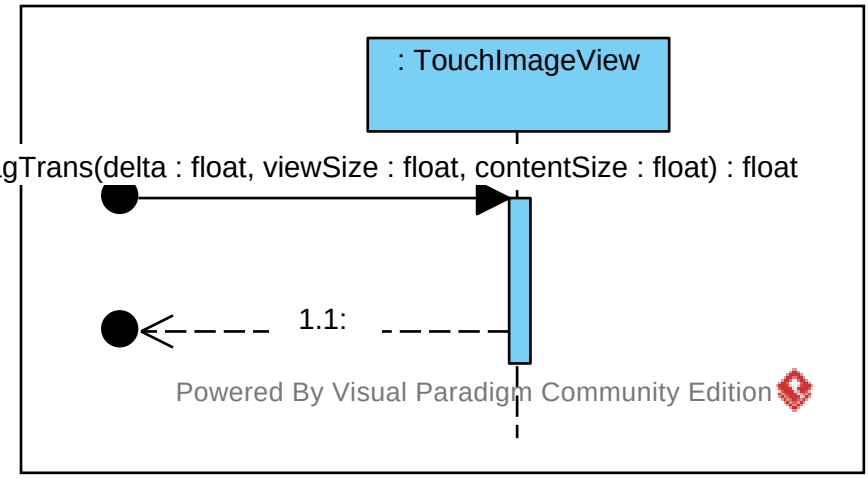

: Successor
1: Successor(sucId : int, description : String)
1.1:
Powered By Visual Paradigm Community Edition

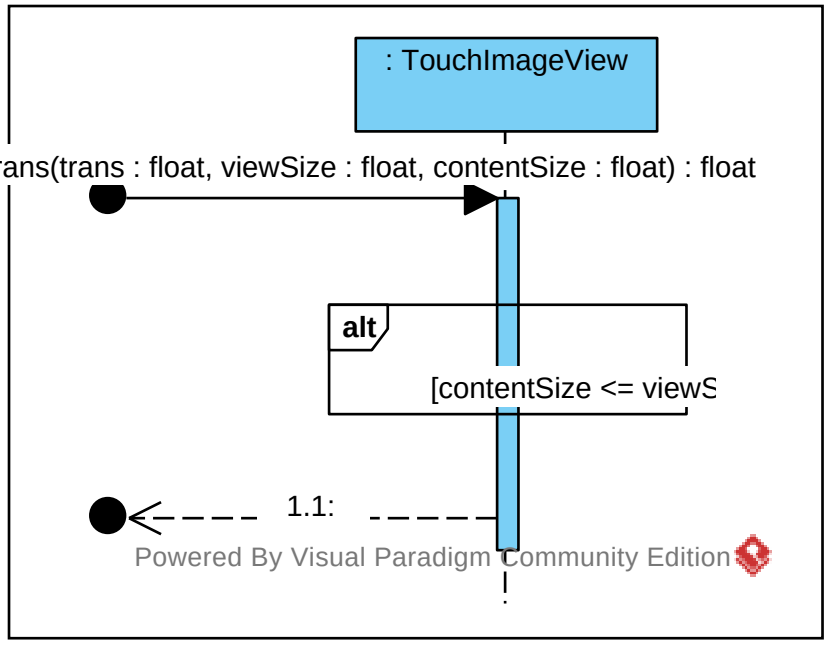

: SwipeAdapter
c : android.content.Context
e1 : java.io.IOException
list : List<AlgorithmElement>
e1 : org.xmlpull.v1.XmlPullParserException
e1 : java.io.IOException
1: getCount() : int
alt
[try]
1.1: getAssets()
[catch IOException]
1.2: printStackTrace()
alt
[try]
1.3:
: XmlParser
1.4: parse(in : InputStream = is) : List
1.5: get(parentID)
[catch XmlPullParserException]
1.6: printStackTrace()
[catch IOException]
1.7: printStackTrace()
1.8:
Powered By Visual Paradigm Community Edition

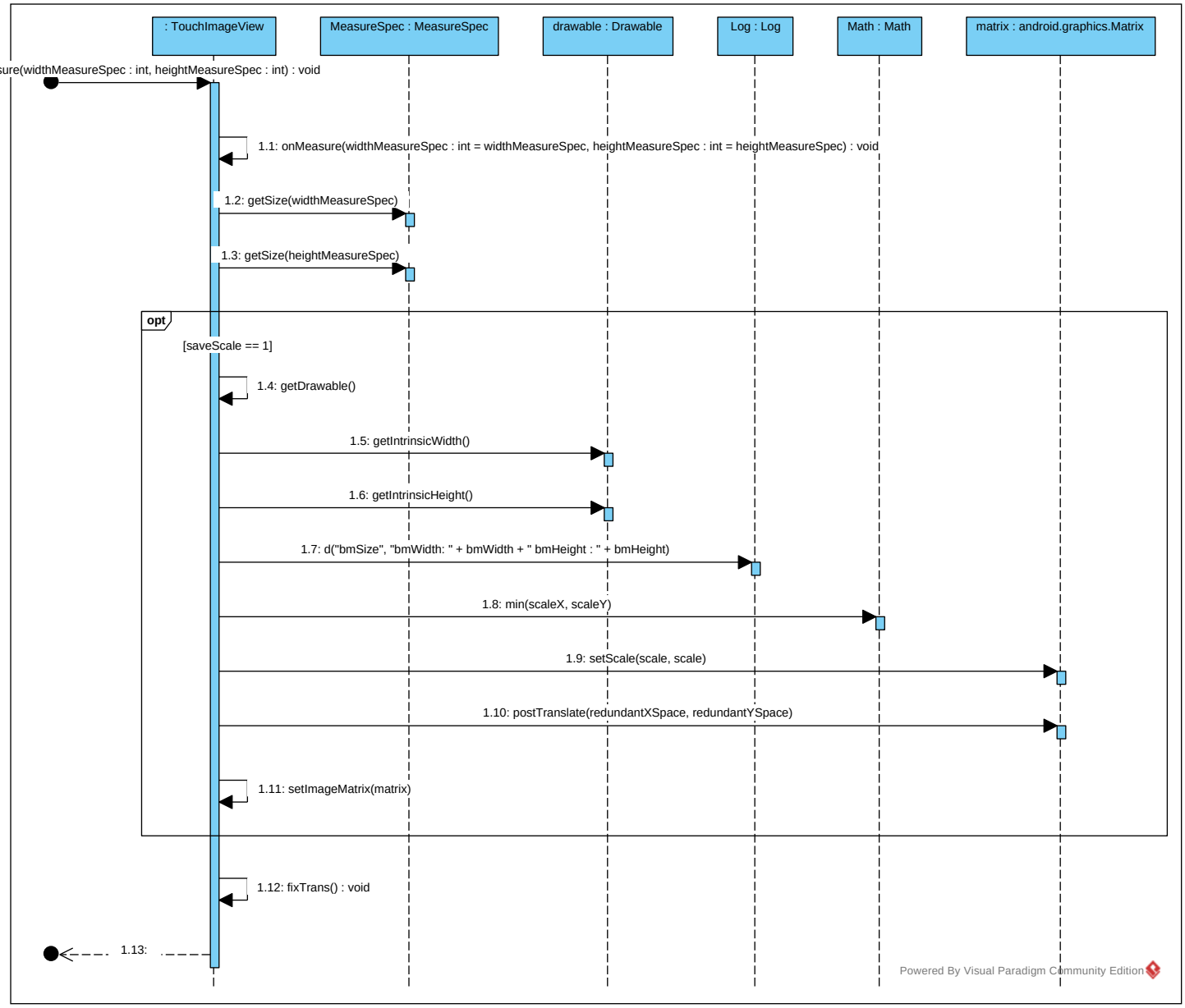

: SwipeAdapter
c : android.content.Context
e1 : java.io.IOException
list : List<AlgorithmElement>
e1 : org.xmlpull.v1.XmlPullParserException
e1 : java.io.IOException
1: getItem(position : int) : Fragment
alt
[try]
1.1: getAssets()
[catch IOException]
1.2: printStackTrace()
alt
[try]
1.3:
: XmlParser
1.4: parse(in : InputStream = is) : List
1.5: get(parentID)
1.6: get(parentID)
1.7: get(parentID)
1.8: get(nextelement-1)
1.9: getResources()
1.10: get(nextelement-1)
[catch XmlPullParserException]
1.11: printStackTrace()
[catch IOException]
1.12: printStackTrace()
1.13:
fragment : PageFragment
1.14:
bundle : android.os.Bundle
1.15: putInt("position",nextelement-1)
1.16: putInt("count_successors",count_successors)
1.17: putString("opis",opis)
1.18: putString("krok",krok)
1.19: putString("extrainfo",extrainfo)
1.20: putInt("image",image)
1.21: putString("xml", xmlfile)
1.22: putString("prev",prev)
1.23: putString("next",next)
1.24: setArguments(bundle)
1.25:
Powered By Visual Paradigm Community Edition

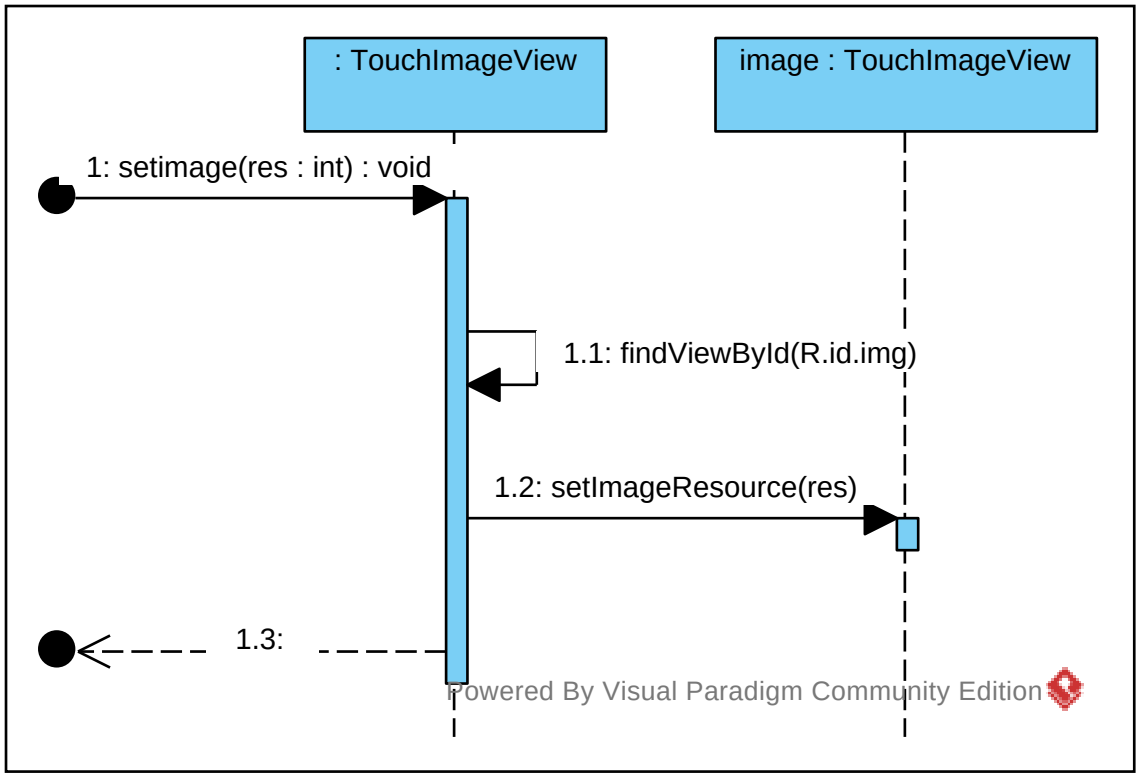

: SwipeAdapter
pter(fm : FragmentManager, c : Context, parentID : int, xmlfile : String)
1.1:
Powered By Visual Paradigm Community Edition

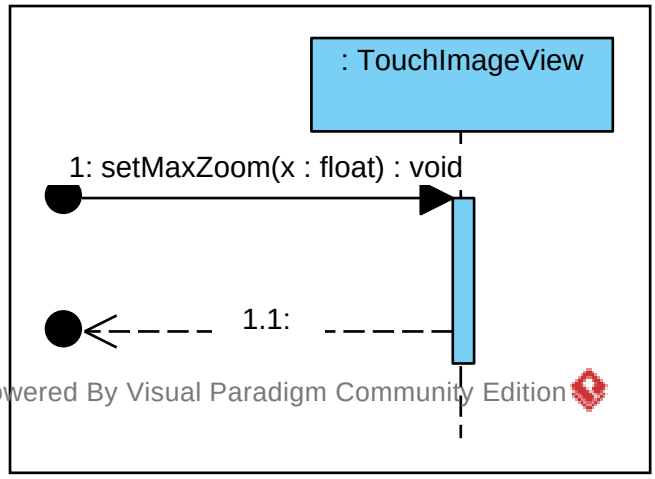

: TouchImageView
matrix : android.graphics.Matrix
1: fixTrans() : void
1.1: getValues(m)
1.2: getFixTrans(trans : float = transX, viewSize : float = viewWidth, contentSize : float = origWidth * saveScale) : float
1.3: getFixTrans(trans : float = transY, viewSize : float = viewHeight, contentSize : float = origHeight
* saveScale) : float
opt
[fixTransX != 0 || fixTransY != 0]
1.4: postTranslate(fixTransX, fixTransY)
1.5:
Powered By Visual Paradigm Community Edition

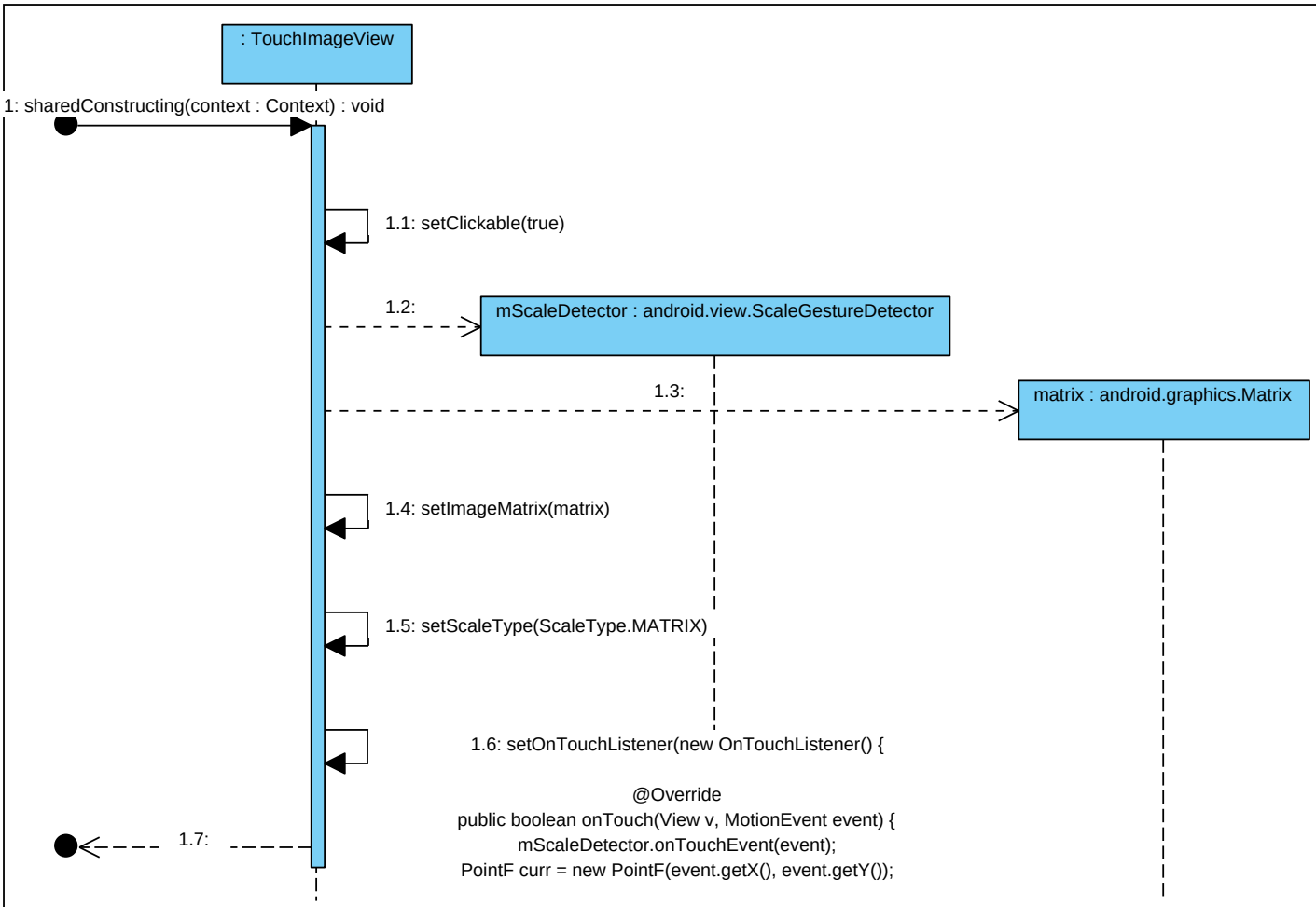

: TouchImageView
xDragTrans(delta : float, viewSize : float, contentSize : float) : float
1.1:
Powered By Visual Paradigm Community Edition

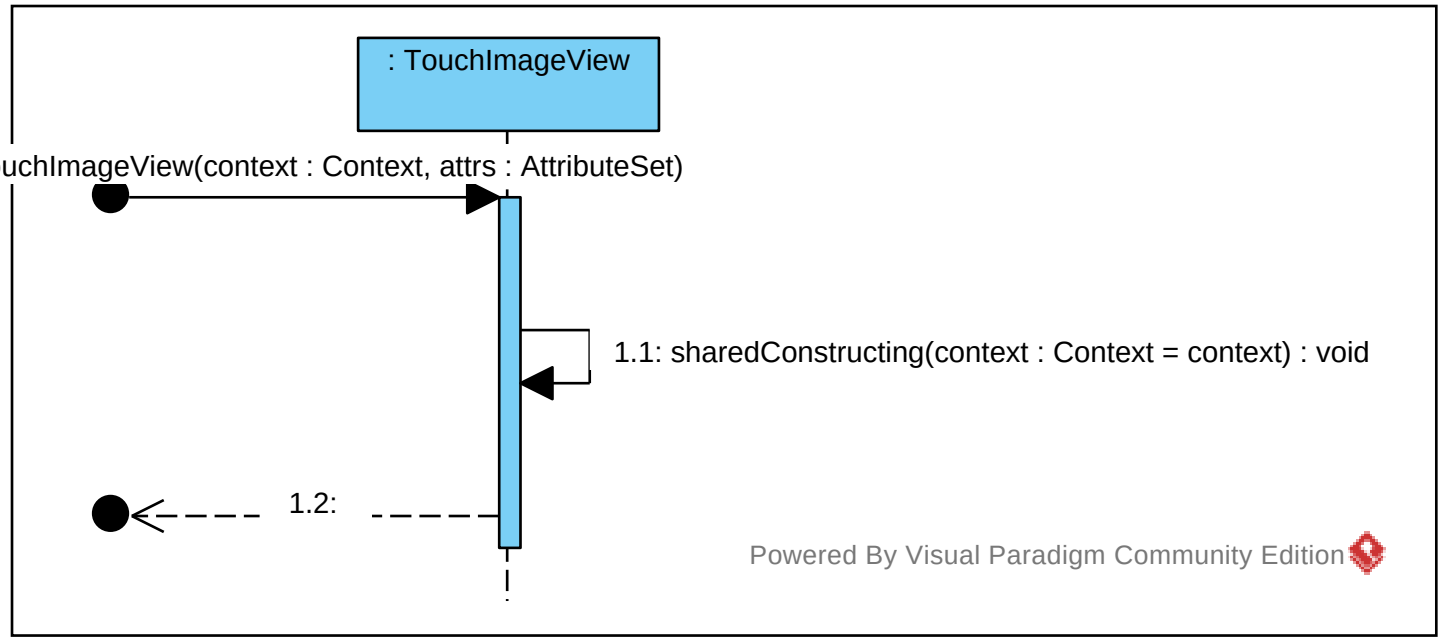

: TouchImageView
tFixTrans(trans : float, viewSize : float, contentSize : float) : float
alt
[contentSize <= viewS
1.1:
Powered By Visual Paradigm Community Edition

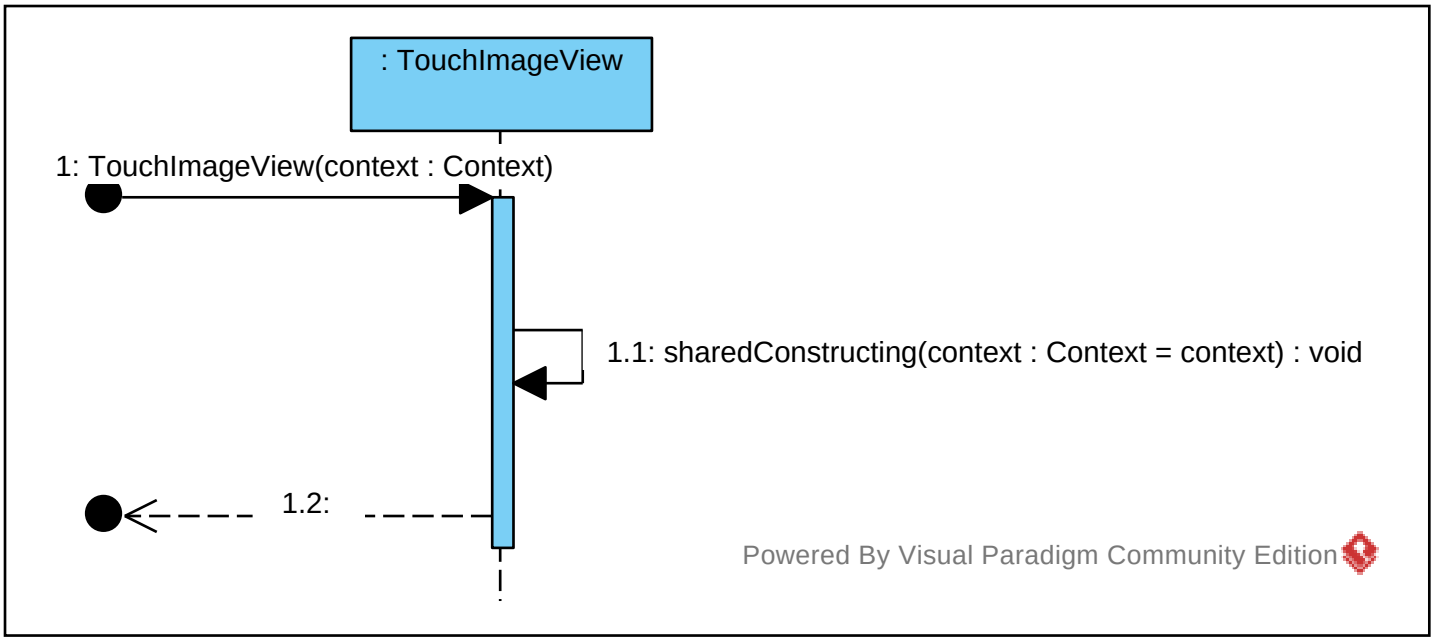

: TouchImageView
MeasureSpec : MeasureSpec
drawable : Drawable
Log : Log
Math : Math
matrix : android.graphics.Matrix
1: onMeasure(widthMeasureSpec : int, heightMeasureSpec : int) : void
1.1: onMeasure(widthMeasureSpec : int = widthMeasureSpec, heightMeasureSpec : int = heightMeasureSpec) : void
1.2: getSize(widthMeasureSpec)
1.3: getSize(heightMeasureSpec)
opt
[saveScale == 1]
1.4: getDrawable()
1.5: getIntrinsicWidth()
1.6: getIntrinsicHeight()
1.7: d("bmSize", "bmWidth: " + bmWidth + " bmHeight : " + bmHeight)
1.8: min(scaleX, scaleY)
1.9: setScale(scale, scale)
1.10: postTranslate(redundantXSpace, redundantYSpace)
1.11: setImageMatrix(matrix)
1.12: fixTrans() : void
1.13:
Powered By Visual Paradigm Community Edition

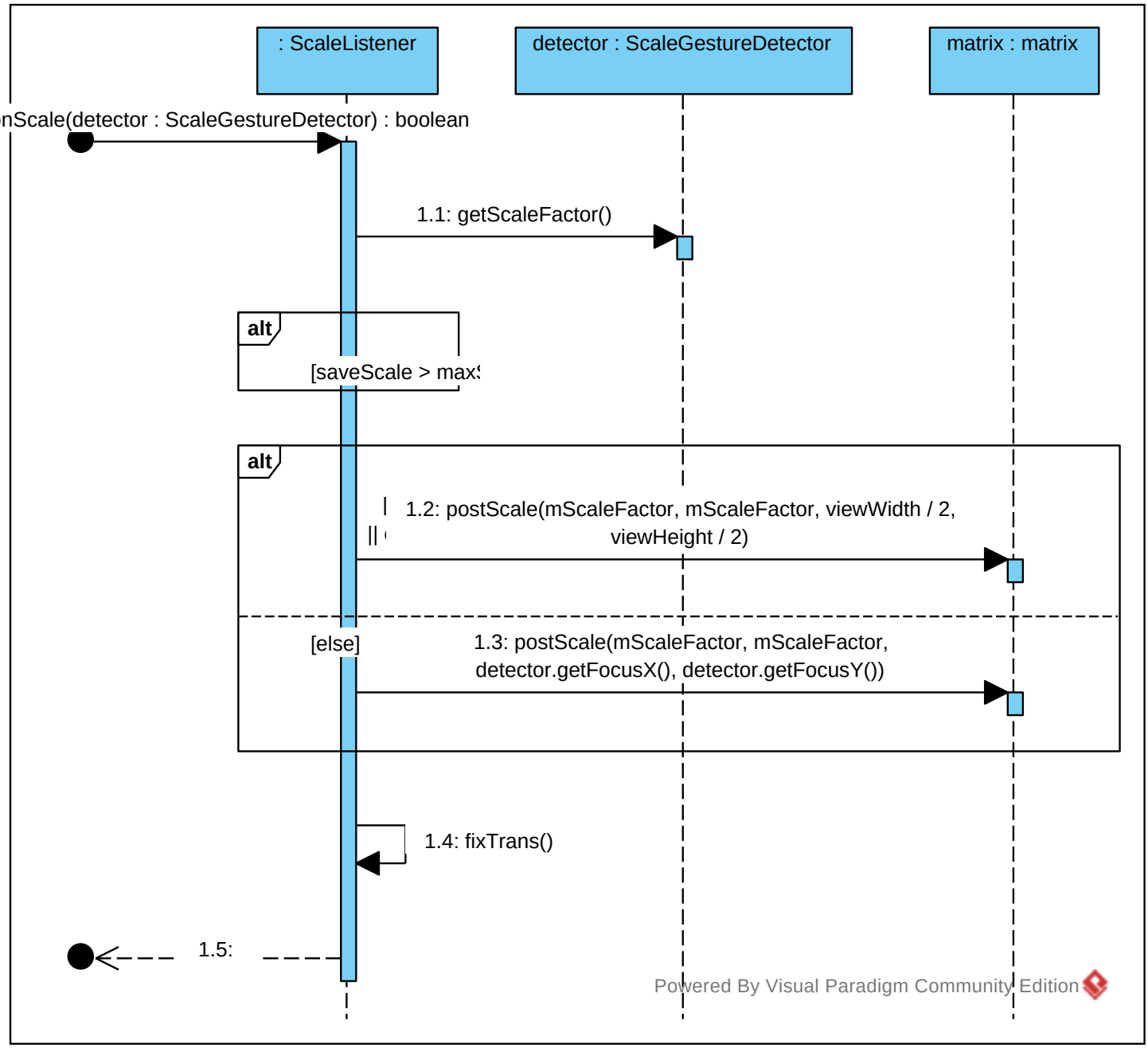

: TouchImageView
image : TouchImageView
1: setimage(res : int) : void
1.1: findViewById(R.id.img)
1.2: setImageResource(res)
1.3:
Powered By Visual Paradigm Community Edition

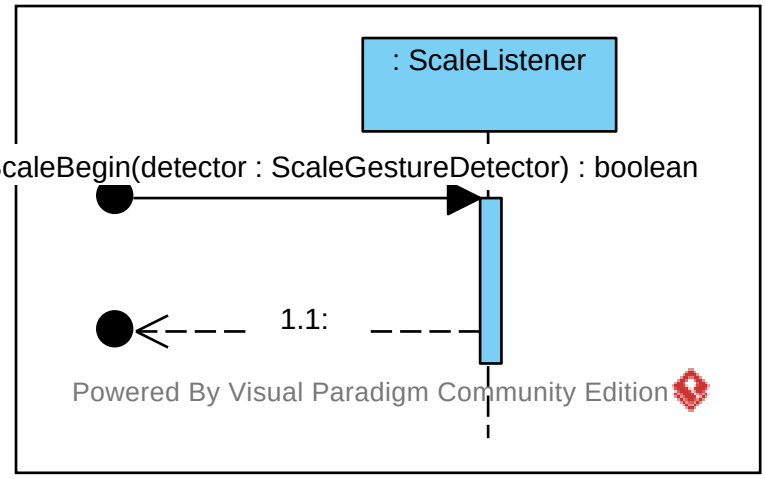

: TouchImageView
1: setMaxZoom(x : float) : void
1.1:
Powered By Visual Paradigm Community Edition

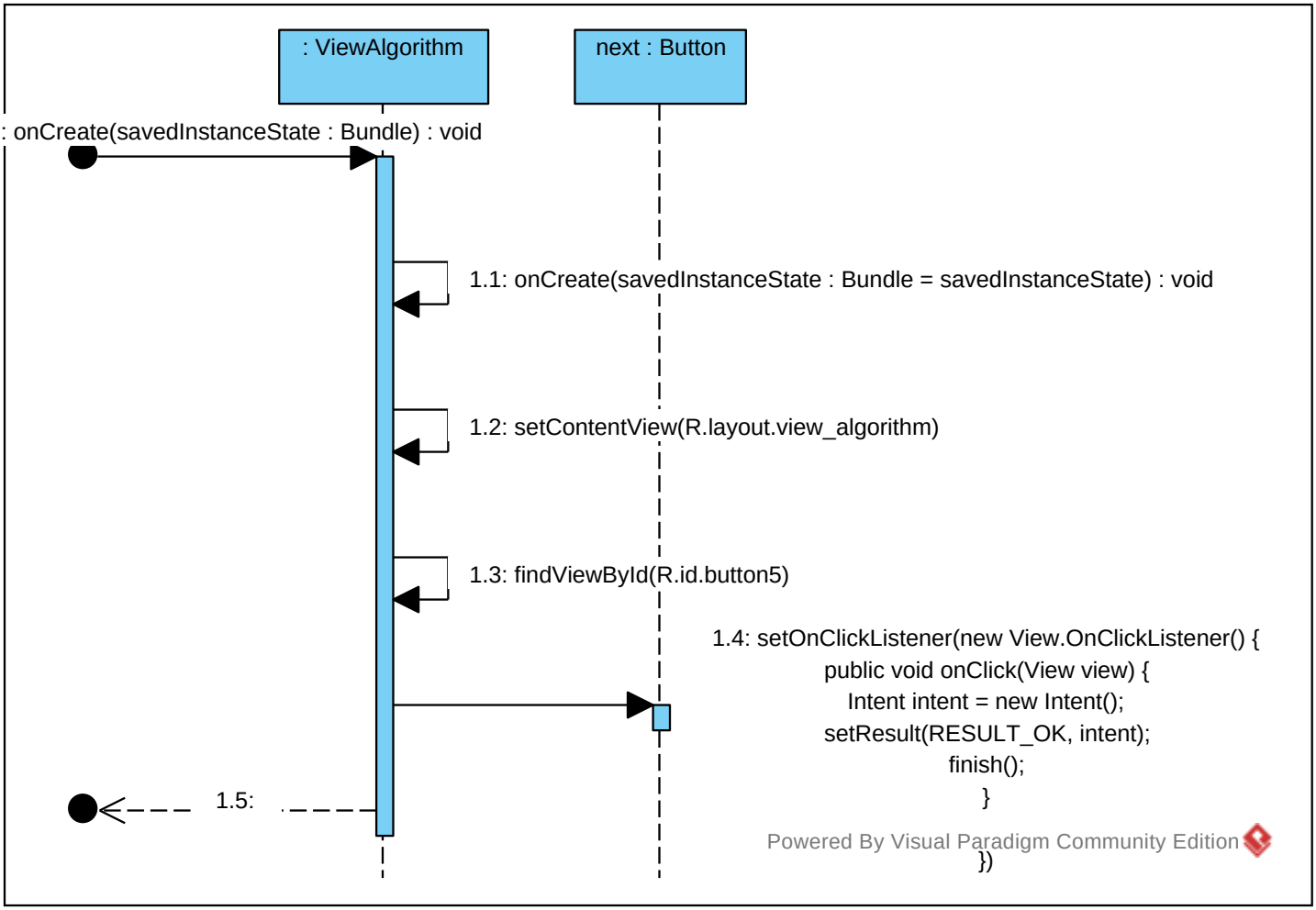

: TouchImageView
1: sharedConstructing(context : Context) : void
1.1: setClickable(true)
1.2:
mScaleDetector : android.view.ScaleGestureDetector
1.3:
matrix : android.graphics.Matrix
1.4: setImageMatrix(matrix)
1.5: setScaleType(ScaleType.MATRIX)
1.6: setOnTouchListener(new OnTouchListener() {
@Override
public boolean onTouch(View v, MotionEvent event) {
mScaleDetector.onTouchEvent(event);
PointF curr = new PointF(event.getX(), event.getY());
switch (event.getAction()) {
case MotionEvent.ACTION_DOWN:
last.set(curr);
start.set(last);
mode = DRAG;
break;
case MotionEvent.ACTION_MOVE:
if (mode == DRAG) {
float deltaX = curr.x - last.x;
float deltaY = curr.y - last.y;
float fixTransX = getFixDragTrans(deltaX, viewWidth,
origWidth * saveScale);
float fixTransY = getFixDragTrans(deltaY, viewHeight,
origHeight * saveScale);
matrix.postTranslate(fixTransX, fixTransY);
fixTrans();
last.set(curr.x, curr.y);
}
break;
case MotionEvent.ACTION_UP:
mode = NONE;
int xDiff = (int) Math.abs(curr.x - start.x);
int yDiff = (int) Math.abs(curr.y - start.y);
if (xDiff < CLICK && yDiff < CLICK)
performClick();
break;
case MotionEvent.ACTION_POINTER_UP:
mode = NONE;
break;
}
setImageMatrix(matrix);
invalidate();
return true; // indicate event was handled
}
})
1.7:
Powered By Visual Paradigm Community Edition

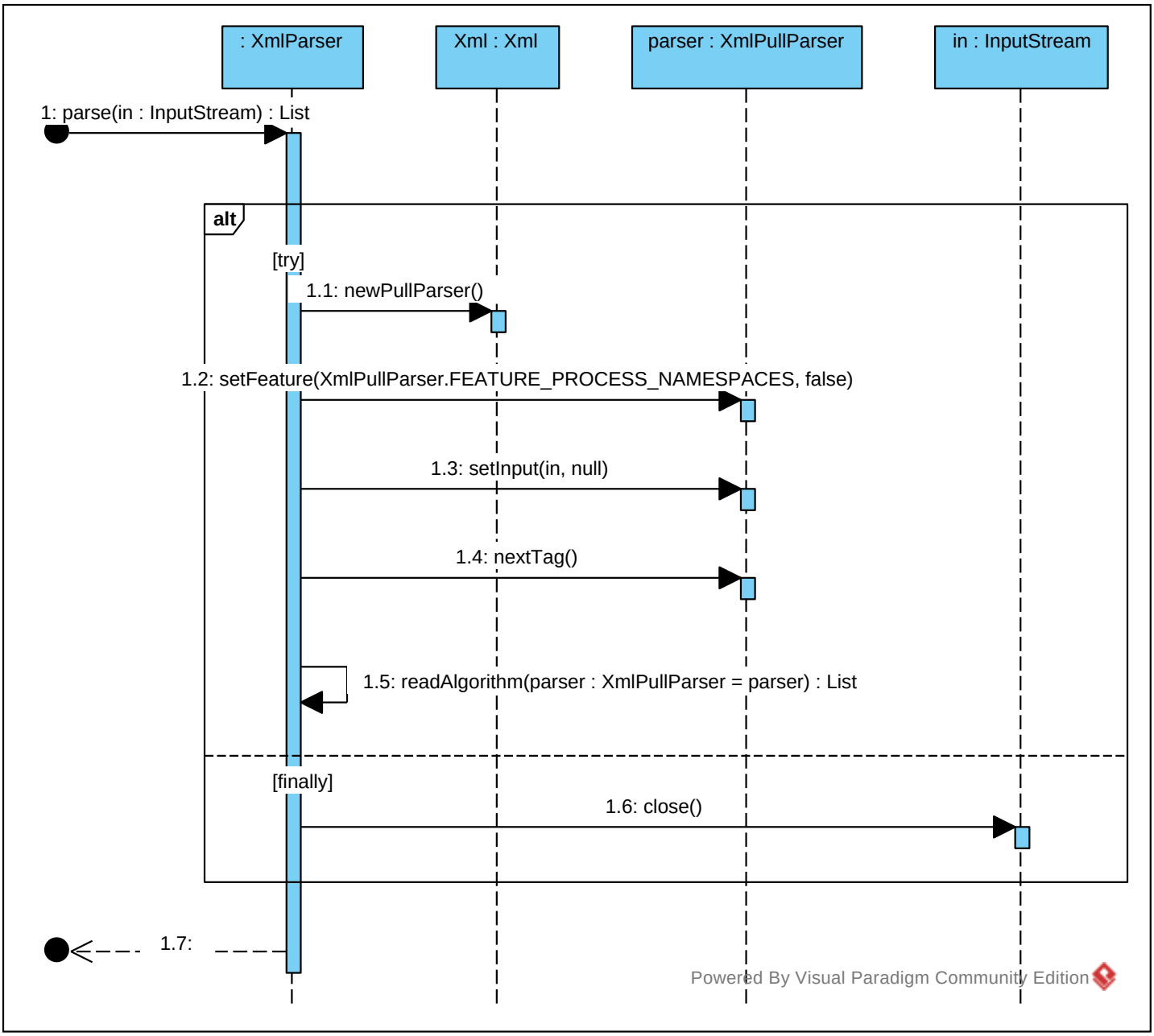

: TouchImageView
1: TouchImageView(context : Context, attrs : AttributeSet)
1.1: sharedConstructing(context : Context = context) : void
1.2:
Powered By Visual Paradigm Community Edition

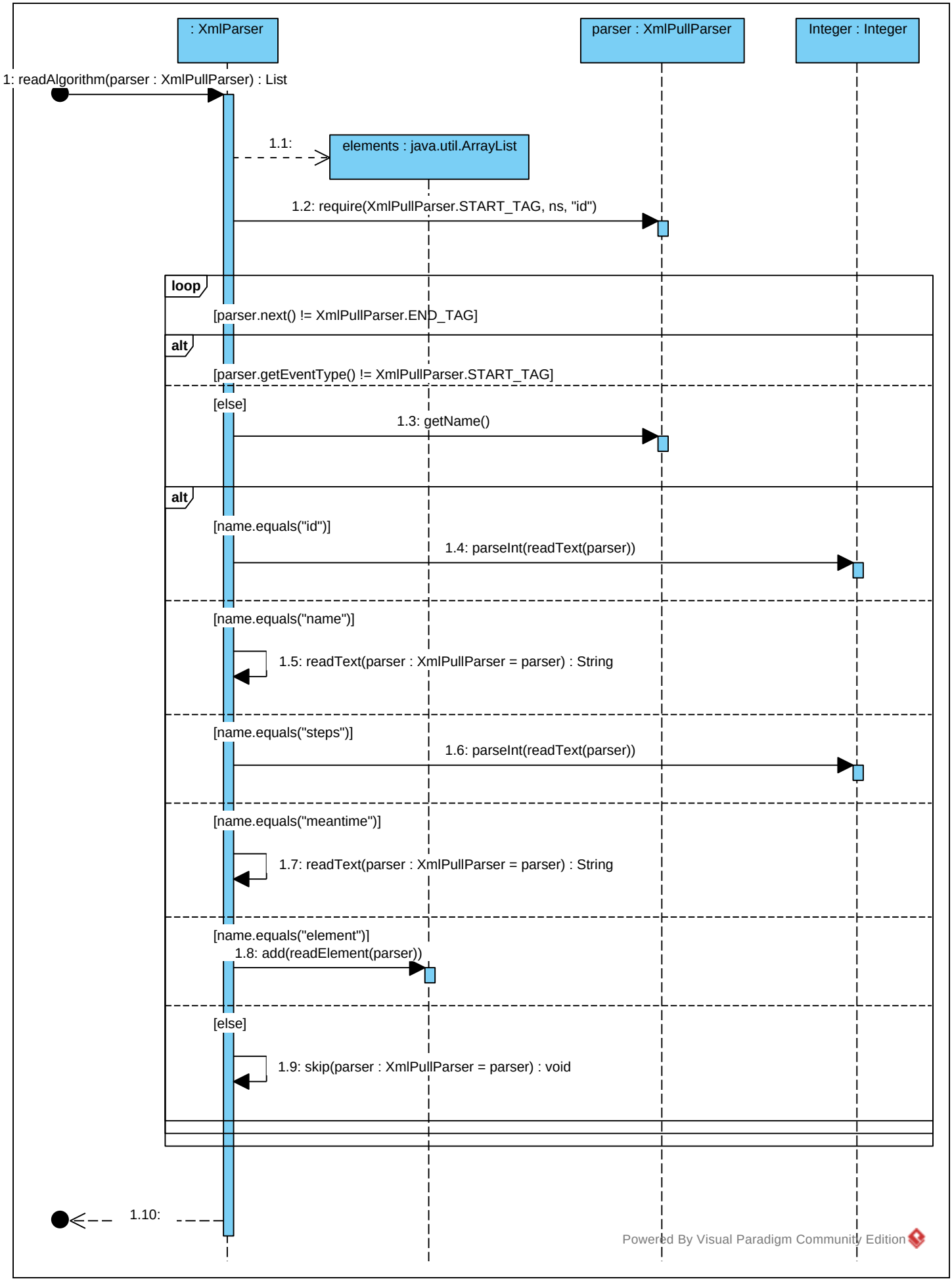

: TouchImageView
1: TouchImageView(context : Context)
1.1: sharedConstructing(context : Context = context) : void
1.2:
Powered By Visual Paradigm Community Edition

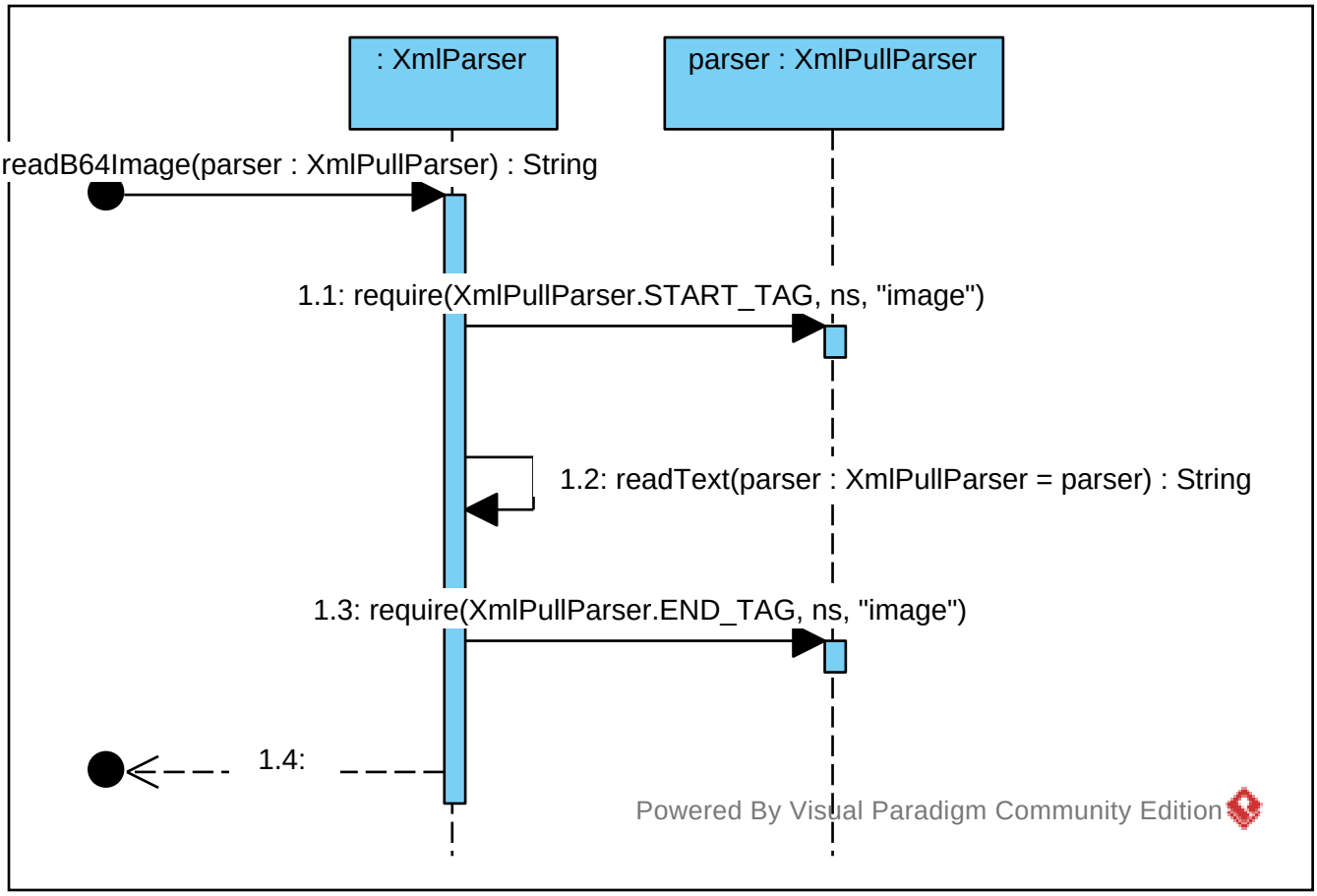

: ScaleListener
detector : ScaleGestureDetector
matrix : matrix
1: onScale(detector : ScaleGestureDetector) : boolean
1.1: getScaleFactor()
alt
[saveScale > maxS
alt
1.2: postScale(mScaleFactor, mScaleFactor, viewWidth / 2,
viewHeight / 2)
[else]
1.3: postScale(mScaleFactor, mScaleFactor,
detector.getFocusX(), detector.getFocusY())
1.4: fixTrans()
1.5:
Powered By Visual Paradigm Community Edition

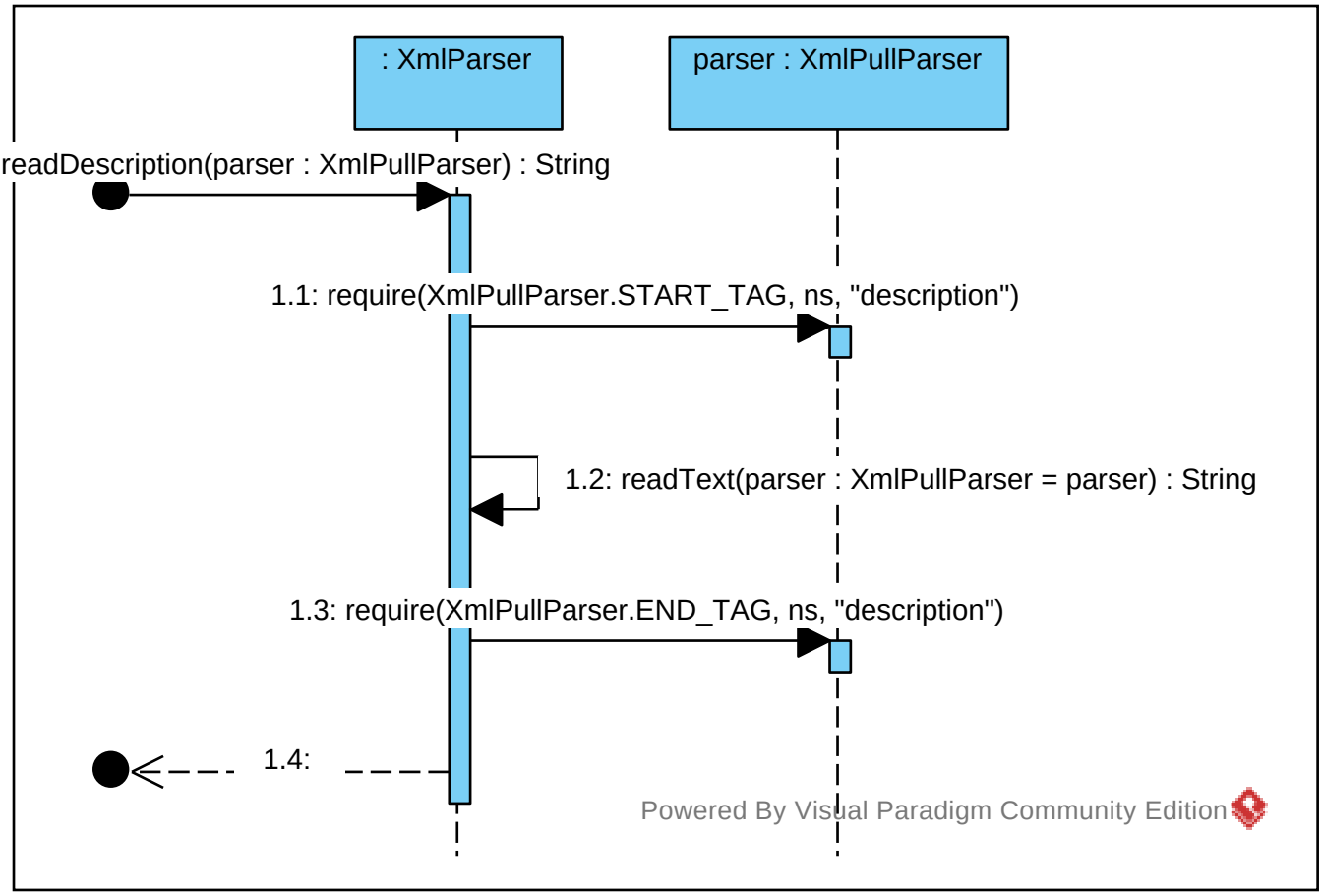

: ScaleListener
: onScaleBegin(detector : ScaleGestureDetector) : boolean
1.1:
Powered By Visual Paradigm Community Edition

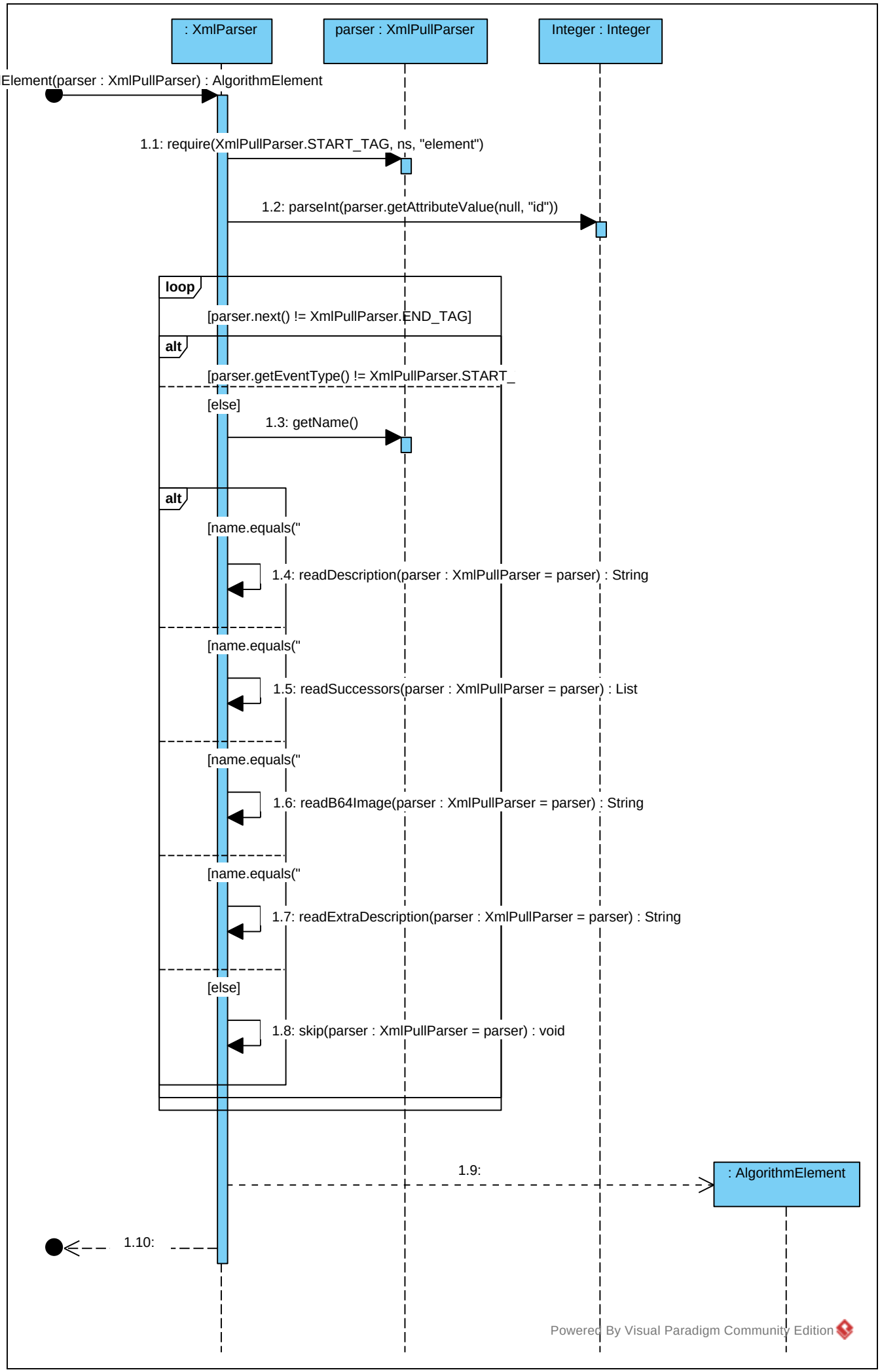

: ViewAlgorithm
next : Button
1: onCreate(savedInstanceState : Bundle) : void
1.1: onCreate(savedInstanceState : Bundle = savedInstanceState) : void
1.2: setContentView(R.layout.view_algorithm)
1.3: findViewById(R.id.button5)
1.4: setOnClickListener(new View.OnClickListener() {
public void onClick(View view) {
Intent intent = new Intent();
setResult(RESULT_OK, intent);
finish();
}
})
1.5:
Powered By Visual Paradigm Community Edition

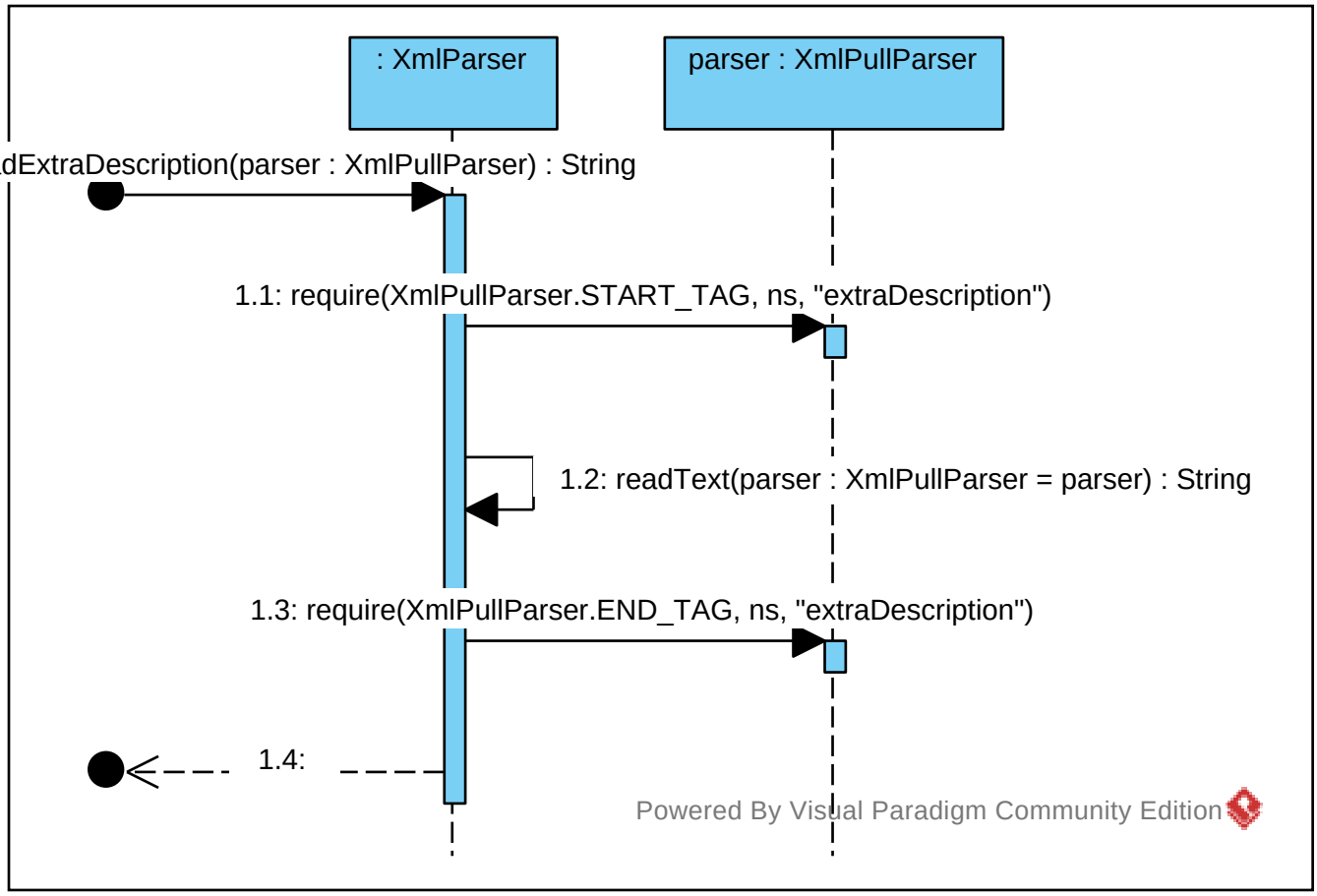

: XmlParser
Xml : Xml
parser : XmlPullParser
in : InputStream
1: parse(in : InputStream) : List
alt
[try]
1.1: newPullParser()
1.2: setFeature(XmlPullParser.FEATURE_PROCESS_NAMESPACES, false)
1.3: setInput(in, null)
1.4: nextTag()
1.5: readAlgorithm(parser : XmlPullParser = parser) : List
[finally]
1.6: close()
1.7:
Powered By Visual Paradigm Community Edition

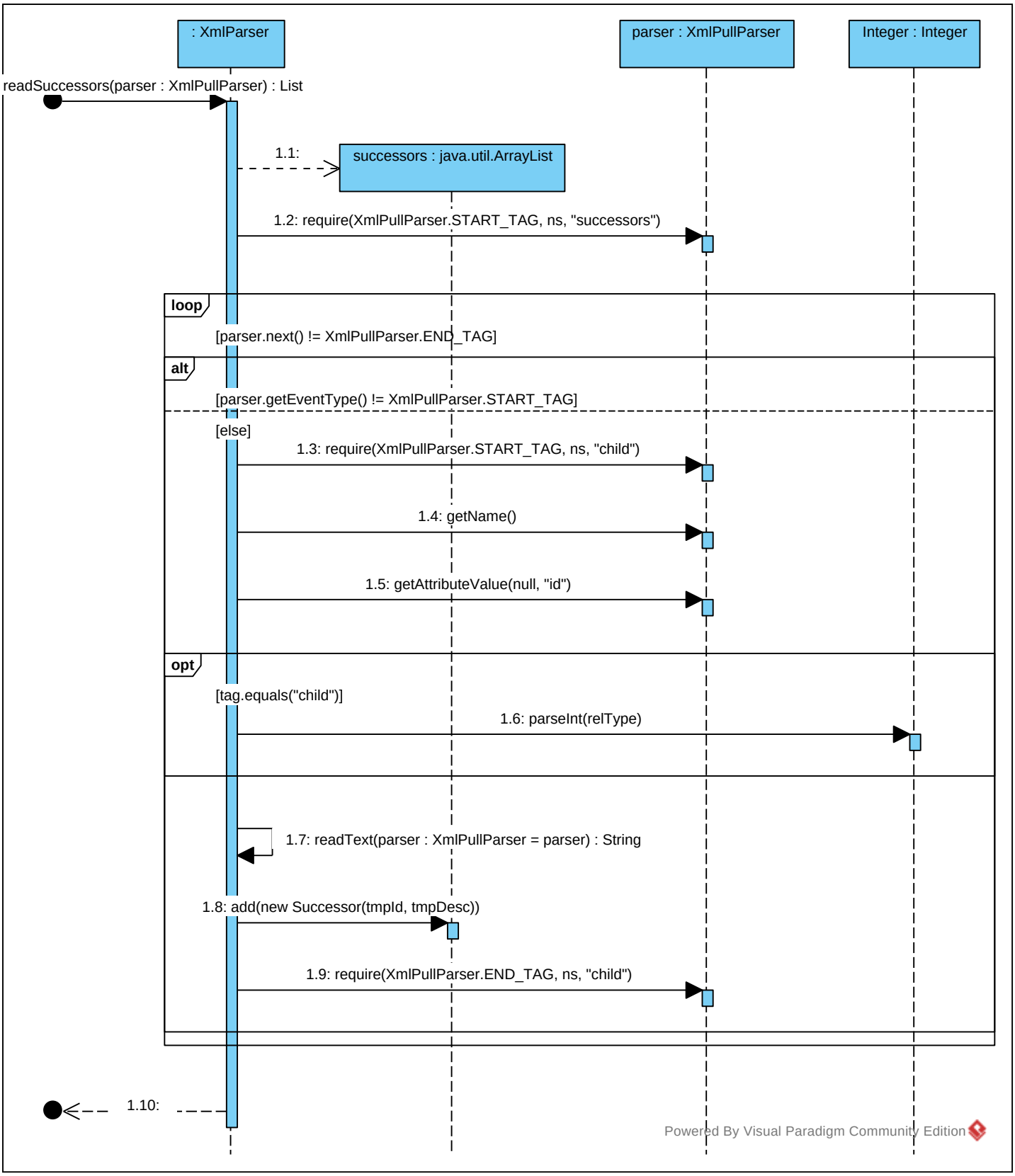

: XmlParser
parser : XmlPullParser
Integer : Integer
1: readAlgorithm(parser : XmlPullParser) : List
1.1:
elements : java.util.ArrayList
1.2: require(XmlPullParser.START_TAG, ns, "id")
loop
[parser.next() != XmlPullParser.END_TAG]
alt
[parser.getEventType() != XmlPullParser.START_TAG]
[else]
1.3: getName()
alt
[name.equals("id")]
1.4: parseInt(readText(parser))
[name.equals("name")]
1.5: readText(parser : XmlPullParser = parser) : String
[name.equals("steps")]
1.6: parseInt(readText(parser))
[name.equals("meantime")]
1.7: readText(parser : XmlPullParser = parser) : String
[name.equals("element")]
1.8: add(readElement(parser))
[else]
1.9: skip(parser : XmlPullParser = parser) : void
1.10:
Powered By Visual Paradigm Community Edition

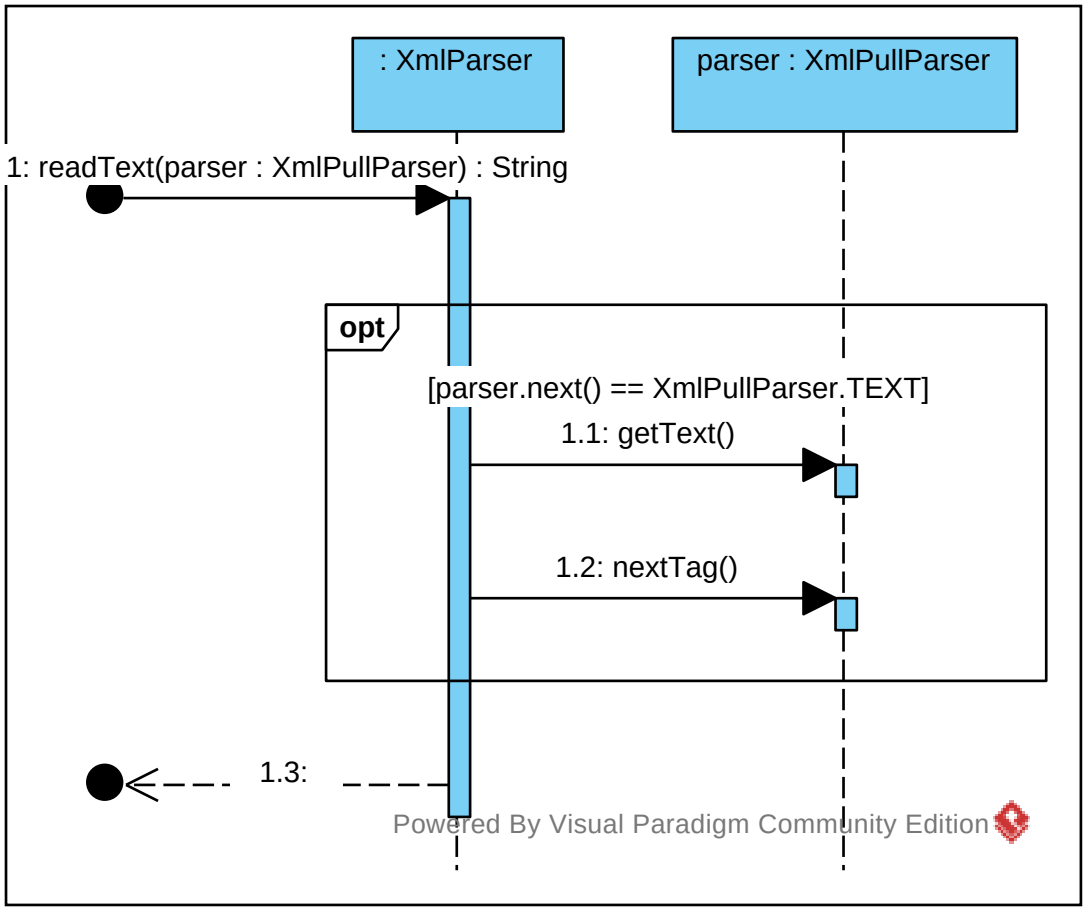

: XmlParser
parser : XmlPullParser
1: readB64Image(parser : XmlPullParser) : String
1.1: require(XmlPullParser.START_TAG, ns, "image")
1.2: readText(parser : XmlPullParser = parser) : String
1.3: require(XmlPullParser.END_TAG, ns, "image")
1.4:
Powered By Visual Paradigm Community Edition

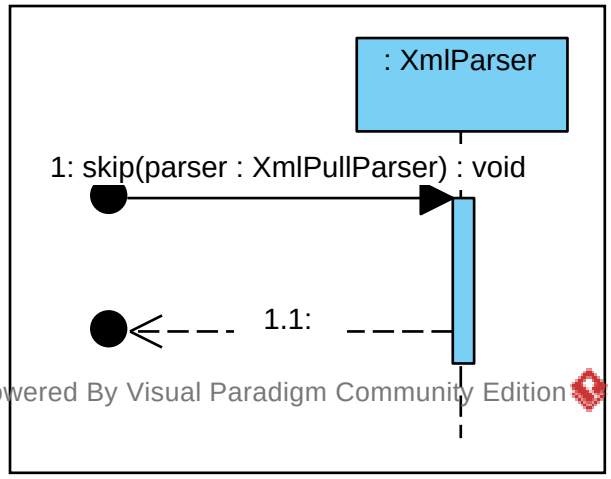

: XmlParser
parser : XmlPullParser
1: readDescription(parser : XmlPullParser) : String
1.1: require(XmlPullParser.START_TAG, ns, "description")
1.2: readText(parser : XmlPullParser = parser) : String
1.3: require(XmlPullParser.END_TAG, ns, "description")
1.4:
Powered By Visual Paradigm Community Edition


: XmlParser
parser : XmlPullParser
Integer : Integer
1: readElement(parser : XmlPullParser) : AlgorithmElement
1.1: require(XmlPullParser.START_TAG, ns, "element")
1.2: parseInt(parser.getAttributeValue(null, "id"))
loop
[parser.next() != XmlPullParser.END_TAG]
alt
[parser.getEventType() != XmlPullParser.START_
[else]
1.3: getName()
alt
[name.equals("
1.4: readDescription(parser : XmlPullParser = parser) : String
[name.equals("
1.5: readSuccessors(parser : XmlPullParser = parser) : List
[name.equals("
1.6: readB64Image(parser : XmlPullParser = parser) : String
[name.equals("
1.7: readExtraDescription(parser : XmlPullParser = parser) : String
[else]
1.8: skip(parser : XmlPullParser = parser) : void
1.9:
: AlgorithmElement
1.10:
Powered By Visual Paradigm Community Edition


: XmlParser
parser : XmlPullParser
1: readExtraDescription(parser : XmlPullParser) : String
1.1: require(XmlPullParser.START_TAG, ns, "extraDescription")
1.2: readText(parser : XmlPullParser = parser) : String
1.3: require(XmlPullParser.END_TAG, ns, "extraDescription")
1.4:
Powered By Visual Paradigm Community Edition


: XmlParser
parser : XmlPullParser
Integer : Integer
1: readSuccessors(parser : XmlPullParser) : List
1.1:
successors : java.util.ArrayList
1.2: require(XmlPullParser.START_TAG, ns, "successors")
loop
[parser.next() != XmlPullParser.END_TAG]
alt
[parser.getEventType() != XmlPullParser.START_TAG]
[else]
1.3: require(XmlPullParser.START_TAG, ns, "child")
1.4: getName()
1.5: getAttributeValue(null, "id")
opt
[tag.equals("child")]
1.6: parseInt(relType)
1.7: readText(parser : XmlPullParser = parser) : String
1.8: add(new Successor(tmpId, tmpDesc))
1.9: require(XmlPullParser.END_TAG, ns, "child")
1.10:
Powered By Visual Paradigm Community Edition


: XmlParser
parser : XmlPullParser
1: readText(parser : XmlPullParser) : String
opt
[parser.next() == XmlPullParser.TEXT]
1.1: getText()
1.2: nextTag()
1.3:
Powered By Visual Paradigm Community Edition


: XmlParser
1: skip(parser : XmlPullParser) : void
1.1:
Powered By Visual Paradigm Community Edition