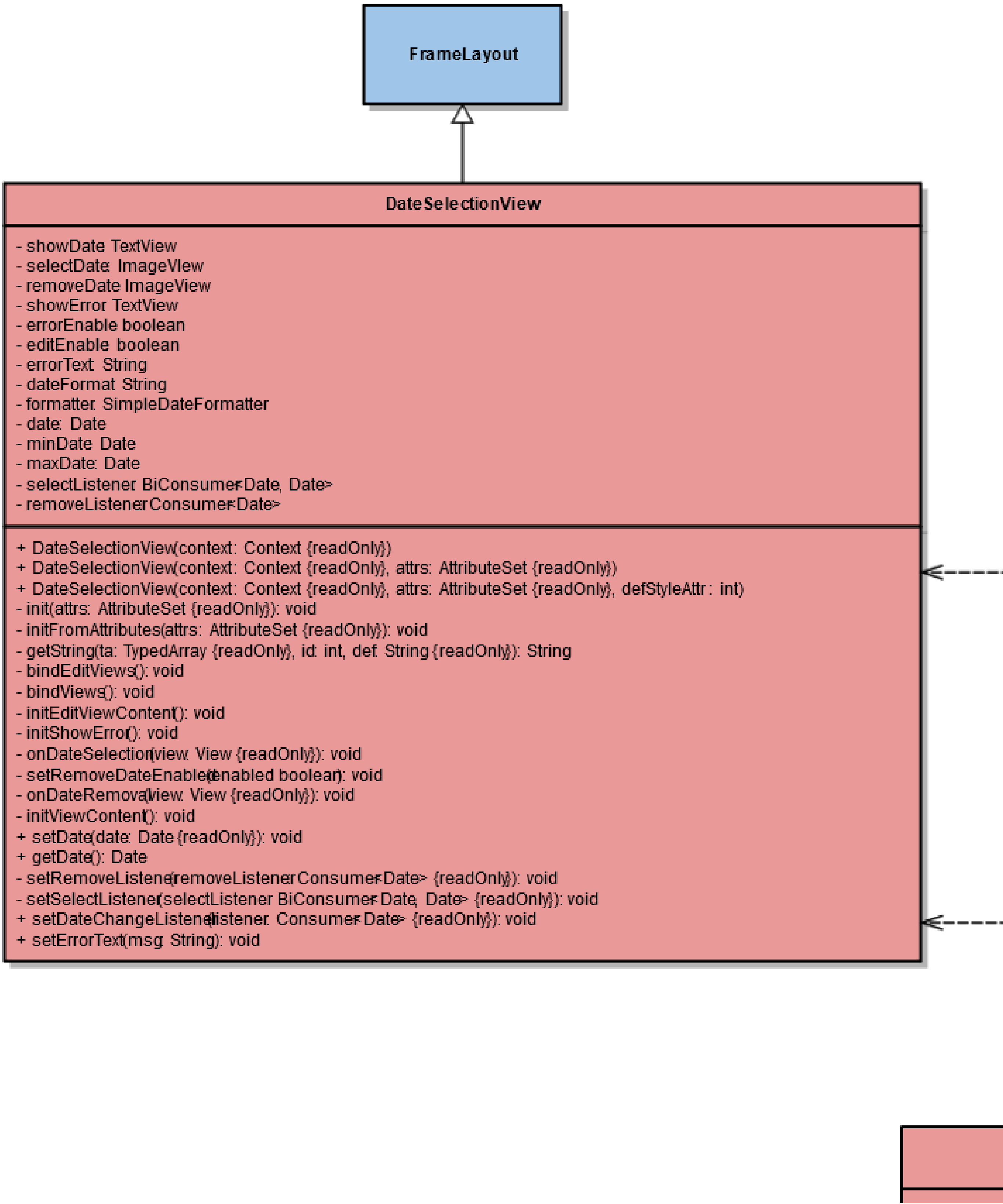
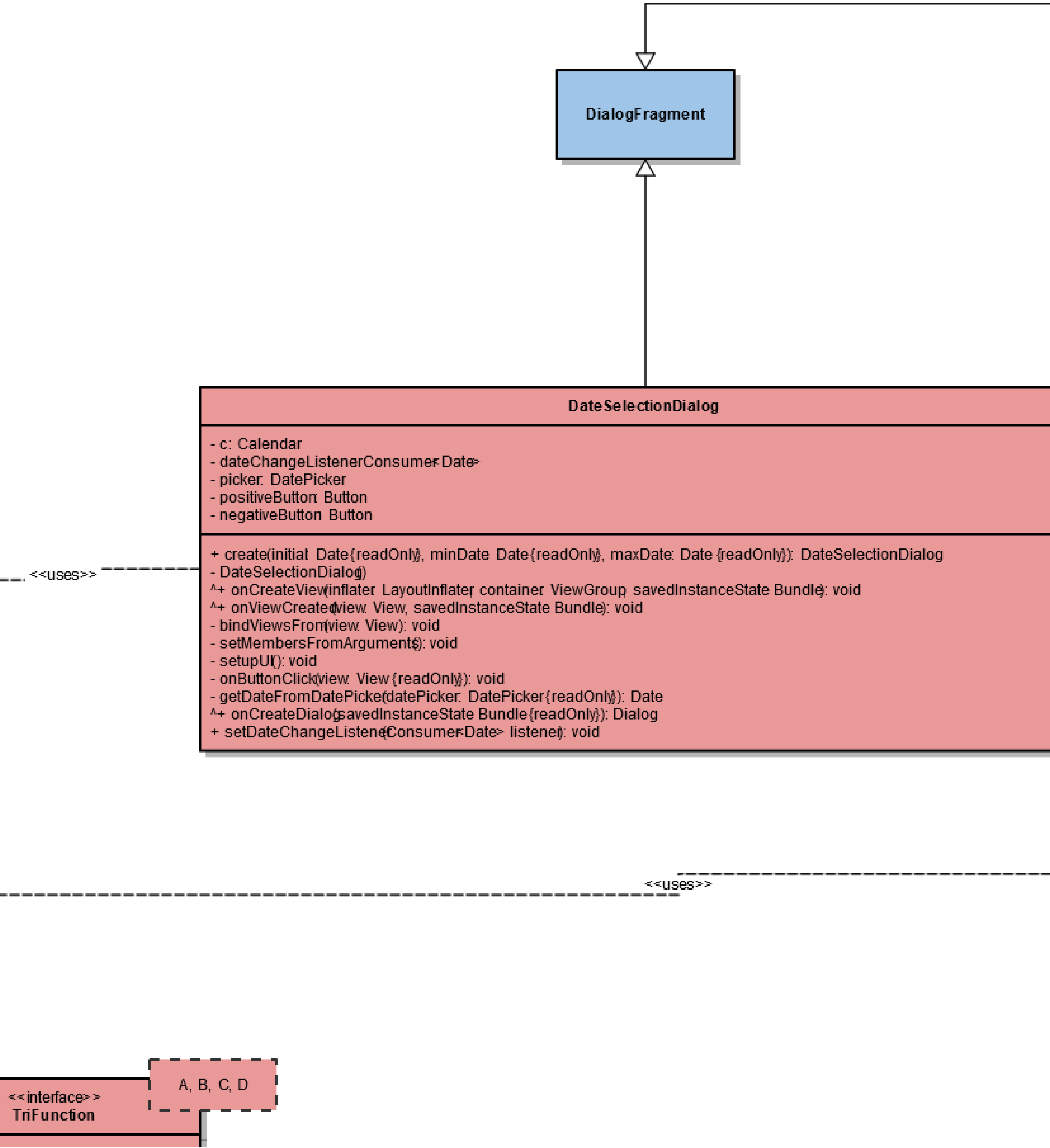
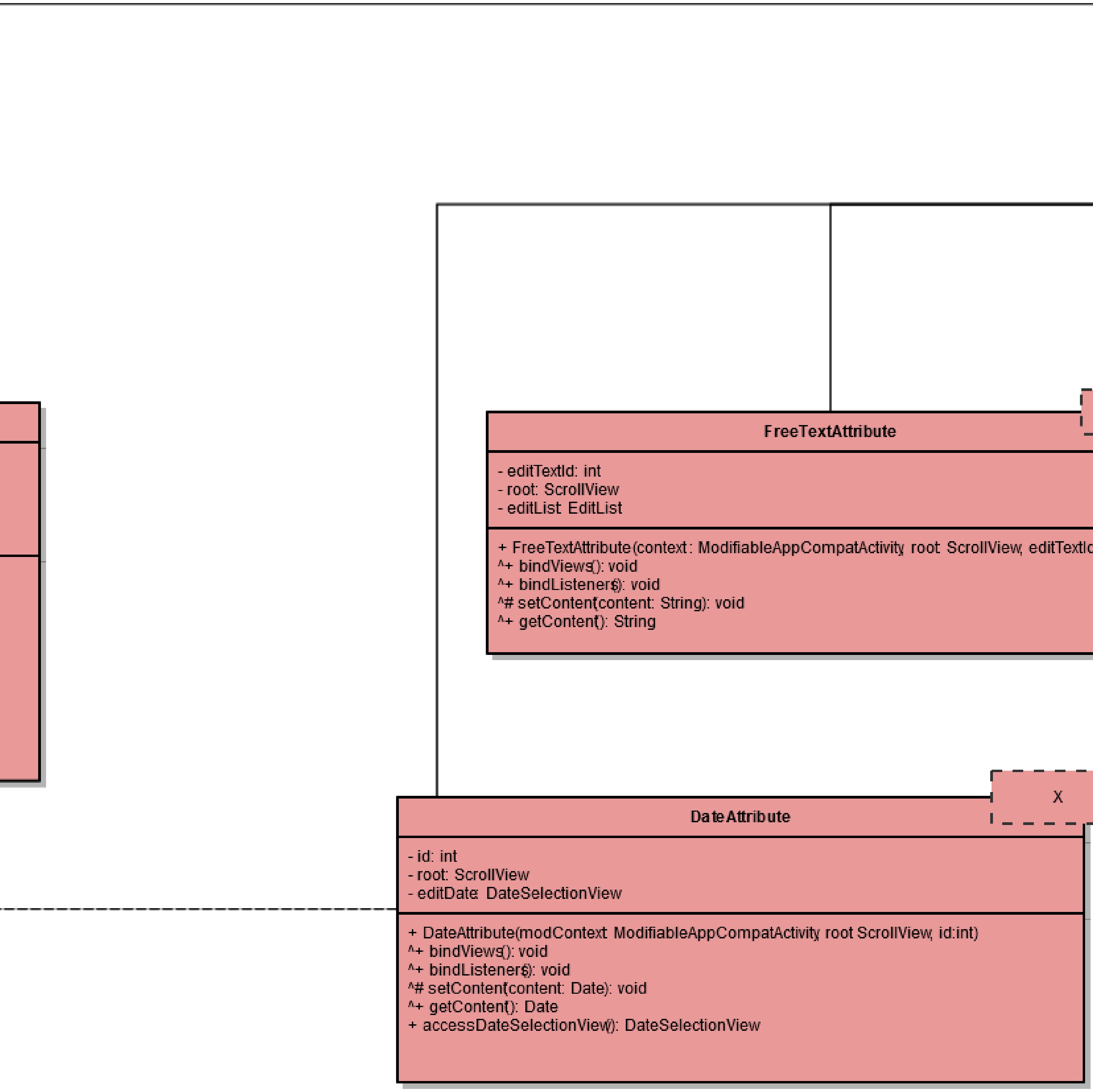
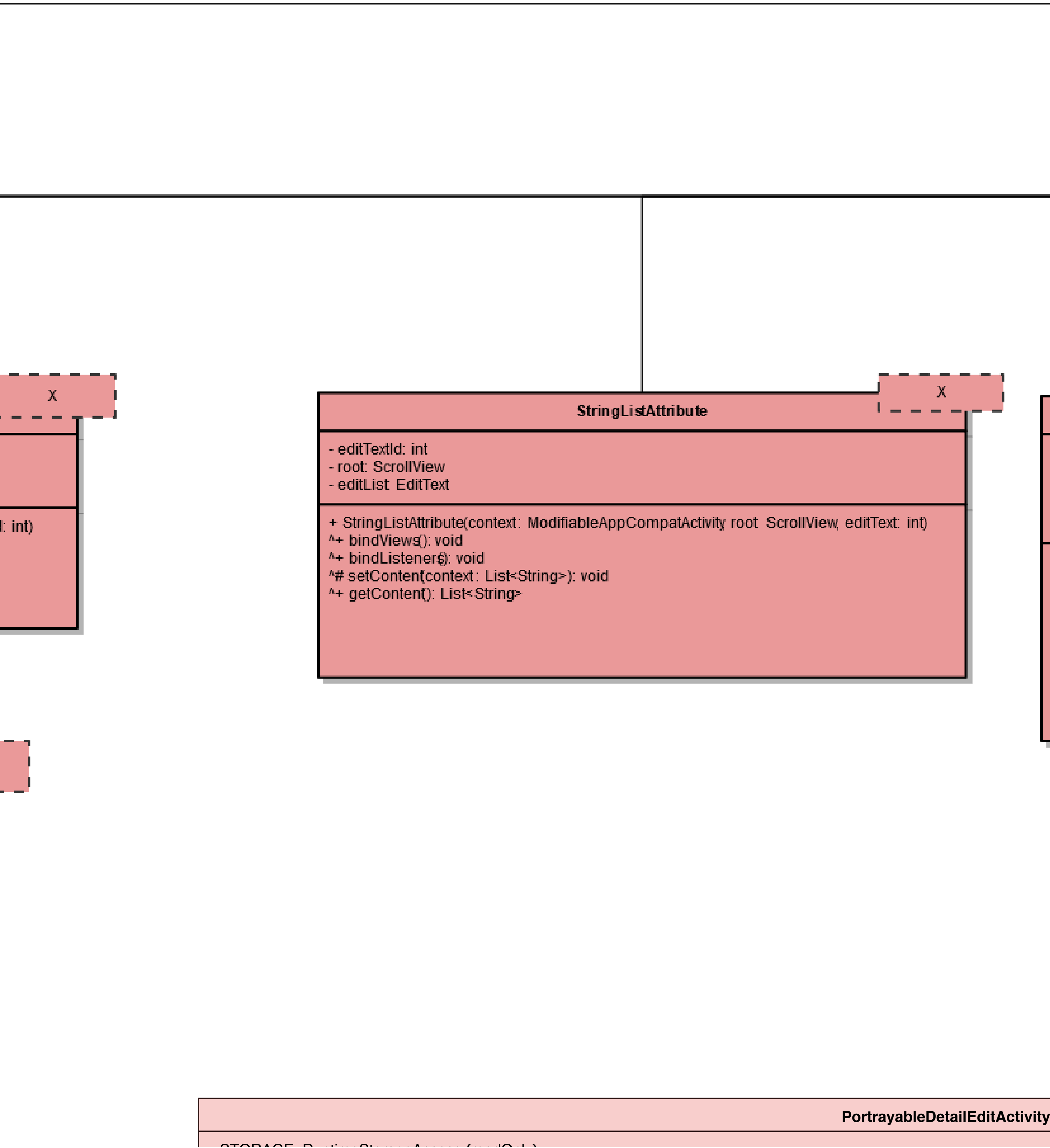
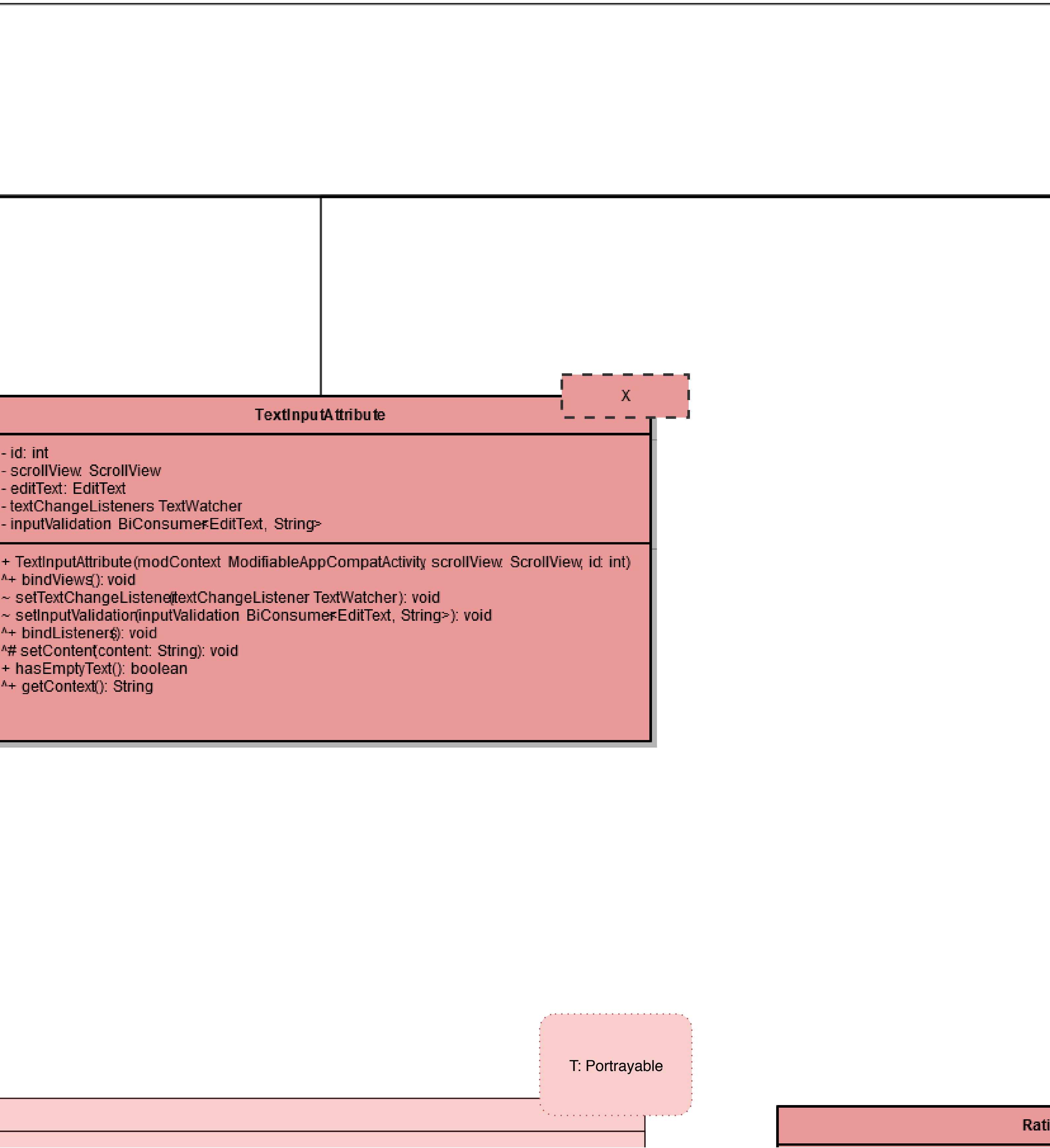

FrameLayout
DateSelectionView
- showDate: TextView
- selectDate: ImageVIew
- removeDate: ImageView
- showError: TextView
- errorEnable: boolean
- editEnable: boolean
- errorText: String
- dateFormat: String
- formatter: SimpleDateFormatter
- date: Date
- minDate: Date
- maxDate: Date
- selectListener: BiConsumer<Date, Date>
- removeListener: Consumer<Date>
+ DateSelectionView(context: Context {readOnly})
+ DateSelectionView(context: Context {readOnly}, attrs: AttributeSet {readOnly})
+ DateSelectionView(context: Context {readOnly}, attrs: AttributeSet {readOnly}, defStyleAttr : int)
- init(attrs: AttributeSet {readOnly}): void
- initFromAttributes(attrs: AttributeSet {readOnly}): void
- getString(ta: TypedArray {readOnly}, id: int, def: String {readOnly}): String
- bindEditViews(): void
- bindViews(): void
- initEditViewContent(): void
- initShowError(): void
- onDateSelection(view: View {readOnly}): void
- setRemoveDateEnabled(enabled boolean): void
- onDateRemoval(view: View {readOnly}): void
- initViewContent(): void
+ setDate(date: Date {readOnly}): void
+ getDate(): Date
- setRemoveListener(removeListener: Consumer<Date> {readOnly}): void
- setSelectListener(selectListener: BiConsumer<Date, Date> {readOnly}): void
+ setDateChangeListener(listener: Consumer<Date> {readOnly}): void
+ setErrorText(msg: String): void


DialogFragment
DateSelectionDialog
- c: Calendar
- dateChangeListener: Consumer<Date>
- picker: DatePicker
- positiveButton: Button
- negativeButton: Button
+ create(initial: Date{readOnly}, minDate: Date{readOnly}, maxDate: Date {readOnly}): DateSelectionDialog
- DateSelectionDialog()
^+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): void
^+ onViewCreated(view: View, savedInstanceState: Bundle): void
- bindViewsFrom(view: View): void
- setMembersFromArguments(): void
- setupUI(): void
- onButtonClick(view: View{readOnly}): void
- getDateFromDatePicker(datePicker: DatePicker{readOnly}): Date
^+ onCreateDialog(savedInstanceState: Bundle{readOnly}): Dialog
+ setDateChangeListener(Consumer<Date> listener): void
<<uses>>
<<uses>>
<<interface>>
TriFunction
A, B, C, D


FreeTextAttribute
- editTextId: int
- root: ScrollView
- editList: EditList
+ FreeTextAttribute(context: ModifiableAppCompatActivity, root: ScrollView, editTextId
^+ bindViews(): void
^+ bindListeners(): void
^# setContent(content: String): void
^+ getContent(): String
X
DateAttribute
- id: int
- root: ScrollView
- editDate: DateSelectionView
+ DateAttribute(modContext: ModifiableAppCompatActivity, root:ScrollView, id:int)
^+ bindViews(): void
^+ bindListeners(): void
^# setContent(content: Date): void
^+ getContent(): Date
+ accessDateSelectionView(): DateSelectionView



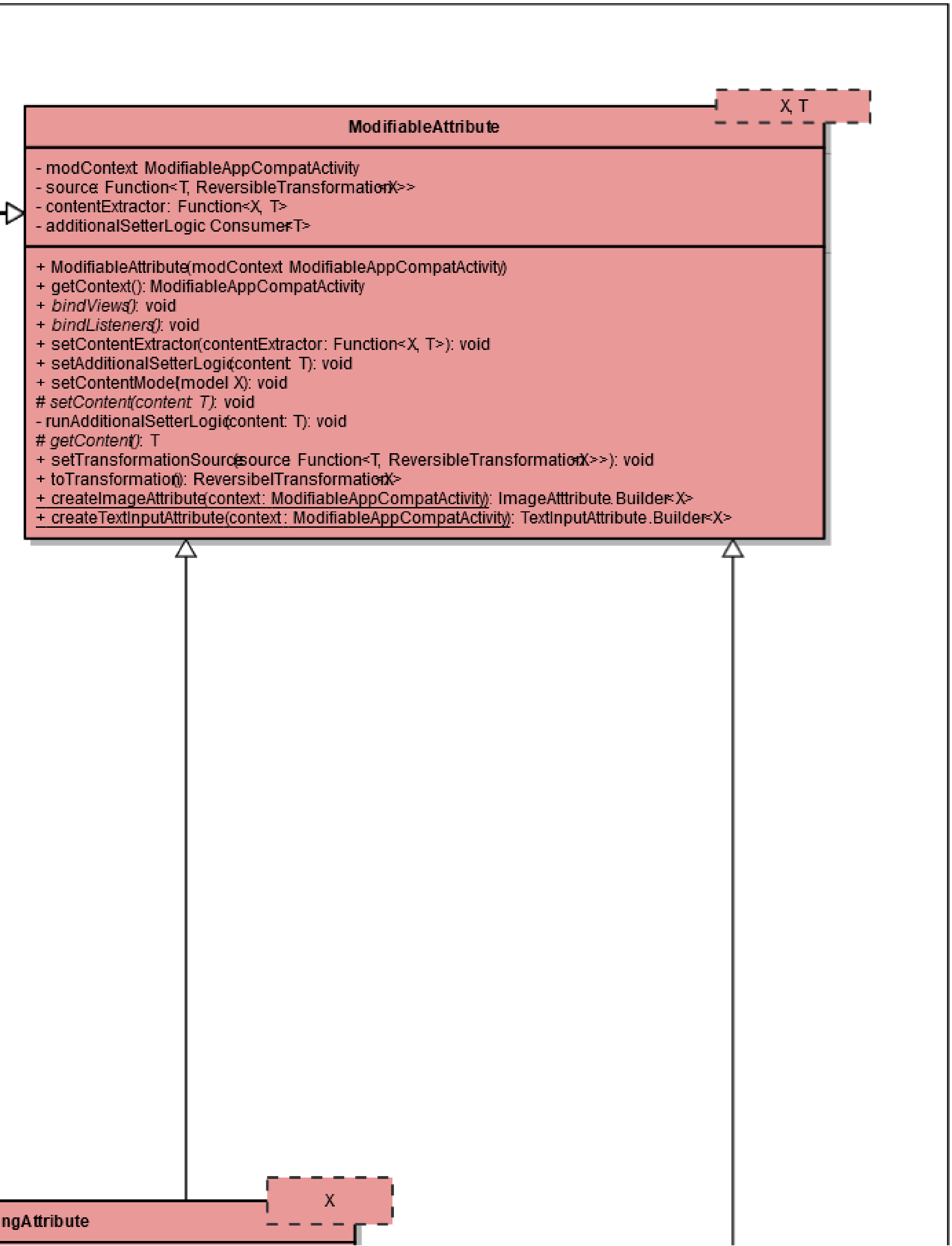
X
: int)
StringListAttribute
X
- editTextId: int
- root: ScrollView
- editList EditText
+ StringListAttribute(context: ModifiableAppCompatActivity, root: ScrollView, editText: int)
^+ bindViews(): void
^+ bindListeners(): void
^# setContent(context: List<String>): void
^+ getContent(): List<String>
PortrayableDetailEditActivity
STORAGE: RuntimeStorageAccess (readOnly)

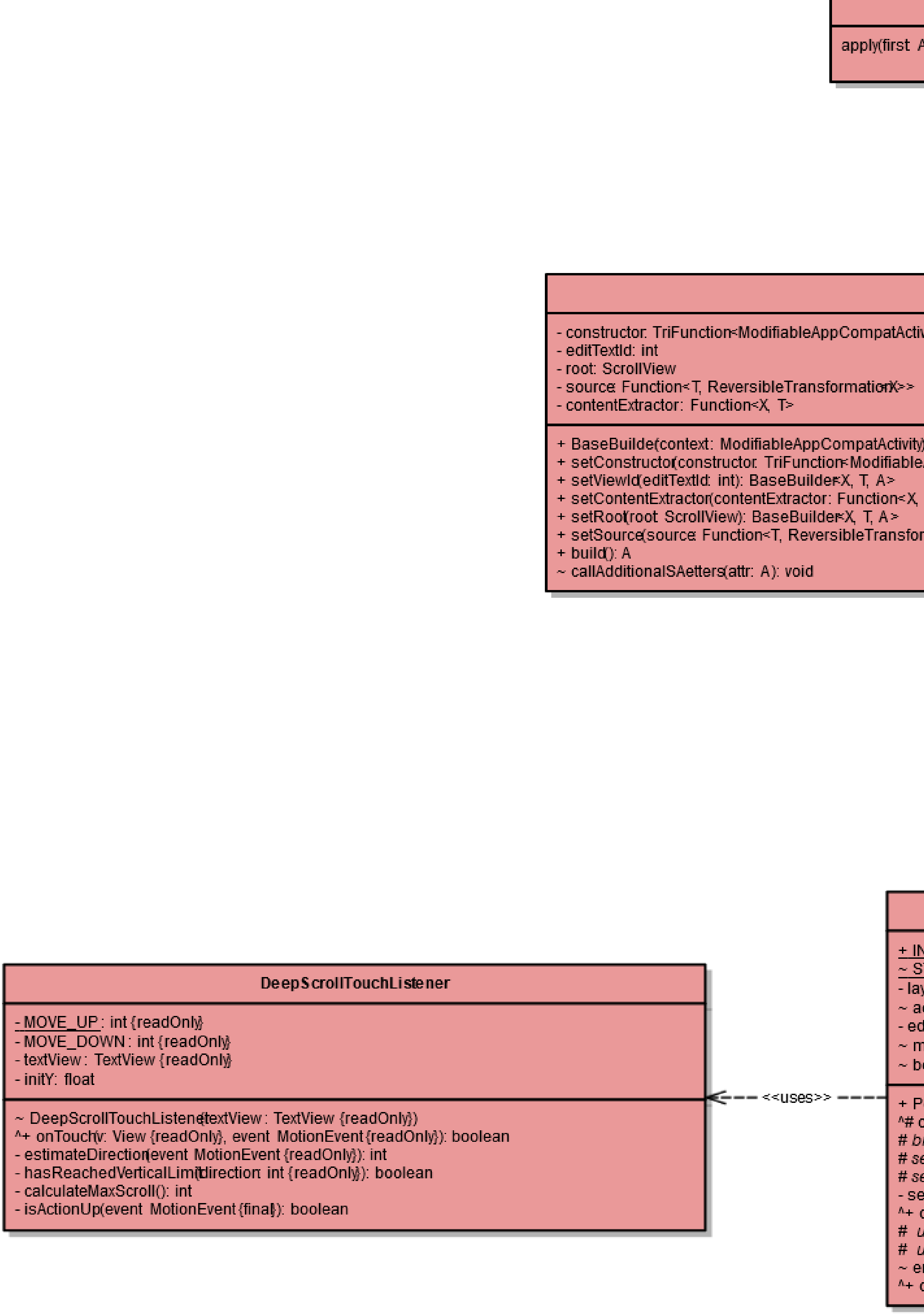


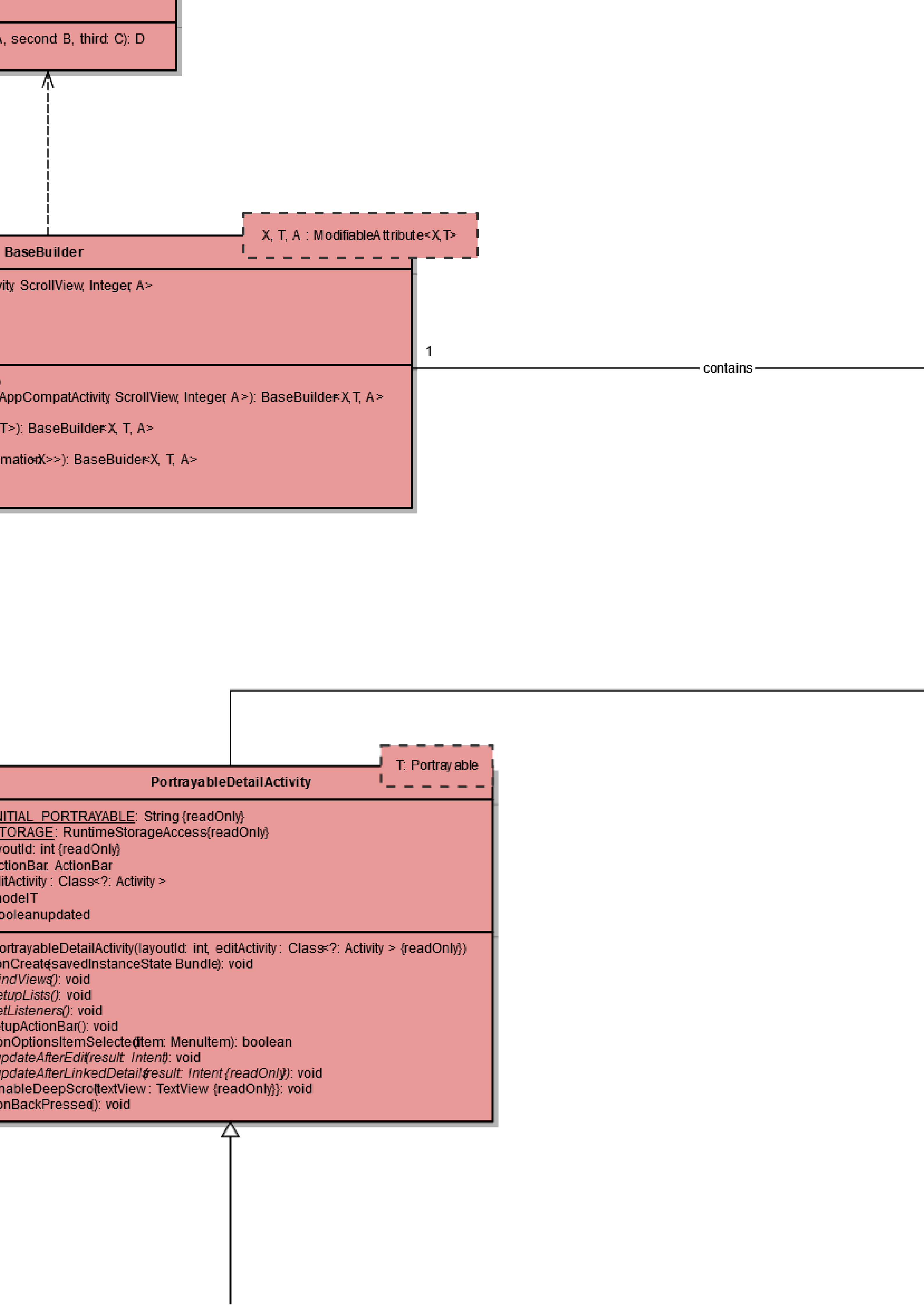
TextInputAttribute

- id: int
- scrollView: ScrollView
- editText: EditText
- textChangeListener: TextWatcher
- inputValidation: BiConsumer<EditText, String>

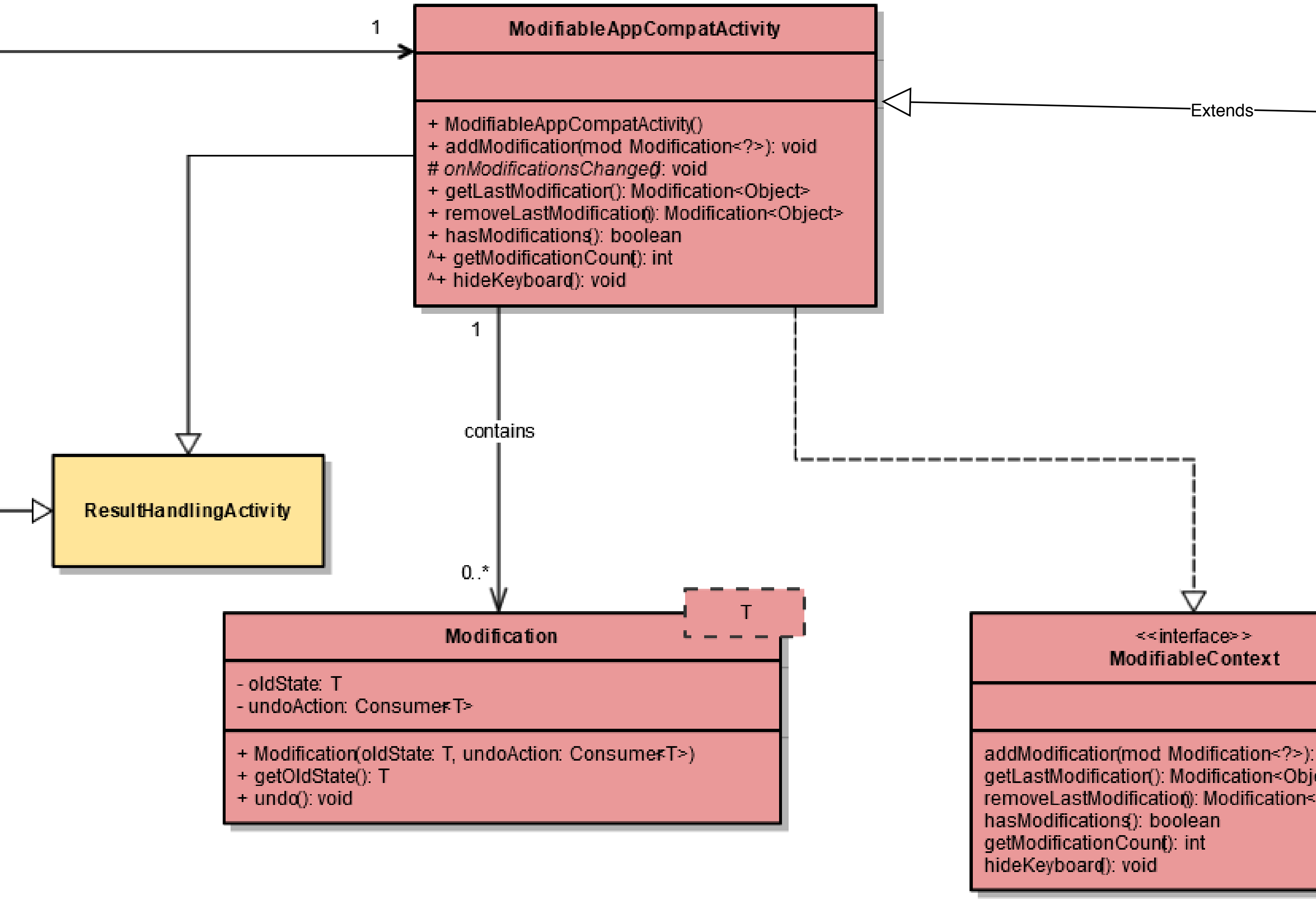
```
+ TextInputAttribute(modContext: ModifiableAppCompatActivity, scrollView: ScrollView, id: int)
^+ bindViews(): void
~ setTextChangeListener(textChangeListener: TextWatcher): void
~ setInputValidation(inputValidation: BiConsumer<EditText, String>): void
^+ bindListeners(): void
^# setContent(content: String): void
+ hasEmptyText(): boolean
^+ getContext(): String
```

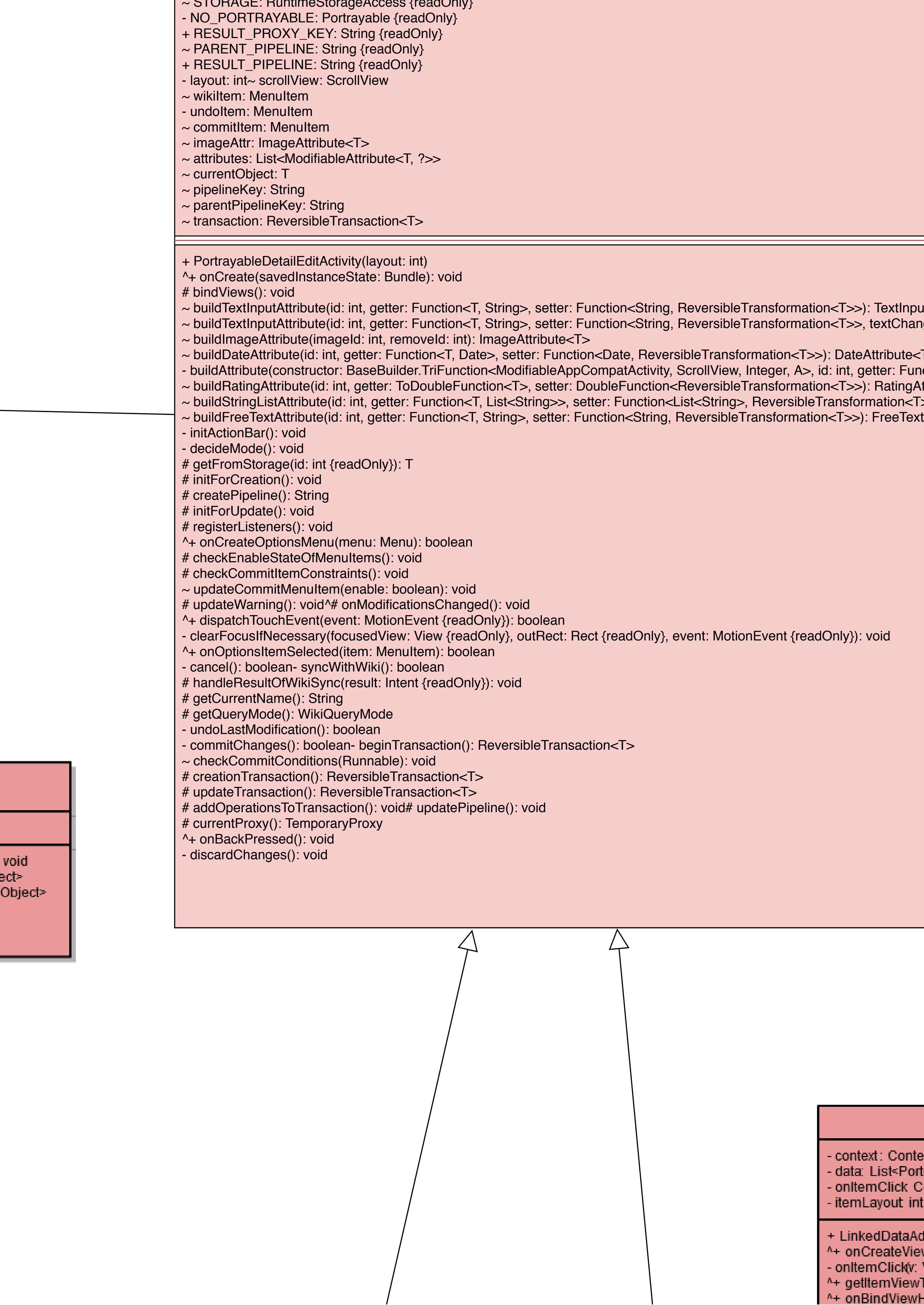

X, T
ModifiableAttribute
- modContext: ModifiableAppCompatActivity
- source: Function<T, ReversibleTransformation<X>>
- contentExtractor: Function<X, T>
- additionalSetterLogic Consumer<T>
+ ModifiableAttribute(modContext: ModifiableAppCompatActivity)
+ getContext(): ModifiableAppCompatActivity
+ *bindViews()*: void
+ *bindListeners()*: void
+ setContentExtractor(contentExtractor: Function<X, T>): void
+ setAdditionalSetterLogic(content: T): void
+ setContentModel(model: X): void
# *setContent(content: T)*: void
- runAdditionalSetterLogic(content: T): void
# *getContent()*: T
+ setTransformationSource(source: Function<T, ReversibleTransformation<X>>): void
+ toTransformation(): ReversibelTransformation<X>
+ createImageAttribute(context: ModifiableAppCompatActivity): ImageAtttribute.Builder<X>
+ createTextInputAttribute(context: ModifiableAppCompatActivity): TextInputAttribute.Builder<X>
X
ngAttribute


apply(first A
- constructor: TriFunction<ModifiableAppCompatActiv
- editTextId: int
- root: ScrollView
- source: Function<T, ReversibleTransformation<X>>
- contentExtractor: Function<X, T>
+ BaseBuilde(context: ModifiableAppCompatActivity)
+ setConstructor(constructor: TriFunction<Modifiable
+ setViewId(editTextId: int): BaseBuilder<X, T, A>
+ setContentExtractor(contentExtractor: Function<X,
+ setRoot(root: ScrollView): BaseBuilder<X, T, A>
+ setSource(source: Function<T, ReversibleTransfor
+ build(): A
~ callAdditionalSAetters(attr: A): void
DeepScrollTouchListener
- MOVE_UP : int {readOnly}
- MOVE_DOWN : int {readOnly}
- textView : TextView {readOnly}
- initY: float
~ DeepScrollTouchListener(textView : TextView {readOnly})
^+ onTouch(v: View {readOnly}, event: MotionEvent {readOnly}): boolean
- estimateDirection(event: MotionEvent {readOnly}): int
- hasReachedVerticalLimit(direction: int {readOnly}): boolean
- calculateMaxScroll(): int
- isActionUp(event: MotionEvent {final}): boolean
<<uses>>
+ IN
~ S
- lay
~ ac
- ed
~ m
~ bo
+ P
^# c
# br
# se
# se
- se
^+ c
# u
# u
~ er
^+ c



, second B, third C): D
X, T, A : ModifiableAttribute<X,T>
BaseBuilder
vity ScrollView, Integer, A>
AppCompatActivity ScrollView, Integer, A>): BaseBuilder<X,T, A>
T>): BaseBuilder<X, T, A>
mation<X>>): BaseBuider<X, T, A>
1
contains
T: Portray able
PortrayableDetailActivity
NITIAL_PORTRAYABLE: String {readOnly}
TORAGE: RuntimeStorageAccess{readOnly}
youtId: int {readOnly}
ctionBar: ActionBar
itActivity : Class<?: Activity >
nodelT
ooleanupdated
ortrayableDetailActivity(layoutId: int, editActivity : Class<?: Activity > {readOnly})
onCreate(savedInstanceState Bundle): void
indViews(): void
etupLists(): void
etListeners(): void
tupActionBar(): void
onOptionsItemSelected(item: MenuItem): boolean
pdateAfterEdit(result: Intent): void
pdateAfterLinkedDetails(result: Intent {readOnly}): void
nableDeepScroll(textView : TextView {readOnly}): void
onBackPressed(): void


1
ModifiableAppCompatActivity
+ ModifiableAppCompatActivity()
+ addModification(mod Modification<?>): void
# onModificationsChanged(): void
+ getLastModification(): Modification<Object>
+ removeLastModification(): Modification<Object>
+ hasModifications(): boolean
^+ getModificationCount(): int
^+ hideKeyboard(): void
Extends
1
contains
0..*
ResultHandlingActivity
T
Modification
- oldState: T
- undoAction: Consumer<T>
+ Modification(oldState: T, undoAction: Consumer<T>)
+ getOldState(): T
+ undo(): void
<<interface>>
ModifiableContext
addModification(mod Modification<?>):
getLastModification(): Modification<Obj
removeLastModification(): Modification<
hasModifications(): boolean
getModificationCount(): int
hideKeyboard(): void



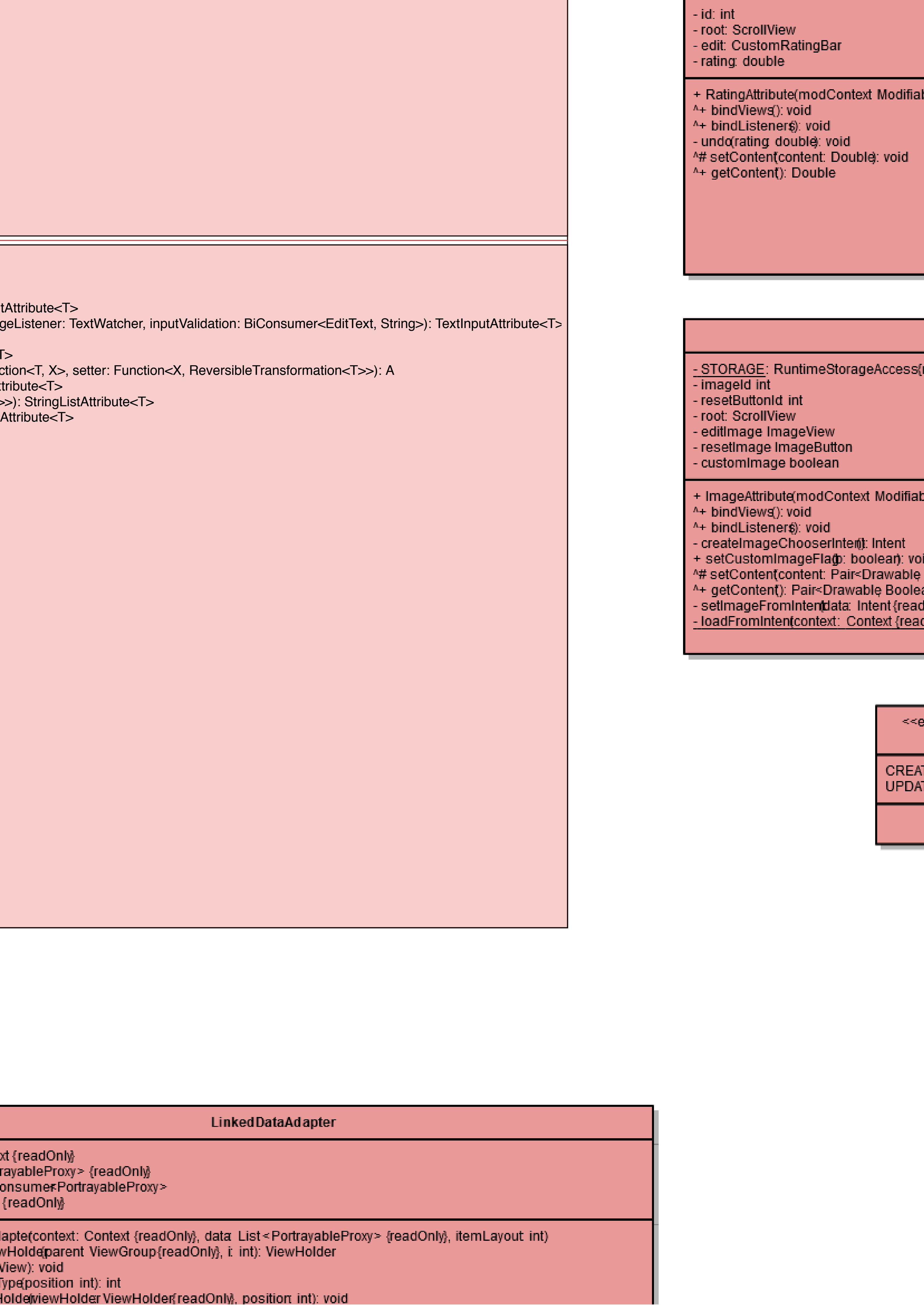
~ STORAGE: RuntimeStorageAccess {readOnly}
- NO_PORTRAYABLE: Portrayable {readOnly}
+ RESULT_PROXY_KEY: String {readOnly}
~ PARENT_PIPELINE: String {readOnly}
+ RESULT_PIPELINE: String {readOnly}
- layout: int~ scrollView: ScrollView
~ wikiItem: MenuItem
- undoItem: MenuItem
~ commitItem: MenuItem
~ imageAttr: ImageAttribute<T>
~ attributes: List<ModifiableAttribute<T, ?>>
~ currentObject: T
~ pipelineKey: String
~ parentPipelineKey: String
~ transaction: ReversibleTransaction<T>

+ PortrayableDetailEditActivity(layout: int)
^+ onCreate(savedInstanceState: Bundle): void
# bindViews(): void
~ buildTextInputAttribute(id: int, getter: Function<T, String>, setter: Function<String, ReversibleTransformation<T>>): TextInpu
~ buildTextInputAttribute(id: int, getter: Function<T, String>, setter: Function<String, ReversibleTransformation<T>>, textChan
~ buildImageAttribute(imageId: int, removeId: int): ImageAttribute<T>
~ buildDateAttribute(id: int, getter: Function<T, Date>, setter: Function<Date, ReversibleTransformation<T>>): DateAttribute<T
- buildAttribute(constructor: BaseBuilder.TriFunction<ModifiableAppCompatActivity, ScrollView, Integer, A>, id: int, getter: Func
~ buildRatingAttribute(id: int, getter: ToDoubleFunction<T>, setter: DoubleFunction<ReversibleTransformation<T>>): RatingAt
~ buildStringListAttribute(id: int, getter: Function<T, List<String>>, setter: Function<List<String>, ReversibleTransformation<T>
~ buildFreeTextAttribute(id: int, getter: Function<T, String>, setter: Function<String, ReversibleTransformation<T>>): FreeText
- initActionBar(): void
- decideMode(): void
# getFromStorage(id: int {readOnly}): T
# initForCreation(): void
# createPipeline(): String
# initForUpdate(): void
# registerListeners(): void
^+ onCreateOptionsMenu(menu: Menu): boolean
# checkEnableStateOfMenuItems(): void
# checkCommitItemConstraints(): void
~ updateCommitMenuItem(enable: boolean): void
# updateWarning(): void^# onModificationsChanged(): void
^+ dispatchTouchEvent(event: MotionEvent {readOnly}): boolean
- clearFocusIfNecessary(focusedView: View {readOnly}, outRect: Rect {readOnly}, event: MotionEvent {readOnly}): void
^+ onOptionsItemSelected(item: MenuItem): boolean
- cancel(): boolean- syncWithWiki(): boolean
# handleResultOfWikiSync(result: Intent {readOnly}): void
# getCurrentName(): String
# getQueryMode(): WikiQueryMode
- undoLastModification(): boolean
- commitChanges(): boolean- beginTransaction(): ReversibleTransaction<T>
~ checkCommitConditions(Runnable): void
# creationTransaction(): ReversibleTransaction<T>
# updateTransaction(): ReversibleTransaction<T>
# addOperationsToTransaction(): void# updatePipeline(): void
# currentProxy(): TemporaryProxy
^+ onBackPressed(): void
- discardChanges(): void

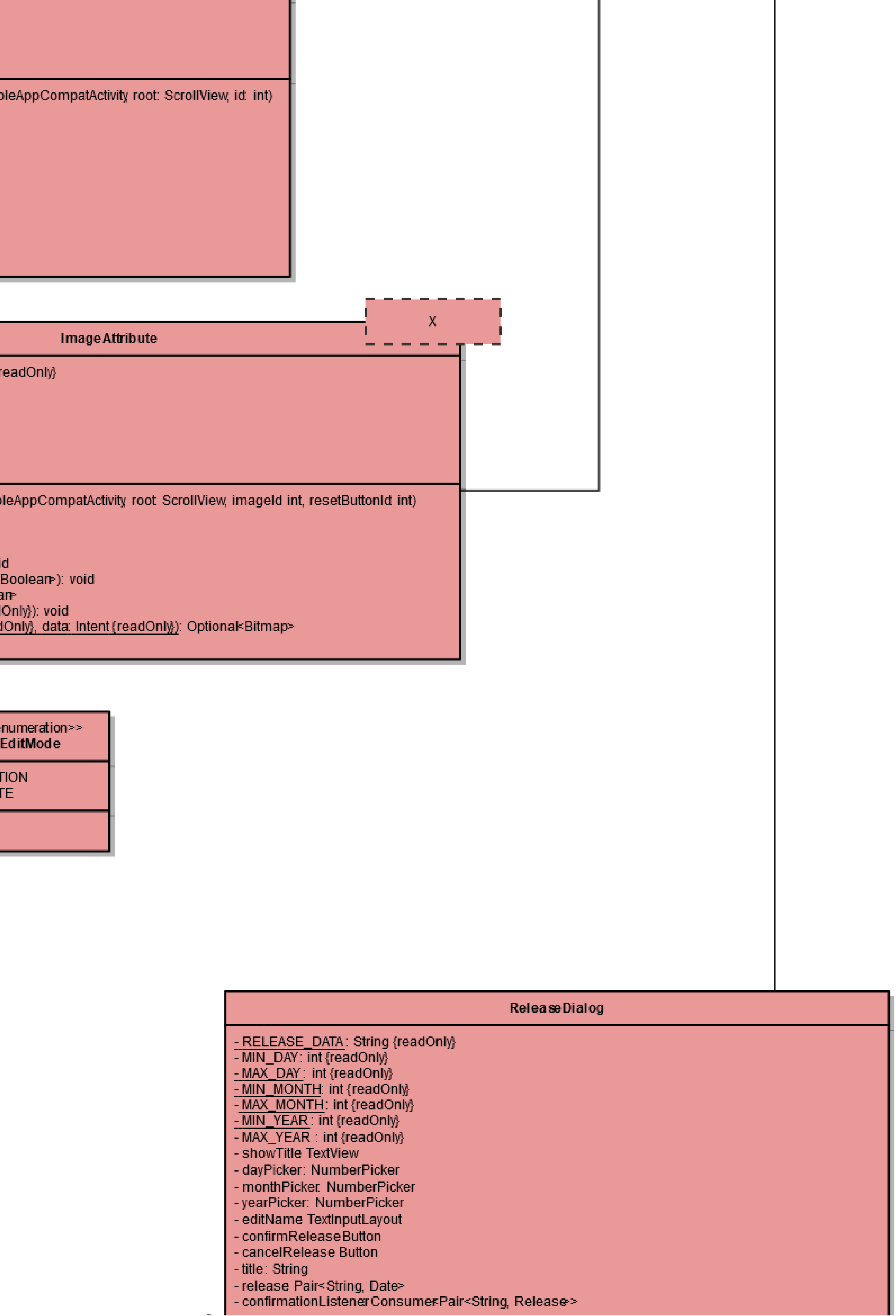
void
ect>
Object>

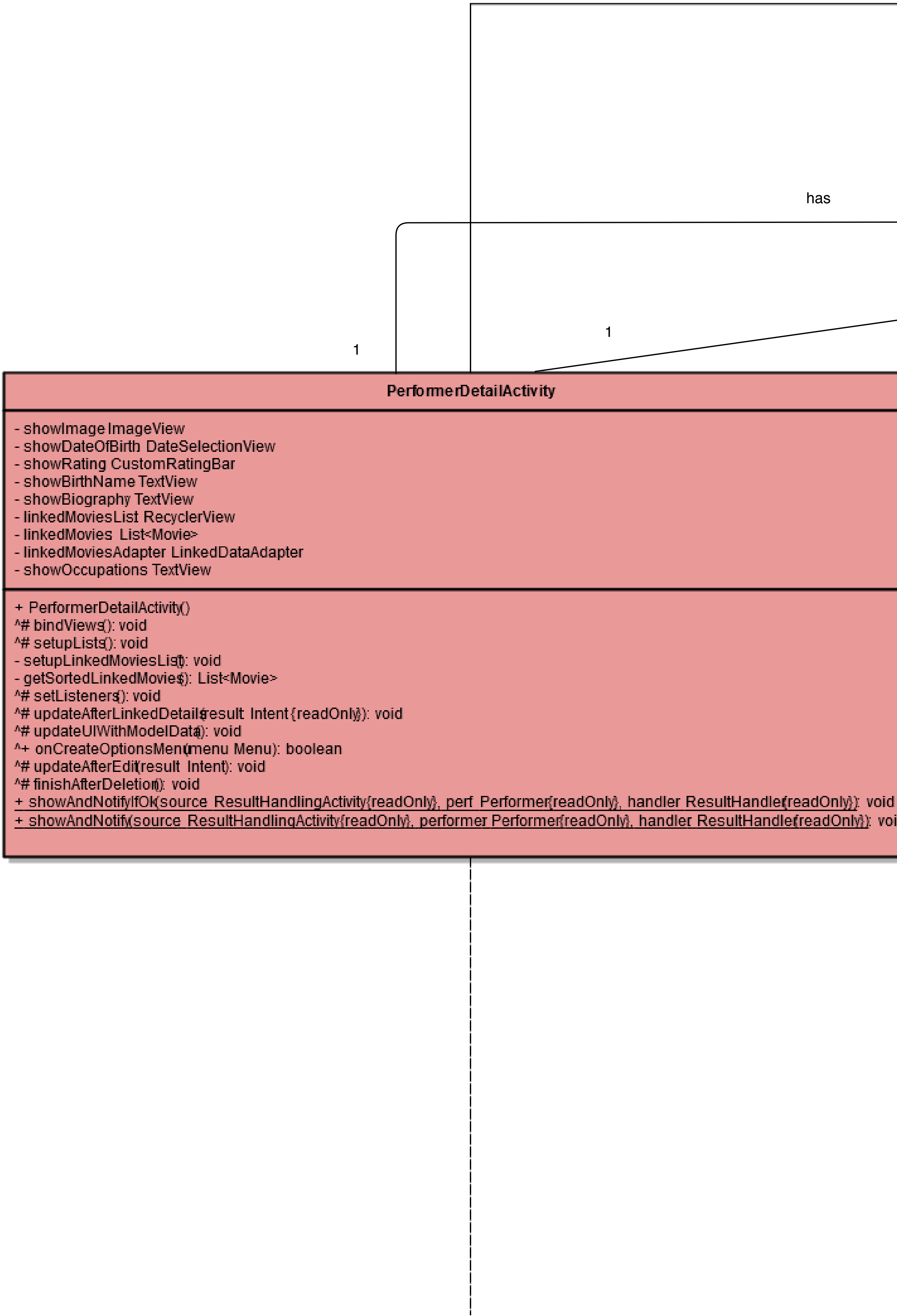
- context: Conte
- data: List<Port
- onItemClick C
- itemLayout int

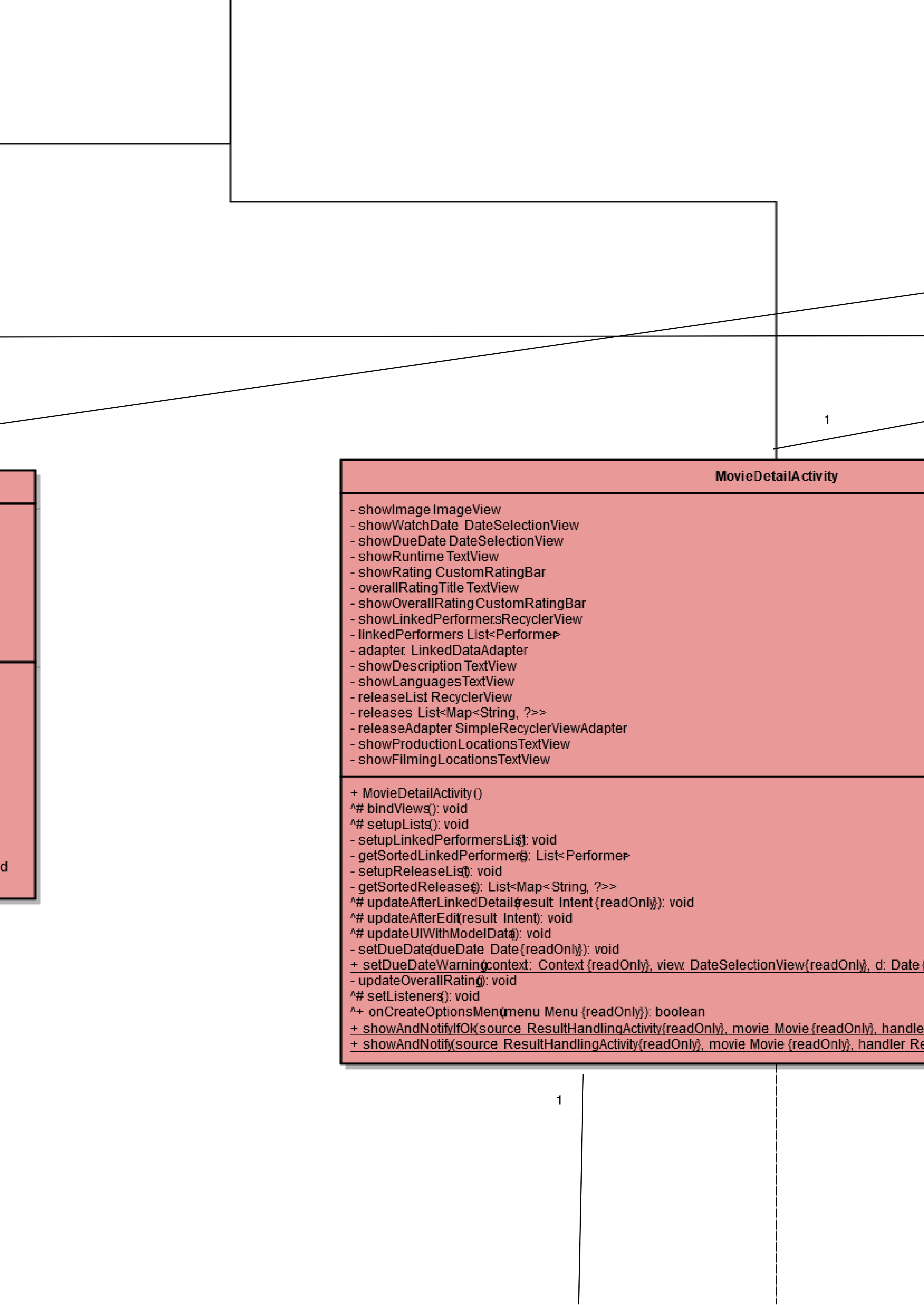
+ LinkedDataAd
^+ onCreateView
- onItemClick(v:
^+ getItemViewT
^+ onBindViewH



- id: int
- root: ScrollView
- edit: CustomRatingBar
- rating: double
+ RatingAttribute(modContext: Modifiab
^+ bindViews(): void
^+ bindListeners(): void
- undo(rating: double): void
^# setContent(content: Double): void
^+ getContent(): Double
tAttribute<T>
geListener: TextWatcher, inputValidation: BiConsumer<EditText, String>): TextInputAttribute<T>
T>
ction<T, X>, setter: Function<X, ReversibleTransformation<T>>): A
tribute<T>
>>): StringListAttribute<T>
Attribute<T>
- STORAGE: RuntimeStorageAccess{r
- imageId int
- resetButtonId int
- root: ScrollView
- editImage ImageView
- resetImage ImageButton
- customImage boolean
+ ImageAttribute(modContext: Modifiab
^+ bindViews(): void
^+ bindListeners(): void
- createImageChooserIntent(): Intent
+ setCustomImageFlag(b: boolean): voi
^# setContent(content: Pair<Drawable
^+ getContent(): Pair<Drawable, Boolea
- setImageFromIntent(data: Intent {read
- loadFromIntent(context: Context {reac
<<e
CREAT
UPDAT
LinkedDataAdapter
xt {readOnly}
rayableProxy> {readOnly}
onsumer<PortrayableProxy>
{readOnly}
lapter(context: Context {readOnly}, data: List<PortrayableProxy> {readOnly}, itemLayout: int)
wHolder(parent: ViewGroup {readOnly}, i: int): ViewHolder
View): void
Type(position: int): int
Holder(viewHolder: ViewHolder {readOnly}, position: int): void


bleAppCompatActivity, root: ScrollView, id: int)
X
ImageAttribute
readOnly}
bleAppCompatActivity, root: ScrollView, imageId: int, resetButtonId: int)
id
Boolean>): void
an>
Only}): void
dOnly}, data: Intent {readOnly}): Optional<Bitmap>
numeration>>
EditMode
TION
TE
ReleaseDialog
- RELEASE_DATA: String {readOnly}
- MIN_DAY: int {readOnly}
- MAX_DAY: int {readOnly}
- MIN_MONTH: int {readOnly}
- MAX_MONTH: int {readOnly}
- MIN_YEAR: int {readOnly}
- MAX_YEAR : int {readOnly}
- showTitle: TextView
- dayPicker: NumberPicker
- monthPicker: NumberPicker
- yearPicker: NumberPicker
- editName: TextInputLayout
- confirmRelease: Button
- cancelRelease: Button
- title: String
- release: Pair<String, Date>
- confirmationListener: Consumer<Pair<String, Release>>

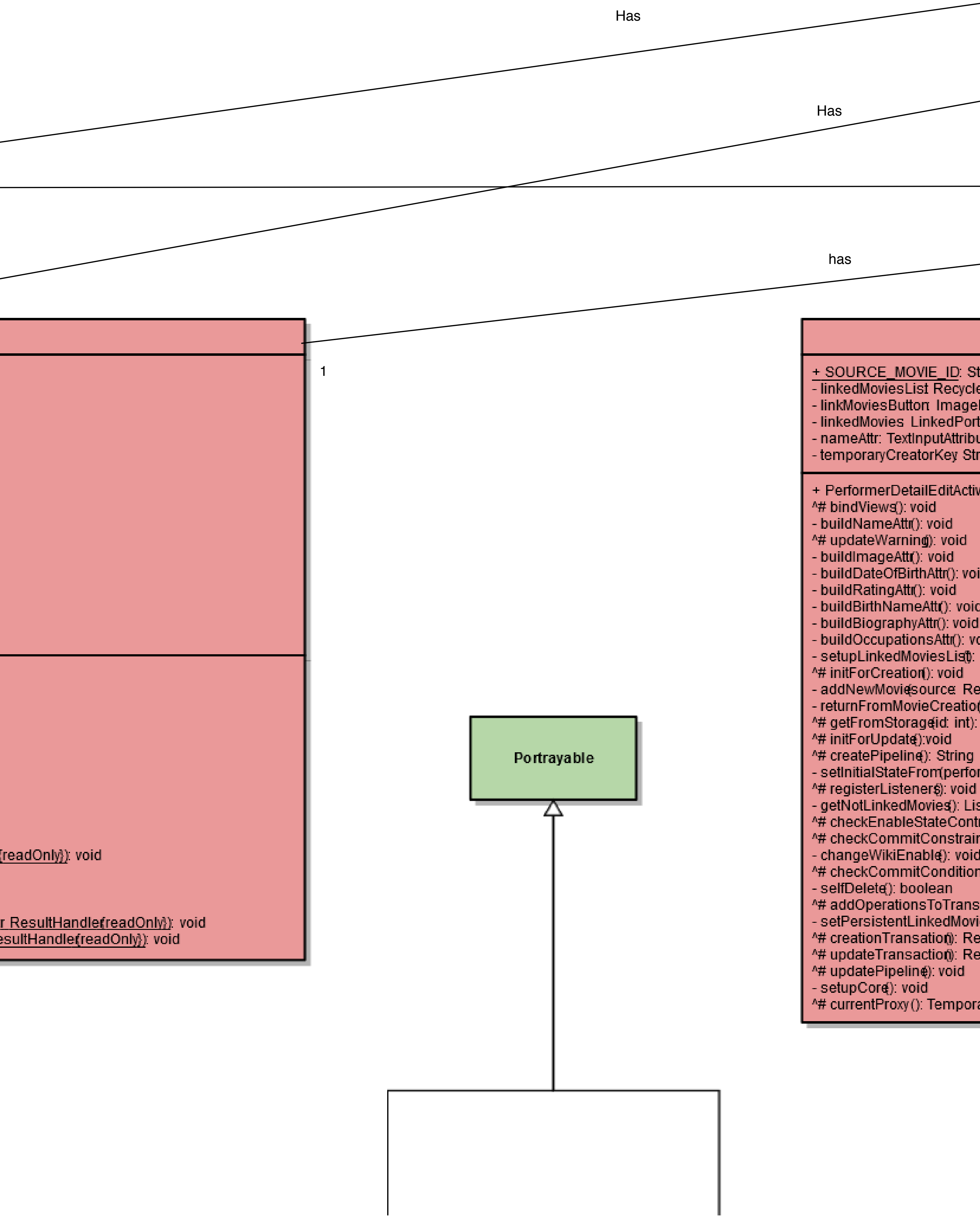

has
1
1
PerformerDetailActivity
- showImage ImageView
- showDateOfBirth DateSelectionView
- showRating CustomRatingBar
- showBirthName TextView
- showBiography TextView
- linkedMoviesList RecyclerView
- linkedMovies List<Movie>
- linkedMoviesAdapter LinkedDataAdapter
- showOccupations TextView
+ PerformerDetailActivity()
^# bindViews(): void
^# setupLists(): void
- setupLinkedMoviesList(): void
- getSortedLinkedMovies(): List<Movie>
^# setListeners(): void
^# updateAfterLinkedDetails(result Intent {readOnly}): void
^# updateUIWithModelData(): void
^+ onCreateOptionsMenu(menu Menu): boolean
^# updateAfterEdit(result Intent): void
^# finishAfterDeletion(): void
+ showAndNotifyIfOk(source ResultHandlingActivity{readOnly}, perf Performer{readOnly}, handler ResultHandler{readOnly}): void
+ showAndNotify(source ResultHandlingActivity{readOnly}, performer Performer{readOnly}, handler ResultHandler{readOnly}): voi



MovieDetailActivity

- showImage ImageView
- showWatchDate DateSelectionView
- showDueDate DateSelectionView
- showRuntime TextView
- showRating CustomRatingBar
- overallRatingTitle TextView
- showOverallRating CustomRatingBar
- showLinkedPerformersRecyclerView
- linkedPerformers List<Performer>
- adapter LinkedDataAdapter
- showDescription TextView
- showLanguagesTextView
- releaseList RecyclerView
- releases List<Map<String, ?>>
- releaseAdapter SimpleRecyclerViewAdapter
- showProductionLocationsTextView
- showFilmingLocationsTextView

```
+ MovieDetailActivity()
^# bindViews(): void
^# setupLists(): void
- setupLinkedPerformersList(): void
- getSortedLinkedPerformers(): List<Performer>
- setupReleaseList(): void
- getSortedReleases(): List<Map<String, ?>>
^# updateAfterLinkedDetails(result Intent {readOnly}): void
^# updateAfterEdit(result Intent): void
^# updateUIWithModelData(): void
- setDueDate(dueDate Date {readOnly}): void
+ setDueDateWarning(context: Context {readOnly}, view: DateSelectionView {readOnly}, d: Date
- updateOverallRating(): void
^# setListeners(): void
^+ onCreateOptionsMenu(menu Menu {readOnly}): boolean
+ showAndNotifyIfOk(source ResultHandlingActivity {readOnly}, movie Movie {readOnly}, handle
+ showAndNotify(source ResultHandlingActivity {readOnly}, movie Movie {readOnly}, handler Re
```

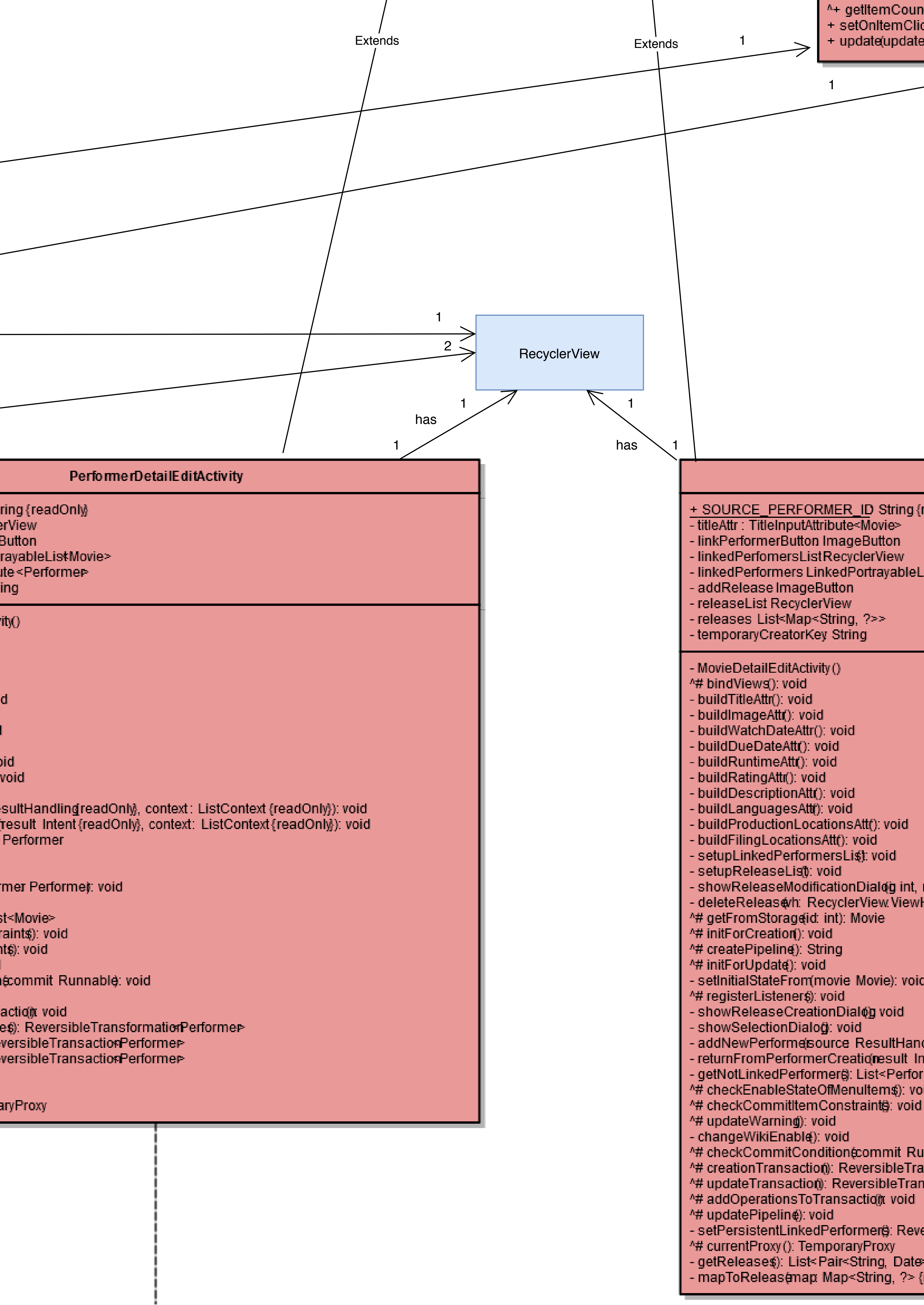


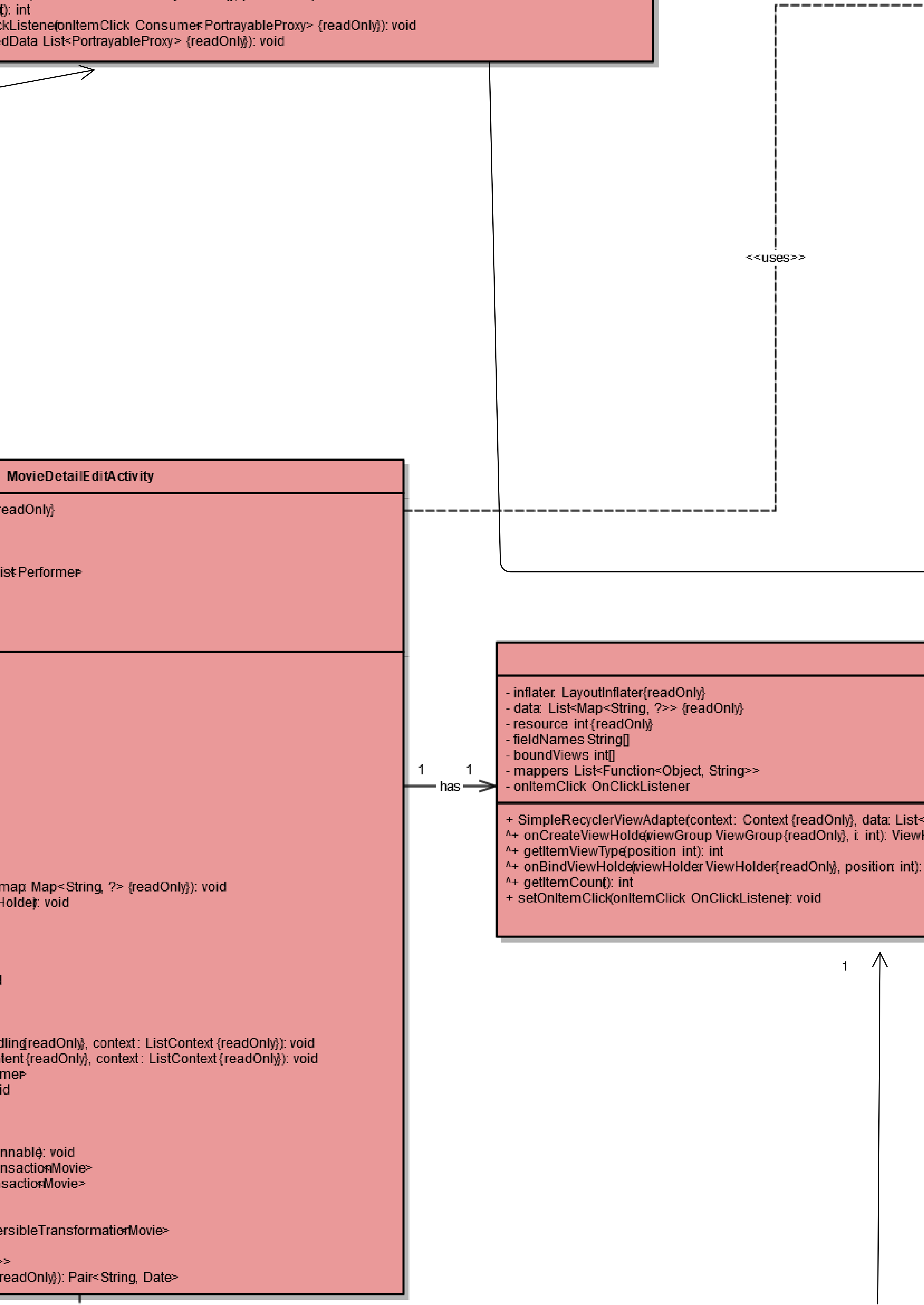
{readOnly}): void

r ResultHandler{readOnly}): void
esultHandler{readOnly}): void

+ SOURCE_MOVIE_ID: St
- linkedMoviesList: Recycle
- linkMoviesButton: ImageI
- linkedMovies: LinkedPort
- nameAttr: TextInputAttribu
- temporaryCreatorKey: Str

+ PerformerDetailEditActiv
^# bindViews(): void
- buildNameAttr(): void
^# updateWarning(): void
- buildImageAttr(): void
- buildDateOfBirthAttr(): voi
- buildRatingAttr(): void
- buildBirthNameAttr(): voic
- buildBiographyAttr(): void
- buildOccupationsAttr(): vo
- setupLinkedMoviesList():
^# initForCreation(): void
- addNewMovie(source: Re
- returnFromMovieCreatio(
^# getFromStorage(id: int):
^# initForUpdate():void
^# createPipeline(): String
- setInitialStateFrom(perfor
^# registerListeners(): void
- getNotLinkedMovies(): Lis
^# checkEnableStateContr
^# checkCommitConstrair
- changeWikiEnable(): void
^# checkCommitCondition
- selfDelete(): boolean
^# addOperationsToTrans
- setPersistentLinkedMovi
^# creationTransation(): Re
^# updateTransaction(): Re
^# updatePipeline(): void
- setupCore(): void
^# currentProxy(): Tempora


^+ getItemCoun
+ setOnItemClic
+ update(update
Extends
Extends
1
1
RecyclerView
1
2
has
1
1
1
has
1
PerformerDetailEditActivity
ring {readOnly}
erView
Button
rayableList<Movie>
ute <Performer>
ing
rity()
d
l
oid
void
esultHandling {readOnly}, context : ListContext {readOnly}): void
result Intent {readOnly}, context: ListContext {readOnly}): void
Performer
rmer Performer): void
st<Movie>
raints(): void
nts(): void
e(commit Runnable): void
action(): void
es(): ReversibleTransformation<Performer>
versibleTransaction<Performer>
versibleTransaction<Performer>
aryProxy
+ SOURCE_PERFORMER_ID String {r
- titleAttr : TitleInputAttribute<Movie>
- linkPerformerButton ImageButton
- linkedPerfomersList RecyclerView
- linkedPerformers LinkedPortrayableL
- addRelease ImageButton
- releaseList RecyclerView
- releases List<Map<String, ?>>
- temporaryCreatorKey String
- MovieDetailEditActivity()
^# bindViews(): void
- buildTitleAttr(): void
- buildImageAttr(): void
- buildWatchDateAttr(): void
- buildDueDateAttr(): void
- buildRuntimeAttr(): void
- buildRatingAttr(): void
- buildDescriptionAttr(): void
- buildLanguagesAtt(): void
- buildProductionLocationsAtt(): void
- buildFilingLocationsAtt(): void
- setupLinkedPerformersList(): void
- setupReleaseList(): void
- showReleaseModificationDialog(i int,
- deleteRelease(vh: RecyclerView.ViewH
^# getFromStorage(id: int): Movie
^# initForCreation(): void
^# createPipeline(): String
^# initForUpdate(): void
- setInitialStateFrom(movie Movie): voic
^# registerListeners(): void
- showReleaseCreationDialog(): void
- showSelectionDialog(): void
- addNewPerformer(source ResultHanc
- returnFromPerformerCreation(result In
- getNotLinkedPerformers(): List<Perfor
^# checkEnableStateOfMenuItems(): vo
^# checkCommitItemConstraints(): void
^# updateWarning(): void
- changeWikiEnable(): void
^# checkCommitConditions(commit Ru
^# creationTransaction(): ReversibleTra
^# updateTransaction(): ReversibleTran
^# addOperationsToTransaction(): void
^# updatePipeline(): void
- setPersistentLinkedPerformers(): Reve
^# currentProxy(): TemporaryProxy
- getReleases(): List<Pair<String, Date
- mapToRelease(map Map<String, ?> {



(): int
ckListener(onItemClick Consumer<PortrayableProxy> {readOnly}): void
dData List<PortrayableProxy> {readOnly}): void

<<uses>>

MovieDetailEditActivity

readOnly}

ist<Performer>

map: Map<String, ?> {readOnly}): void
Holder): void

dling{readOnly}, context: ListContext {readOnly}): void
tent{readOnly}, context: ListContext{readOnly}): void
mer>
id

nnable): void
nsaction<Movie>
saction<Movie>

ersibleTransformation<Movie>

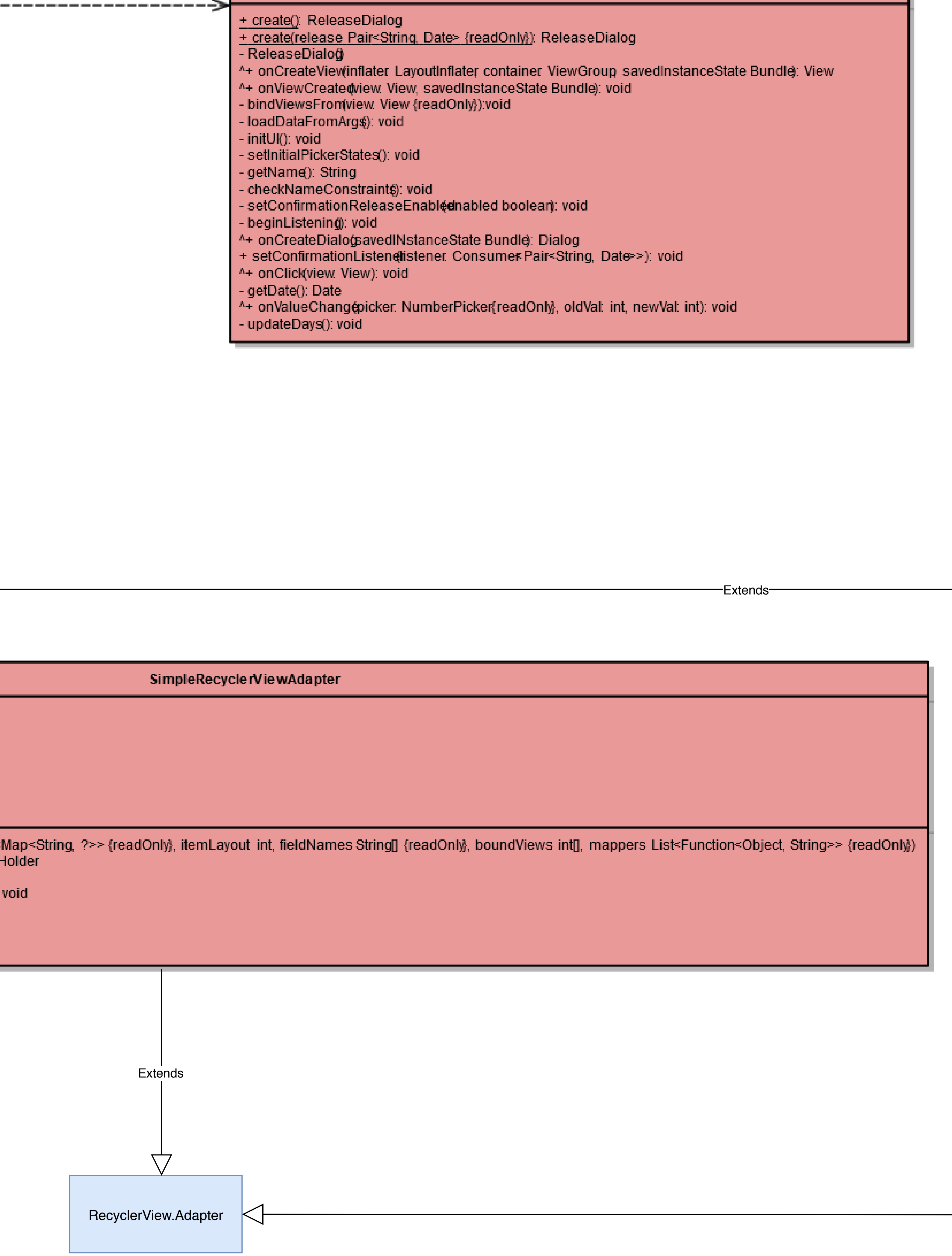
>>
readOnly}): Pair<String, Date>

1 has 1

- inflater: LayoutInflater{readOnly}
- data: List<Map<String, ?>> {readOnly}
- resource: int{readOnly}
- fieldNames: String[]
- boundViews: int[]
- mappers: List<Function<Object, String>>
- onItemClick: OnClickListener

+ SimpleRecyclerViewAdapter(context: Context {readOnly}, data: List<
^+ onCreateViewHolder(viewGroup: ViewGroup{readOnly}, i: int): ViewH
^+ getItemViewType(position: int): int
^+ onBindViewHolder(viewHolder: ViewHolder{readOnly}, position: int):
^+ getItemCount(): int
+ setOnItemClick(onItemClick: OnClickListener): void

1

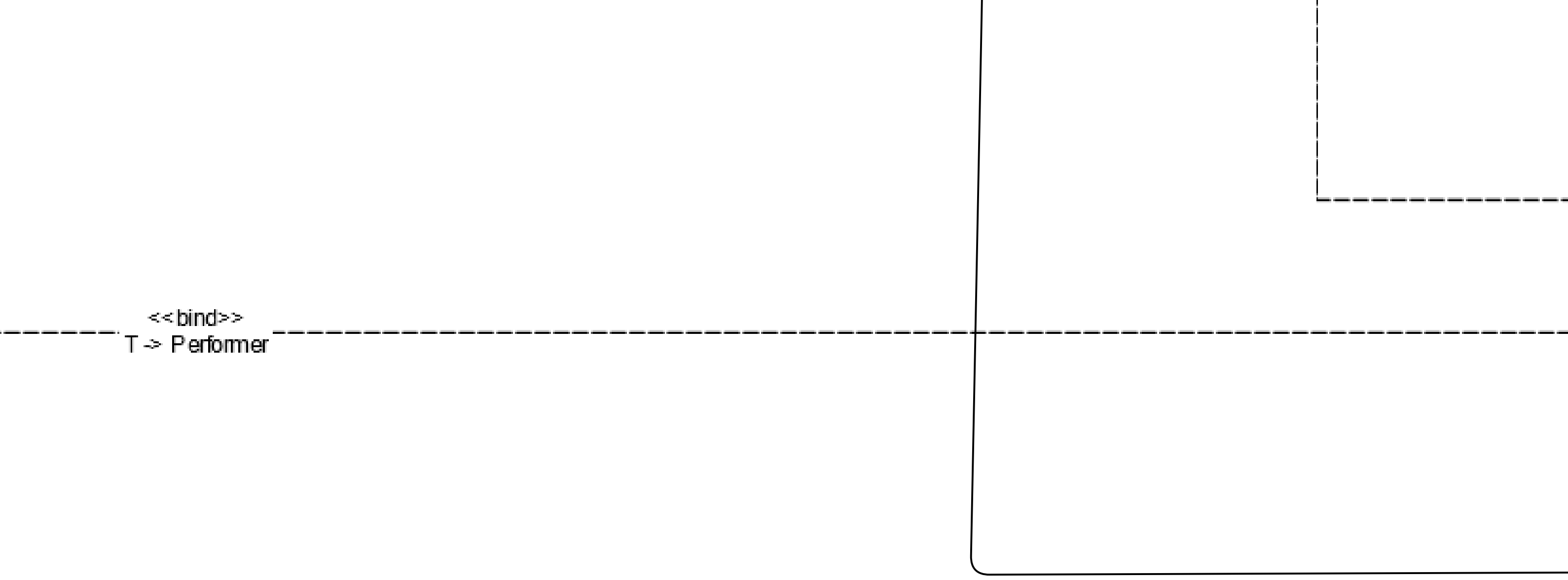


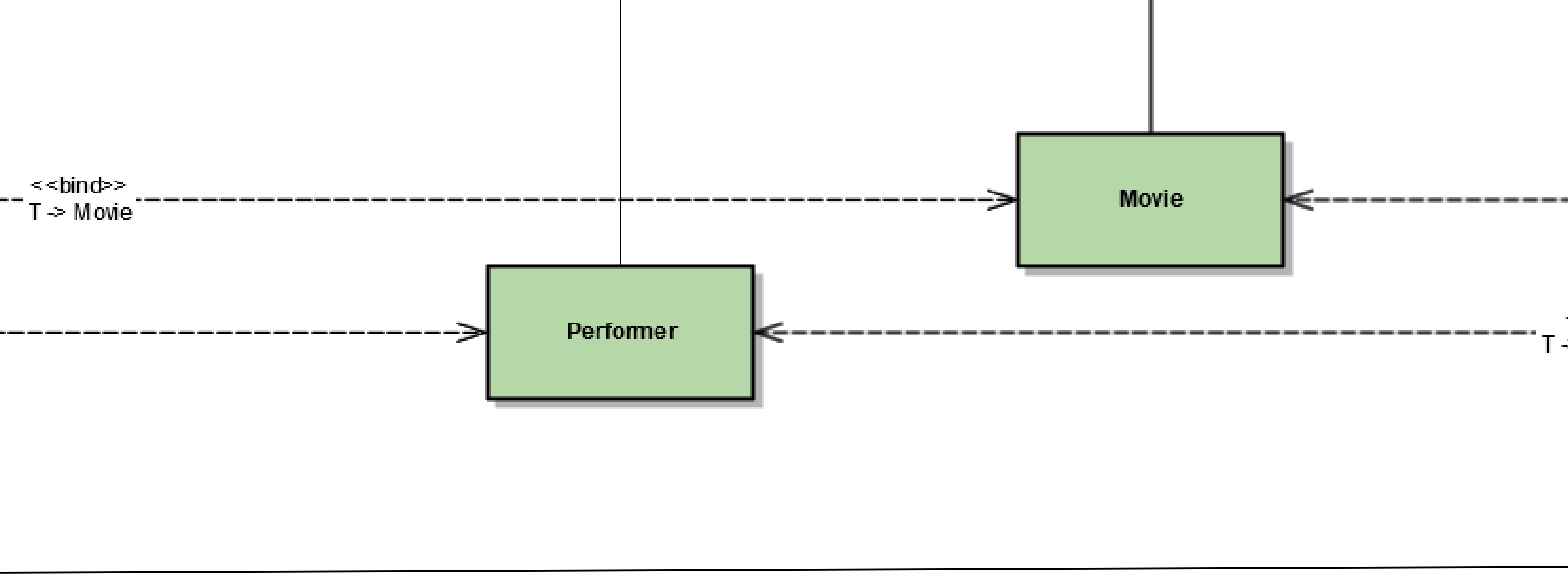
+ create(): ReleaseDialog
+ create(release: Pair<String, Date> {readOnly}): ReleaseDialog
- ReleaseDialog()
^+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View
^+ onViewCreated(view: View, savedInstanceState: Bundle): void
- bindViewsFrom(view: View {readOnly}):void
- loadDataFromArgs(): void
- initUI(): void
- setInitialPickerStates(): void
- getName(): String
- checkNameConstraints(): void
- setConfirmationReleaseEnabled(enabled: boolean): void
- beginListening(): void
^+ onCreateDialog(savedINstanceState: Bundle): Dialog
+ setConfirmationListener(listener: Consumer<Pair<String, Date>>): void
^+ onClick(view: View): void
- getDate(): Date
^+ onValueChange(picker: NumberPicker{readOnly}, oldVal: int, newVal: int): void
- updateDays(): void

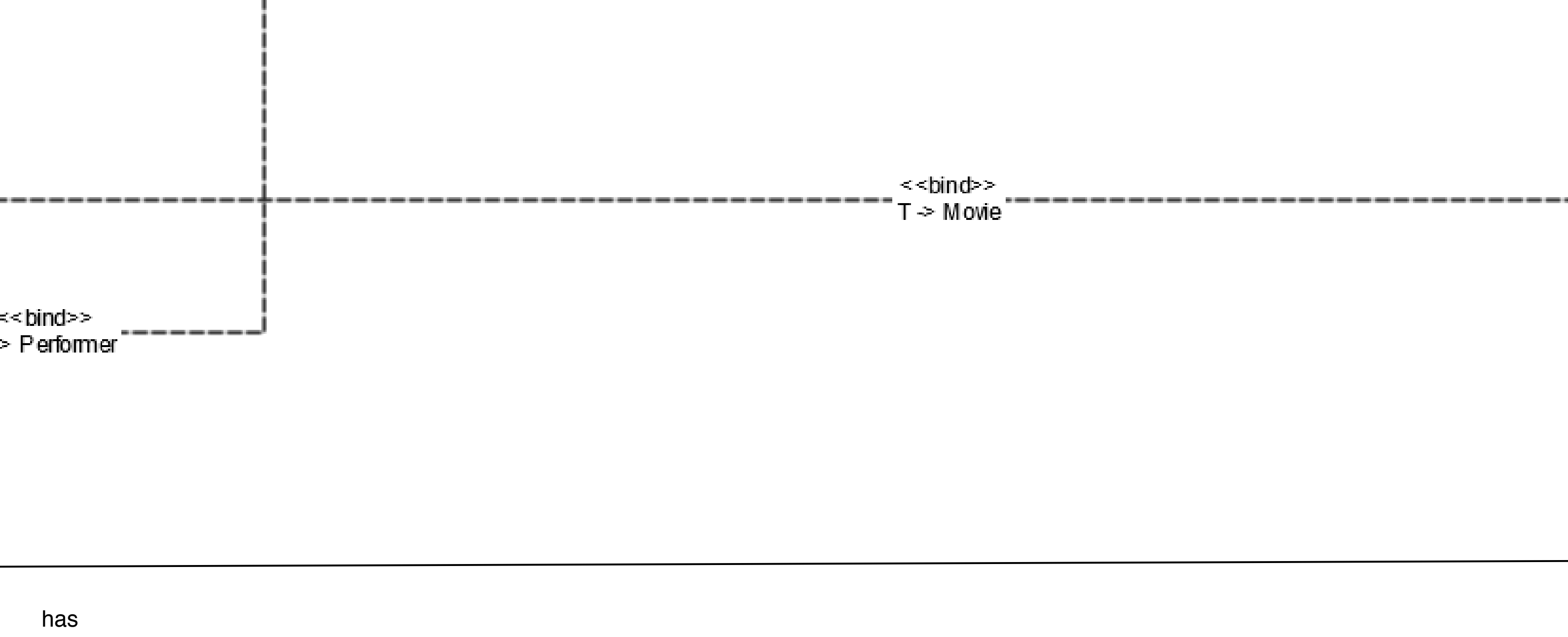
SimpleRecyclerViewAdapter

Map<String, ?>> {readOnly}, itemLayout: int, fieldNames: String[] {readOnly}, boundViews: int[], mappers: List<Function<Object, String>> {readOnly})
Holder

void


<<bind>>
T -> Performer


<<bind>>
T -> Movie
Movie
Performer
T -


<<bind>>
T -> Movie
<<bind>>
> Performer
has