Diagram 4.4-E. Constant pool relations

New items abbreviated Anchor, Linkage are fully spelled CONSTANT_SpecializationAnchor, CONSTANT_SpecializationLinkage

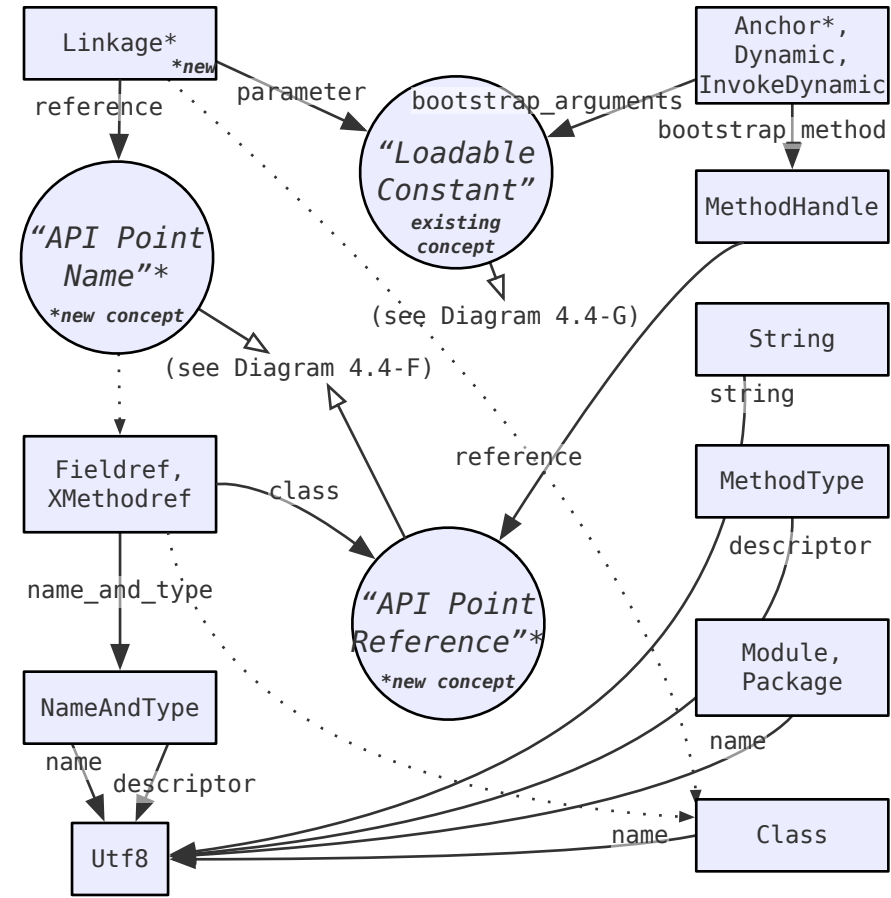


Diagram 4.4-F(a). API Point relations

Note: A Linkage constant is usable anywhere its reference item may be used.

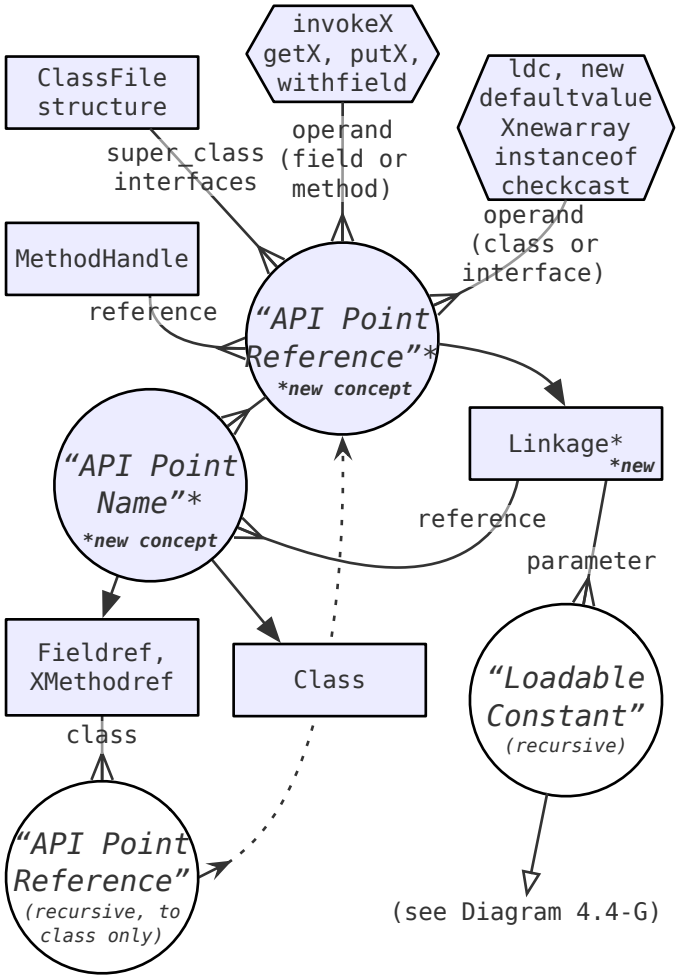


Diagram 4.4-F(b). “API Point References”, all configurations
(These are also “API Point Names”, except Linkage constants.)

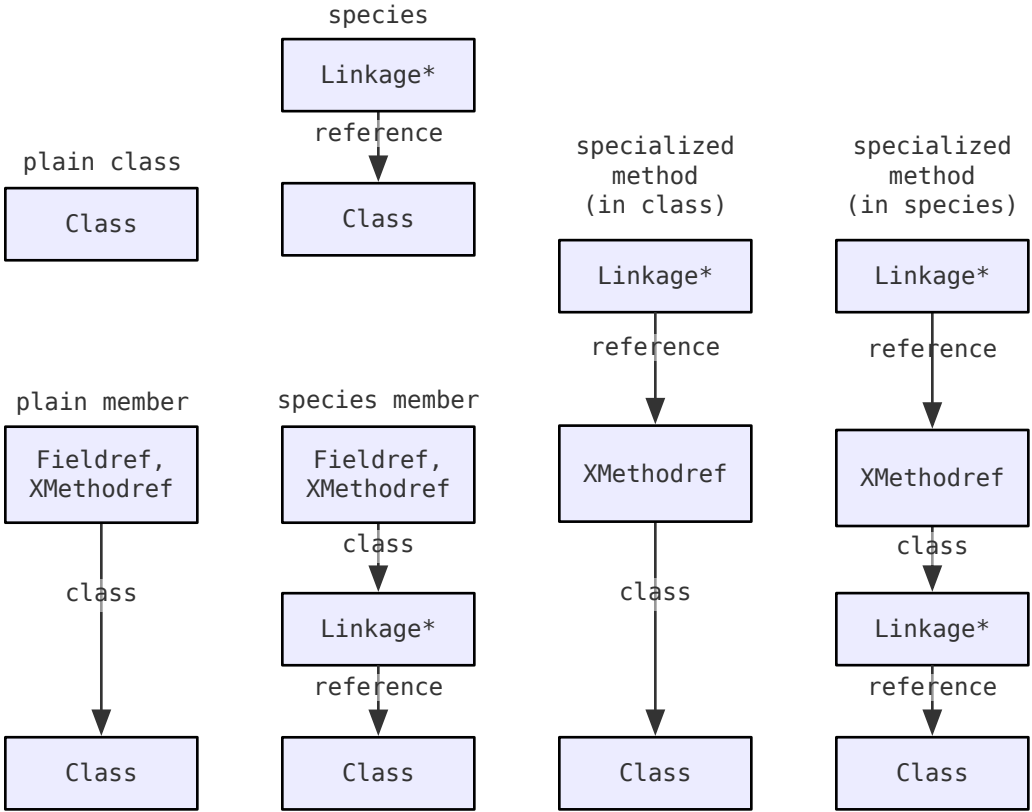
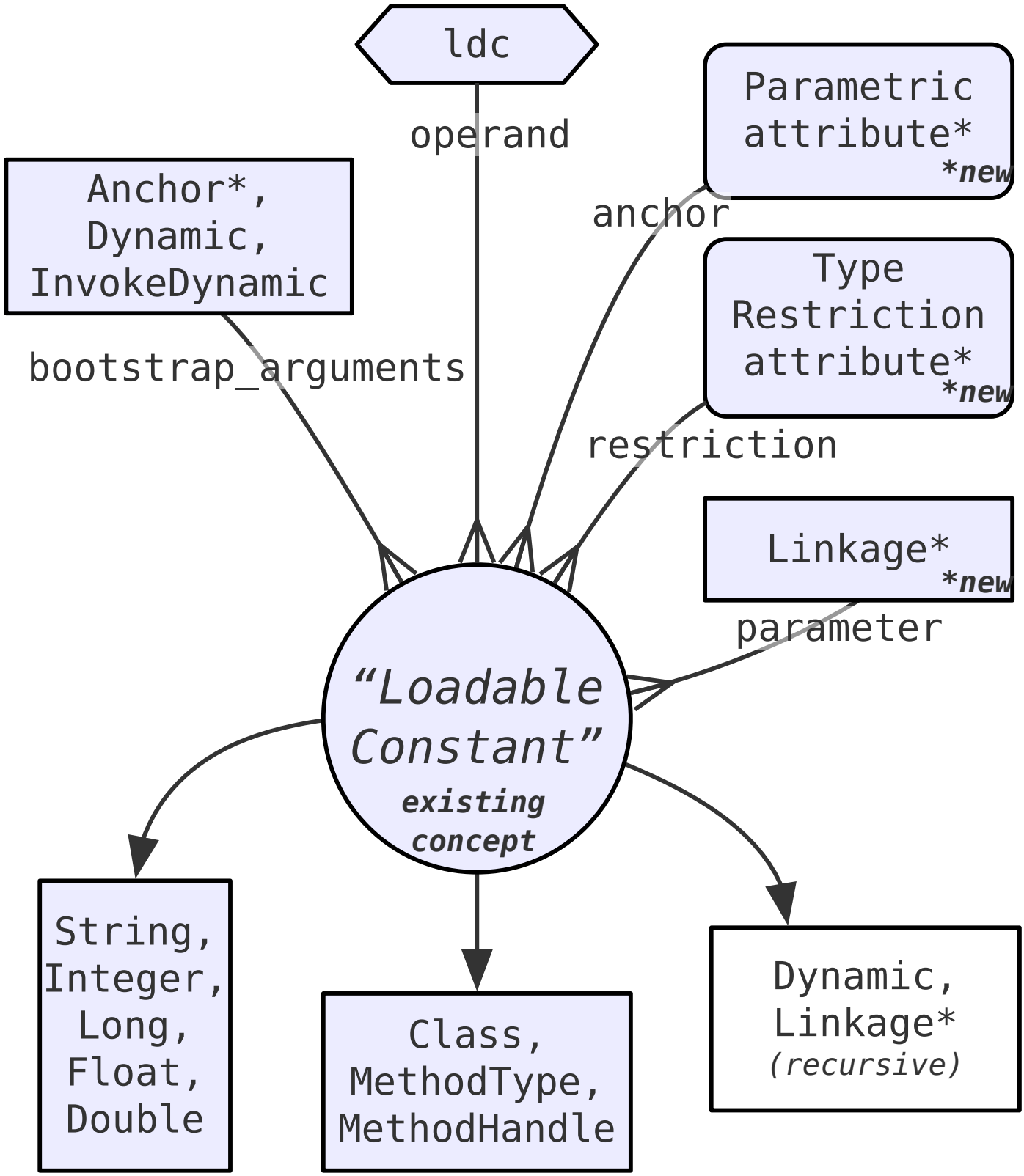


Diagram 4.4-G. “Loadable Constant” relations



Note: A Linkage constant is only loadable if its reference is loadable, i.e., a Class. Other Linkage constants (e.g., of a Fieldref) are not loadable.

Diagram 4.4-H(a). Example constant pool:
non-parametric client of List<Point>
(resolution states are at right; all are invariant)

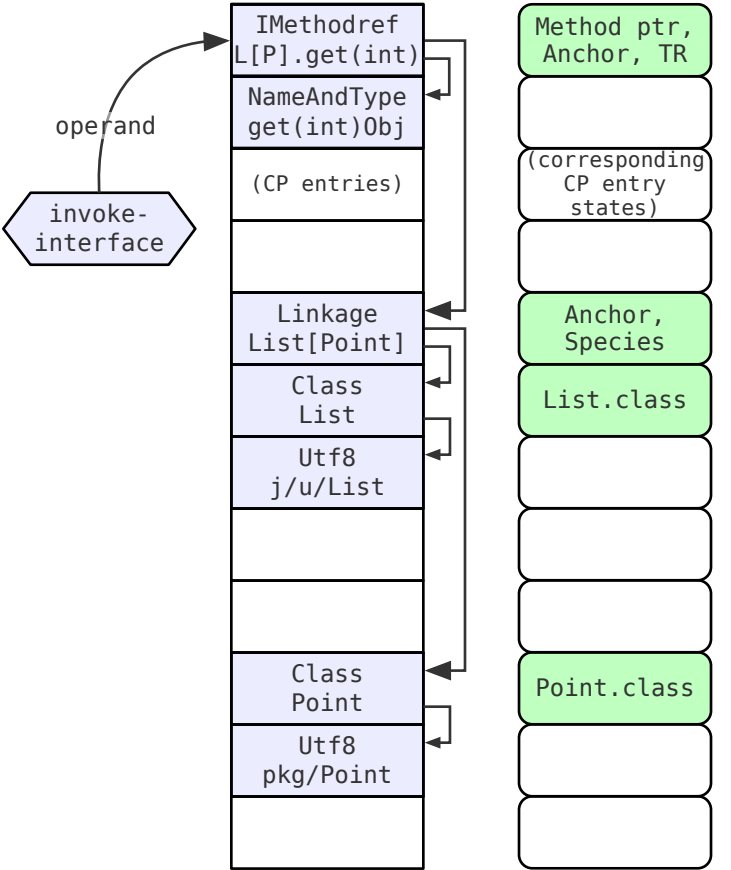


Diagram 4.4-H(b). Example parametric interface

public interface List<T> { …get… }

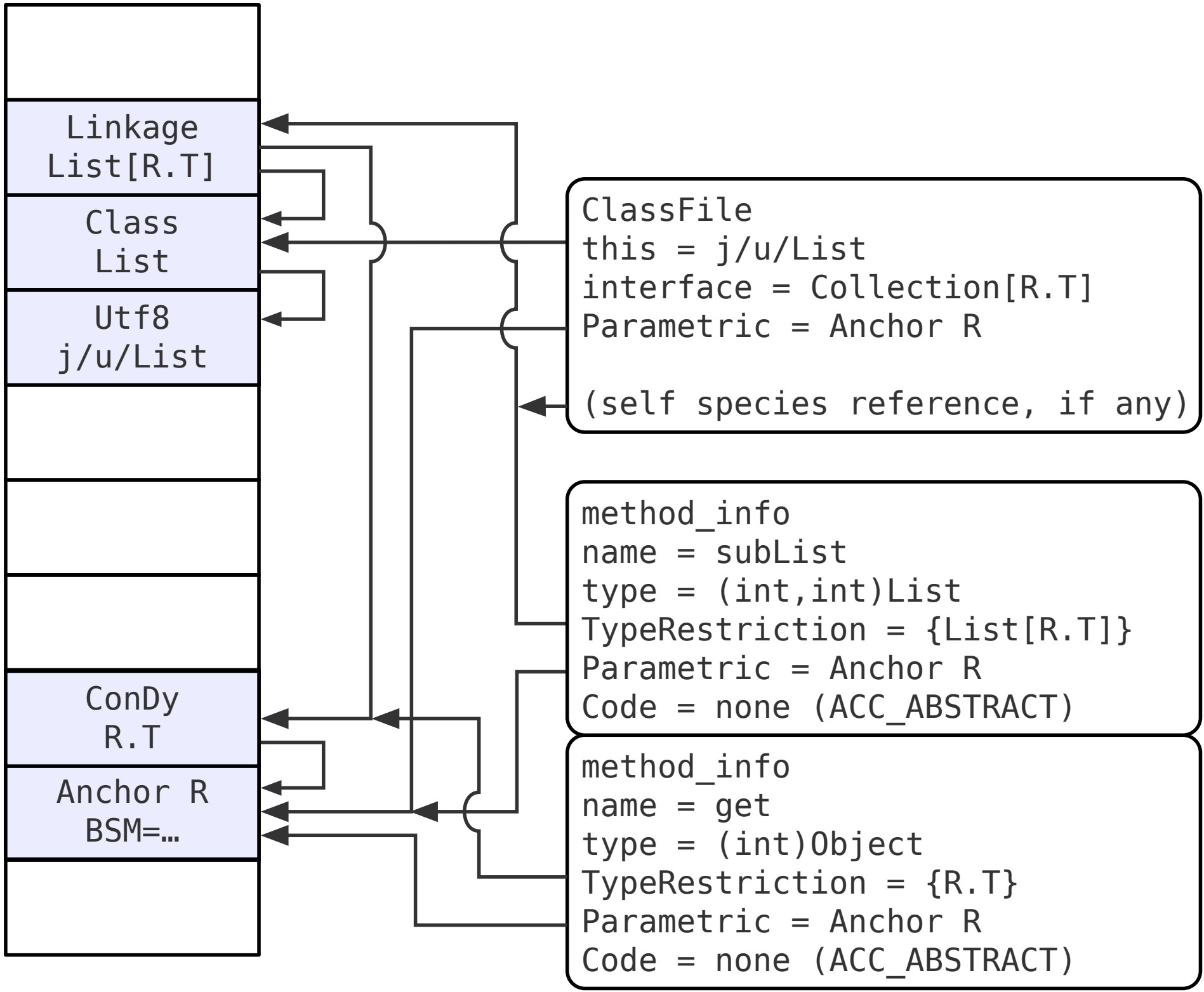


Diagram 4.4-H(c). Example parametric implementation
class ArrayList<T> implements List<T> { …get… }

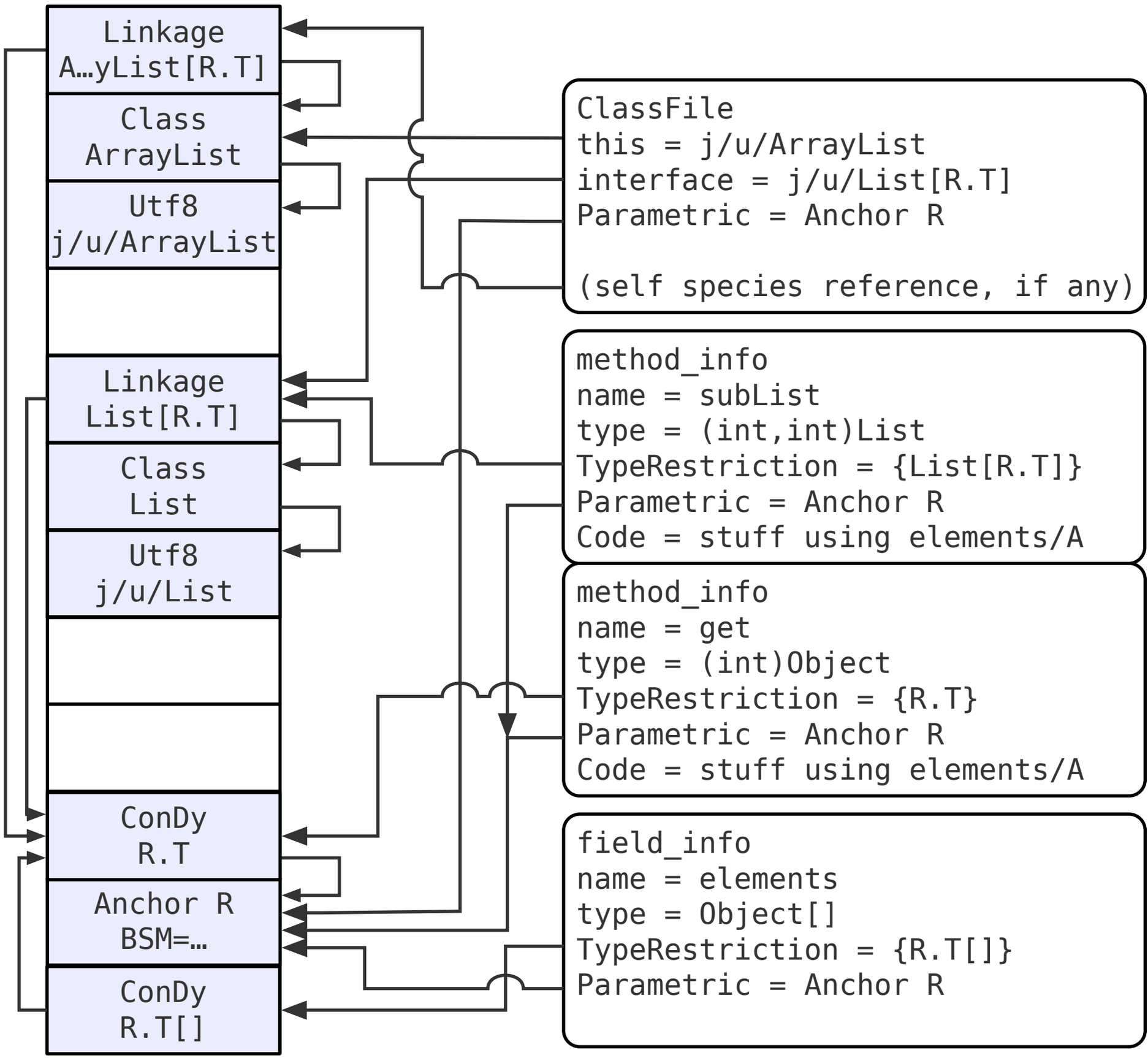


Diagram 4.4-H(d). Example parametric subclass

```
class MyVector<T> extends ju.Vector<T> { …get… }
class Vector<T> { … protected T[] elementData; … }
```

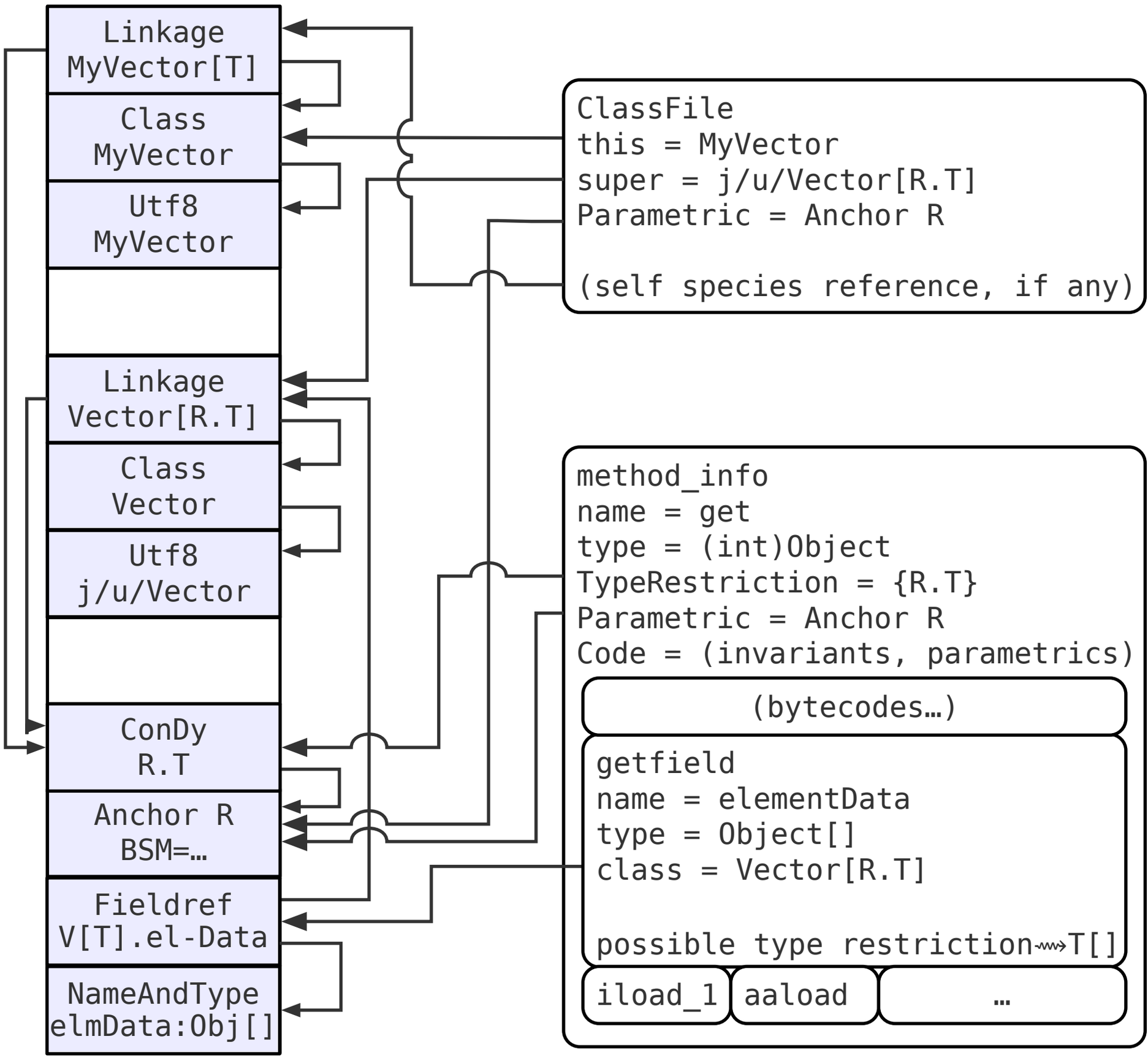


Diagram 4.7-D(a). Parametric attribute relations

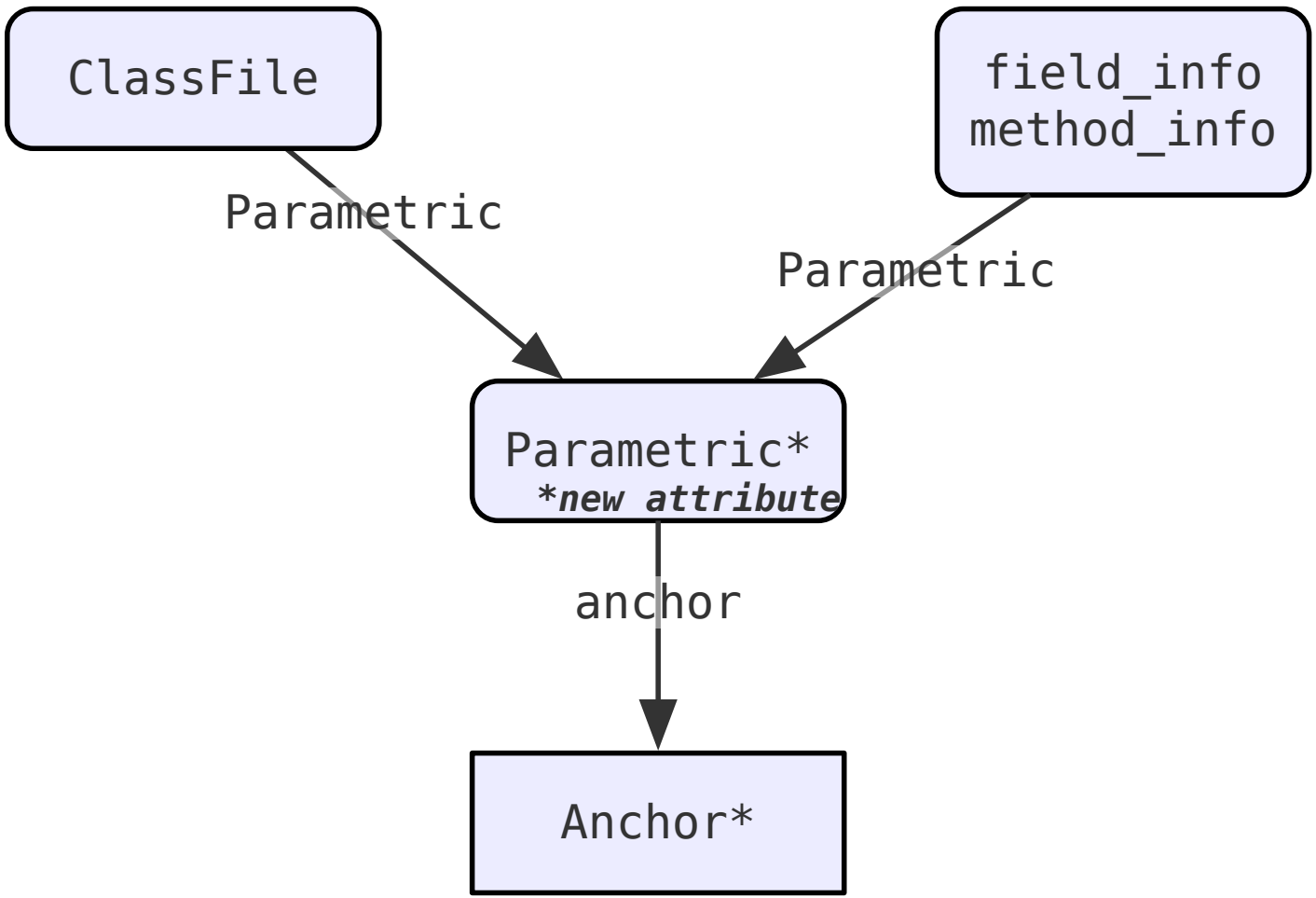


Diagram 4.7-D(b). TypeRestriction attribute relations

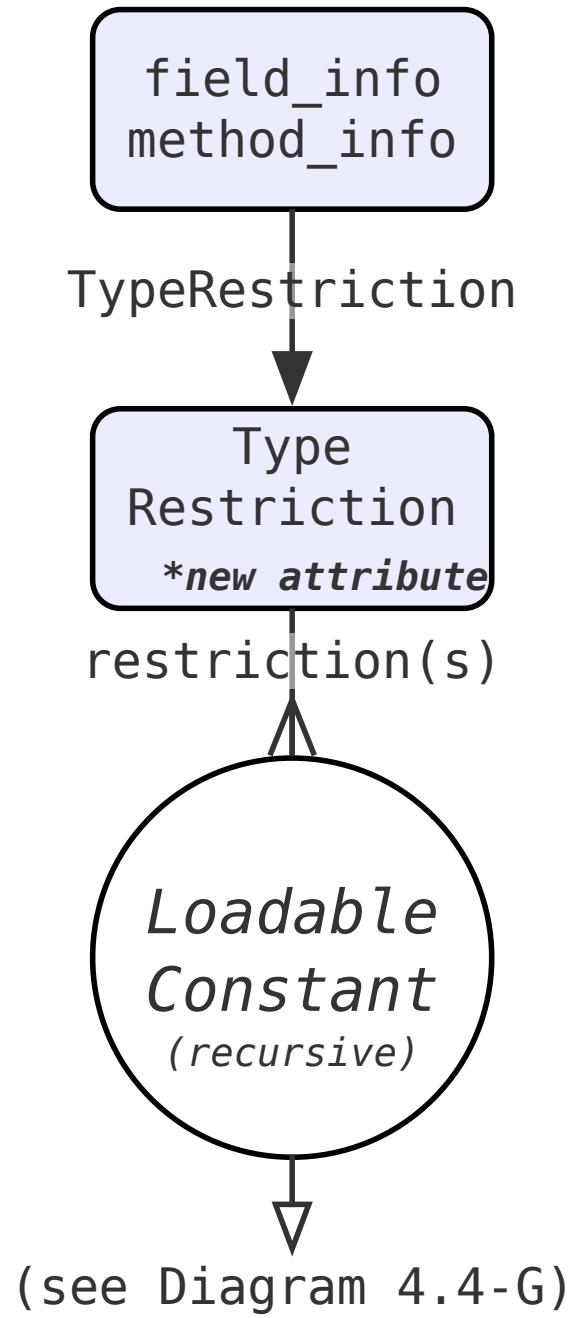
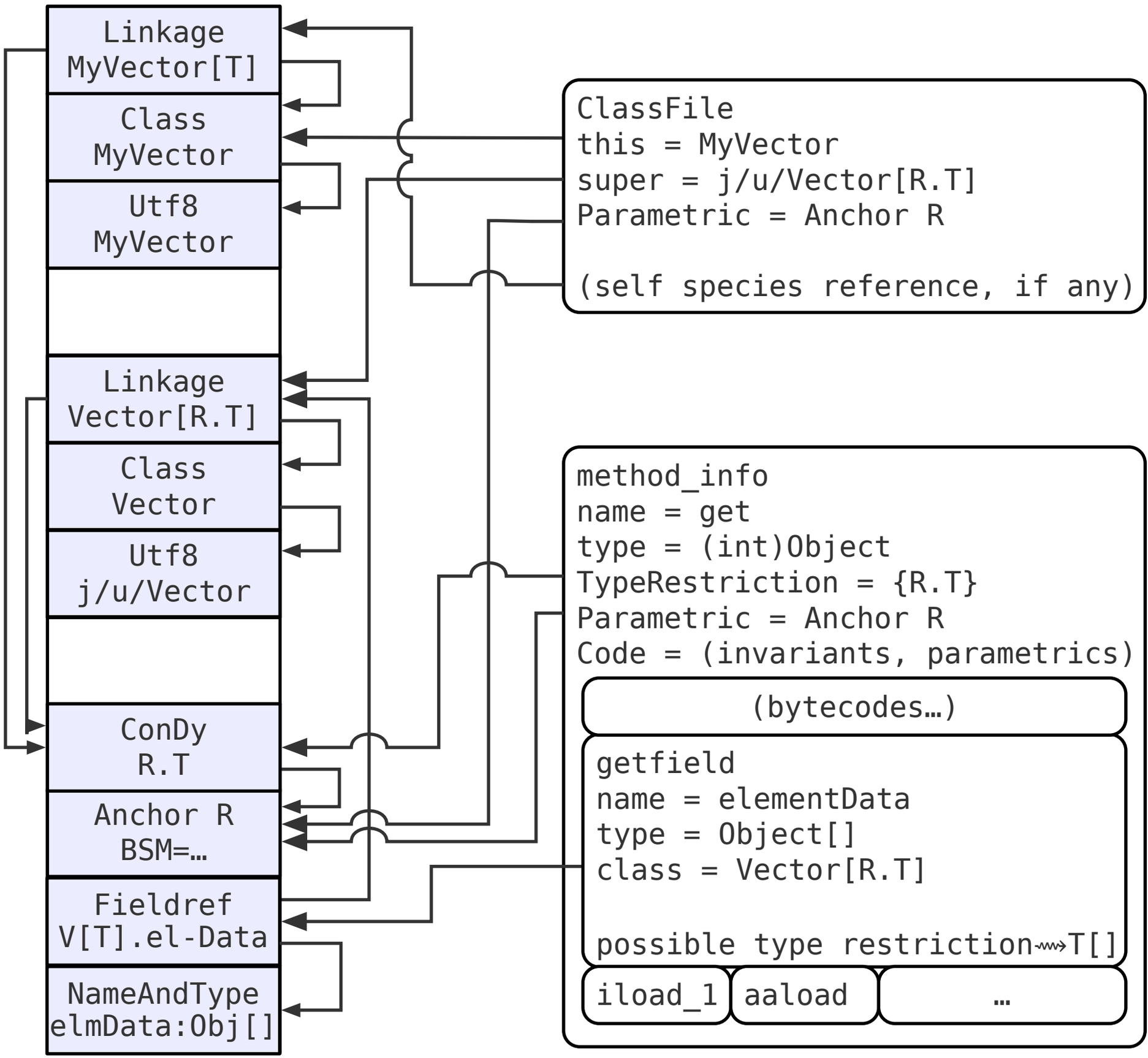


Diagram 4.4-H(d). Example parametric subclass

```
class MyVector<T> extends ju.Vector<T> { …get… }
class Vector<T> { … protected T[] elementData; … }
```



Graph of existing and proposed relations among constant pool structures, to support the “Parametric VM”.

Legend:

- A rectangular box shows one or more constant types. For example, a box labeled `Class` represents a `CONSTANT_Class_info` structure.
- An arrow shows how one entity depends directly on another. For example, `String` depends directly on the `Utf8` which specifies its characters.
- Arrows from boxes are labeled to show which item in a given constant pool structure provides the index for the relation indicated by the arrow.
- A circle shows, not a single constant type, but a conceptual group of types, which are (for some uses) interchangeable. The conceptual groups are *Loadable Constant, API Point Reference,* and *API Point Name*.
- Box-headed arrows into a group circle show users of all the types in the group, while arrows out of the circle indicate the group’s types. Small dotted arrows show some routes through a group.
- A starred footnote of *new* indicates a proposed new constant type or conceptual group of types. The new types are `SpecializationAnchor` and `SpecializationLinkage`, or `Anchor` and `Linkage` for short. The new conceptual groups are *API Point Reference* and (a subset) *API Point Name*.
- An arrow with a blank head redirects to a different diagram.
- Recursion points are white, and stand in place of the corresponding colored concept or constant type.
- A lozenge shaped box shows one or more instruction types.
- A box with rounded corners shows some other structure, such as the new `Parametric` and `TypeRestriction` attributes.