Variables (local variables and class fields)

In this section we are considering only plain Java variables.

Depending on where a variable is declared it can be either a:

- *Local variable* – an auxiliary temporary variable that exists only while a particular function or a block of statements is executed, or
- *Class variable* (or *class field* – more correct term in Java) – a variable that is present in any object of a class, and whose lifetime is the same as the object lifetime.

Local (temporary) variables

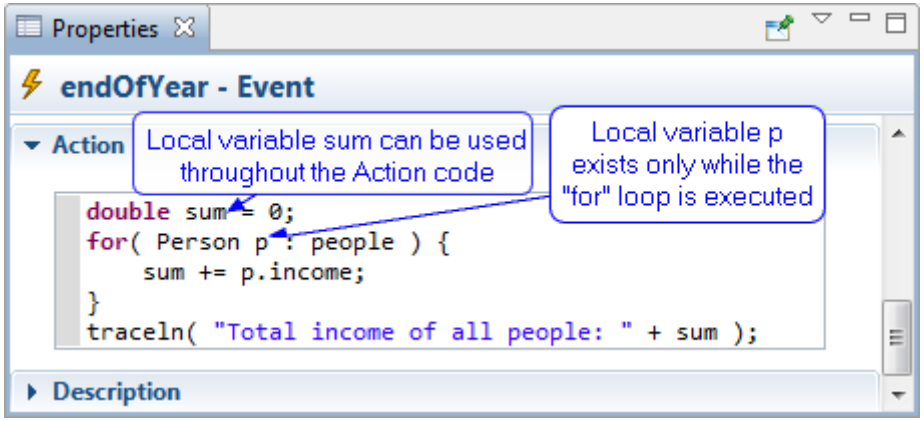
Local variables are declared in sections of Java code such as a block, a loop statement, or a function body. They are created and initialized when the code section execution starts and disappear when it finishes. The declaration consists of the variable type, name, and optional initialization. The declaration is a statement, so it should end with a semicolon. For example:

```
double sum = 0; //double variable sum initially 0

int k; //integer variable k, not initialized

String msg = ok ? "OK" : "Not OK"; //string variable msg initialized with expression
```

Local variables can be declared and used in AnyLogic fields where you enter actions (sequences of statements), such as **On startup** code of agent type, **Action** field of events or transitions, **Entry action** and **Exit action** of state, **On enter** and **On exit** fields of flowchart objects. The variables `sum` and `p` are declared in the action code of the event `endOfYear` in the Figure and exist only while this portion of code is being executed.



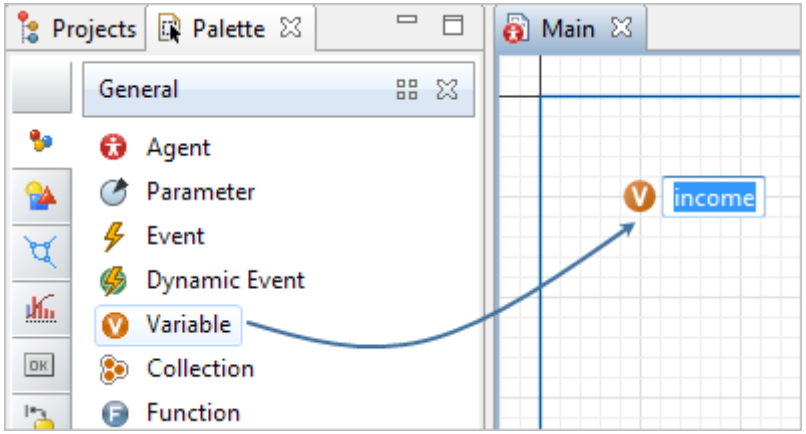
Local variables declared in the Action code of an event

Class variables (fields)

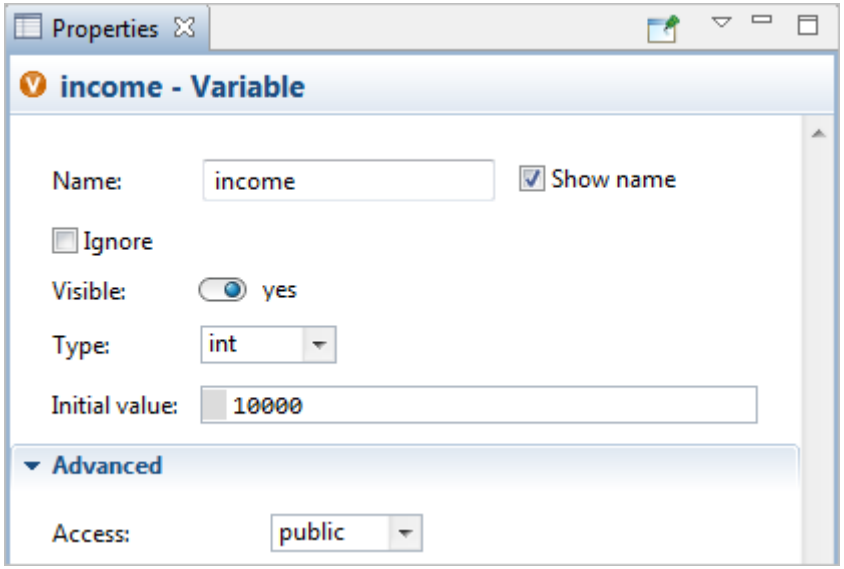
Java variables (fields) of agent class are part of "memory" or "state" of agents. They can be declared graphically or in code.

To declare a variable of agent (or experiment) class

1. Open the **Agent** palette and drag the **Variable** element from the palette to the canvas.
2. Type the variable name in the in-place editor or in the variable properties.

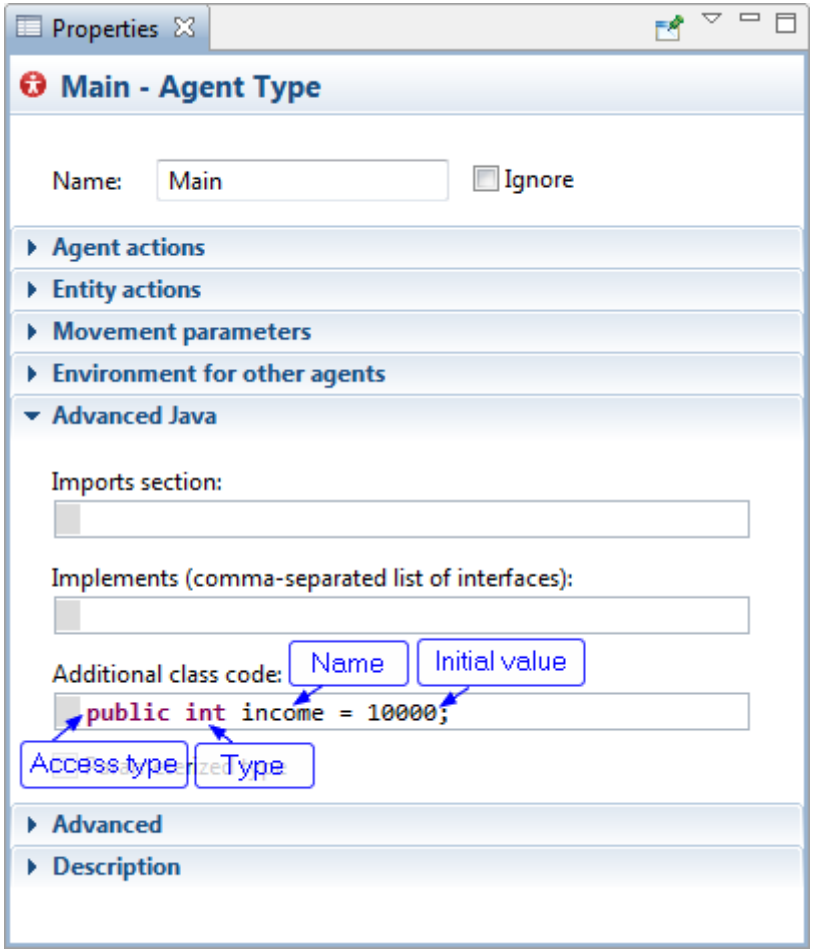


3. Choose the variable **Access** type in the variable properties. In most cases you can leave **default**, which means the variable will be visible within the current model. **Public** opens the access to the variable from other models, and **private** limits the access to this agent only.
4. Choose the variable type. If the type is not one of the primitive types, you should choose **Other** and enter the type name in the field nearby.
5. Optionally you can enter the variable **Initial value**.
6. If you do not specify the initial value, it will be `false` for `boolean` type, 0 for numeric variables, and `null` ("nothing") for all other classes including `String`.



A variable of an agent declared in the graphical editor

In the Figure a variable `income` of type `int` is declared in an agent (or experiment) type. Its access type is public, therefore it will be accessible from anywhere. The initial value of the variable is 10000. The graphical declaration above is equivalent to a line of code of the class, which you can write in **Additional class code** field in the **Advanced Java** property section of the agent type, see the Figure:



The same variable declared in the Additional class code field

Graphical declaration of a variable allows you to visually group it together with related functions or objects, and also to view or change the variable value at runtime with one click.