The Z notation is a formal language for writing mathematical specifications. It can be used to specify a wide range of systems, including the specification of a birthday book.

Here is an example of a simple specification for a birthday book using the Z notation:

Sorts

Person

Date

Operations

add\_person(p: Person, b: Date)

remove\_person(p: Person)

get\_age(p: Person, d: Date): Nat

get\_birthday(p: Person): Date

get\_person(b: Date): Person

Predicates

is\_birthday(d: Date, p: Person)

is\_older(p1: Person, p2: Person, d: Date)

Axioms

forall p: Person, b: Date. add\_person(p, b) <=>

not(is\_birthday(b, \_))

forall p: Person. remove\_person(p) =>

not(is\_birthday(\_, p))

forall p: Person, d: Date. get\_age(p, d) = d - get\_birthday(p)

forall p: Person, b: Date. is\_birthday(b, p) <=>

get\_birthday(p) = b

forall p1, p2: Person, d: Date. is\_older(p1, p2, d) <=>

get\_age(p1, d) > get\_age(p2, d)

This specification defines a birthday book as a system with a set of people (Person) and dates (Date). It includes operations for adding and removing people from the book, as well as for retrieving information about a person's age and birthday. It also includes predicates for checking whether a given date is someone's birthday and for comparing the ages of two people. Finally, it includes axioms that specify the relationships between these elements and constrain their behavior.