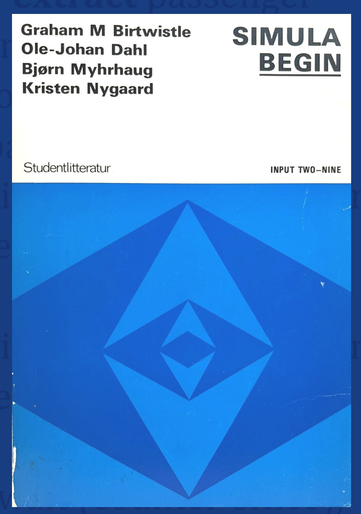
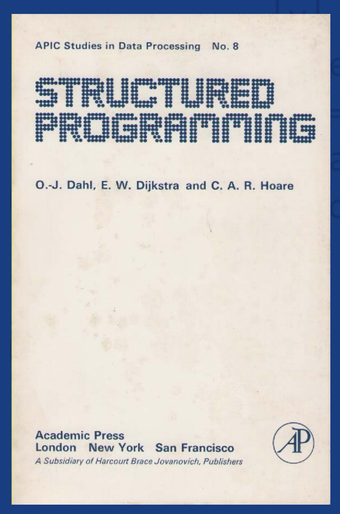
Object-oriented programming

1960s—**Simula** anticipates many aspects of later object-oriented languages in the context of mathematical simulations.



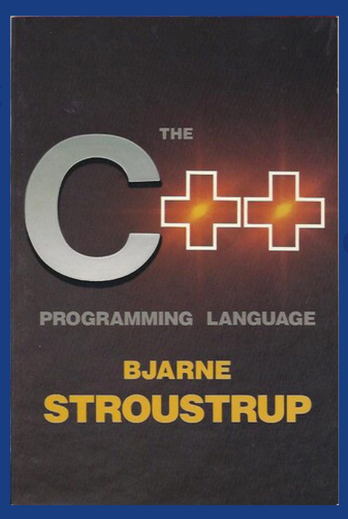


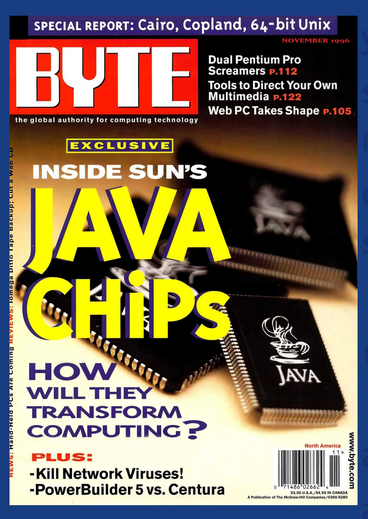

```
SIMULA begin comment airport departure;
set q counter, q fee, q control, lobby (passenger);
  counter office (clerk); ...
activity passenger; Boolean fee paid;
  begin fee paid := random (0, 1) < 0.5; ... ;
    wait (q counter) end;
...
end of SIMULA;
```

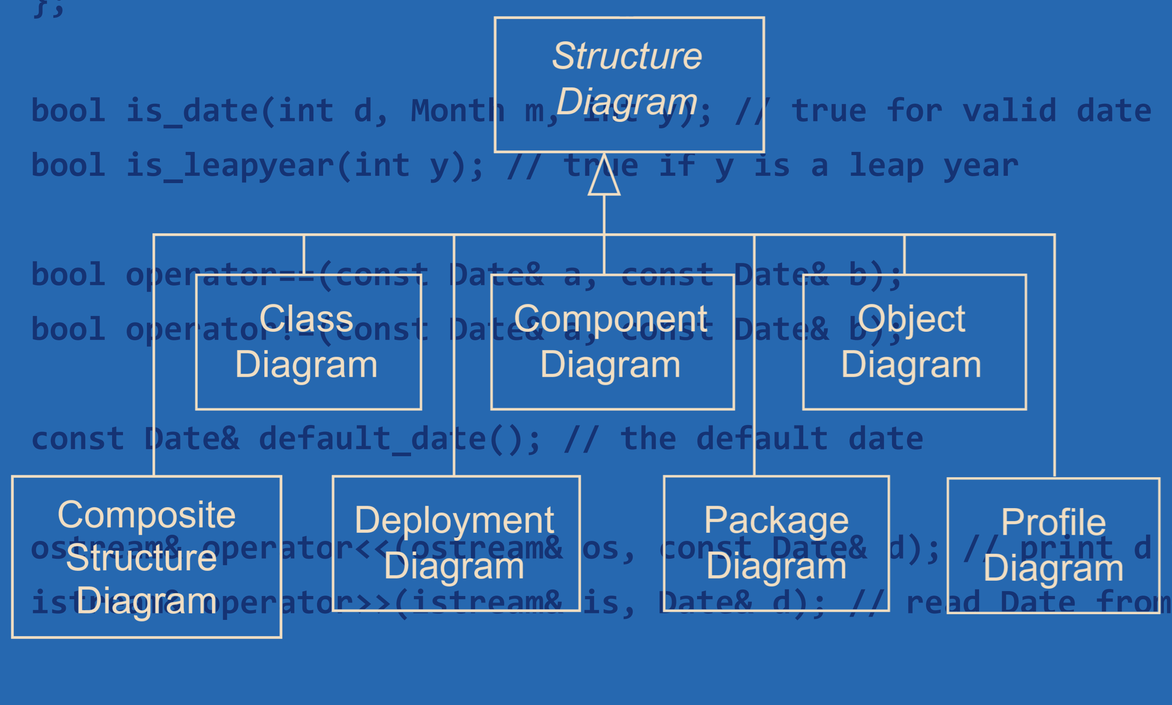
```
exampleWithNumber: x
|y|
true & false not & (nil isNil) ifFalse: [self halt].
y := self size + super size.
#($a #a 'a' 1 1.0)
  do: [:each | Transcript
                 show: (each class name);
                 show: (each printString);
                 show: ' '].
```

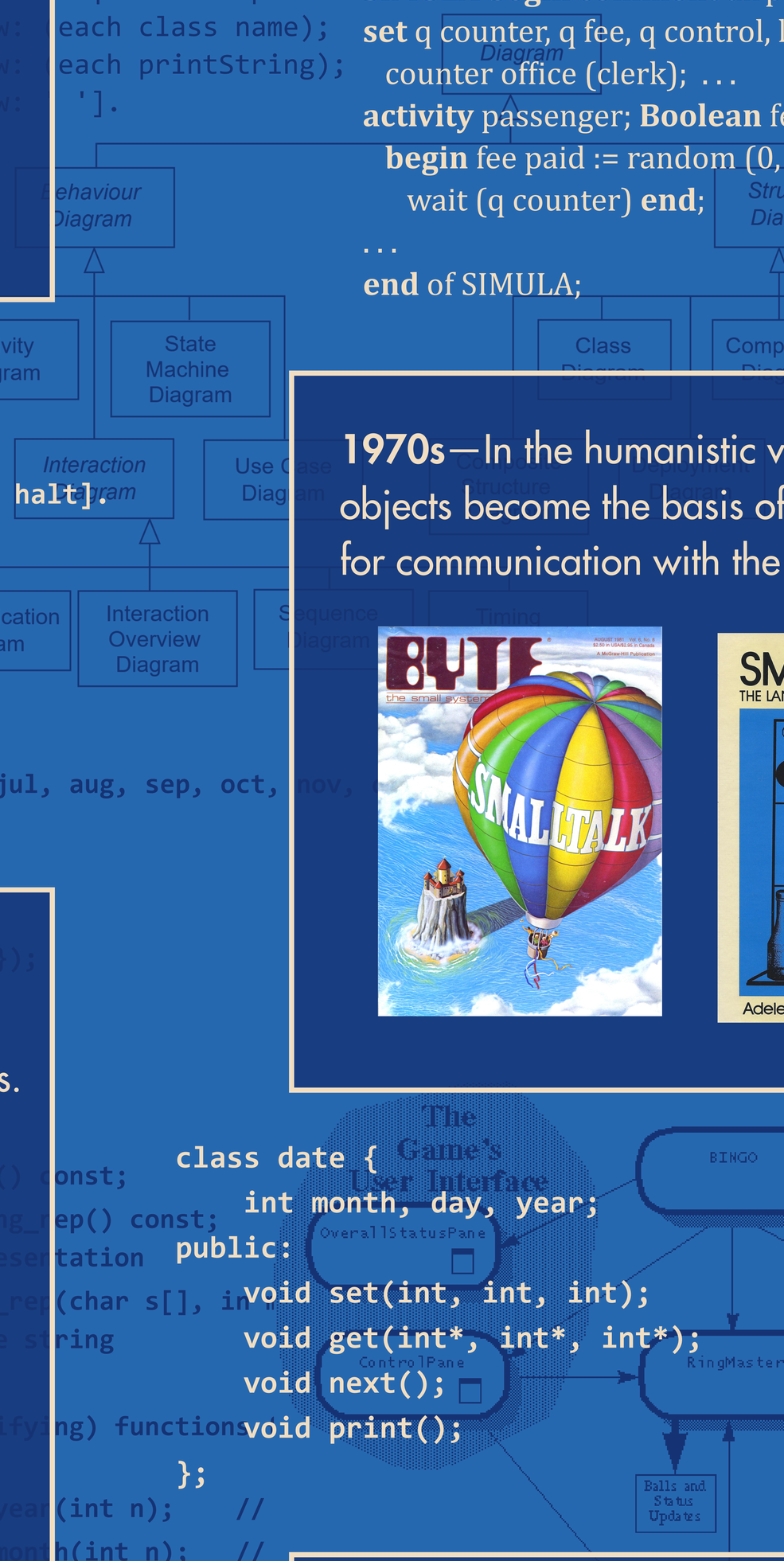
```
namespace Chrono {
  enum class Month { jan=1, feb, mar, apr, may, jun, jul, aug, sep, oct, nov,
  class Date {
  public: // public interface:
    class Bad_date { }; // exception class
    Date(int dd =0, Month mm =0, int yy =0);
    // nonmodifying functions for examining the Date:
    int day() const;
    Month month() const;
    int year() const;
    string string_rep() const;
    void char_rep(char s[], int) const;
    // C-style string
    // (modifying) functions:
    Date& add_year(int n);    // add n years
    Date& add_month(int n);   //
    Date& add_day(int n);     // add n days
  private:
    bool is_valid();          // check if this Date represents a date
    int d, m, y;              // representation
  };

  bool is_date(int d, Month m, int y); // true for valid date
  bool is_leapyear(int y); // true if y is a leap year

  bool operator==(const Date& a, const Date& b);
  bool operator!=(const Date& a, const Date& b);

  const Date& default_date(); // the default date

  ostream& operator<<(ostream& os, const Date& d); // print d to os
  istream& operator>>(istream& is, Date& d); // read Date from is into d
```

1980s—Object-oriented programming becomes a household name. **C++** and later **Java** focus on practical engineering concerns.

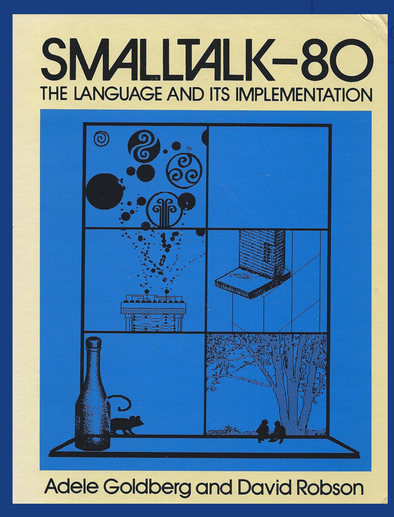


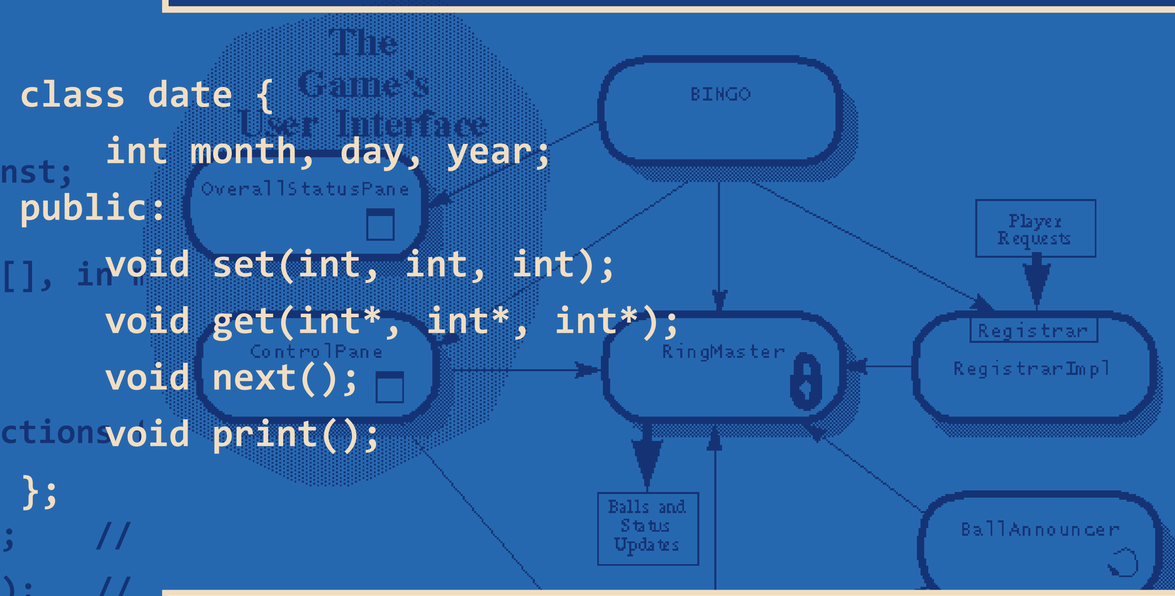


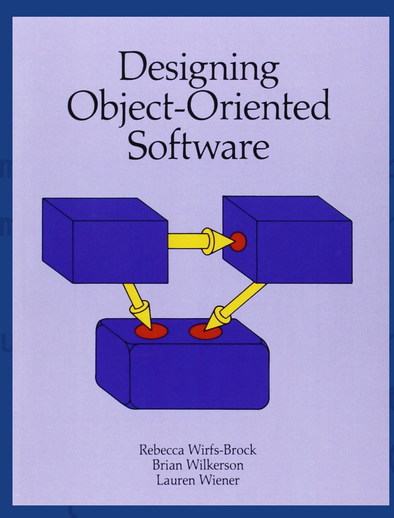
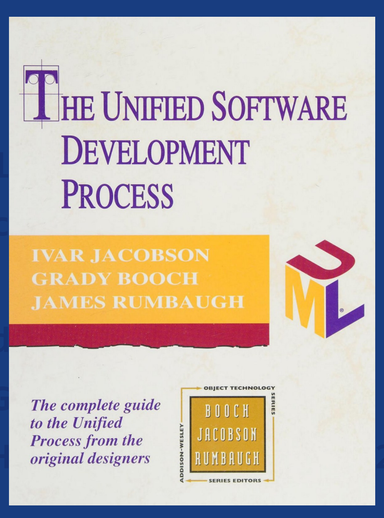




1970s—In the humanistic vision of **Smalltalk**, objects become the basis of a new medium for communication with the computer.






```
class date {
    int month, day, year;
public:
    void set(int, int, int);
    void get(int*, int*, int*);
    void next();
    void print();
};
```

1990s—**UML** and new software development processes shift attention to the managerial problem of structuring systems and teams.