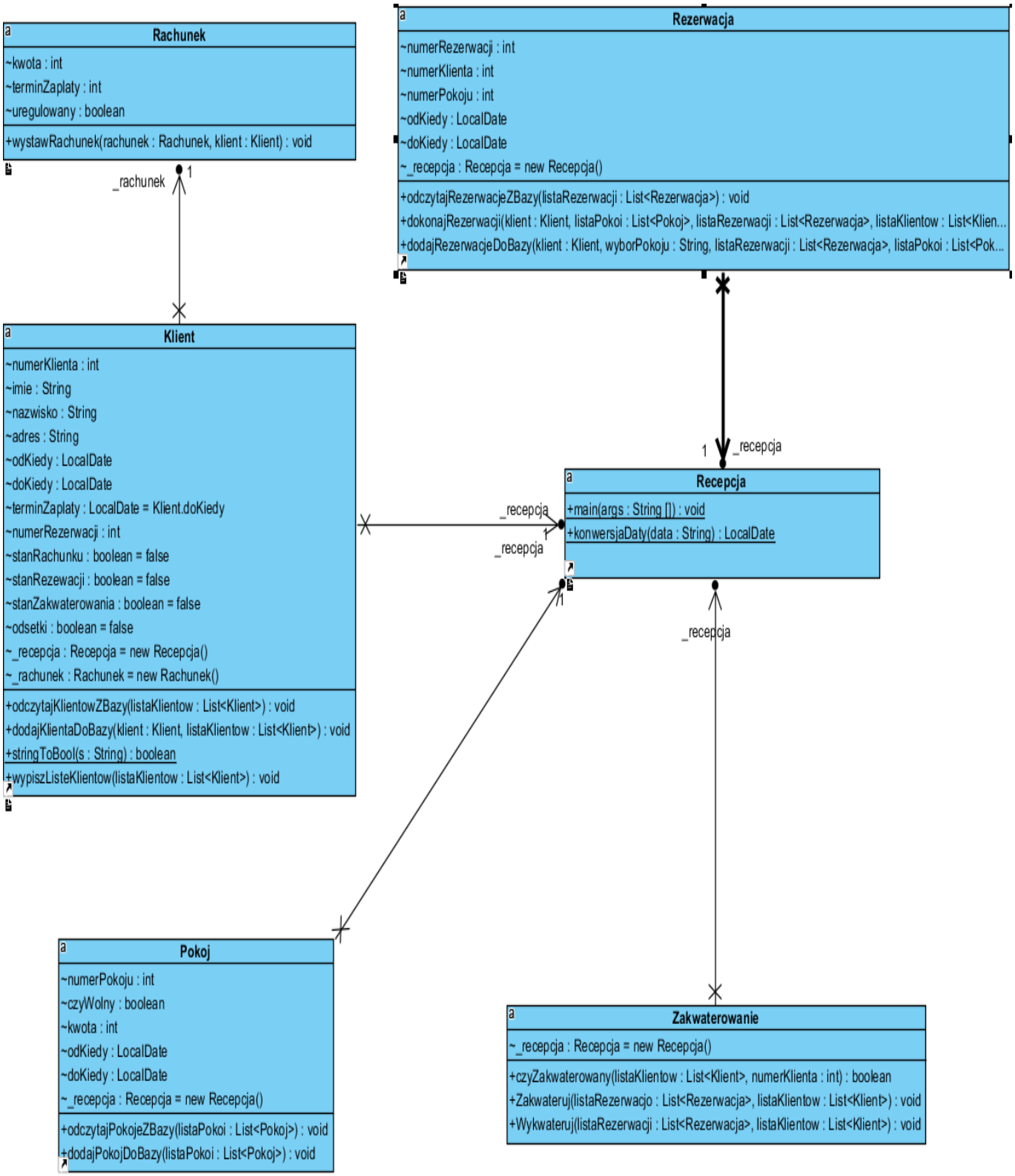
Anna Jankowska 218187
Kacper Cierzniewski 218233

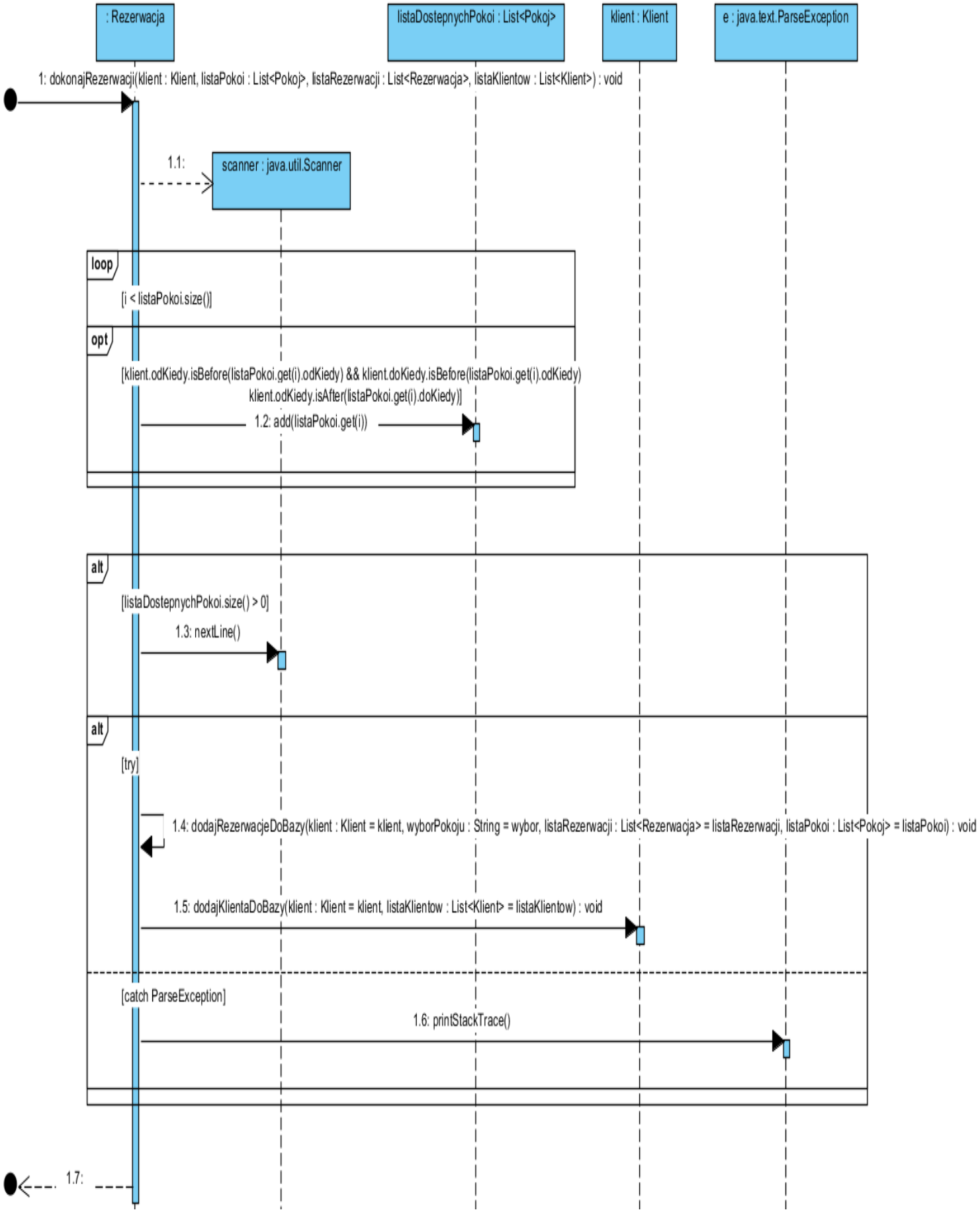
Etap 5,6,7: Identyfikacja klas reprezentujących logikę biznesową projektowanego oprogramowania. pracowanie diagramów sekwencji dla wybranych przypadków użycia reprezentujących usługi oprogramowania wynikających również z wykonanych diagramów czynności.

1. Diagram klas

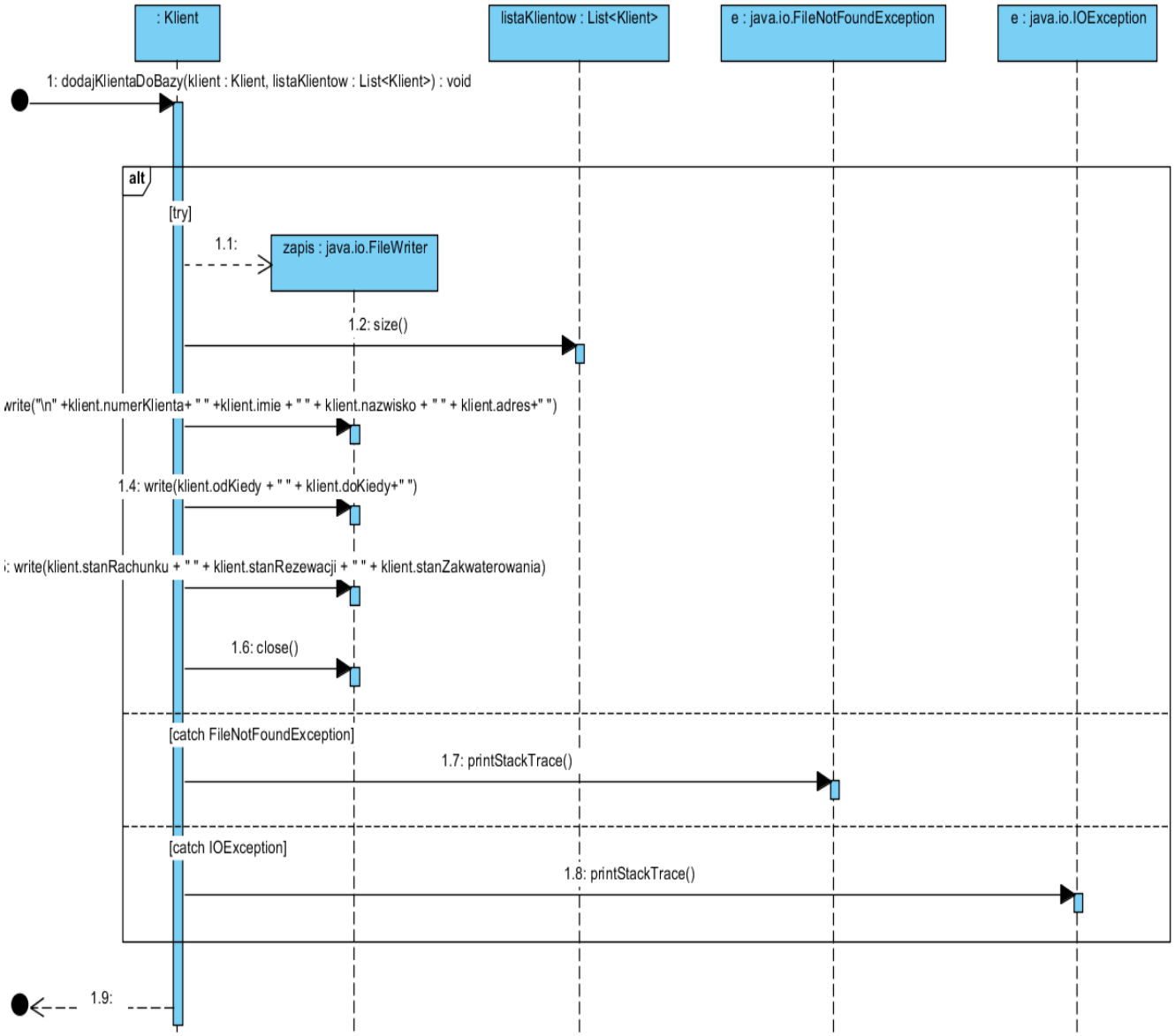


2. Diagramy sekwencji

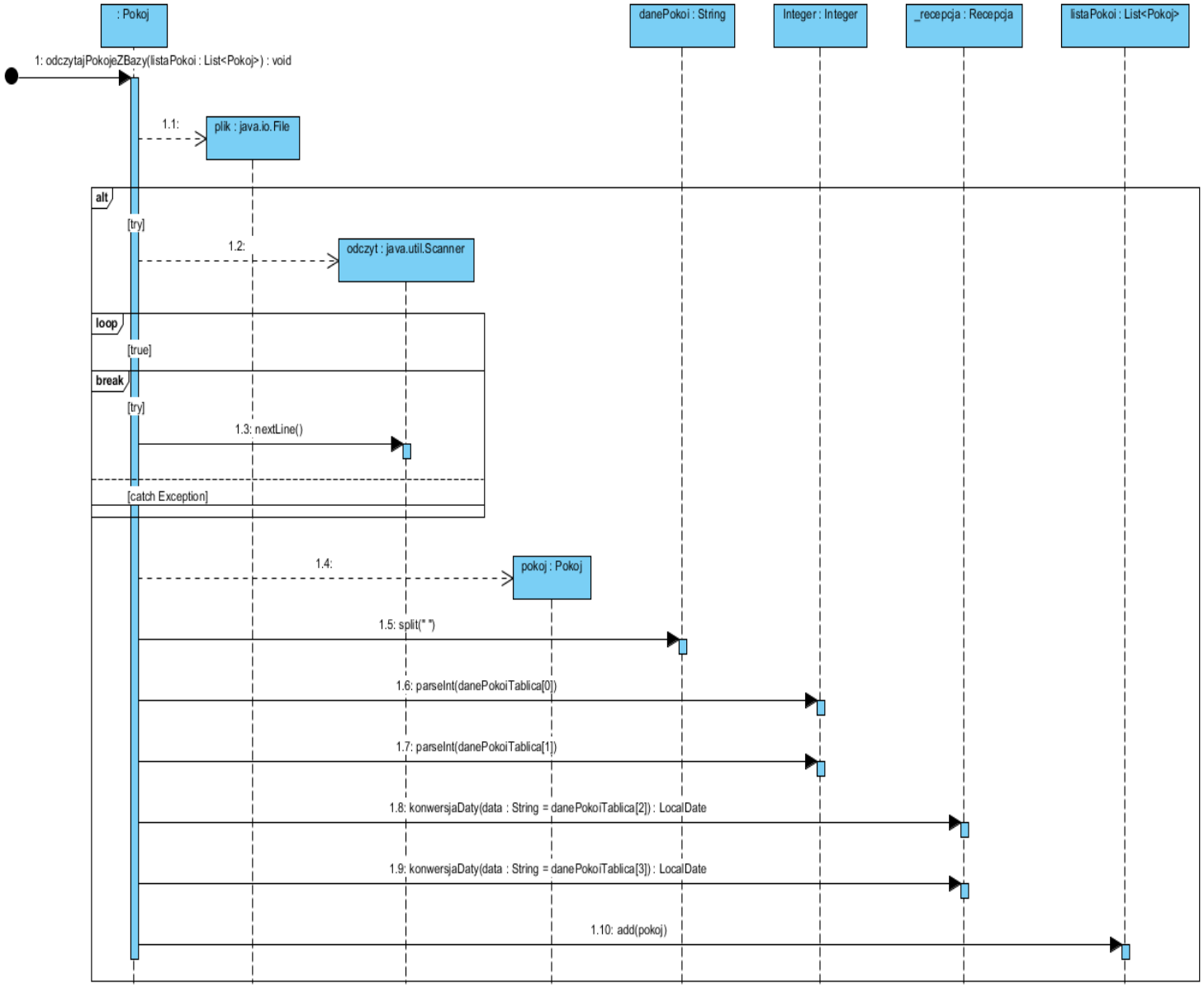
1) **Dokonanie rezerwacji klienta**



2) **Dodanie klienta i jego danych do bazy**



3) **Odczytanie informacji o pokojach z bazy**



3. Kody dla powyższych diagramów sekwencji

1) Dokonanie rezerwacji klienta

```
public void dokonajRezerwacji(Klient klient, List<Pokoj> listaPokoi, List<Rezerwa-
cja> listaRezerwacji, List<Klient> listaKlientow) {
    Scanner scanner = new Scanner(System.in);
    String wybor;
    List<Pokoj> listaDostepnychPokoi = null;
    System.out.println("Lista dostępnych pokoi: ");
    for (int i = 0; i < listaPokoi.size(); i++)
        if (klient.odKiedy.isBefore(listaPokoi.get(i).odKiedy) && klient.do-
Kiedy.isBefore(listaPokoi.get(i).odKiedy) ||
                klient.odKiedy.isAfter(listaPokoi.get(i).doKiedy))

        {
            listaDostepnychPokoi.add(listaPokoi.get(i));
            System.out.println("Numer pokoju: " + listaPokoi.get(i).numerPokoju +
" " + listaPokoi.get(i).kwota + "  zł");

        }
    if (listaDostepnychPokoi.size() > 0) {
        System.out.println("Wybierz pokój do rezerwacji: ");
        wybor = scanner.nextLine();

        try {
            dodajRezerwacjeDoBazy(klient, wybor,listaRezerwacji,listaPokoi );
            klient.dodajKlientaDoBazy(klient,listaKlientow);
            System.out.println("Rezerwacja dokonana pomyślnie! ");
        } catch (ParseException e) {
            e.printStackTrace();
        }


    } else {
        System.out.println("Brak dostępnych pokoi w wybranym terminie!");
    }


}
```

2) **Dodanie klienta i jego danych do bazy**

```
public  void dodajKlientaDoBazy(Klient klient, List<Klient> listaKlientow) {
    FileWriter zapis ;
    try {
        zapis = new FileWriter("klienci.txt", true);
        klient.numerKlienta=listaKlientow.size()+1;
        zapis.write("\n" +klient.numerKlienta+ " " +klient.imie + " " + klient.na-
zwisko + " " + klient.adres+" ");
        zapis.write(klient.odKiedy + " " + klient.doKiedy+" ");
        zapis.write(klient.stanRachunku + " " + klient.stanRezewacji + " " +
klient.stanZakwaterowania);
        zapis.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

3) Odczytanie informacji o pokojach z bazy

```
public void odczytajPokojeZBazy(List<Pokoj> listaPokoi) throws ParseException {
    File plik = new File("pokoje.txt");
    String danePokoi;
    int i=0;
    try {
        Scanner odczyt = new Scanner(plik);

        while (true) {
            try {
                danePokoi = odczyt.nextLine();
            }
            catch(Exception e){
                break;
            }
            Pokoj pokoj = new Pokoj();

            String danePokoiTablica[] = danePokoi.split(" ");
            pokoj.numerPokoju = Integer.parseInt(danePokoiTablica[0]);
            pokoj.kwota =Integer.parseInt(danePokoiTablica[1]);
            pokoj.odKiedy = _recepcja.konwersjaDaty(danePokoiTablica[2]);
            pokoj.doKiedy = _recepcja.konwersjaDaty(danePokoiTablica[3]);
            listaPokoi.add(pokoj);
        }
    } catch (FileNotFoundException e) {
        System.out.println("ERROR");
    }

}
```