Banking Exercise No. 2

This exercise is the continuation of **Banking Exercise No. 1**, from session 2
It involves several classes, but only one object per class

1. Add a new class Customer to the Bank project. Customer objects are to have the following private attributes: name, street, postal code, town and telephone number. Write a constructor with parameters which initialize the attributes to the values given as arguments to the constructor. Test the constructor.
2. Add an attribute of type BankAccount to the Customer class. Name the attribute myBankAccount. This attribute associates objects of type Customer to an object of type BankAccount. The attribute represents that a customer owns a bank account and makes it possible to call methods on BankAccount from Customer. (In Customer one may write myBankAccount.foo(), where foo() is some method on BankAccount
3. Now add a method to class Customer that takes an object of type BankAccount as argument and assigns the argument to the attribute myBankAccount. You may want to use the following method heading: *public void addAccount(BankAccount account);*
4. We are now ready to add a method to Customer that prints all information about the customer including the balance of the customer's account. Hint: The method is to include a call to the getBalance()-method on BankAccount.
5. The banking system is now to be prepared for handling different types of accounts such as saving accounts, credit accounts, wage accounts etc. To handle information about different types of accounts a new class AccountType is to be added to the project. This class should have two attributes: a string to hold the name of the account type and an interest rate (all accounts of given type have the same interest rate). Write this class including a constructor that takes type and interest rate as arguments and accessor methods for the attributes.
6. Add an attribute to class BankAccount with type AccountType. Name the attribute accountType. Add a new constructor to class BankAccount that takes an account type as argument and assigns the value of the argument to the attribute accountType. The constructor is still to assign 0 to balance.
7. Add a method to class BankAccount that returns the type of the account. Use this method heading: *public AccountType getAccountType()*
8. Modify the method created in 4. so the method also prints information about the type of the account and the interest rate.