1. Now a software system is going to be developed, and for the system it is necessary to support different types of database, abstract factory pattern can be chosen to implement the system. Here, we take two types of database such as SQL Server, Access and a table such as Department in the system for an example. The class diagram is as follows



Please fill the following blanks numbered (1) – (6)

[Java code]

import java.util.\*;

class Department {/\* abbreviated code\*/}

interface IDepartment {  
(1) ;  
(2) ;  
}

class SqlserverDepartment (3) {  
public void Insert(Department department) {  
System.out.println("Insert a record into Department in SQL Server!");  
// abbreviated code  
}

public Department GetDepartment(int id) {  
/\* abbreviated code \*/  
}  
}

class AccessDepartment (4) {  
public void Insert(Department department) {  
System.out.println("Insert a record into Department in Access!");  
// abbreviated code  
}  
public Department GetDepartment(int id) {  
/\* abbreviated code \*/  
}  
}

(5) {  
(6) ;  
}

class SqlserverFactory implements IFactory {  
public IDepartment CreateDepartment() {  
return new SqlserverDepartment();

}  
// abbreviated code  
}

class AccessFactory implements IFactory {  
public IDepartment CreateDepartment() {  
return new AccessDepartment();  
}

// abbreviated code

}