|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**КУРСОВАЯ РАБОТА**

по дисциплине: Объектно-ориентированное программирование

по профилю: Информатизация предприятий

направления профессиональной подготовки: Прикладная информатика, бакалавриат

Тема: «Разработка программы для управления деятельностью агентства по подбору персонала»

Студент: Ватулин Дмитрий Сергеевич

Группа: ИНБО-04-18

Работа представлена к защите\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Руководитель: Ассистент кафедры ИиППО Хлебникова В.Л

Работа допущена к защите\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Оценка по итогам защиты: \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

М. МИРЭА. 2019г.

# УДК 004.432.4

# Ватулин Д.С. Проектирование и разработка программы для управления деятельностью агентства по подбору персонала» / Курсовая работа по дисциплине «Объектно-ориентированное программирование» профиля «Информатизация предприятия» направления профессиональной подготовки бакалавриата 09.03.03 «Прикладная информатика» (2-ой семестр) / руководитель асс. Хлебникова В.Л. / кафедра ИиППО Института ИТ РТУ МИРЭА – с. 54, табл. 1, илл. 7, ист. 4, (в т.ч. 1 на англ. яз.).

# Целью данной работы является написание программы для управления деятельностью агентства по подбору персоналас использованием знаний в области объектно-ориентированного программирования, полученных в рамках лекционных и семинарских занятий по соответствующей дисциплине.

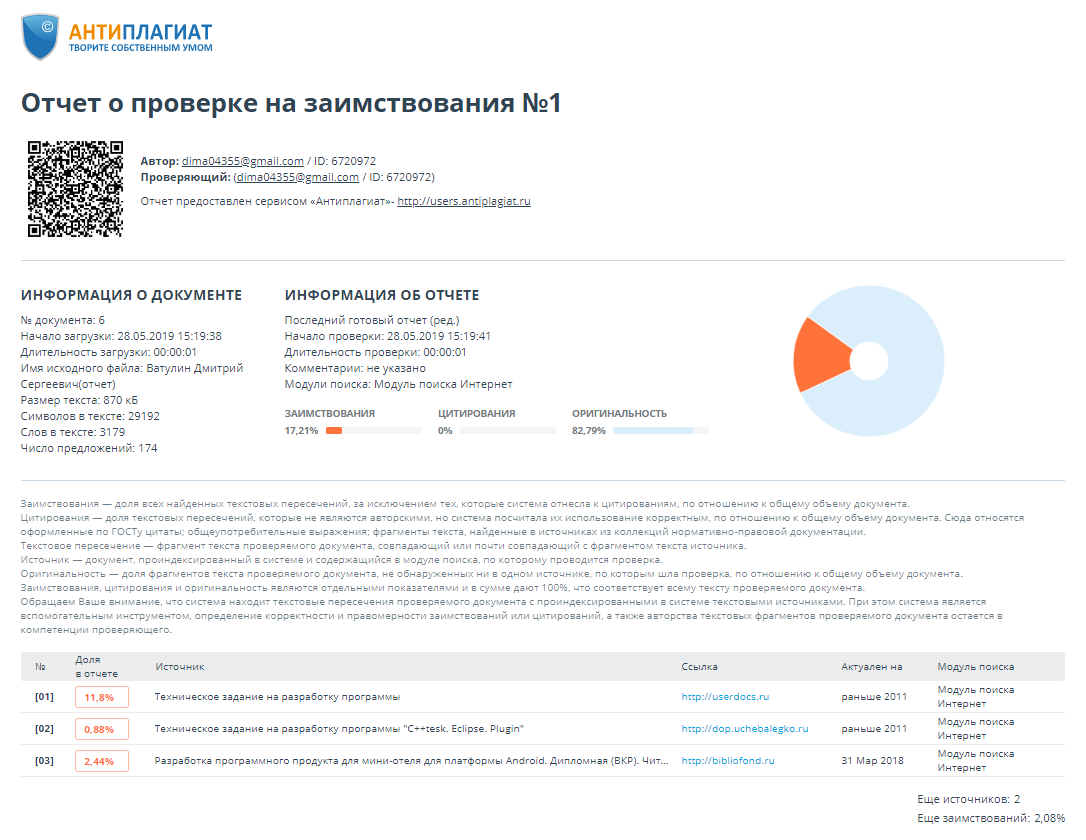
# В рамках данной работы проведено моделирование нескольких классов, а также была создана программа на языке C++, представляющая собой реализацию программы для управления деятельностью агентства по подбору персонала.

Vatulin D.S. Design and development of a program to manage the activities of the recruitment agency ”/ Course work in the discipline“ Object-Oriented Programming ”of the profile“ Informatization of an Enterprise ”Bachelor's Degree in Professional Education 09.03.03“ Applied Informatics ”(2nd semester) / Head Ass. Khlebnikov V.L. / Department IiPPO Institute IT RTU MIREA - p. 54, tab. 1, ill. 7, ist. 4, (including 1 in English. Lang.).

The purpose of this work is to write a program to manage the activities of the recruitment agency using knowledge in the field of object-oriented programming, obtained in the framework of lectures and seminars on the relevant discipline.

As part of this work, several classes were modeled, and a C ++ program was created, representing the implementation of a program for managing recruitment agency activities.

М. МИРЭА, Ин-т ИТ, каф. ИиППО. 2019. @ Ватулин Д.С.



**ОТЗЫВ РУКОВОДИТЕЛЯ КУРСОВОЙ РАБОТЫ**

**ПО КАФЕДРЕ ИиППО ИНСТИТУТА ИТ МИРЭА**

Настоящий отзыв составлен руководителем КР студента Ватулина Д.С. ИНБО-04-18 шифр 09.03.03 по итогам выполнения им на кафедре ИиППО института ИТ КР по дисциплине «Объектно-ориентированное программирование» во 2 семестре 1 курса обучения в бакалавриате на основании ФГОС ВО и действующих в РТУ МИРЭА нормативных актов: «Инструкция по организации и проведению курсового проектирования (СМКО МИРЭА 7.5.1/04.И.05)»; «Временное положение о текущем контроле успеваемости и промежуточной аттестации по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры (Положение. СМКО МИРЭА 7.5.1/03.П.08)»; «Рекомендации по оформлению письменных работ обучающимися (СМКО МИРЭА 7.5.1/03.П.69)»; а также Рабочей программы поименованной дисциплины.

Руководителем КР отмечается, что задание на курсовую работу выдано в первые две недели семестра. В это же время со студентом проведено собеседование по вопросам организации, учебно-трудовой дисциплины, графика работ и взаимодействия с руководителем, поставлены, обеспечены информационными ресурсами и обсуждены цель, основная задача, содержание, технологии, информационно-методическое обеспечение, формы и сущности отчётности и её защиты под дифференцированную аттестацию. Проведён инструктаж по соблюдению требований охраны труда, техники безопасности (ТБ), а также правилам внутреннего трудового (учебного) распорядка (ПВР).

За студентом закреплен тьютор - стажёр из числа актива подразделения СНК при кафедре ИиППО, выход в кафедральный тематический мини-портал Кабинета дипломного проектирования и магистерской подготовки при кафедре ИиППО (далее: «Кабинет»); в портале создана соответствующая директория курсовой работы, в распоряжение проектанта предоставлены стартовые информационные консисты тематических библиотек портала, выданы методические указания и типовые шаблоны документального оформления пояснительной записки (ПЗ), относящиеся к конкретной дисциплине – в совокупности позволяющие эффективно использовать средства Онтонет управления образовательным макромедиа контентом, а также всемирных технологий WorkBooks. Обозначено и реализовано обретение обучающимся профессиональных компетенций, а именно:ОПК-2, ПК-8

Отчётные материалы студента в целом отвечают заданной теме, поставленной задаче, требованиям по оформлению. Предложенные решения обладают полнотой и качественной глубиной раскрытия изучаемых вопросов, демонстрируя признаки новизны, креативности и меры самостоятельности, а также проявление проектантом дисциплинированности, элементов профессиональной этики, умения работать в коллективе.

Качество, полнота и технико-эстетический уровень оформления ПЗ удовлетворяют типовым требованиям. Проверенная руководителем на работоспособность, соответствие и незашумлённость электронная версия ПЗ в форматах Word и Pdf удовлетворительна.

По итогам курсовой работы, представления ПЗ руководитель согласно перечисленным выше нормативам подводит следующие итоги:

1. Качественная сторона выполненных работ - \_\_\_\_\_\_\_\_\_\_\_\_(отл., хор., удовл., неудовл.).
2. Количественная сторона (характеризующая охват, объём, детализацию и т.п.) выполненных работ \_\_\_\_\_\_\_\_ (отл., хор., удовл., неудовл.).
3. Содержание планово-отчётной документации проектанта\_\_\_\_\_\_\_\_ (отл., хор., удовл., неудовл.).
4. Качество оформления документации на бумажном носителе и в электронной форме (в том числе, соблюдение требований стандартов и упомянутых выше нормативов) \_\_\_\_\_\_\_\_\_\_\_(отл., хор., удовл., неудовл.).
5. Учебно-производственная дисциплина, включая соблюдение установленных сроков выполнения, отчётности и аттестаций, а также норм учебно-профессиональной этики \_\_\_\_\_\_\_\_\_(отл., хор., удовл., неудовл.).

Итоговая оценка, рекомендованная руководителем:\_\_\_\_\_\_\_\_\_\_\_\_ (отл., хор., удовл., неудовл.).

Таким образом, курсовая работа успешно и в срок завершено. Курсовая работа рекомендована к защите.

Руководитель:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(должность, ф.и.о., дата)

Подпись отв. лица. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Должность. Дата.

Оглавление

[ГЛАВА 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 5](#_Toc9971836)

[1.1. Введение 5](#_Toc9971837)

[1.1.1. Наименование программы 5](#_Toc9971838)

[1.1.2. Назначение и область применения 5](#_Toc9971839)

[1.1.3. Требования к функциональным характеристикам 5](#_Toc9971840)

[1.2. Условия эксплуатации 5](#_Toc9971841)

[1.2.1. Требования к составу и параметрам технических средств 5](#_Toc9971842)

[1.2.2. Требования к информационной и программной совместимости 5](#_Toc9971843)

[1.3. Требования к программной документации 6](#_Toc9971844)

[1.3.1. Предварительный состав программной документации 6](#_Toc9971845)

[1.4. Стадии и этапы разработки 6](#_Toc9971846)

[1.4.1. Стадии разработки 6](#_Toc9971847)

[1.4.2. Этапы разработки 6](#_Toc9971848)

[1.4.3. Содержание работ по этапам 7](#_Toc9971849)

[1.5. Порядок контроля и приемки 7](#_Toc9971850)

[1.5.1. Виды испытаний 7](#_Toc9971851)

[1.5.2. Общие требования к приемке работы 8](#_Toc9971852)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММЫ 8](#_Toc9971853)

[2.1. Проектирование программы 8](#_Toc9971854)

[2.1.1. Функциональные требования к программе 8](#_Toc9971855)

[2.1.2. Проектирование классов 8](#_Toc9971856)

[2.1.3. Описание проектного решения 8](#_Toc9971857)

[Глава 3. Руководство пользователя 11](#_Toc9971858)

[Заключение 13](#_Toc9971859)

[Список литературы 14](#_Toc9971862)

[Приложение №1. Исходный код программы 15](#_Toc9971863)

[Приложение №2. UML-диаграмма 27](#_Toc9971864)

**Цель работы:**

Целью работы является разработать программу, для управления деятельностью агентства по подбору персонала Необходимо разработать программу, для заказа персонала в Агентстве по подбору персонала, с использованием функции поиска нужной услуги по заданным критериям поиска.

# Введение:

В ходе выполнения Курсовой работы должна быть реализована программа для заказа персонала в агентство по подбору персонала, а так же должна быть использована функция поиска нужной услуги по заданным критериям поиска с помощью языка Объективно-Ориентированного ПрограммированияC++. Язык C++ — компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование.

Программное предложение«Программа для заказа персонала в агенство по подбору персонала» представляет собой систему, с помощью которой можновыбрать необходимую услугу и заказать человека, который соответствует вашим критериям поиска.

# ГЛАВА 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## Введение

### Наименование программы

Наименование программы – «Программа для заказа персонала в агентство по подбору персонала».

### Назначение и область применения

«Программа для управления заказа персонала в агентство по подбору персонала» заказа персонала в агентство по подбору персонала, с использованием функции поиска нужной услуги по заданным критериям поиска.

Требования к программе или программному изделию

### Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

* функцию выбора услуги, которая необходима покупателю.
* функции показа всех работников, относящихся к выбранной услуге.
* Функции выбора подходящего работника.
* функции выхода из программы после завершения работы.

## Условия эксплуатации

### Требования к составу и параметрам технических средств

В состав технических средств должен входить IВМ-совместимый персональный компьютер (ПЭВМ), включающий в себя:

* процессор Pentium-2.0Hz, не менее;
* оперативную память объемом, 1 Гигабайт, не менее;
* любую операционную систему;
* стандартный пакет С++;

### Требования к информационной и программной совместимости

#### Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке C++. В качестве интегрированной среды разработки программы должна быть использована среда Qt.

#### Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией любой операционной системы и ПО C++.

## Требования к программной документации

### Предварительный состав программной документации

Состав программной документации должен включать в себя:

* техническое задание;
* программу и методики испытаний;
* руководство пользователя.

## Стадии и этапы разработки

### Стадии разработки

Разработка должна быть проведена в три стадии:

* разработка технического задания;
* исследование предметной области;
* рабочее проектирование;
* внедрение.

### Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

* разработка программы;
* разработка программной документации;
* испытания программы.

На стадии внедрения должен быть выполнен этап разработки - подготовка и передача программы.

### Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

* постановка задачи;
* определение и уточнение требований к техническим средствам;
* определение требований к программе;
* определение стадий, этапов и сроков разработки программы и документации на неё;
* согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

* разработка, согласование и утверждение программы и методики испытаний;
* проведение приемо-сдаточных испытаний;
* корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию.

## Порядок контроля и приемки

### Виды испытаний

Демонстрация работы программы и ее функционала.

### Общие требования к приемке работы

Программа должна соответствовать всем вышесказанным требованиям.

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММЫ

## Проектирование программы

### Функциональные требования к программе

Функциональные требования к программе описаны в пункте № 1.2 Технического задания.

## Проектирование классов

Для реализации проекта будет использован язык программирования C++. Данный язык является одним из самых наиболее распространённых объектно-ориентированных языков программирования (ООП).

Целью данной работы является написание приложения “Программа для заказа персонала в агентство по подбору персонала”. Для удобства пользования все действия будут реализованы через кнопки, находящиеся в графическом окне, разработанном при помощи фреймворкаQt.

## Описание проектного решения

**Plumber** – класс, содержащий имя, фамилию, стаж работы, возраст, поля и методы (Рис. 1.).

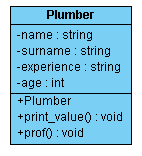


Рис. 1. UML диаграмма класса **Plumber**

**Electrician**– класс, содержащий имя, фамилию, стаж работы, возраст, поля и методы (Рис. 2.).

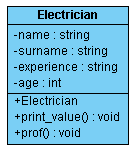
****

Рис. 2. UMLдиаграмма класса **Electrician**

**Driver**– класс, содержащий имя, фамилию, стаж работы, возраст, поля и методы(Рис. 3.).

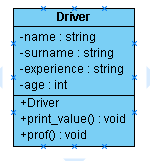
****

Рис. 3. UMLдиаграммакласса**Driver**

**Nanny**– класс, содержащий имя, фамилию, стаж работы, возраст, поля и методы(Рис. 4.).

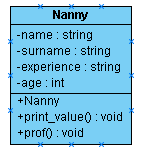
****

Рис. 4. UMLдиаграммакласса**Nanny**

**Tutor**– класс, содержащий имя, фамилию, стаж работы, возраст, поля и методы(Рис. 5.).

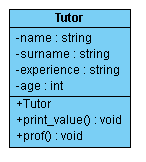
****

Рис. 5. UMLдиаграммакласса**Tutor**

**Waiter**– класс, содержащий имя, фамилию, стаж работы, возраст, поля и методы(Рис. 6.).

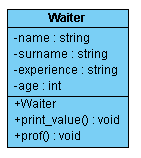
****

Рис. 6. UMLдиаграммакласса**Waiter**

**Designer**– класс, содержащий имя, фамилию, стаж работы, возраст, поля и методы(Рис. 7.).

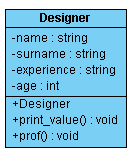
****

Рис. 7. UMLдиаграммакласса**Designer**

**Pharmacist**– класс, содержащий имя, фамилию, стаж работы, возраст, поля и методы(Рис. 8.).

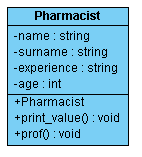
****

Рис. 8. UMLдиаграммакласса**Pharmacist**

**Lawyer**– класс, содержащий имя, фамилию, стаж работы, возраст, поля и методы(Рис. 9.).

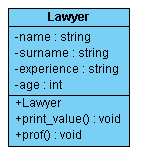
****

Рис. 9. UMLдиаграммакласса**Lawyer**

**Manager**– класс, содержащий имя, фамилию, стаж работы, возраст, поля и методы(Рис. 10.).

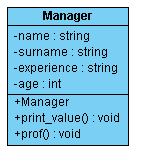
****

Рис. 10. UMLдиаграммакласса**Manager**

# Глава 3. Руководство пользователя

Запуск программы осуществляется через исполняемый файл.exe.

После запуска программы в открытом менювам потребуется ввести необходимую вам услугу (Рис. 11.).

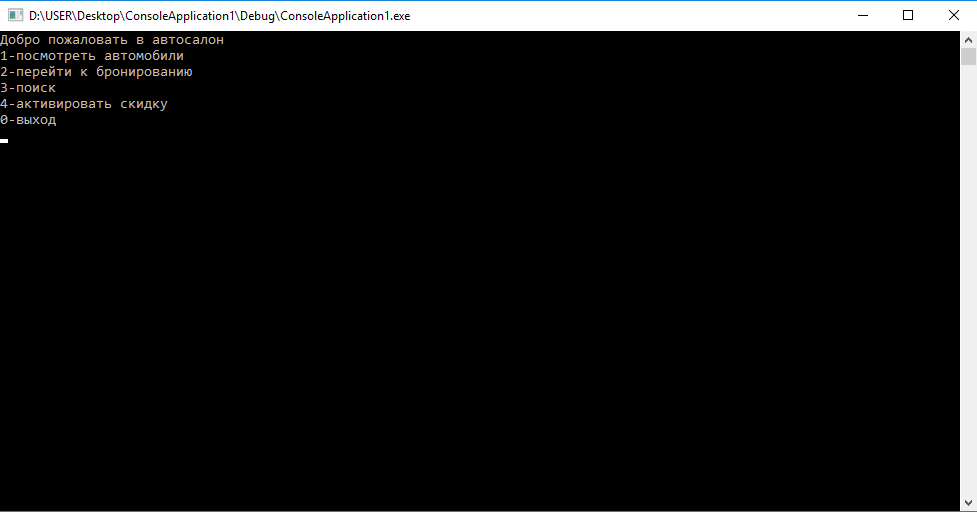


Рис.11. Скриншот окна “ Меню”.

Если вы хотите посмотреть список всех услуг,необходимо ввести цифру 1 (Рис. 12.).

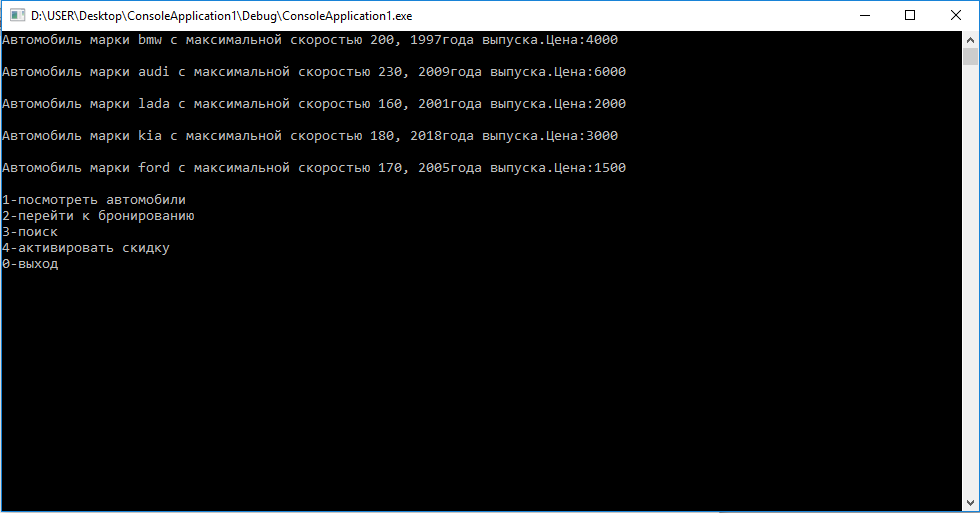


Рис. 12. Скриншот окна “Список всех услуг ”.

После ввода необходимой услуги вы сможете увидеть список работников, относящихся к этой сфере деятельности (Рис. 13.).

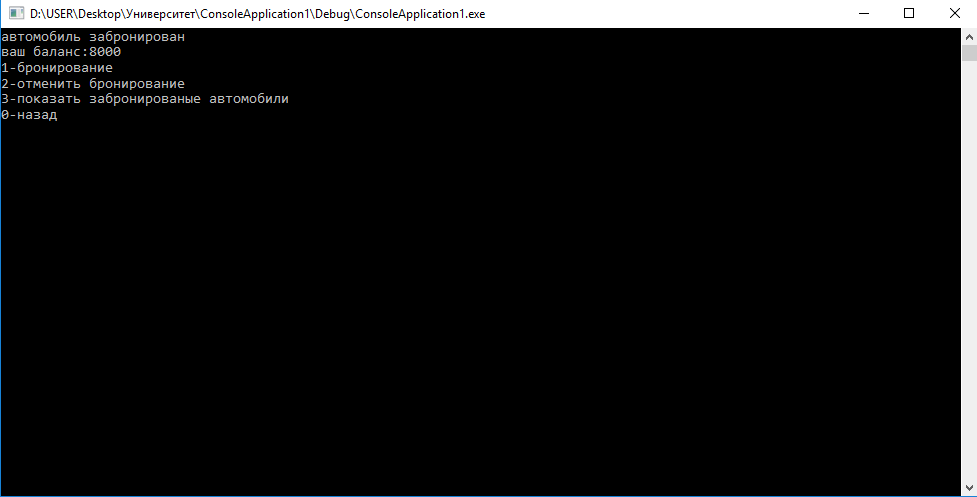


Рис. 13. Скриншот окна “Список работников”.

Если вы определились с подходящим для вас работником, введите его имя (Рис. 14.).

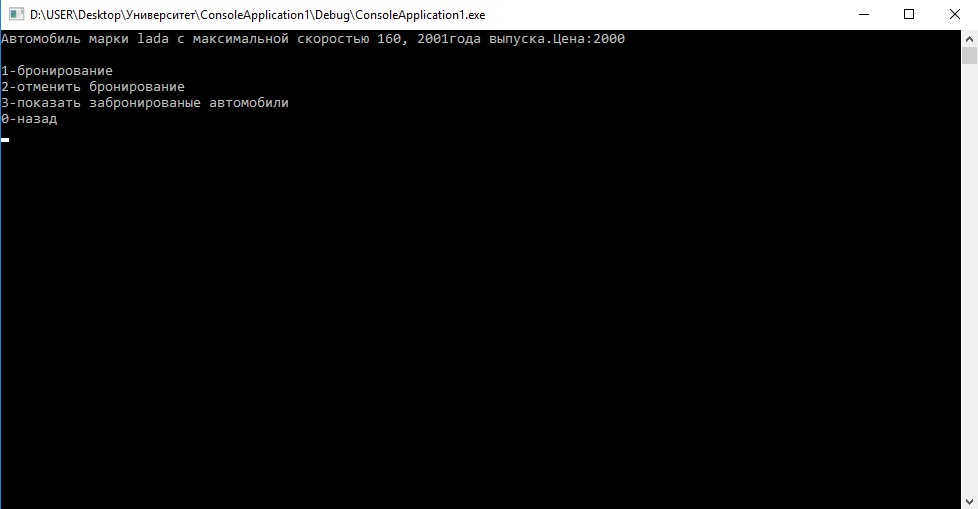


Рис. 14. Скриншот окна “Время прибытия выбранного работника”.

# Заключение

# В результате выполнения курсовой работы была полностью реализована программа «Программа для заказа персонала в агентство по подбору персонала», используя объектно-ориентированное программирование. Написание программы способствовало закреплению теоретического материала на практических занятиях.

# В ходе выполнения курсовой работы были более подробно изучены базовые элементы управления, их свойства и методы

# Список литературы

1. А. Александреску. Современное программирование на C++. Обобщенное программирование и прикладные шаблоны проектирования. Книга для опытных программистов не С++. 2002 год, 330 стр.
2. Т. А. Павловская. C/C++. Программирование на языке высокого уровня. Из серии "300 лучших учебников".2003 год. 461 стр.
3. Прата Стивен. Язык программирования С++. Лекции и упражнения. Учебник. 2005 год. 1100 стр.
4. Бланшет, Саммерфилд - Qt4 Программирование GUI на С++. 2ed. 2008
5. Шлее Макс - Профессиональное программирование на C++. +CD. Qt 4.8. 2012
6. Марк Саммерфилд - Qt Профессиональное программирование (Hightech). 2011

# Приложение №1. Исходный код программы

|  |
| --- |
| #include "stdafx.h" |
|  |

|  |
| --- |
| #include <iostream> |
|  |

|  |
| --- |
| #include <string> |
|  |

|  |
| --- |
| #include <Windows.h> |
|  |

|  |
| --- |
| #include <stdlib.h> |
|  |

|  |
| --- |
| using namespace std; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| class Plumber { |
|  |

|  |
| --- |
| private: |
|  |

|  |
| --- |
| string pr; |
|  |

|  |
| --- |
| string name; |
|  |

|  |
| --- |
| string surname; |
|  |

|  |
| --- |
| string experience; |
|  |

|  |
| --- |
| int age; |
|  |

|  |
| --- |
| public:Plumber() {} |
|  |

|  |
| --- |
| int a; |
|  |

|  |
| --- |
| Plumber(string name, string surname, string experience, int age) { |
|  |

|  |
| --- |
| this->age = age; |
|  |

|  |
| --- |
| this->name = name; |
|  |

|  |
| --- |
| this->surname = surname; |
|  |

|  |
| --- |
| this->experience = experience; |
|  |

|  |
| --- |
| pr = "plumber"; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void print\_value() { |
|  |

|  |
| --- |
| cout << "Имя: " << name << endl; |
|  |

|  |
| --- |
| cout << "Фамилия: " << surname << endl; |
|  |

|  |
| --- |
| cout << "Стаж работы: " << experience << endl; |
|  |

|  |
| --- |
| cout << "Возраст: " << age << endl; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| void prof() { |
|  |

|  |
| --- |
| cout << "plumber" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bool check\_pr(string& a) { |
|  |

|  |
| --- |
| if (a == pr)return true; |
|  |

|  |
| --- |
| else return false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| class Electrician { |
|  |

|  |
| --- |
| string pr; |
|  |

|  |
| --- |
| string name; |
|  |

|  |
| --- |
| string surname; |
|  |

|  |
| --- |
| string experience; |
|  |

|  |
| --- |
| int age; |
|  |

|  |
| --- |
| public:Electrician() {} |
|  |

|  |
| --- |
| int a; |
|  |

|  |
| --- |
| Electrician(string name, string surname, string experience, int age) { |
|  |

|  |
| --- |
| this->age = age; |
|  |

|  |
| --- |
| this->name = name; |
|  |

|  |
| --- |
| this->surname = surname; |
|  |

|  |
| --- |
| this->experience = experience; |
|  |

|  |
| --- |
| pr = "electrician"; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void print\_value() { |
|  |

|  |
| --- |
| cout << "Имя: " << name << endl; |
|  |

|  |
| --- |
| cout << "Фамилия: " << surname << endl; |
|  |

|  |
| --- |
| cout << "Стаж работы: " << experience << endl; |
|  |

|  |
| --- |
| cout << "Возраст: " << age << endl; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| void prof() { |
|  |

|  |
| --- |
| cout << "electrician" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bool check\_pr(string& a) { |
|  |

|  |
| --- |
| if (a == pr)return true; |
|  |

|  |
| --- |
| else return false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| class Driver { |
|  |

|  |
| --- |
| string pr; |
|  |

|  |
| --- |
| string name; |
|  |

|  |
| --- |
| string surname; |
|  |

|  |
| --- |
| string experience; |
|  |

|  |
| --- |
| int age; |
|  |

|  |
| --- |
| public:Driver() {} |
|  |

|  |
| --- |
| int a; |
|  |

|  |
| --- |
| Driver(string name, string surname, string experience, int age) { |
|  |

|  |
| --- |
| this->age = age; |
|  |

|  |
| --- |
| this->name = name; |
|  |

|  |
| --- |
| this->surname = surname; |
|  |

|  |
| --- |
| this->experience = experience; |
|  |

|  |
| --- |
| pr = "driver"; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void print\_value() { |
|  |

|  |
| --- |
| cout << "Имя: " << name << endl; |
|  |

|  |
| --- |
| cout << "Фамилия: " << surname << endl; |
|  |

|  |
| --- |
| cout << "Стаж работы: " << experience << endl; |
|  |

|  |
| --- |
| cout << "Возраст: " << age << endl; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| void prof() { |
|  |

|  |
| --- |
| cout << "driver" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bool check\_pr(string& a) { |
|  |

|  |
| --- |
| if (a == pr)return true; |
|  |

|  |
| --- |
| else return false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| class Nanny { |
|  |

|  |
| --- |
| string name; |
|  |

|  |
| --- |
| string surname; |
|  |

|  |
| --- |
| string experience; |
|  |

|  |
| --- |
| string pr; |
|  |

|  |
| --- |
| int age; |
|  |

|  |
| --- |
| public:Nanny() {} |
|  |

|  |
| --- |
| int a; |
|  |

|  |
| --- |
| Nanny(string name, string surname, string experience, int age) { |
|  |

|  |
| --- |
| this->age = age; |
|  |

|  |
| --- |
| this->name = name; |
|  |

|  |
| --- |
| this->surname = surname; |
|  |

|  |
| --- |
| this->experience = experience; |
|  |

|  |
| --- |
| pr = "nanny"; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void print\_value() { |
|  |

|  |
| --- |
| cout << "Имя: " << name << endl; |
|  |

|  |
| --- |
| cout << "Фамилия: " << surname << endl; |
|  |

|  |
| --- |
| cout << "Стаж работы: " << experience << endl; |
|  |

|  |
| --- |
| cout << "Возраст: " << age << endl; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| void prof() { |
|  |

|  |
| --- |
| cout << "nanny" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bool check\_pr(string& a) { |
|  |

|  |
| --- |
| if (a == pr)return true; |
|  |

|  |
| --- |
| else return false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| class Tutor { |
|  |

|  |
| --- |
| string name; |
|  |

|  |
| --- |
| string surname; |
|  |

|  |
| --- |
| string experience; |
|  |

|  |
| --- |
| string pr; |
|  |

|  |
| --- |
| int age; |
|  |

|  |
| --- |
| public:Tutor() {} |
|  |

|  |
| --- |
| int a; |
|  |

|  |
| --- |
| Tutor(string name, string surname, string experience, int age) { |
|  |

|  |
| --- |
| this->age = age; |
|  |

|  |
| --- |
| this->name = name; |
|  |

|  |
| --- |
| this->surname = surname; |
|  |

|  |
| --- |
| this->experience = experience; |
|  |

|  |
| --- |
| pr = "tutor"; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void print\_value() { |
|  |

|  |
| --- |
| cout << "Имя: " << name << endl; |
|  |

|  |
| --- |
| cout << "Фамилия: " << surname << endl; |
|  |

|  |
| --- |
| cout << "Стаж работы: " << experience << endl; |
|  |

|  |
| --- |
| cout << "Возраст: " << age << endl; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| void prof() { |
|  |

|  |
| --- |
| cout << "tutor" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bool check\_pr(string& a) { |
|  |

|  |
| --- |
| if (a == pr)return true; |
|  |

|  |
| --- |
| else return false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| class Waiter { |
|  |

|  |
| --- |
| string name; |
|  |

|  |
| --- |
| string surname; |
|  |

|  |
| --- |
| string experience; |
|  |

|  |
| --- |
| string pr; |
|  |

|  |
| --- |
| int age; |
|  |

|  |
| --- |
| public:Waiter() {} |
|  |

|  |
| --- |
| int a; |
|  |

|  |
| --- |
| Waiter(string name, string surname, string experience, int age) { |
|  |

|  |
| --- |
| this->age = age; |
|  |

|  |
| --- |
| this->name = name; |
|  |

|  |
| --- |
| this->surname = surname; |
|  |

|  |
| --- |
| this->experience = experience; |
|  |

|  |
| --- |
| pr = "waiter"; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void print\_value() { |
|  |

|  |
| --- |
| cout << "Имя: " << name << endl; |
|  |

|  |
| --- |
| cout << "Фамилия: " << surname << endl; |
|  |

|  |
| --- |
| cout << "Стаж работы: " << experience << endl; |
|  |

|  |
| --- |
| cout << "Возраст: " << age << endl; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| void prof() { |
|  |

|  |
| --- |
| cout << "waiter" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bool check\_pr(string& a) { |
|  |

|  |
| --- |
| if (a == pr)return true; |
|  |

|  |
| --- |
| else return false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| class Designer { |
|  |

|  |
| --- |
| string name; |
|  |

|  |
| --- |
| string surname; |
|  |

|  |
| --- |
| string experience; |
|  |

|  |
| --- |
| string pr; |
|  |

|  |
| --- |
| int age; |
|  |

|  |
| --- |
| public:Designer() {} |
|  |

|  |
| --- |
| int a; |
|  |

|  |
| --- |
| Designer(string name, string surname, string experience, int age) { |
|  |

|  |
| --- |
| this->age = age; |
|  |

|  |
| --- |
| this->name = name; |
|  |

|  |
| --- |
| this->surname = surname; |
|  |

|  |
| --- |
| this->experience = experience; |
|  |

|  |
| --- |
| pr = "designer"; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void print\_value() { |
|  |

|  |
| --- |
| cout << "Имя: " << name << endl; |
|  |

|  |
| --- |
| cout << "Фамилия: " << surname << endl; |
|  |

|  |
| --- |
| cout << "Стаж работы: " << experience << endl; |
|  |

|  |
| --- |
| cout << "Возраст: " << age << endl; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| void prof() { |
|  |

|  |
| --- |
| cout << "designer" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bool check\_pr(string& a) { |
|  |

|  |
| --- |
| if (a == pr)return true; |
|  |

|  |
| --- |
| else return false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| class Pharmacist { |
|  |

|  |
| --- |
| string name; |
|  |

|  |
| --- |
| string surname; |
|  |

|  |
| --- |
| string experience; |
|  |

|  |
| --- |
| string pr; |
|  |

|  |
| --- |
| int age; |
|  |

|  |
| --- |
| public:Pharmacist() {} |
|  |

|  |
| --- |
| int a; |
|  |

|  |
| --- |
| Pharmacist(string name, string surname, string experience, int age) { |
|  |

|  |
| --- |
| this->age = age; |
|  |

|  |
| --- |
| this->name = name; |
|  |

|  |
| --- |
| this->surname = surname; |
|  |

|  |
| --- |
| this->experience = experience; |
|  |

|  |
| --- |
| pr = "pharmasist"; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void print\_value() { |
|  |

|  |
| --- |
| cout << "Имя: " << name << endl; |
|  |

|  |
| --- |
| cout << "Фамилия: " << surname << endl; |
|  |

|  |
| --- |
| cout << "Стаж работы: " << experience << endl; |
|  |

|  |
| --- |
| cout << "Возраст: " << age << endl; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| void prof() { |
|  |

|  |
| --- |
| cout << "pharmasist" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bool check\_pr(string& a) { |
|  |

|  |
| --- |
| if (a == pr)return true; |
|  |

|  |
| --- |
| else return false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| class Lawyer { |
|  |

|  |
| --- |
| string name; |
|  |

|  |
| --- |
| string surname; |
|  |

|  |
| --- |
| string experience; |
|  |

|  |
| --- |
| string pr; |
|  |

|  |
| --- |
| int age; |
|  |

|  |
| --- |
| public:Lawyer() {} |
|  |

|  |
| --- |
| int a; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Lawyer(string name, string surname, string experience, int age) { |
|  |

|  |
| --- |
| this->age = age; |
|  |

|  |
| --- |
| this->name = name; |
|  |

|  |
| --- |
| this->surname = surname; |
|  |

|  |
| --- |
| this->experience = experience; |
|  |

|  |
| --- |
| pr = "lawyer"; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void print\_value() { |
|  |

|  |
| --- |
| cout << "Имя: " << name << endl; |
|  |

|  |
| --- |
| cout << "Фамилия: " << surname << endl; |
|  |

|  |
| --- |
| cout << "Стаж работы: " << experience << endl; |
|  |

|  |
| --- |
| cout << "Возраст: " << age << endl; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| void prof() { |
|  |

|  |
| --- |
| cout << "lawyer" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bool check\_pr(string& a) { |
|  |

|  |
| --- |
| if (a == pr)return true; |
|  |

|  |
| --- |
| else return false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| class Manager { |
|  |

|  |
| --- |
| string name; |
|  |

|  |
| --- |
| string surname; |
|  |

|  |
| --- |
| string experience; |
|  |

|  |
| --- |
| string pr; |
|  |

|  |
| --- |
| int age; |
|  |

|  |
| --- |
| public:Manager() {} |
|  |

|  |
| --- |
| int a; |
|  |

|  |
| --- |
| Manager(string name, string surname, string experience, int age) { |
|  |

|  |
| --- |
| this->age = age; |
|  |

|  |
| --- |
| this->name = name; |
|  |

|  |
| --- |
| this->surname = surname; |
|  |

|  |
| --- |
| this->experience = experience; |
|  |

|  |
| --- |
| pr = "manager"; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void print\_value() { |
|  |

|  |
| --- |
| cout << "Имя: " << name << endl; |
|  |

|  |
| --- |
| cout << "Фамилия: " << surname << endl; |
|  |

|  |
| --- |
| cout << "Стаж работы: " << experience << endl; |
|  |

|  |
| --- |
| cout << "Возраст: " << age << endl; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| void prof() { |
|  |

|  |
| --- |
| cout << "manager" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bool check\_pr(string& a) { |
|  |

|  |
| --- |
| if (a == pr)return true; |
|  |

|  |
| --- |
| else return false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| int main(int argc, char const\* argv[]) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| setlocale(LC\_ALL, "Russian"); |
|  |

|  |
| --- |
| int b; |
|  |

|  |
| --- |
| Plumber a1("Иван", "Кузнецов", "12 лет", 36); |
|  |

|  |
| --- |
| Electrician a2("Николай", "Юдин", "8 лет", 30); |
|  |

|  |
| --- |
| Driver a3("Даниил", "Крюков", "15 лет", 40); |
|  |

|  |
| --- |
| Nanny a4("Евгения", "Баженова", "11 лет", 48); |
|  |

|  |
| --- |
| Waiter a5("Ксения", "Кулакова", "7 лет", 31); |
|  |

|  |
| --- |
| Tutor a6("Галина", "Лукина", "20 лет", 50); |
|  |

|  |
| --- |
| Designer a7("Стас", "Грачев", "10 лет", 34); |
|  |

|  |
| --- |
| Pharmacist a8("Анастасия", "Костина", "7 лет", 50); |
|  |

|  |
| --- |
| Lawyer a9("Алексей", "Бородин", "15 лет", 43); |
|  |

|  |
| --- |
| Manager a10("Глеб", "Литвинов", "9 лет", 40); |
|  |

|  |
| --- |
| do { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| cout << "Пожалуйста,введите 1 чтобы посмотреть весь список услуг\n" << "Введить 0 для выхода" << endl; |
|  |

|  |
| --- |
| cin >> b; |
|  |

|  |
| --- |
| system("cls"); |
|  |

|  |
| --- |
| switch (b) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| case 1: { |
|  |

|  |
| --- |
| a1.prof(); |
|  |

|  |
| --- |
| a2.prof(); |
|  |

|  |
| --- |
| a3.prof(); |
|  |

|  |
| --- |
| a4.prof(); |
|  |

|  |
| --- |
| a5.prof(); |
|  |

|  |
| --- |
| a6.prof(); |
|  |

|  |
| --- |
| a7.prof(); |
|  |

|  |
| --- |
| a8.prof(); |
|  |

|  |
| --- |
| a9.prof(); |
|  |

|  |
| --- |
| a10.prof(); |
|  |

|  |
| --- |
| string q; |
|  |

|  |
| --- |
| cout << "введите название услуги" << endl; |
|  |

|  |
| --- |
| cin >> q; |
|  |

|  |
| --- |
| if (a1.check\_pr(q))a1.print\_value(); else |
|  |

|  |
| --- |
| if (a2.check\_pr(q))a2.print\_value(); else |
|  |

|  |
| --- |
| if (a3.check\_pr(q))a3.print\_value(); else |
|  |

|  |
| --- |
| if (a4.check\_pr(q))a4.print\_value(); else |
|  |

|  |
| --- |
| if (a5.check\_pr(q))a5.print\_value(); else |
|  |

|  |
| --- |
| if (a6.check\_pr(q))a6.print\_value(); else |
|  |

|  |
| --- |
| if |
|  |

|  |
| --- |
| (a7.check\_pr(q))a7.print\_value(); else |
|  |

|  |
| --- |
| if (a8.check\_pr(q))a8.print\_value(); else |
|  |

|  |
| --- |
| if (a9.check\_pr(q))a9.print\_value(); else |
|  |

|  |
| --- |
| if (a10.check\_pr(q))a10.print\_value(); else |
|  |

|  |
| --- |
| cout << "не найдено" << endl; |
|  |

|  |
| --- |
| if (a1.check\_pr(q)) cout <<"Спасибо что выбрали нашу компанию\n" << "Иван Кузнецов выйдет на работу в течении следующих 24 часов" << endl; else |
|  |

|  |
| --- |
| if (a2.check\_pr(q)) cout << "Спасибо что выбрали нашу компанию\n" << "Николай Юдин выйдет на работу в течении следующих 24 часов" << endl; else |
|  |

|  |
| --- |
| if(a3.check\_pr(q)) cout << "Спасибо что выбрали нашу компанию\n" << "Даниил Крюков выйдет на работу в течении следующих 24 часов" << endl; else |
|  |

|  |
| --- |
| if(a4.check\_pr(q)) cout << "Спасибо что выбрали нашу компанию\n" << "Евгения Баженова выйдет на работу в течении следующих 24 часов" << endl; else |
|  |

|  |
| --- |
| if(a5.check\_pr(q)) cout << "Спасибо что выбрали нашу компанию\n" << "Ксения Кулакова выйдет на работу в течении следующих 24 часов" << endl; else |
|  |

|  |
| --- |
| if(a6.check\_pr(q)) cout << "Спасибо что выбрали нашу компанию\n" << "Галина Лукина выйдет на работу в течении следующих 24 часов" << endl; else |
|  |

|  |
| --- |
| if(a7.check\_pr(q)) cout << "Спасибо что выбрали нашу компанию\n" << "Стас Грачев выйдет на работу в течении следующих 24 часов" << endl; else |
|  |

|  |
| --- |
| if(a8.check\_pr(q)) cout << "Спасибо что выбрали нашу компанию\n" << "Анастасия Костина выйдет на работу в течении следующих 24 часов" << endl; else |
|  |

|  |
| --- |
| if(a9.check\_pr(q)) cout << "Спасибо что выбрали нашу компанию\n" << "Алексей Бородин выйдет на работу в течении следующих 24 часов" << endl; else |
|  |

|  |
| --- |
| if(a10.check\_pr(q)) cout << "Спасибо что выбрали нашу компанию\n" << "Глеб Литвинов выйдет на работу в течении следующих 24 часов" << endl; |
|  |

|  |
| --- |
| cout << "Введите 0 для выхода" << endl; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| default: |
|  |

|  |
| --- |
| break; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } while (b != 0); |
|  |

}

# Приложение №2. UML-диаграмма

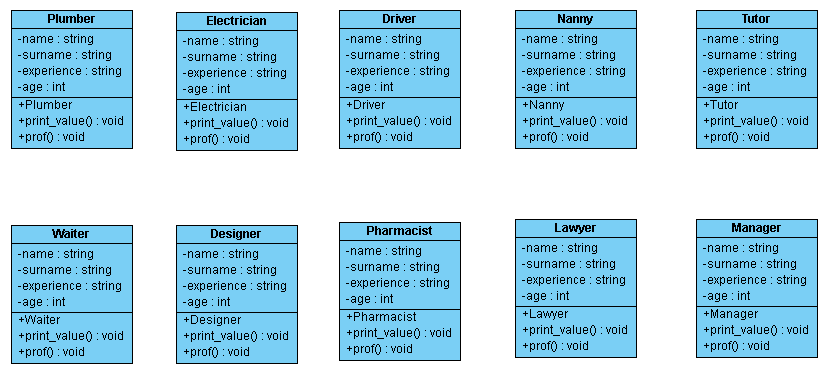


Рис. 15. Общая UML диаграмма