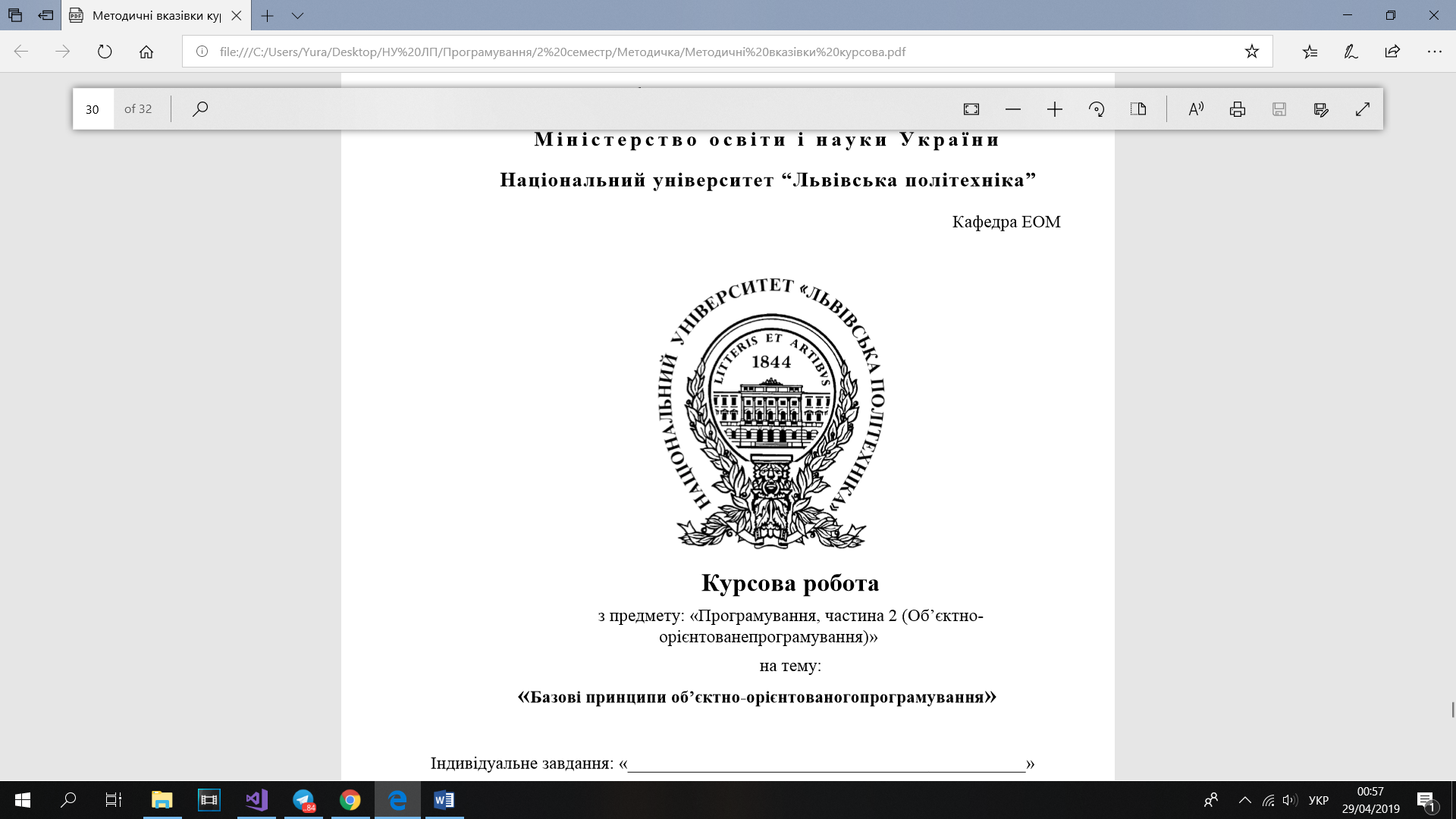
Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



**Курсова робота**

з предмету: «Програмування, частина 2 (Об’єктно-орієнтоване програмування)» на тему: «Базові принципи об’єктно-орієнтованого програмування»

**Індивідуальне завдання: «ІТ-компанія»**

Виконав:

ст. гр. КІ-15

Аркавенко Л.С.

Прийняв:

Козак Н.Б.

Львів-2019

**Анотація**

У цій курсовій роботі було виконано написання програмного забезпечення на мові програмування С++ з використанням основних принципів об’єктно-орієнтованого програмування. Програма являє собою спрощену модель функціонування ІТ-компанії та включає у себе роботу з текстовими файлами, а саме: запис, зчитування, пошук, видалення та додавання нових даних, а також сортування, за певною характеристикою, яке виводиться на екран. Програма має кілька рівнів доступу, які спрощують її використання, оскільки можливо одразу визначити, який саме користувач увійшов у систему та які операції він може виконувати. Інтерфейс програмного забезпечення написаний англійською мовою.

**ЗМІСТ**

[**ВСТУП** 4](#_Toc41261445)

[**РОЗДІЛ 1** 5](#_Toc41261446)

[**Визначення поняття Об’єктно орієнтованого програмування(ООП), його основні принципи та обгрунтування його технологій.** 5](#_Toc41261447)

[**1.1. ООП як основа сучасного програмування** 5](#_Toc41261448)

[**1.2. Визначення ООП** 5](#_Toc41261449)

[**1.3. Фундаментальні поняття ООП** 6](#_Toc41261450)

[**1.4. Переваги вибраного методу** 7](#_Toc41261451)

[**РОЗДІЛ 2. Аналіз та розробка алгоритму згідно індивідуального завдання** 8](#_Toc41261452)

[**2.1. Загальний опис поставленої задачі** 8](#_Toc41261453)

[**2.2 Процес взаємодії розробника та споживача ІТ-продукту** 10](#_Toc41261454)

[**2.3. Визначення основних задач покладених в основу програми** 12](#_Toc41261455)

[**2.4.** **Алгоритм виконання роботи програми** 13](#_Toc41261456)

[**РОЗДІЛ 3. Реалізація алгоритму у програмному коді та тестування програми** 16](#_Toc41261457)

[**Висновок** 24](#_Toc41261458)

[**Список використаної літератури** 25](#_Toc41261459)

[**ДОДАТОК А (Лістинг програмного коду проекту)** 26](#_Toc41261460)

**ВСТУП**

Об'є́ктно-орієнто́ване програмува́ння (ООП) — одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Основу ООП складають три основні концепції: інкапсуляція, успадкування та поліморфізм. Одною з переваг ООП є краща модульність програмного забезпечення (тисячу функцій процедурної мови, в ООП можна замінити кількома десятками класів із своїми методами). Попри те, що ця парадигма з'явилась в 1960-тих роках, вона не мала широкого застосування до 1990-тих, коли розвиток комп'ютерів та комп'ютерних мереж дозволив писати надзвичайно об'ємне і складне програмне забезпечення, що змусило переглянути підходи до написання програм. Однією з таких мов програмування стала мова С++. У 1990-х р. дана мова стала однією з найуживаніших мов програмування загального призначення. ЇЇ розробником є професор Техаського університету А&М Б'ярн Страуструп. На початку свого існування ця мова дістала назву «Сі з класами». Згодом Страуструп перейменував мову на C++ у 1983 р. У 1998 році ратифіковано міжнародний стандарт мови С++: ISO/IEC 14882 «Standard for the C++ Programming Language». Поточна версія цього стандарту — ISO/IEC 14882:2011. На сьогоднішній день цю мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм таких як відеоігри. Також мова С++ суттєво вплинула на інші, популярні сьогодні, мови програмування такі як С# та Java.

Метою виконання цієї курсової роботи є засвоєння основних методів та алгоритмів об’єктно-орієнтованого програмування мови С++, та практичне застосування їх у цій сфері.

**РОЗДІЛ 1**

**Визначення поняття Об’єктно орієнтованого програмування(ООП), його основні принципи та обгрунтування його технологій.**

**1.1. ООП як основа сучасного програмування**

Із розвитком комп’ютерних технологій, коли програми поступово почали ускладнюватися, виникла потреба для нових засобів програмування, які б могли впорядкувати та спростити сприйняття та написання програм, які могли б містити тисячі стрічок коду. Для подолання цієї проблеми було розроблено структурне програмування. Структурне програмування, на відміну від його попередника — стихійного програмування, має певні правила та засоби, які суттєво вдосконалили процес написання програм. Але за допомогою структурного програмування дуже важко писати програми ще більшої складності, які вимагає сучасність. Тому була створена парадигма Об’єктно-орієнтованого програмування. Першою мовою програмування, в якій були запропоновані основні поняття, які згодом склалися в парадигму, була Симула, але термін «об'єктна орієнтованість" не використовувався в контексті використання цієї мови. У момент його появи в 1967 році в ньому були запропоновані революційні ідеї: об'єкти, класи, віртуальні методи і ін., проте це все не було сприйнято сучасниками як щось грандіозне. Фактично, Симула була «Алголом з класами», що спрощує вираз в процедурному програмуванні багатьох складних концепцій. Поняття класу в Симулі може бути повністю визначено через композицію конструкцій Алгола (тобто клас в Симулі - це щось складне, що описується за допомогою примітивів).

Погляд на програмування «під новим кутом» (відмінним від процедурного) запропонували Алан Кей і Ден Інгаллс в мові Smalltalk. Тут поняття класу стало основотвірною ідеєю для всіх інших конструкцій мови (тобто клас в Смолток є примітивом, за допомогою якого описані більш складні конструкції). Саме ця мова стала першою широко поширеною об'єктно-орієнтованою мовою програмування.

В даний час кількість прикладних мов програмування (список мов), що реалізують об'єктно-орієнтовану парадигму, є найбільшою по відношенню до інших парадигм. Найбільш поширені в промисловості такі мови як С ++, Delphi, C #, Java які втілюють об'єктну модель Симула. Прикладами мов, що спираються на модель Смолтока, є Objective-C, Python, Ruby.

**1.2. Визначення ООП**

Об'єктно-орієнтоване програмування (ООП) - методологія програмування, заснована на представленні програми у вигляді сукупності об'єктів, кожен з яких є екземпляром певного класу, а класи утворюють ієрархію спадкування. Лука Карделла і Мартін Абаді побудували теоретичне обгрунтування ООП і класифікацію на основі цього обгрунтування. Вони відзначають, що виділені ними поняття і категорії разом зустрічаються далеко не у всіх ОО-мовах, більшість мов підтримують лише підмножини теорії, а часом і своєрідні відхилення від неї.

Ключові поняття:

• квантифікація змінних типу (універсальна, екзистенційна, обмежена);

• підтипізація (англ. subtyping - відносини «супертип-підтип»);

• включення (англ. subsumption) - окремий випадок підтипізації;

• об'єкт;

• об'єктний тип (фактично, спеціальна форма «запису», в число полів якої входять функції першого класу);

• клас.

Клас описує абстрактну поведінку. Об'єктні типи будуються на основі класу за допомогою додавання різних приватних полів і методів. Об'єкт (тобто значення об'єктного типу, традиційно називається «екземпляром класу») породжується конструктором на основі початкових параметрів.

**1.3. Фундаментальні поняття ООП**

До фундаментальних понять (принципів), використаних у переважній більшості визначень об'єктно-орієнтованого програмування належать:

• Абстракція даних – виділення значущої інформації і виключення її з розгляду незначною. В ООП розглядають лише абстракцію даних (нерідко називаючи її просто «абстракцією»), маючи на увазі набір значущих характеристик об'єкта, доступних решті частини програми.

• Інкапсуляція – властивість системи, що дозволяє об'єднати дані і методи, що працюють з ними, в класі. Одні мови (наприклад, С ++, Java або Ruby) ототожнюють інкапсуляцію з приховуванням, але інші (Smalltalk, Eiffel, OCaml) розрізняють ці поняття.

• Спадкування – властивість системи, що дозволяє описати новий клас на основі вже існуючого з частково або повністю запозичиною функціональністю. Клас, від якого відбувається спадкування, називається базовим, батьківським або суперкласом. Новий клас – нащадком, спадкоємцем, дочірнім або похідним класом.

• Поліморфізм підтипів – властивість системи, що дозволяє використовувати об'єкти з однаковим інтерфейсом без інформації про тип і внутрішню структуру об'єкта. Інший вид поліморфізму – параметричний - в ООП називають узагальненим програмуванням.

• Клас – абстрактні характеристики деякої сутності, включаючи характеристики самої сутності (її атрибути або властивості) та дії, які вона здатна виконувати (її поведінки, методи або можливості). Класи вносять модульність та структурованість в об'єктно-орієнтовану програму. Як правило, клас має бути зрозумілим для не-програмістів, що знаються на предметній області, що, у свою чергу, значить, що клас повинен мати значення в контексті. Також, код реалізації класу має бути досить самодостатнім. Властивості та методи класу, разом називаються його членами.

• Об’єкт – сутність в адресному просторі обчислювальної системи, що з'являється при створенні екземпляра класу (наприклад, після запуску результатів компіляції і зв'язування вихідного коду на виконання).

**1.4. Переваги вибраного методу**

Від будь-якого методу програмування ми чекаємо, що він допоможе нам у вирішенні наших проблем. Але однією з найбільш значних проблем у програмуванні є складність. Чим більша і складніша програма, тим важливіше стає розбити її на невеликі, чітко окреслені частини. Щоб побороти складність, ми повинні абстрагуватися від дрібних деталей. У цьому сенсі класи являють собою дуже зручний інструмент. Класи дозволяють проводити конструювання з корисних компонент, що володіють простими інструментами, що дає можливість абстрагуватися від деталей реалізації. Також ООП дає можливість створювати розширювані системи. Це одна з найбільш значних переваг ООП і саме вона відрізняє даний підхід від традиційних методів програмування. Розширюваність означає, що існуючу систему можна змусити працювати з новими компонентами, причому без внесення до неї будь-яких змін. Компоненти можуть бути додані на етапі виконання. Багаторазового використання програмного забезпечення на практиці досягти не вдається з-за того, що існуючі компоненти вже не відповідають новим вимогам. ООП допомагає досягти цього без порушення роботи вже наявних клієнтів, що дозволяє нам витягти максимум з багаторазового використання компонент. Завдяки чому ми скорочуємо час на розробку, яке з вигодою може бути віддано іншим проектам. Також компоненти багаторазового використання звичайно містять набагато менше помилок, ніж знову розроблені, адже вони вже не раз піддавалися перевірці.

**РОЗДІЛ 2. Аналіз та розробка алгоритму згідно індивідуального завдання**

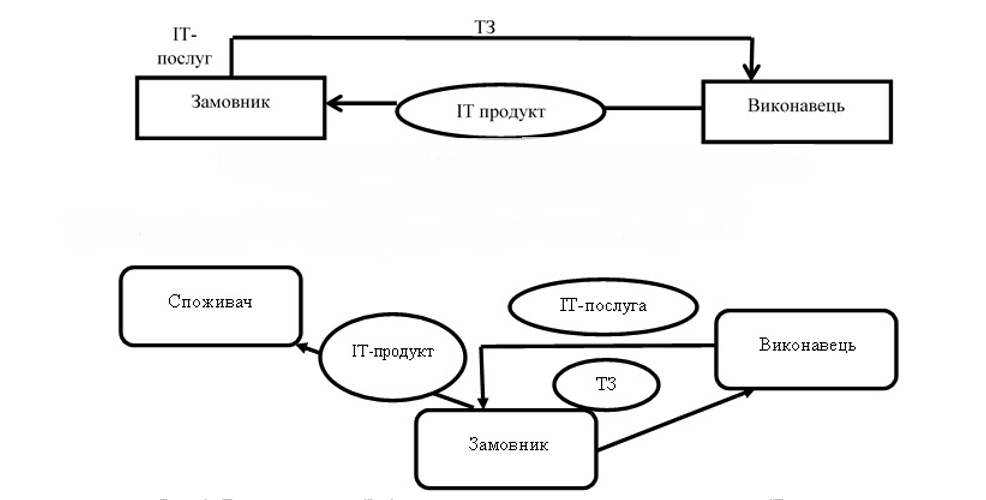
**2.1. Загальний опис поставленої задачі**

Задачею мого курсового проекту є ІТ-компанія. Це поняття можна досить широко розглядати як сукупність багатьох важливих складових, зокрема сама система споживачів програмних продуктів ділиться на чотири рівні.

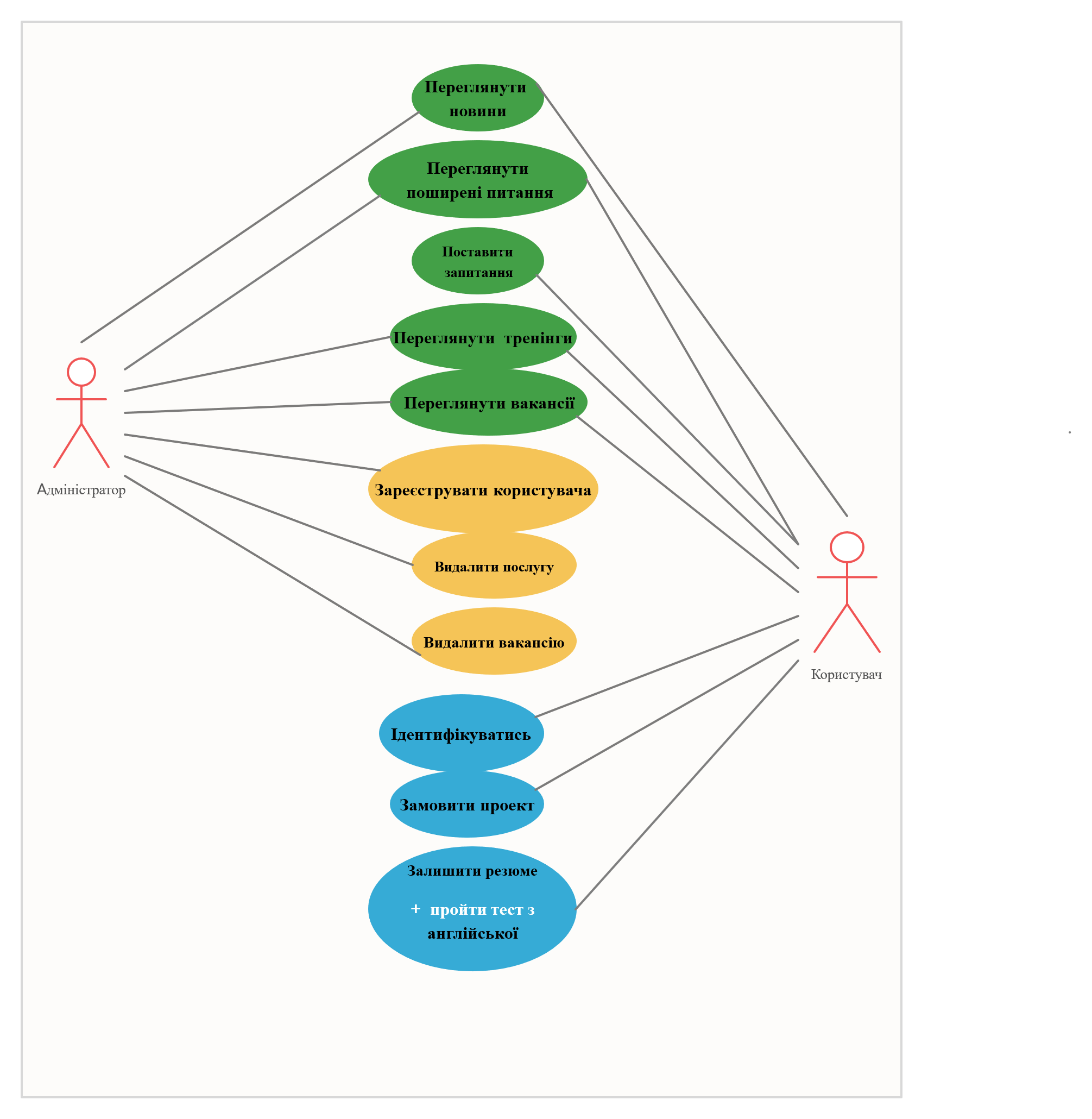
|  |  |
| --- | --- |
| **Рівень** | **Споживачі програмних продуктів** |
| Перший | Користувач |
| Другий | Інженерно-технічний склад (програмісти- розробники) |
| Третій | Провідні фахівці підрозділів |
| Четвертий | Керівники підприємств, їхні заступники за напрямками |

Надаючи послуги, IT-консалтингова організація аналізує бізнес-завдання (потреби) клієнта крізь "призму" інформаційних технологій.

Складність ІТ-послуги може залежати від багатьох чинників, зокрема, специфіки діяльності споживача, кількості учасників процесу надання ІТ-послуги, характеру їх взаємодії тощо. Придбання і використання товарів IT-ринку висуває особливі вимог до знань споживачів. Часто споживачеві ІТ-послуг не вистачає знань в галузі ІТ-технологій щоб кваліфіковано сформулювати свій запит, зробити вибір ІТ-послуги, доцільно визначити її склад та вибрати виконавця.



Проте, виходячи з базових понять про взаємодію працівника і користувача продукту ІТ-компаній, за допомогою діаграми прецедентів можна зобразити залежність вищевказаних об’єктів на прикладі написаної мною програми даним чином.



«Діаграма прецедентів»

**2.2 Процес взаємодії розробника та споживача ІТ-продукту**

Досвід робити ІТ-компаній показує, що вкрай важко реалізувати абсолютно «закінчений» ІТпродукт через, те що виявити і усунути недоліки якості IT-продукту, зокрема помилки кодування, набагато складніше ніж в інших областях виробництва. З метою виявлення та аналізу особливостей маркетингового забезпечення, і зокрема, просування ІТ-послуг розглянемо процеси взаємодії споживача та розробника ІТ-продукту в процесі надання ІТ-послуги. Для цього простежимо у часі процес створення закінченого ІТ-продукту.

Рисунок відображає процес надання ІТ-послуги та розробки ІТ-продукту в часі від усвідомлення потреби споживачем до супроводу після процесу впровадження. На першому етапі «Підготовка» споживач усвідомлює власну потребу та займається пошуком виконавця. На етапі «Переговори» укладається договір з обраним виконавцем, формулюються цілі та завдання проекту, виділяються базові сутності та взаємозв’язку між ними. Тобто, створюється основа для подальшого проектування системи. Також визначаються терміни і вартість розробки

ІТ-продукту, формується і підписується ТЗ на розробку програмного забезпечення. В ході третього етапу, базуючись на результатах попередньої роботи, проводиться проектування системи. Методологія проектування поєднує в собі сучасні програмні рішення. Під час проектування розробляється функціональна специфікація: вибирається архітектура системи, обумовлюються вимоги до апаратного забезпечення, визначається набір організаційних заходів, які необхідні для впровадження ІТ-продукту, а також перелік документів, що регламентують його використання. На етапі «Створення» будуються прототипи, як цілої системи, так і її частин. У результаті з’являється робоча версія продукту. Проходить тестування системи на відповідність вимог до неї, перевірку оформлення і наявність необхідного пакету документації. Результатом тестування є усунення всіх недоліків системи і висновок про її якість. Етап «Впровадження» ІТ-продукту зазвичай передбачає установку системи, навчання користувачів та експлуатацію створеного програмного продукту. До будь-якої розробки додається повний пакет документації, який включає в себе опис системи, посібники та алгоритми роботи. Підтримка функціонування ІТ-продукта повинна здійснюватися групою технічної підтримки розробника, яка здатна відреагувати на побажання в зміні продукту, усунення недоліків в процесі використання, оновлення тощо. Слід зазначити, що процес надання ІТ-послуги, або процес розробки ІТ-продукту є творчим процесом, але в свою чергу потребує системного підходу, щоб направляти процеси, необхідні для отримання нового продукту на ринку та задовольнити потреби споживача. Далі з метою виявлення та аналізу особливостей маркетингового забезпечення, і зокрема, просування ІТ-послуг визначимо та проаналізуємо деякі особливості притаманні ІТ - послугам. В силу нематеріальності послуги споживачеві складно оцінити її якість та отримані ним результати не тільки до отримання ІТ-послуги, але і протягом певного часу після цього. Це зумовлено тим, що корисність складної ІТ-послуги та її складових не очевидна споживачеві не тільки до замовлення, в процесі купівлі та надання ІТ-послуги, але навіть і після її надання. І тільки через певний проміжок часу отримання споживацького досвіду споживач стає спроможним оцінити своє задоволення та, відповідно, якість отриманої послуги. Чим складніша і технологічніша ІТ-послуга, тим важче споживачеві оцінити її якість та свою задоволеність отриманим. Складність ІТ-послуги може залежати від багатьох чинників, зокрема, специфіки діяльності споживача, кількості учасників процесу надання ІТ-послуги, характеру їх взаємодії тощо. Придбання і використання товарів IT-ринку висуває особливі вимог до знань споживачів. Часто споживачеві ІТ-послуг не вистачає знань в галузі ІТ-технологій щоб кваліфіковано сформулювати свій запит, зробити вибір ІТ-послуги, доцільно визначити її склад та вибрати виконавця. Це відбивається на вимогах до маркетингового комунікаційного забезпечення ІТ-послуг.

Зокрема на етапах: усвідомлення потреби, формування вимог, підбору виконавця, переговорів, підписання контракту. Іншою характерною особливістю є низька ступінь залучення споживача в процес створення ІТ-продукта. Властивість проявляється на початкових етапах виробництва ІТ-послуги, де її створенням займаються фахівці-розробники. Однак наступні етапи передбачають збільшення ступеня залучення споживача, так як установка, настройка і адаптація ПЗ здійснюються при постійній взаємодії замовника та виконавця. Необхідність в покупці ІТ-послуг у клієнта з’являється тільки після додаткового інформування клієнта про кваліфікацію розробника або демонстрації попередніх подібних виконаних робіт (портфоліо) в процесі підбору виконавця. Наступною особливістю притаманною ІТ-послугам варто зазначити можливість віддаленого взаємодії споживача з ІТ-компанією. Це позбавляє споживача необхідності відвідувати сервісне підприємство, і він може ніколи не зустрітися особисто з обслуговуючою компанією. Важливим є те, що кількість контактів між споживачем і виробником ІТ-послуги скорочується, а взаємодія відбувається в основному по телефону, Скайпом, електронною поштою. Результат такої сервісної діяльності залишається надзвичайно важливим, але сам процес надання послуги може бути частково і навіть повністю прихований від споживача через віддалений технологічний супровід, консультування і навчання, які є поширеними на ринку ІТ.

**2.3. Визначення основних задач покладених в основу програми**

Дана програма написана для вирішення певного роду задач які повністю пов’язані з такими об’єктами ІТ-компанії як користувачі та працівники. Основною задачею, яку повинна виконувати ця програма є операції над замовленням проекту, рекрутингом, реєстрацією на тренінги, організовані компанією, зворотнім зв’язком та ознайомленням з інформацією про ІТ-компанію. В результаті я отримав набір функцій, які мені потрібно реалізувати для виконання поставленої перед мною задачі:

• Запис і зберігання даних у файлах;

• Зчитування даних з файлу;

• Додавання або видалення даних;

• Сортування даних для виведення їх екран;

• Пошук потрібних даних по структурованих текстових файлах.

Замовлення самого проекту може реалізовуватись двома способами:

1. Зі сталою ціною, що встановила компанія. Це включає можливість швидкого укладення договору і виключно прозору співпрацю.
2. З ціною, яка укладається в процесі переговорів та уточнення всіх деталей стосовно проекту, якщо вони є, для більшого розуміння працівників чого прагне клієнт. Даний спосіб включає можливість детально описати задачу поставлену до виконання та власне надає обізнаність розробнику в процесі виконання.

Опції адміністратора відносно даної програми:

1. Реєструвати користувачів.
2. Видаляти старі вакансії та додавати нові.
3. Видаляти послуги, які вже не актуальні.
4. Оновлювати новини.

Опції користувача програми:

1. Можлива реєстрація та вхід
2. Замовлення проекту
3. Перегляд інформації
4. Реєстрація на тренінги(курси)
5. Можна подати заявку на вакансію

Заявка на вакансію включає ввід наступних даних:

1. ПІП
2. Вакансія, на яку людина претендує
3. Вік
4. Email

Також є можливість зареєструватись на тренінги, запропоновані компанією :

1. NET
2. JAVA
3. PHP
4. DevOps
5. JavaScript
6. UX Design

Реєстрація на тренінг можлива лише за умови успішного проходження тесту з англійської мови, оскільки це є невід'ємною частиною навчання на курсах.

* 1. **Алгоритм виконання роботи програми**

**Алгоритм** — це точний і зрозумілий опис послідовності дій над заданими об’єктами, що дозволяє одержати кінцевий результат.

Властивості алгоритмів

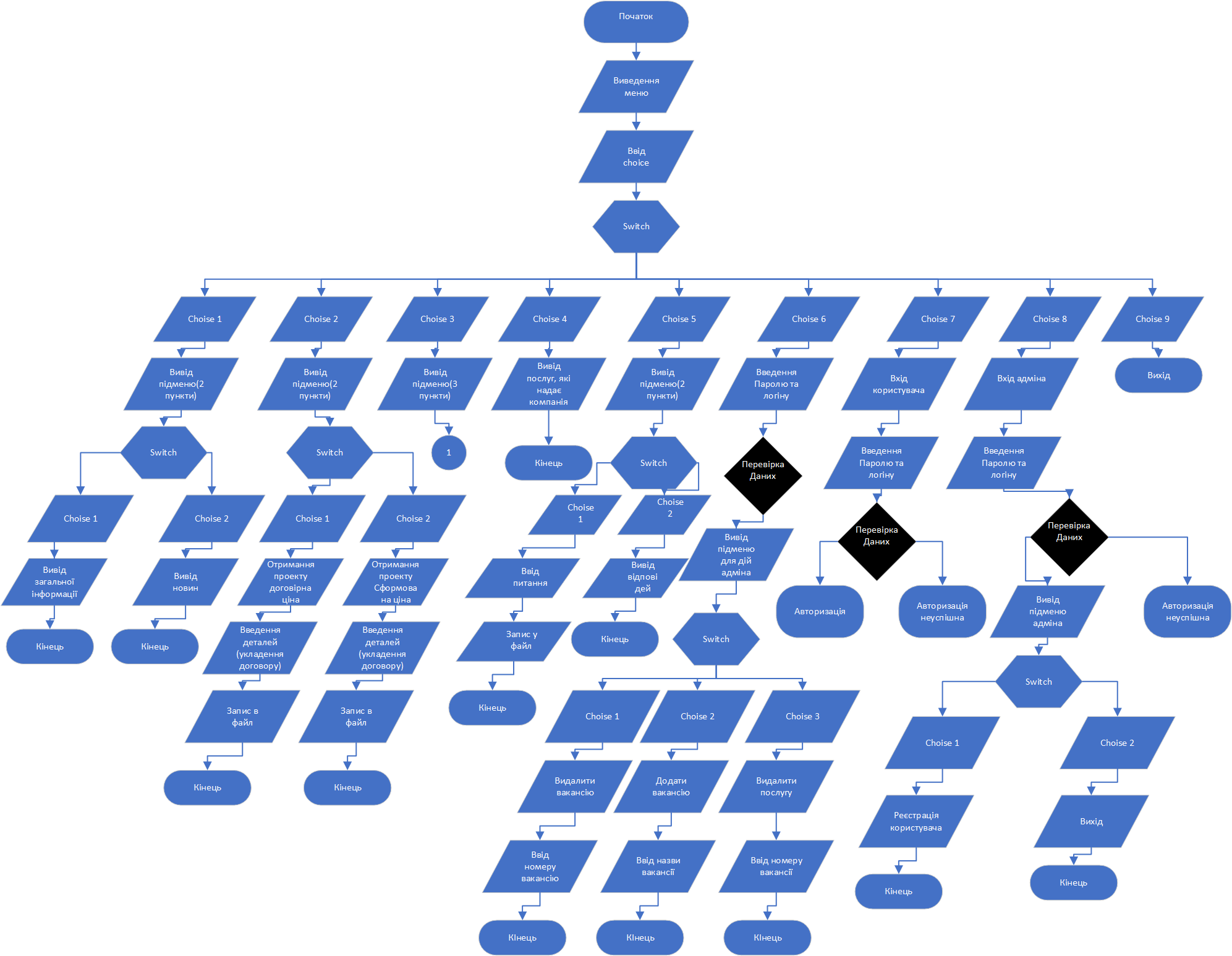
1. Скінченність

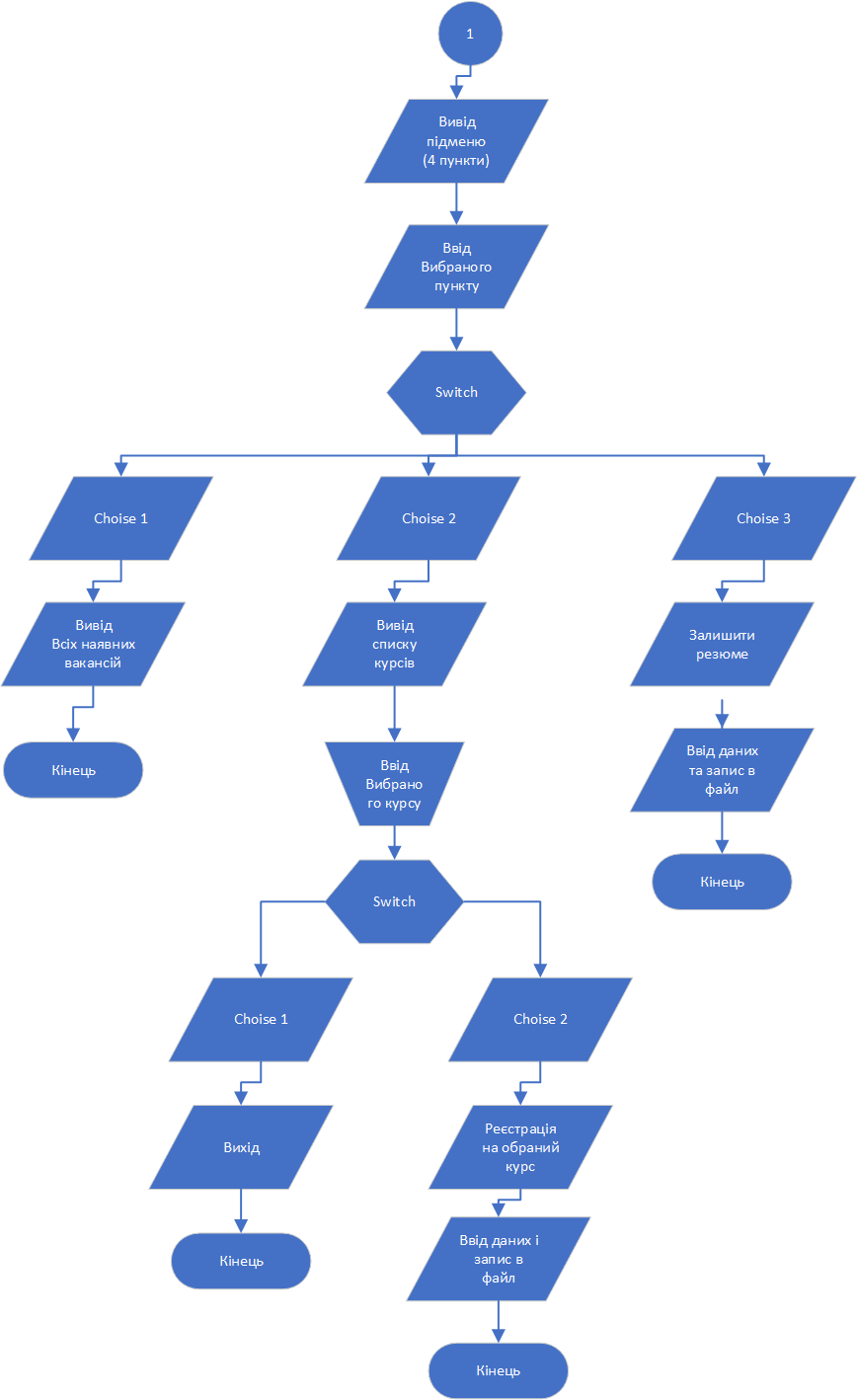
2. Результативність

3. Формальність

4. Визначеність

5. Масовість

На даному рисунку зображено алгоритм виконання моєї програми.



**РОЗДІЛ 3. Реалізація алгоритму у програмному коді та тестування програми**

Після складання діаграми прецедентів та алгоритму програми, я приступила до написання програмного коду.

Для реалізація я використала 3 батьківських класи:

1. class Company

Даний клас містить 7 методів та 2 дочірніх класи.

* метод для новин та загальної інформації;
* метод для того, щоб користувач міг поставити питання;
* метод, щоб залишити резюме(заявку на вакансію);
* метод, щоб зареєструватись на обраний тренінг;
* метод, щоб дізнатись про послуги компанії;
* метод, щоб замовити проект з фіксованою ціною;
* метод, щоб замовити проект з ціною за договором;
* дочірній клас Resume
* дочірній клас Trainings

1. class Vacancies
2. class RegLogin

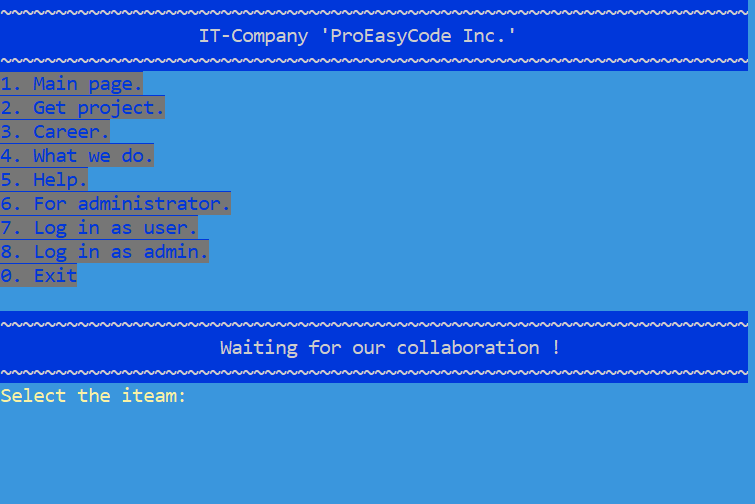
На рис.1 зображено головне меню, яке містить в 9 пунктів.

Рис.1 «Головне меню»

Окремо опишемо кожну вкладку, для огляду роботи програми.

Перша вкладка - «Головна сторінка»

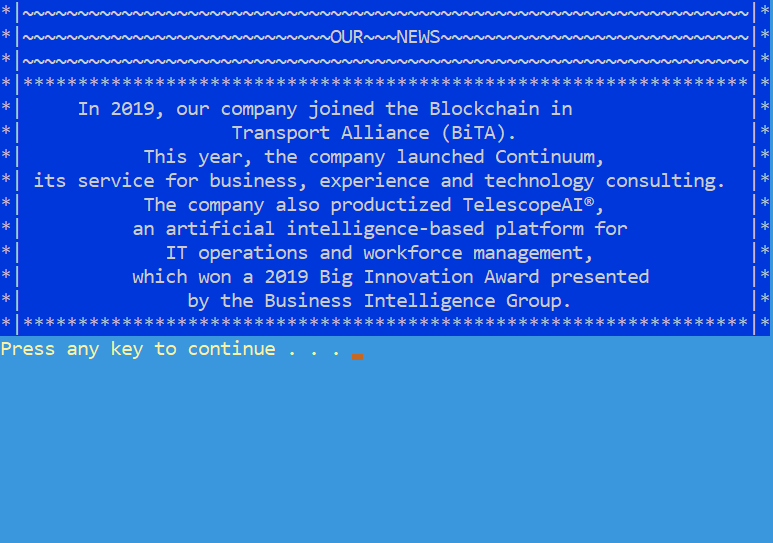
Містить в собі загальну інформацію про ІТ-компанію та новини.

Рис.2 «Новини»

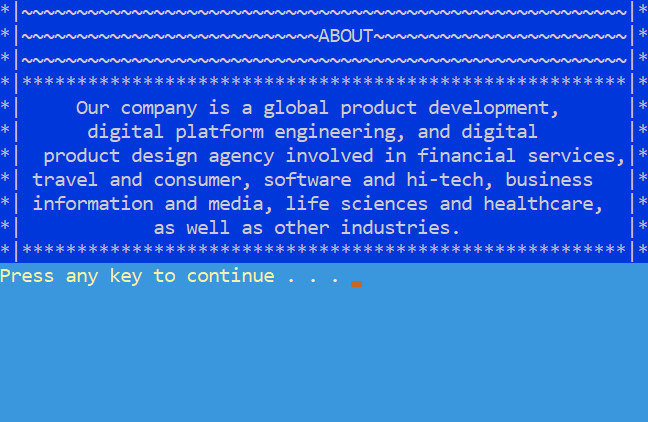


Рис.3 «Загальна інформація»

Друга вкладка - «Замовлення проекту»

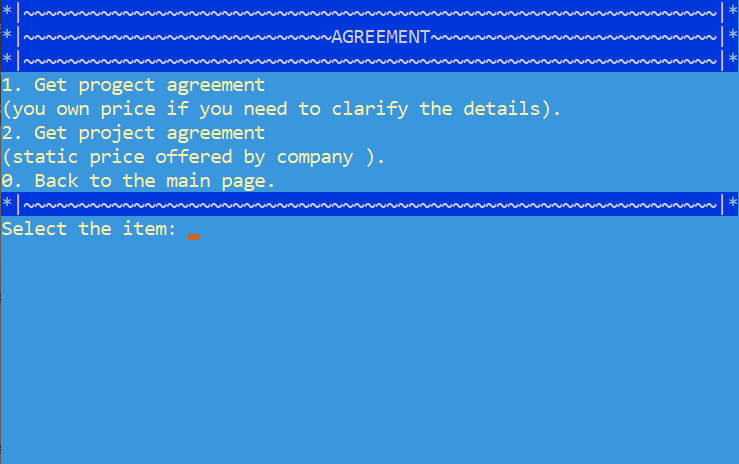


Рис. 4 «Замовлення проекту»



Рис. 5 «Замовлення проекту»

Як можна побачити, є два види для замовлення. В першому пропонується залишити лише свої контакти, щоб обговорити проект особисто. В другому користувач вказує, який його проект цікавить, скільки днів на виконання відведено. Програма обраховує приблизну ціну, щоб зорієнтувати замовника.

Третя вкладка - «Кар’єра»

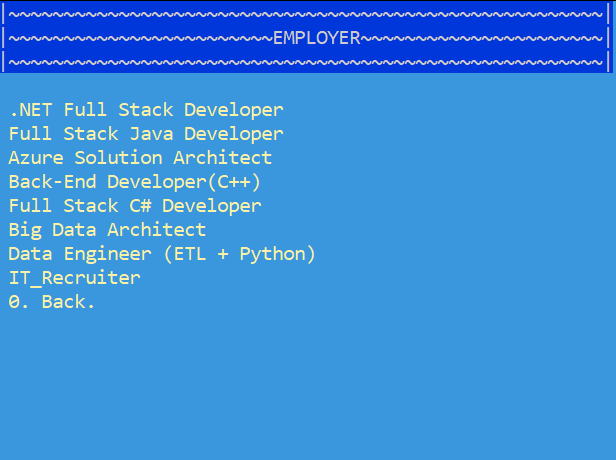
Тут пропонується перегляд наявних вакансій, тренінгів, реєстрація на тренінги та можливість залишити резюме для обраної вакансії.

Рис. 6 «Наявні вакансії»

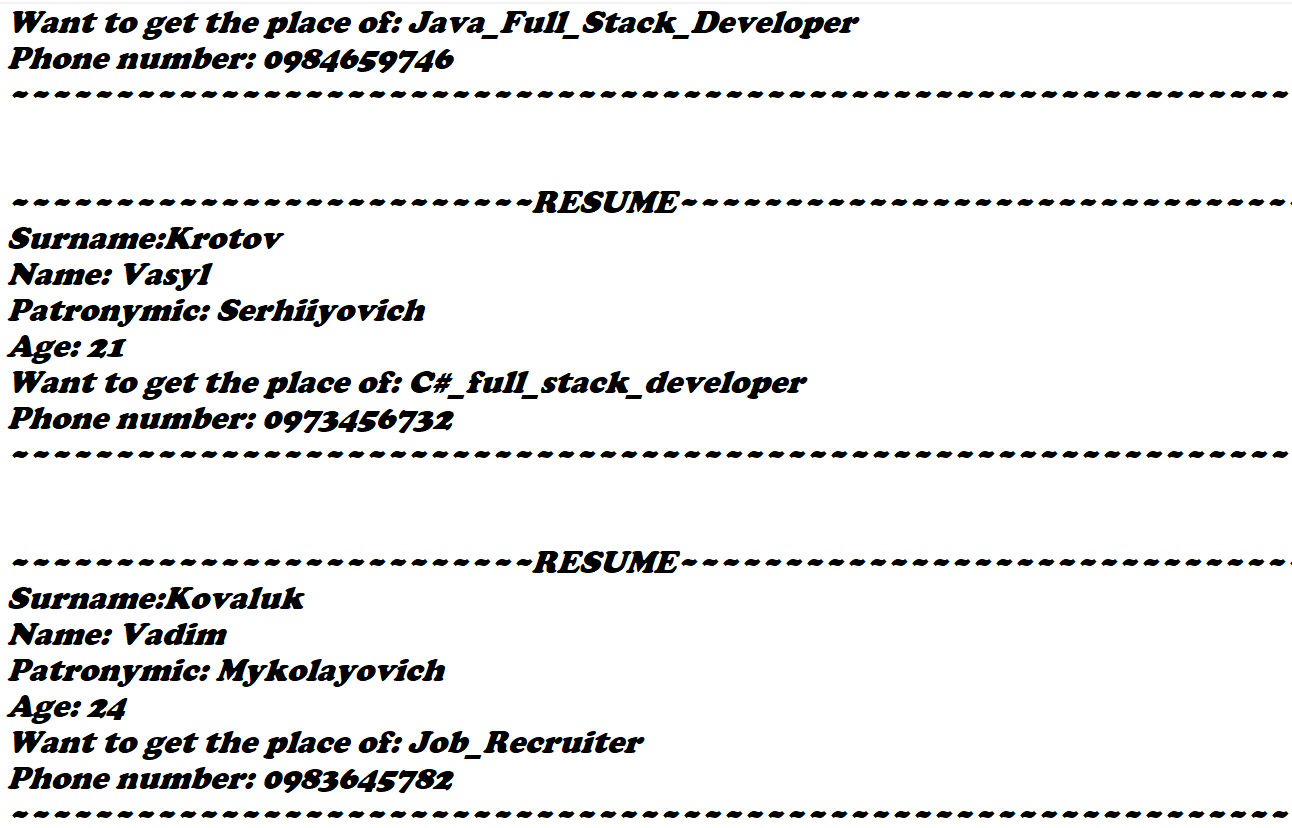


Рис. 7 «Приклад інформації, залишеної в резюме»

Після перегляду вакансій, користувач може залишити резюме, вказавши свої дані. Для розгляду вони записуються у відповідний файл.

Також кількість вакансій з плином часу роботи програми може збільшуватись або зменшуватись, тобто адміністрація може оперувати їх кількістю.

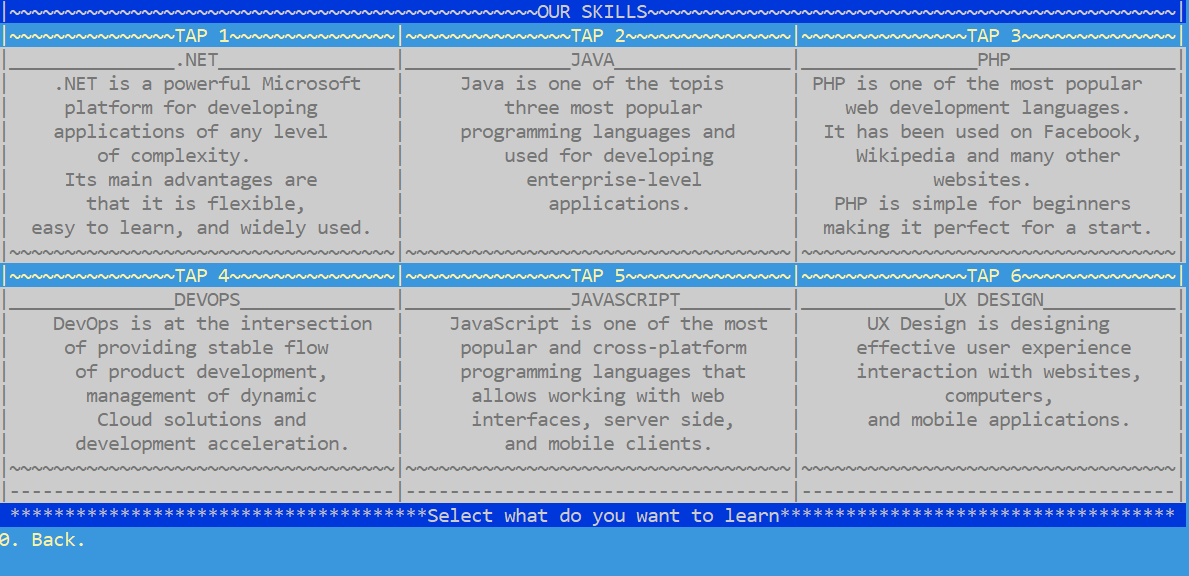


Рис. 8 «Тренінги на базі компанії»

Користувач може обрати тренінг та зареєструватись на нього, попередньо пройшовши тест з англійської. В тесті пропонується 5 питань, тільки правильні відповіді на всі п’ять питань надають можливість зареєструватись. Кількість спроб на тест не обмежена. Інформація про реєстрацію на тренінг записується у відповідний файл.

Четверта вкладка – «Послуги, які надає компанія»

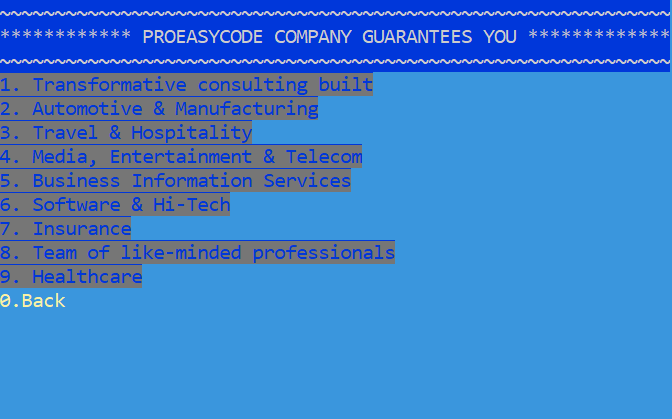


Рис. 9 «Послуги компанії»

П’ята вкладка - «Допомога»

На даній вкладці користувач може поставити питання та переглянути відповіді на поширенні питання.

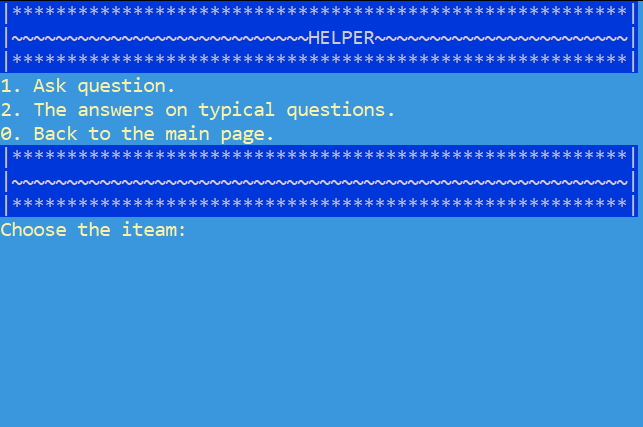


Рис.10 «Допомога»

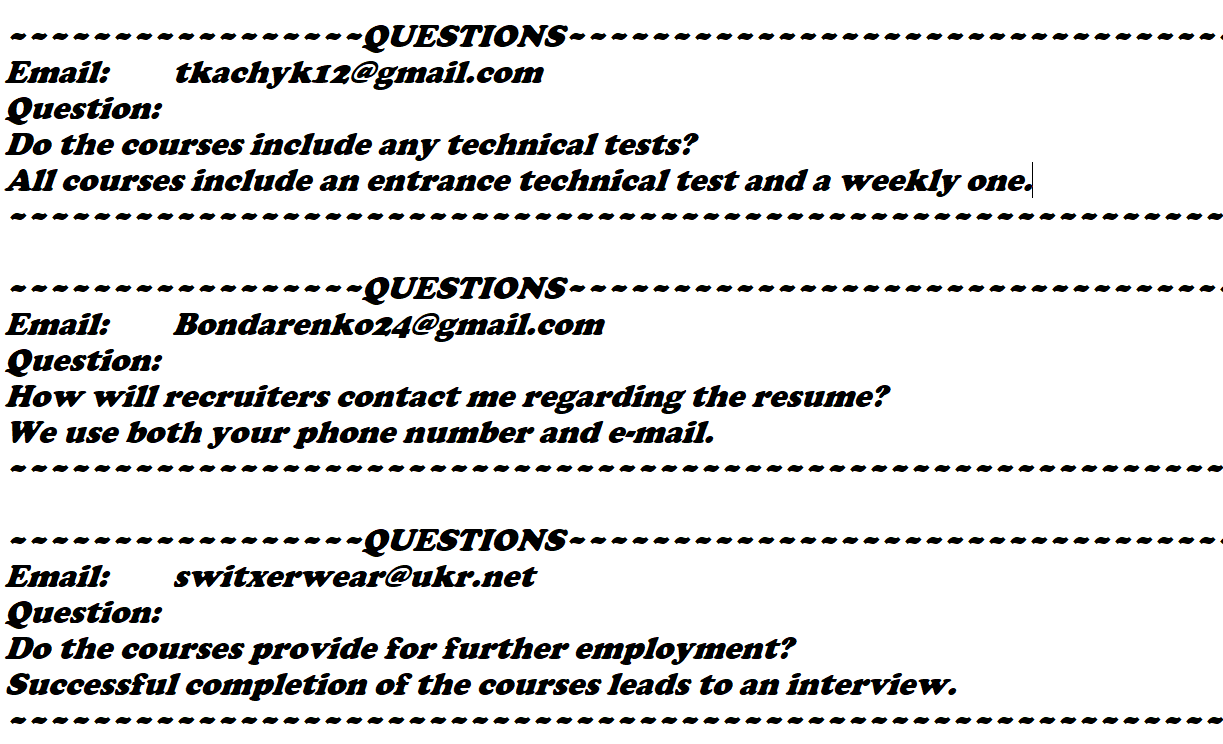


Рис. 11 «Поставлені питання та відповіді на них»

Шоста вкладка - «Адміністрування»

В цій вкладці відображаються повноваження адміністратора. Він може видалити або додати вакансію, послугу.

Для цього відбувається його ідентифікація за допомогою логіна і паролю.

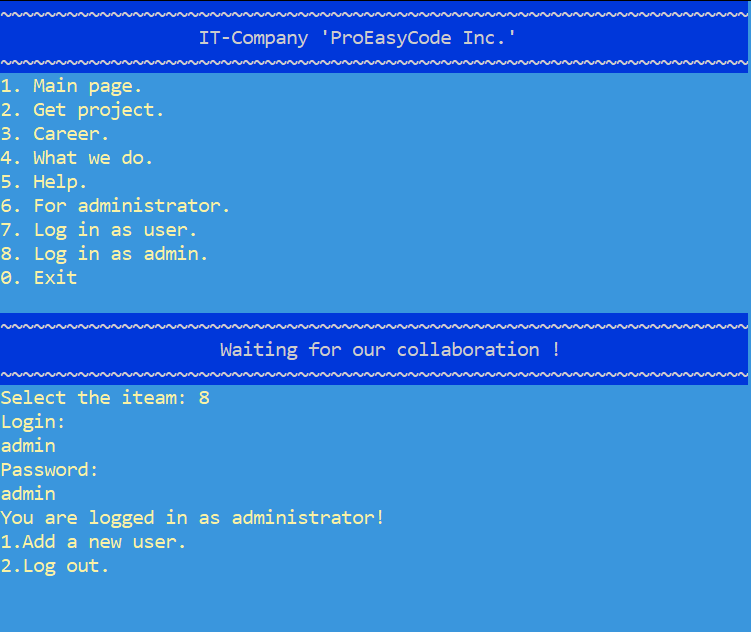


Рис. 12 «Ідентифікація адміністратора»

Сьома вкладка - «Вхід користувача»

Рис.13 «Успішний вхід користувача»



Рис. 14 «Неуспішний вхід користувача»

Восьма вкладка - «Вхід для адміністратора»

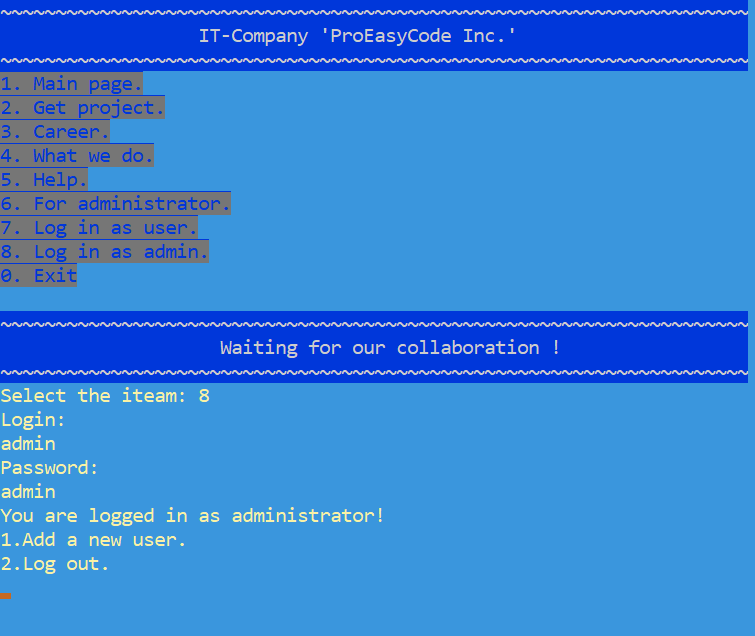
Дана кладка назначена для реєстрації нових користувачів. 

Рис. 15 «Вхід адміністратора»

Дев’ята вкладка – «Вихід з програми»

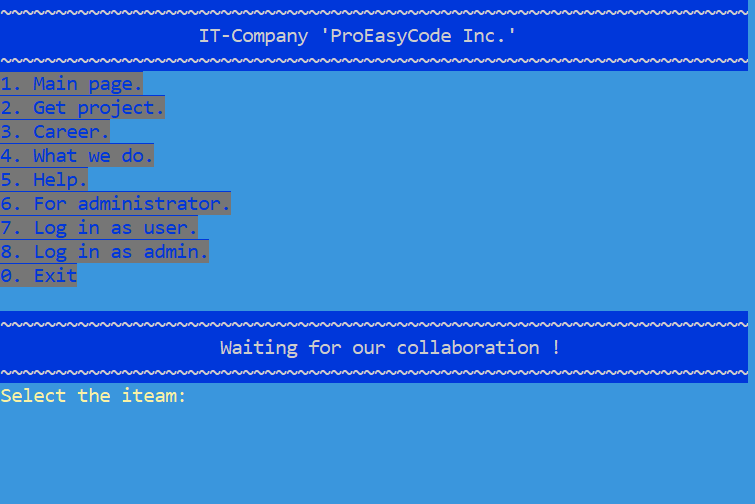


Рис. 15 «Головне меню»

**Висновок**

У процесі виконання поставленого переді мною завдання, а саме реалізація такої предметної галузі як ІТ-компанія, я закріпила свої знання з вивченого за один семестр курсу ООП.

Засвоїла всі правила використання класів. Провела детальний аналіз предметної області, вивчила проблеми і актуальність створення програмного забезпечення. Під час написання програми зіткнулась з багатьма проблемами, тому отримала знання і навички для їх вирішення. Покращила свої вміння у програмуванні.

Навчилася знаходити потрібну інформацію з різних джерел, вміти читати і швидко розбиратись у чужих кодах. Застосовувати вже готові алгоритми у власних цілях і вносити в них певні корективи для коректної роботи програмного забезпечення. Закріпила на практиці знання із створення класів, утворення одинарного наслідування та інкапсуляцію даних.

З плюсів моєї програми можна виділити те , що вона є безкоштовною, а також має високу швидкодію, наявність простого у використанні інтерфейсу який полегшує керування нею.

З точки зору програмування однією із позитивних сторін моєї програми є те, що я використовував масиви об’єктів – так звані вектори. Адже у сучасному світі ООП саме такий метод опрацювання об’єктів є пріоритетним.

Недоліком моєї програми можна вважати те, що вона не охоплює усіх аспектів керування ІТ-компанії, оскільки дане поняття являє собою досить широку сферу .

**Список використаної літератури**

1. <http://programming.in.ua/programming/basisprogramming/25-oop.html>
2. <https://uk.wikipedia.org/wiki/Об%27єктноорієнтоване_програмування#.D0.86.D1.81.D1.82.D0.BE.D1.80.D1.96.D1.8F>
3. <https://uk.wikipedia.org/wiki/C%2B%2B>
4. <https://ru.wikipedia.org/wiki/Объектноориентированное_программирование#.D0.98.D1.81.D1.82.D0.BE.D1.80.D0.B8.D1.8F>
5. <https://uk.wikipedia.org/wiki/Об%27єктно-орієнтоване_програмування>
6. <http://www.progs.in.ua/Progs/rozdilu/OOP.php>
7. <http://damp.biz/plyusi-i-minusi-obyektno-oriyentovanogo-programuvannya/>
8. Armstrong, Deborah J. (February 2006). The Quarks of Object-Oriented Development. Communications of the ACM 49 (2). с. 123–128. ISSN 0001-0782.
9. Meyer, Bertrand (1997). Object-Oriented Software Construction. Prentice Hall. ISBN 0-13-629155-4.
10. Основы информатики и программирования: Учебное пособие// Е. А. Роганов — М.: МГИУ, 2001. — Вузол VI.(рос.)
11. Лавріщева К.М. Програмна інженерія—К.: Академперіодика, 2008.- 319 с.

12. Кнут Д. «Мистецтво програмування». І.Д. Вільямс, 2007.

13. Є.В. Пишкін. «Основні концепції та механізми ООП». БХВ-Петербург, 2005.

14. Греді Буч, Роберт А. Максимчук, Майкл У. Енгл, Боббі Дж. Янг, Джим Коналл, Келлі А. Х'юстон. «Об'єктно-орієнтований аналіз та проектування», І.Д. Вільямс, 2010.

15. Бьярн Страуструп «Мова програмування С++» (четверте видання), Addison–Wesley, 2013

**ДОДАТОК А (Лістинг програмного коду проекту)**

**Company.h**

#include <fstream>

#include <Windows.h>

#include <vector>

#include <ctime>

#include <sstream>

#include <conio.h>

#include <list>

#include <iterator>

#include<iomanip>

#include"Training.h"

#include "Resume.h"

using namespace std;

class Company

{

public:

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

void info()

{

int num = -1;

while (num != 0)

{

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "1. Common information.\n2. News.\n0. Back to the main page.\n" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "Choose the iteam: ";

cin >> num;

if (num > 3 || num < 0)

{

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 3));

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

cout << "|~~~~~The page was deleted or something went wrong!~~~~~~|" << endl;

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

break;

}

switch (num)

{

case 1:

{

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|\*" << endl;

cout << "\*|~~~~~~~~~~~~~~~~~~~~~~~~~~~ABOUT~~~~~~~~~~~~~~~~~~~~~~~|\*" << endl;

cout << "\*|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|\*" << endl;

cout << "\*|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\*" << endl;

cout << "\*| Our company is a global product development, |\*" << endl;

cout << "\*| digital platform engineering, and digital |\*" << endl;

cout << "\*| product design agency involved in financial services,|\*" << endl;

cout << "\*| travel and consumer, software and hi-tech, business |\*" << endl;

cout << "\*| information and media, life sciences and healthcare, |\*" << endl;

cout << "\*| as well as other industries. |\*" << endl;

cout << "\*|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\*" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

break;

}

case 2:

{

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|\*" << endl;

cout << "\*|~~~~~~~~~~~~~~~~~~~~~~~~~~~~OUR~~~NEWS~~~~~~~~~~~~~~~~~~~~~~~~~~~~|\*" << endl;

cout << "\*|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|\*" << endl;

cout << "\*|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\*" << endl;

cout << "\*| In 2019, our company joined the Blockchain in |\*" << endl;

cout << "\*| Transport Alliance (BiTA). |\*" << endl;

cout << "\*| This year, the company launched Continuum, |\*" << endl;

cout << "\*| its service for business, experience and technology consulting. |\*" << endl;

cout << "\*| The company also productized TelescopeAI®, |\*" << endl;

cout << "\*| an artificial intelligence-based platform for |\*" << endl;

cout << "\*| IT operations and workforce management, |\*" << endl;

cout << "\*| which won a 2019 Big Innovation Award presented |\*" << endl;

cout << "\*| by the Business Intelligence Group. |\*" << endl;

cout << "\*|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\*" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

break;

}

}

}

system("cls");

}

void WithYourPrice(int price, string project1) {

fstream file;

file.open("C:\\Users\\luda\\Desktop\\courseworkfiles\\pull.txt", fstream::app);

string text;

for (int i = 0; i < 5; i++) {

if (i = 1) {

system("cls");

cout << "Enter your surname: ";

cin >> text;

file << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";

file << "Ordering a project: \n\n";

file << project1;

file << "\nSurname:" + text + "\n";

}

if (i = 2)

{

cout << "Enter your name: ";

cin >> text;

file << "Name: " + text + "\n";

}

if (i = 3)

{

cout << "Enter how many days you give for implementation: ";

cin >> text;

file << "Your proposition: " + text + "\n";

}

if (i = 4)

{

cout << "Leave your email: ";

cin >> text;

file << "Email:" + text + "\n";

file << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";

}

}

file.close();

system("C:\\Users\\luda\\Desktop\\courseworkfiles\\pull.txt");

}

void WithStaticPrice() {

int num = -1;

while (num != 0) {

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|\*" << endl;

cout << "\*|~~~~~~~~~~~~~~~~~~~~~~~~~~~~AGREEMENT~~~~~~~~~~~~~~~~~~~~~~~~~~|\*" << endl;

cout << "\*|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|\*" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "1. Get progect agreement \n(you own price if you need to clarify the details).\n2. Get project agreement\n(static price offered by company ).\n0. Back to the main page." << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|\*" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "Select the item: ";

cin >> num;

if (num > 4 || num < 0)

{

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 3));

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

cout << "|~~~~~The page was deleted or something went wrong!~~~~~~|" << endl;

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

}

switch (num)

{

case 1:

{

system("cls");

fstream file;

file.open("C:\\Users\\luda\\Desktop\\courseworkfiles\\pull.txt", fstream::app);

string text;

for (int i = 0; i < 5; i++)

{

if (i = 1) {

system("cls");

cout << "Enter your surname: ";

cin >> text;

file << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";

file << "Ordering a project: \n\n";

file << "Surname:" + text + "\n";

}

if (i = 2)

{

cout << "Enter your name: ";

cin >> text;

file << "Name: " + text + "\n";

}

if (i = 3)

{

cout << "Enter how many days you give for implementation: ";

cin >> text;

file << "Your proposition: " + text + "\n";

}

if (i = 4)

{

cout << "Leave your email: ";

cin >> text;

file << "Email:" + text + "\n";

file << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";

}

break;

}

file.close();

system("C:\\Users\\luda\\Desktop\\courseworkfiles\\pull.txt");

break;

}

case 2:

{

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|~~~~~~~~~~~~~~ORDERING A PROJECT~~~~~~~~~~~~~~|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "1.Website\n";

cout << "2.Application\n";

cout << "3.Website + phone application\n";

cout << "4.Establishment for application\website\n";

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "Select specific of your project: ";

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

int n;

int price;

int time1;

while (true) {

cin >> n;

if (n >= 1 && n <= 4)

break;

else

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "| We don't have more proposals for you! Sorry! |\n";

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

break;

}

}

system("cls");

switch (n) {

case 1:

{

price = 300;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|~~~~~~~~~~~~~~ORDERING A PROJECT~~~~~~~~~~~~~~|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "Price per one day to make your project: " << price << " $ \n";

cout << "Specify how many days will you give us(from 6 to 26)>>";

cin >> time1;

if (time1 < 6)

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "~~~~~~~~~~~~~~~~~~~SORRY~~~~~~~~~~~~~~~" << endl;

cout << "~We cannot do your project so quickly!~" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0. Back.\n";

cin >> num;

break;

}

else if (time1 > 26)

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "~~~~~~~~~~~~~~~~~~~SORRY~~~~~~~~~~~~~~~" << endl;

cout << "~~~We cannot realise so big project!~~~" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0. Back.\n";

cin >> num;

break;

}

else

{

system("cls");

string project1;

project1 = "Website";

price = price \* time1;

cout << "Price: " << price << endl;

cout << "Are you ready to pay us? ";

cout << "\n1 - Yes!\n";

cout << "2 - I'm not sure!\n";

int k;

cin >> k;

if (k == 1) {

WithYourPrice(price, project1);

}

if (k == 2)

break;

break;

}

}

case 2:

{

price = 400;

cout << "Price per one day to make your project: " << price << " $ ";

cout << "\n\n" << "Specify how many days will you give us(from 6 to 26)>>";

cin >> time1;

if (time1 < 6)

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "~~~~~~~~~~~~~~~~~~~SORRY~~~~~~~~~~~~~~~" << endl;

cout << "~We cannot do your project so quickly!~" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0. Back.\n";

cin >> num;

break;

}

else if (time1 > 26)

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "~~~~~~~~~~~~~~~~~~~SORRY~~~~~~~~~~~~~~~" << endl;

cout << "~~~We cannot realise so big project!~~~" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0. Back.\n";

cin >> num;

break;

}

else

{

system("cls");

string project1;

project1 = "Application";

price = price \* time1;

cout << "Price: " << price << endl;

cout << "Are you ready to pay us? ";

cout << "\n1 - Yes!\n";

cout << "2 - I'm not sure!\n";

int k;

cin >> k;

if (k == 1) {

WithYourPrice(price, project1);

}

if (k == 2)

break;

break;

}

}

case 3:

{

price = 500;

cout << "Price per one day to make your project: " << price << " $ ";

cout << "\n" << "Specify how many days will you give us(from 6 to 26)>>";

cin >> time1;

if (time1 < 6)

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "~~~~~~~~~~~~~~~~~~~SORRY~~~~~~~~~~~~~~~" << endl;

cout << "~We cannot do your project so quickly!~" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0. Back.\n";

cin >> num;

break;

}

else if (time1 > 26)

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "~~~~~~~~~~~~~~~~~~~SORRY~~~~~~~~~~~~~~~" << endl;

cout << "~~~We cannot realise so big project!~~~" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0. Back.\n";

cin >> num;

break;

}

else

{

system("cls");

string project1;

project1 = "Website + phone application";

price = price \* time1;

cout << "Price: " << price << endl;

cout << "Are you ready to pay us? ";

cout << "\n1 - Yes!\n";

cout << "2 - I'm not sure!\n";

int k;

cin >> k;

if (k == 1) {

WithYourPrice(price, project1);

}

if (k == 2)

break;

break;

}

}

case 4:

{

price = 100;

cout << "Price per one day to make your project: " << price << " $ ";

cout << "\n\n" << "Specify how many days will you give us(from 6 to 26)";

cin >> time1;

if (time1 < 6)

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "~~~~~~~~~~~~~~~~~~~SORRY~~~~~~~~~~~~~~~" << endl;

cout << "~We cannot do your project so quickly!~" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0. Back.\n";

cin >> num;

break;

}

else if (time1 > 26)

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "~~~~~~~~~~~~~~~~~~~SORRY~~~~~~~~~~~~~~~" << endl;

cout << "~~~We cannot realise so big project!~~~" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0. Back.\n";

cin >> num;

break;

}

else

{

system("cls");

price = price \* time1;

string project1;

project1 = "Establishment for website or phone application";

cout << "Price: " << price << endl;

cout << "Are you ready to pay us? ";

cout << "\n1 - Yes!\n";

cout << "2 - I'm not sure!\n";

int k;

cin >> k;

if (k == 1) {

WithYourPrice(price, project1);

}

if (k == 2)

break;

break;

}

}

}

break;

}

}

system("cls");

}

}

void actshow(string actions[], int length)

{

int num = -1;

while (num != 0)

{

system("cls");

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\* PROEASYCODE COMPANY GUARANTEES YOU \*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((11 << 7) | 1));

for (int i = 0; i < length; i++)

{

cout << i + 1 << ". " << actions[i] << endl;

}

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0.Back \n";

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cin >> num;

break;

}

}

void StayAPart()

{

Resume XXX;

XXX.Template();

}

void LeaveDet()

{

Training XXX;

XXX.train();

}

string \* Admin(string actions[], int length)

{

int num = -1;

string password = "admin";

string enter;

while (num != 0)

{

system("cls");

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" << endl;

cout << "Enter password: " << endl;

cin >> enter;

if (enter != password)

{

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" << endl;

cout << "~~~~~~~~~~~~~~~Incorect input! Try again~~~~~~~~~~~~~~~~~" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl << endl;

system("pause");

}

else

{

num = -1;

while (num != 0)

{

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\* PROEASYCODE COMPANY GUARANTEES YOU \*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

for (int i = 0; i < length; i++)

{

cout << i + 1 << "." << actions[i] << "\n";

}

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "Delete one of our actions : " << endl;

cin >> num;

if (num > 10 || num < 1)

{

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "~~~~~~~~~~~~~~~Incorect input! Try again~~~~~~~~~~~~~~~~~" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

break;

}

string \* newActions = new string[length - 1];

for (int i = num - 1; i < length - 1; i++)

{

actions[i] = actions[i + 1];

}

for (int i = 0; i < length - 1; i++)

{

newActions[i] = actions[i];

}

system("pause");

system("cls");

return newActions;

}

}

}

}

void question()

{

int num = -1;

while (num != 0)

{

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~HELPER~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((11 << 7) | 1));

cout << "1. Ask question.\n2. The answers on typical questions.\n0. Back to the main page." << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "Choose the iteam: ";

cin >> num;

if (num >= 3 || num < 0)

{

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

cout << "|~~~~~~~~~~~~~~~Incorect input! Try again~~~~~~~~~~~~~~~~|" << endl;

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

break;

}

switch (num)

{

case 1: {

vector<string>ask;

string text;

char text1[150];

fstream file;

fstream file1;

file.open("C:\\Users\\luda\\Desktop\\courseworkfiles\\questions.txt", fstream::app);

file1.open("questions.txt", fstream::app);

for (int i = 0; i < 3; i++)

{

if (i = 1)

{

cout << "Enter your e-mail: ";

cin >> text;

}

if (i = 2)

{

cout << "What is your question? " << endl;

getchar();

cin.getline(text1, 150);

}

}

string all = "~~~~~~~~~~~~~~~~~QUESTIONS~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\nEmail:\t" + text + "\n" + "Question: \n" + text1 + "\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~n\n\n";

ask.push\_back(all);

file << all;

file1 << all;

file.close();

file1.close();

system("notepad C:\\Users\\luda\\Desktop\\courseworkfiles\\questions.txt");

system("cls");

break;

}

case 2:

{

system("C:\\Users\\luda\\Desktop\\courseworkfiles\\questions.txt");

break;

}

}

}

system("cls");

}

};

**RegLogin.h**

#include <iostream>

#include <string>

#include <fstream>

#include <Windows.h>

#include <vector>

#include <ctime>

#include <sstream>

#include <conio.h>

#include <list>

#include <iterator>

#include<iomanip>

using namespace std;

class RegLogin

{

public:

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

struct Users

{

string login\_new;

string password\_new;

}

users[100];

bool check(string login, string password)

{

if (login == "admin" && password == "admin")

{

return true;

}

else

{

return false;

}

}

bool check\_1(string login\_admin, string password\_admin)

{

if (login\_admin == "admin" && password\_admin == "admin")

{

return true;

}

else

{

return false;

}

}

void register\_1()

{

string login\_new, password\_new;

fstream file;

fstream file1;

file.open("users.txt", fstream::app);

file1.open("try.txt", fstream::app);

cout << "Enter new user's email: " << endl;

cin >> login\_new;

cout << "Add password: " << endl;

cin >> password\_new;

file << login\_new << "\nYour password to log in: " << password\_new << endl;

file1 << " " << login\_new << " " << password\_new << endl;

file.close();

file1.close();

cout << "Awesome!The system will send the password to the user. " << endl;

system("pause");

}

void admin()

{

bool flag\_1 = false;

string login\_admin, password\_admin;

cout << "Login: " << endl;

cin >> login\_admin;

cout << "Password: " << endl;

cin >> password\_admin;

flag\_1 = check\_1(login\_admin, password\_admin);

if (flag\_1 == true)

{

cout << "You are logged in as administrator!" << endl;

cout << "1.Add a new user. " << endl;

cout << "2.Log out. " << endl;

int ch;

cin >> ch;

switch (ch)

{

case 1:

{

register\_1();

break;

}

case 2:

{

break;

}

}

}

else if (flag\_1 == false)

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "|#####################################################|" << endl;

cout << "| Wrong password! Try again! |" << endl;

cout << "|#####################################################|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

}

}

void read(string login\_new, string password\_new)

{

ifstream fin("try.txt");

for (int i = 0; i < 100; i++)

{

fin >> users[1].login\_new >> users[1].password\_new;

}

}

void login()

{

bool show = false;

string login\_new, password\_new;

cout << "Enter login: " << endl;

cin >> login\_new;

cout << "Enter password: " << endl;

cin >> password\_new;

read(login\_new, password\_new);

for (int i = 0; i < 100; i++)

{

if (login\_new == users[1].login\_new && password\_new == users[1].password\_new)

{

show = true;

}

}

if (show == true)

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 4) | 5));

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|~~Your are logged in succesfully!~~|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

}

else

{

SetConsoleTextAttribute(hConsole, (WORD)((10 << 5) | 7));

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|~~~~~Wrong password! Try again!~~~~|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

}

}

};

**Vacancies.h**

#include <iostream>

#include <string>

#include <fstream>

#include <Windows.h>

#include <vector>

#include <ctime>

#include <sstream>

#include <conio.h>

#include <list>

#include <iterator>

#include<iomanip>

using namespace std;

class Vacancies

{

public:

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

void vacshow()

{

int num = -1;

while (num != 0)

{

system("cls");

list<string> vaclist;

ifstream fin("vacancy.txt", fstream::in);

for (std::string line; std::getline(fin, line); )

{

vaclist.push\_back(line);

}

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~EMPLOYER~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

setw(3);

copy(vaclist.begin(), vaclist.end(), ostream\_iterator<string>(cout, " \n "));

cout << "0. Back." << endl;

cin >> num;

}

}

void vacdelete()

{

system("cls");

string password = "admin";

string enter;

cout << "Enter password: " << endl;

cin >> enter;

if (enter == password)

{

int length1 = 10;

std::string \* vacancy = new std::string[256];

char \* buff = new char[500];

std::ifstream fil("vacancy.txt");

int i = 0;

while (!fil.eof())

{

fil.getline(buff, 500);

vacancy[i] = std::string(buff);

std::cout << vacancy[i] << std::endl;

i++;

}

int num;

std::cout << "Delete one of our vacansies : " << std::endl;

std::cin >> num;

std::fstream file;

std::string \* newVacancy = new std::string[length1 - 1];

for (int i = num - 1; i < length1 - 1; i++)

{

vacancy[i] = vacancy[i + 1];

}

file.open("vacancy.txt", std::fstream::in | std::fstream::out);

for (int i = 0; i < length1 - 1; i++)

{

newVacancy[i] = vacancy[i];

std::cout << newVacancy[i] << std::endl;

file << newVacancy[i] << std::endl;

}

file.close();

}

else

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)((12 << 4) | 5));

cout << "|#####################################################|" << endl;

cout << "| Wrong password! Try again! |" << endl;

cout << "|#####################################################|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

}

}

void addvacancy()

{

int num = -1;

while (num != 0)

{

string password = "admin";

string enter;

cout << "Enter password: " << endl;

cin >> enter;

if (enter == password)

{

system("cls");

fstream file;

list<string> vaclist;

ifstream fin("vacancy.txt", fstream::in);

for (std::string line; std::getline(fin, line); )

{

vaclist.push\_back(line);

}

copy(vaclist.begin(), vaclist.end(), ostream\_iterator<string>(cout, "\n"));

string newvac;

cout << "Enter the name of vacancy you want to add: " << endl;

cin >> newvac;

vaclist.push\_back(newvac);

copy(vaclist.begin(), vaclist.end(), ostream\_iterator<string>(cout, " \n"));

std:ofstream fout("vacancy.txt");

std::copy(vaclist.begin(), vaclist.end(), std::ostream\_iterator<string>(fout, "\n"));

system("pause");

break;

}

else

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)((12 << 4) | 5));

cout << "|#####################################################|" << endl;

cout << "| Wrong password! Try again! |" << endl;

cout << "|#####################################################|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

}

}

}

};

**Resume.h**

#include <fstream>

#include <Windows.h>

#include <vector>

#include <ctime>

#include <sstream>

#include <conio.h>

#include <list>

#include <iterator>

#include<iomanip>

using namespace std;

class Resume

{

public:

void Template()

{

system("cls");

fstream file;

file.open("C:\\Users\\luda\\Desktop\\courseworkfiles\\application.txt", fstream::app);

for (int i = 0; i < 7; i++)

{

if (i = 1)

{

cout << "Surname: ";

SetConsoleCP(1251);

cin >> sname;

SetConsoleCP(866);

}

if (i = 2)

{

cout << "Name: ";

SetConsoleCP(1251);

cin >> name;

SetConsoleCP(866);

}

if (i = 3)

{

std::cout << "Patronymic: ";

SetConsoleCP(1251);

cin >> patronymic;

SetConsoleCP(866);

}

if (i = 4)

{

cout << "Job you want to get: ";

SetConsoleCP(1251);

cin >> vac;

SetConsoleCP(866);

}

if (i = 5)

{

cout << "Age: ";

SetConsoleCP(1251);

cin >> age;

SetConsoleCP(866);

}

if (i = 6)

{

cout << "Phone number: ";

SetConsoleCP(1251);

cin >> numb;

SetConsoleCP(866);

}

}

string all = "~~~~~~~~~~~~~~~~~~~~~~~~~RESUME~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\nSurname:" + sname + "\n" + "Name: " + name + "\n" + "Patronymic: " + patronymic + "\n" + "Age: " + age + "\n" + "Want to get the place of: " + vac + "\n" + "Phone number: " + numb + "\n" + "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n\n";

rsm.push\_back(all);

file << all;

file.close();

system("notepad C:\\Users\\luda\\Desktop\\courseworkfiles\\application.txt");

system("cls");

}

private:

vector<string>rsm;

string name;

string sname;

string patronymic;

string vac;

string age;

string numb;

};

**Training.h**

#include <fstream>

#include <Windows.h>

#include <vector>

#include <ctime>

#include <sstream>

#include <conio.h>

#include <list>

#include <iterator>

#include<iomanip>

using namespace std;

class Training

{

public:

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

void engtest()

{

system("cls");

int a, b, c, d, e;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~EXAMINATOR~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "1.My father ...... his new car for two weeks now. " << endl;

cout << " 1.has had\n 2.has\n 3.is having\n 4.had" << endl;

cin >> a;

cout << "2.She is now taking a more positive ...... to her studies and should do well. " << endl;

cout << " 1.attitude\n 2.behaviour\n 3.manner\n 4.style" << endl;

cin >> b;

cout << "3.According to Richard's ...... the train leaves at 7 o'clock. " << endl;

cout << " 1.opinion\n 2.advice\n 3.knowledge\n 4.information" << endl;

cin >> c;

cout << "4.You should not have a dog if you are not ...... to look after it. " << endl;

cout << " 1.prepared\n 2.adapted\n 3.arranged\n 4.decided" << endl;

cin >> d;

cout << "5.Marianne seemed to take ...... at my comments on her work " << endl;

cout << " 1.annoyance\n 2.insult\n 3.offence\n 4.indigation" << endl;

cin >> e;

if (a == 1 && b == 1 && c == 4 && d == 1 && e == 3)

{

system("cls");

fstream file;

file.open("C:\\Users\\luda\\Desktop\\courseworkfiles\\students.txt", fstream::app);

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

cout << "|~~~~~~~~~Awesome! Your English level is sufficient!~~~~~|" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "Leave us your details below." << endl;

vector<string>leave;

string yname;

string ysurname;

string email;

string all1;

string sub;

cout << "Course you are going to learn: " << endl;

cin >> sub;

cout << "Name: " << endl;

cin >> yname;

cout << "Surname: " << endl;

cin >> ysurname;

cout << "E-mail: " << endl;

cin >> email;

system("cls");

all1 = "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~STUDENTS~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\nField: " + sub + "\nEmail:" + email + "\n" + "Name: " + yname + " " + ysurname + "\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n\n";

leave.push\_back(all1);

file << all1;

cout << "0. Back.\n";

file.close();

system("notepad C:\\Users\\luda\\Desktop\\courseworkfiles\\students.txt");

}

else

{

system("cls");

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)((12 << 4) | 5));

cout << "|#####################################################|" << endl;

cout << "| Your English level is not sufficient! |" << endl;

cout << "| Your registration is impossible! |" << endl;

cout << "|#####################################################|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0. Back.\n";

}

}

void train()

{

system("cls");

int tap = -1;

while (tap != 0)

{

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~OUR SKILLS~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "|~~~~~~~~~~~~~~~TAP 1~~~~~~~~~~~~~~~|~~~~~~~~~~~~~~~TAP 2~~~~~~~~~~~~~~~|~~~~~~~~~~~~~~~TAP 3~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 3) | 8));

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.NET\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_JAVA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_PHP\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cout << "| .NET is a powerful Microsoft | Java is one of the topis | PHP is one of the most popular |" << endl;

cout << "| platform for developing | three most popular | web development languages. |" << endl;

cout << "| applications of any level | programming languages and | It has been used on Facebook, |" << endl;

cout << "| of complexity. | used for developing | Wikipedia and many other |" << endl;

cout << "| Its main advantages are | enterprise-level | websites. |" << endl;

cout << "| that it is flexible, | applications. | PHP is simple for beginners |" << endl;

cout << "| easy to learn, and widely used. | | making it perfect for a start. |" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "|~~~~~~~~~~~~~~~TAP 4~~~~~~~~~~~~~~~|~~~~~~~~~~~~~~~TAP 5~~~~~~~~~~~~~~~|~~~~~~~~~~~~~~~TAP 6~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 3) | 8));

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_DEVOPS\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_JAVASCRIPT\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_UX DESIGN\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cout << "| DevOps is at the intersection | JavaScript is one of the most | UX Design is designing |" << endl;

cout << "| of providing stable flow | popular and cross-platform | effective user experience |" << endl;

cout << "| of product development, | programming languages that | interaction with websites, |" << endl;

cout << "| management of dynamic | allows working with web | computers, |" << endl;

cout << "| Cloud solutions and | interfaces, server side, | and mobile applications. |" << endl;

cout << "| development acceleration. | and mobile clients. | |" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

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

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Select what do you want to learn\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* " << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "0. Back." << endl;

cin >> tap;

system("cls");

switch (tap)

{

case 1:

{

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.NET\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((10 << 4) | 9));

cout << "| .NET is a powerful Microsoft |" << endl;

cout << "| platform for developing |" << endl;

cout << "| applications of any level |" << endl;

cout << "| of complexity. |" << endl;

cout << "| Its main advantages are |" << endl;

cout << "| that it is flexible, |" << endl;

cout << "| easy to learn, and widely used. |" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_START DATE\_\_\_|\_\_\_\_END DATE \_\_|\_\_\_\_\_FREQUENCY\_\_\_\_\_\_\_\_|\_\_LOCATION\_\_\_|" << endl;

cout << "|\_\_\_\_\_\_2020 MAY\_\_\_|\_\_\_\_2020 AUG\_\_\_|\_\_2-3 TIMES PER WEEK\_\_|\_\_\_UKRAINE\_\_\_|" << endl;

cout << "| |REGISTRATION - 1| |BACK - 0| |" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

int call = -1;

while (call != 0)

{

cin >> call;

system("cls");

switch (call)

{

case 1:

{

cout << "WE MUST CHECK THE LEVEL OF YOUR ENGISH" << endl;

cout << "Tap 1 to pass it" << endl;

cout << "Tap 0 to cancel if you aren't ready" << endl;

cin >> call;

system("cls");

switch (call)

{

case 1:

{

engtest();

break;

}

break;

}

}

break;

}

}

break;

}

case 2:

{

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_JAVA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((10 << 4) | 9));

cout << "| Java is one of the topis |" << endl;

cout << "| three most popular |" << endl;

cout << "| programming languages and |" << endl;

cout << "| used for developing |" << endl;

cout << "| enterprise-level |" << endl;

cout << "| applications. |" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_START DATE\_\_\_|\_\_\_\_END DATE \_\_|\_\_\_\_\_FREQUENCY\_\_\_\_\_\_\_\_|\_\_LOCATION\_\_\_|" << endl;

cout << "|\_\_\_\_\_\_2020 MAY\_\_\_|\_\_\_\_2020 AUG\_\_\_|\_\_2-3 TIMES PER WEEK\_\_|\_\_\_UKRAINE\_\_\_|" << endl;

cout << "| |REGISTRATION - 1| |BACK - 0| |" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

int call = -1;

while (call != 0)

{

cin >> call;

system("cls");

switch (call)

{

case 1:

{

cout << "WE MUST CHECK THE LEVEL OF YOUR ENGISH" << endl;

cout << "Tap 1 to pass it" << endl;

cout << "Tap 0 to cancel if you aren't ready" << endl;

cin >> call;

system("cls");

switch (call)

{

case 1:

{

engtest();

}

break;

}

}

break;

}

}

break;

}

case 3:

{ system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_PHP\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((10 << 4) | 9));

cout << "| PHP is one of the most popular |" << endl;

cout << "| web development languages. |" << endl;

cout << "| It has been used on Facebook, |" << endl;

cout << "| Wikipedia and many other |" << endl;

cout << "| PHP is simple for beginners |" << endl;

cout << "| making it perfect for a start. |" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_START DATE\_\_\_|\_\_\_\_END DATE \_\_|\_\_\_\_\_FREQUENCY\_\_\_\_\_\_\_\_|\_\_LOCATION\_\_\_|" << endl;

cout << "|\_\_\_\_\_\_2020 MAY\_\_\_|\_\_\_\_2020 AUG\_\_\_|\_\_2-3 TIMES PER WEEK\_\_|\_\_\_UKRAINE\_\_\_|" << endl;

cout << "| |REGISTRATION - 1| |BACK - 0| |" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

int call = -1;

while (call != 0)

{

cin >> call;

system("cls");

switch (call)

{

case 1:

{

cout << "WE MUST CHECK THE LEVEL OF YOUR ENGISH" << endl;

cout << "Tap 1 to pass it" << endl;

cout << "Tap 0 to cancel if you aren't ready" << endl;

cin >> call;

system("cls");

switch (call)

{

case 1:

{

engtest();

}

break;

}

}

break;

}

}

break;

}

case 4:

{ system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_DEVOPS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((10 << 4) | 9));

cout << "| DevOps is at the intersection |" << endl;

cout << "| of providing stable flow |" << endl;

cout << "| of product development, |" << endl;

cout << "| management of dynamic |" << endl;

cout << "| Cloud solutions and |" << endl;

cout << "| development acceleration. |" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_START DATE\_\_\_|\_\_\_\_END DATE \_\_|\_\_\_\_\_FREQUENCY\_\_\_\_\_\_\_\_|\_\_LOCATION\_\_\_|" << endl;

cout << "|\_\_\_\_\_\_2020 MAY\_\_\_|\_\_\_\_2020 AUG\_\_\_|\_\_2-3 TIMES PER WEEK\_\_|\_\_\_UKRAINE\_\_\_|" << endl;

cout << "| |REGISTRATION - 1| |BACK - 0| |" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

int call = -1;

while (call != 0)

{

cin >> call;

system("cls");

switch (call)

{

case 1:

{

cout << "WE MUST CHECK THE LEVEL OF YOUR ENGISH" << endl;

cout << "Tap 1 to pass it" << endl;

cout << "Tap 0 to cancel if you aren't ready" << endl;

cin >> call;

system("cls");

switch (call)

{

case 1:

{

engtest();

}

break;

}

}

break;

}

}

break;

}

case 5:

{

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_JAVASCRIPT\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((10 << 4) | 9));

cout << "| JavaScript is one of the most |" << endl;

cout << "| popular and cross-platform |" << endl;

cout << "| programming languages that |" << endl;

cout << "| allows working with web |" << endl;

cout << "| interfaces, server side, |" << endl;

cout << "| and mobile clients |" << endl;

cout << "| development acceleration. |" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_START DATE\_\_\_|\_\_\_\_END DATE \_\_|\_\_\_\_\_FREQUENCY\_\_\_\_\_\_\_\_|\_\_LOCATION\_\_\_|" << endl;

cout << "|\_\_\_\_\_\_2020 MAY\_\_\_|\_\_\_\_2020 AUG\_\_\_|\_\_2-3 TIMES PER WEEK\_\_|\_\_\_UKRAINE\_\_\_|" << endl;

cout << "| |REGISTRATION - 1| |BACK - 0| |" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

int call = -1;

while (call != 0)

{

cin >> call;

system("cls");

switch (call)

{

case 1:

{

cout << "WE MUST CHECK THE LEVEL OF YOUR ENGISH" << endl;

cout << "Tap 1 to pass it" << endl;

cout << "Tap 0 to cancel if you aren't ready" << endl;

cin >> call;

system("cls");

switch (call)

{

case 1:

{

engtest();

}

break;

}

}

break;

}

}

break;

}

case 6:

{

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_UX DESIGN\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((10 << 4) | 9));

cout << "| UX Design is designing |" << endl;

cout << "| effective user experience |" << endl;

cout << "| interaction with websites, |" << endl;

cout << "| computers, |" << endl;

cout << "| and mobile applications. |" << endl;

cout << "|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "|\_\_\_\_\_\_START DATE\_\_\_|\_\_\_\_END DATE \_\_|\_\_\_\_\_FREQUENCY\_\_\_\_\_\_\_\_|\_\_LOCATION\_\_\_|" << endl;

cout << "|\_\_\_\_\_\_2020 MAY\_\_\_|\_\_\_\_2020 AUG\_\_\_|\_\_2-3 TIMES PER WEEK\_\_|\_\_\_UKRAINE\_\_\_|" << endl;

cout << "| |REGISTRATION - 1| |BACK - 0| |" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

int call = -1;

while (call != 0)

{

cin >> call;

system("cls");

switch (call)

{

case 1:

{

cout << "WE MUST CHECK THE LEVEL OF YOUR ENGISH" << endl;

cout << "Tap 1 to pass it" << endl;

cout << "Tap 0 to cancel if you aren't ready" << endl;

cin >> call;

system("cls");

switch (call)

{

case 1:

{

engtest();

}

break;

}

}

break;

}

}

break;

}

}

}

}

};

**main.cpp**

#include <iostream>

#include <string>

#include <fstream>

#include <Windows.h>

#include <vector>

#include <ctime>

#include <sstream>

#include <conio.h>

#include <list>

#include <iterator>

#include<iomanip>

#include"Company.h"

#include"RegLogin.h"

#include"Vacancies.h"

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

enum ConsoleColor {

Black = 0,

Blue = 1,

Green = 2,

Cyan = 3,

Red = 4,

Magenta = 5,

Brown = 6,

LightGray = 7,

DarkGray = 8,

LightBlue = 9,

LightGreen = 10,

LightCyan = 11,

LightRed = 12,

LightMagenta = 13,

Yellow = 14,

White = 15

};

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

srand(time(NULL));

Company UZ;

RegLogin DO;

Vacancies XXX;

string actions[] = { "Transformative consulting built","Automotive & Manufacturing","Travel & Hospitality","Media, Entertainment & Telecom","Business Information Services","Software & Hi-Tech","Insurance","Team of like-minded professionals","Healthcare" };

string vacancy[] = { "HR manger",".NET DevOps Engineer",".NET Full Stack Developer","Azure Solution Architect","Front-End Developer","Full Stack Java Developer","IT Support Analyst","Java Backend Engineer" };

int length = (sizeof(actions) / sizeof(string));

int num = -1;

while (num != 0)

{

system("cls");

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << " IT-Company 'ProEasyCode Inc.' " << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((11 << 7) | 1));

cout << "1. Main page.\n2. Get project.\n3. Career.\n4. What we do.\n5. Help.\n6. For administrator.\n7. Log in as user.\n8. Log in as admin.\n0. Exit\n" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << " Waiting for our collaboration ! " << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "Select the iteam: ";

cin >> num;

if (num > 8 || num < 0)

{

system("cls");

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)((12 << 4) | 5));

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

cout << "|~~~~~The page was deleted or something went wrong!~~~~~~|" << endl;

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

system("pause");

}

else

{

switch (num)

{

case 1:

{

UZ.info();

break;

}

case 2:

{

UZ.WithStaticPrice();

break;

}

case 3:

{

system("cls");

int num = -1;

while (num != 0)

{

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((11 << 7) | 1));

cout << "1. Vacancy.\n2. Training list.\n3. Leave resume.\n0. Back to the main page.\n" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cout << "Select the item: ";

cin >> num;

if (num > 3 || num < 0)

{

system("cls");

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)((12 << 4) | 5));

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

cout << "|~~~~~The page was deleted or something went wrong!~~~~~~|" << endl;

cout << "|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|" << endl;

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

break;

}

switch (num)

{

case 1:

{

XXX.vacshow();

break;

}

case 2:

{

system("cls");

UZ.LeaveDet();

break;

}

case 3:

{

UZ.StayAPart();

break;

}

case 0:

{

system("cls");

break;

}

}

}

system("cls");

break;

}

case 4:

{

UZ.actshow(actions, length);

system("cls");

break;

}

case 5:

{

UZ.question();

break;

}

case 6:

{

int num = -1;

while (num != 0)

{

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

SetConsoleTextAttribute(hConsole, (WORD)((11 << 7) | 1));

cout << "1.Delete vacancy.\n2.Add new vacancy.\n3.Delete actions.\n0.Back." << endl;

SetConsoleTextAttribute(hConsole, (WORD)((1 << 4) | 7));

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

SetConsoleTextAttribute(hConsole, (WORD)((15 << 2) | 6));

cin >> num;

switch (num)

{

case 1:

{

XXX.vacdelete();

system("pause");

break;

}

case 2:

{

XXX.addvacancy();

break;

}

case 3:

{

string \*newactions;

cout << length;

newactions = UZ.Admin(actions, length);

for (int i = 0; i < length - 1; i++)

{

actions[i] = newactions[i];

}

length--;

break;

}

break;

}

break;

}

break;

}

case 7:

{

DO.login();

break;

}

case 8:

{

DO.admin();

break;

}

}

}

}

}