Міністерство освіти і науки України

Інженерний інститут

Запорізького національного університету

Кафедра програмного забезпечення автоматизованих систем

**КУРСОВА РОБОТА**

**з курсу Операційні системи**

**тема:** Програма збору інформації про систему

Спеціальність 121 „Інженерія програмного забезпечення”

Виконав студент

групи ІПЗ-16-бд. Розов Б.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник роботи

Лимаренко Ю.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії: доц. Лимаренко Ю.О.

Запоріжжя

2018 р.

# Реферат

* Об'єм роботи –
* Кількість рисунків –
* Таблиць –
* Додатків –
* Кількість використаних джерел –

Метою написання курсової роботи з предмету "Операційні системи" було створення утиліти, що дає можливість переглянути різну інформацію про систему, таку як: наявні девайси, встановлене програмне забезпечення та інше.

В результаті розробки був отриманий програмний продукт, створений на мові програмування C++ , що дозволяє виводити інформацію про систему

* ключові слова, написані прописними буквами через кому (близько 10 слів та словосполучень, що найчастіше зустрічаються в тексті та відповідають темі роботи).

# Зміст

[1. Реферат 2](#_Toc11672277)

[2. Зміст 3](#_Toc11672278)

[4. Вступ 4](#_Toc11672279)

[4.1. Глосарій 4](#_Toc11672280)

[4.2. Опис наочної області 4](#_Toc11672281)

[4.2.1. Дослідницька проблема 4](#_Toc11672282)

[4.3. Постановка задачі 4](#_Toc11672283)

[4.4. Огляд існуючих методів рішення 4](#_Toc11672284)

[4.4.1. Аналогічні рішення 4](#_Toc11672285)

[4.4.2. Висновок 4](#_Toc11672286)

[4.5. План робіт 4](#_Toc11672287)

[5. Вимоги до оточення 5](#_Toc11672288)

[5.1. Вимоги до апаратного забезпечення 5](#_Toc11672289)

[5.2. Вимоги до програмного забезпечення 5](#_Toc11672290)

[5.3. Вимоги до користувачів 5](#_Toc11672291)

[5.4. Організаційні вимоги 5](#_Toc11672292)

[6. Архітектура системи 6](#_Toc11672293)

[7. Специфікація даних 7](#_Toc11672294)

[7.1. Опис формату та/або структури даних 7](#_Toc11672295)

[7.2. Опис сутностей 7](#_Toc11672296)

[7.3. Опис протоколу 7](#_Toc11672297)

[8. Функціональні вимоги 8](#_Toc11672298)

[8.1. Загальні вимоги 8](#_Toc11672299)

[8.2. Бібліотека підпрограм (класів) 8](#_Toc11672300)

[9. Вимоги до інтерфейсу 9](#_Toc11672301)

[10. Інші вимоги 10](#_Toc11672302)

[10.1. Вимоги до надійності 10](#_Toc11672303)

[10.2. Вимоги до продуктивності 10](#_Toc11672304)

[11. Проект програмної системи 11](#_Toc11672305)

[11.1. Засоби реалізації 11](#_Toc11672306)

[11.2. Модулі і алгоритми 11](#_Toc11672307)

[11.3. Структури даних 11](#_Toc11672308)

[11.4. Проект інтерфейсу 11](#_Toc11672309)

[11.5. Реалізація і тестування 11](#_Toc11672310)

[12. Висновки 12](#_Toc11672311)

[13. Список літератури 13](#_Toc11672312)

[14. Додатки 14](#_Toc11672313)

# Вступ

Існує декілька вбудованих програм що дозволяють переглянути необхідну користувачу інформацію про систему: Програми та компоненти, диспетчер задач, диспетчер приладів.

На відміну від вбудованих програмних засобів, розроблюваний програмний продукт дає можливість переглянути усю необхідну у більшості випадків інформацію використовуючи лише одну застосунок.

## Глосарій

**Common Language Runtime**, скорочено **CLR** — «загальномовне виконуюче середовище» — це компонент пакету Microsoft .NET Framework, віртуальна машина, на якій виконуються всі мови платформи .NET Framework.

CLR транслює початковий код в байт-код на мові IL, реалізація компіляції якого компанією Microsoft називається MSIL, а також надає MSIL-програмам (а отже, і програмам, написаним на мовах високого рівня, що підтримують .NET Framework) доступ до бібліотеки класів .NET Framework, або так званою .NET FCL (англ. Framework Class Library).

Середовище CLR є реалізацією специфікації CLI (англ. Common Language Infrastructure), специфікації загальномовної інфраструктури, компанією Microsoft.

Віртуальна машина CLR дозволяє програмістам забути про багато деталей конкретного процесора, на якому виконуватиметься програма. CLR також забезпечує такі важливі служби як:

* управління пам'яттю
* управління потоками
* обробка винятків
* збірка сміття
* безпека виконання

**Windows Management Instrumentation** (**WMI**) в дослівному перекладі — інструментарій управління Windows. WMI — це одна з базових технологій для централізованого управління і стеження за роботою різних частин комп'ютерної інфраструктури під управлінням платформи Windows.

В основі структури даних в WBEM лежить Common Information Model (CIM), що реалізує об'єктно-орієнтований підхід до подання компонентів системи. CIM є розширюваною моделлю, що дозволяє програмам, системам і драйверам додавати в неї свої класи, об'єкти, методи і властивості.

WMI, заснований на CIM, також є відкритою уніфікованою системою інтерфейсів доступу до будь-яких параметрах операційної системи, пристроїв і додатків, які функціонують у ній.

Тест продуктивності, або **бенчмарк** (англ. benchmark) — контрольне завдання, необхідне для визначення порівняльних характеристик продуктивності комп'ютерної системи. Існують програми для тесту продуктивності системи, що тестують час автономної роботи ноутбуків і КПК, радіус дії бездротової мережі, пропускну здатність каналів передачі даних, АЧХ звукового тракту та інші доступні для вимірювання характеристики, що не пов'язані напряму з продуктивністю.

## Опис наочної області

### Дослідницька проблема

## Постановка задачі

Програма збору інформації про систему. Можливості її включають в себе повний набір відомостей про комп'ютер.

Функціонально присутні закладки:

* System,
* Memory,
* Software,
* Drives,
* I/O Devices
* Summary

Розділ System містить відомості про вміст комп'ютера:

* процесор,
* оперативна пам'ять,
* відеосистемі,
* диск,
* шині PCI,
* Plug and Play-пристроях,
* BIOS і CMOS.

Закладка Memory містить інформацію про оперативну пам'ять:

* пам'яті, займаної різними процесами;
* розподіл фізичної пам'яті;
* системної і розділяється пам'яті;
* вміст першого адресуемого мегабайта.

Закладка Software показує програмне забезпечення, встановлене на комп'ютері.

Summary покаже версію операційної системи, загальну інформацію по запущеним процесам, ниткам, віртуальним машинам.

Наступна закладка Drives містить інформацію про диски:

* флоппі,
* вінчестер,
* CD-ROM та ін.

Закладка I/O це дані про мережу, звукових адаптерах, послідовних і паралельних портах, принтерах.

## Огляд існуючих методів рішення

### Аналогічні рішення

**Aida64** – застосунок компанії FinalWire Ltd. для тестування та ідентифікації компонентів персонального комп'ютера під управлінням операційних систем Windows, що надає детальні відомості про апаратне та програмне забезпечення.

* Основні можливості - детальний аналіз системних пристроїв, моніторинг температури та енергії, різні бенчмарки для перевірки потужності процесору.
* Користувацький інтерфейс - складається переважно с верхнього меню, бокової ієрархічної навігації зліва та основної відображаїмої інформації справа.
* Підтримувані платформи - Windows, Anіdroid, iOS, Windows Phone, Tizen, Chrome OS and Sailfish OS operating systems
* 30 днів пробного періоду, ціна базової версії складає $39.95

**CrystalDiskInfo –** безкоштовна утиліта з відкритим вихідним кодом, що розробляється програмістом з Японії Noriyuki Miyazaki. Призначена для діагностики роботи жорстких дисків ПК. У процесі роботи програми відображається загальна інформація про HDD, ведеться моніторинг значень S.M.A.R.T., а також здійснюється постійний контроль температури диска.

* Основні можливості - детальних моніторинг фізичних (вбудованих та зовнішніх) носіїв
* Користувацький інтерфейс - Меню зверху, перелік системних дисків, відображення стану та інформації про обраний диск.
* Підтримувані платформи - Операційна система Windows починаючи з версії XP
* Open source

**CCleaner (раніше Crap Cleaner) —** безкоштовна утиліта із закритим вихідним кодом, яка надає користувачам потужний і простий у використанні інструмент для очищення і оптимізації 32- та 64-розрядних операційних систем Microsoft Windows. Утиліта була створена британською приватною фірмою Piriform Limited і написана на C++.

* Основні можливості – моніторинг встановлених програм с можливостю видалення, очистка реєстру та кешу браузерів.
* Користувацький інтерфейс – головне меню зліва та основна частина по центру
* Підтримувані платформи – Windows, Mac OS, Android
* Безкоштовна базова версія, вартість повної - 570.04 гривні

### Висновок

На даний момент існую декілька крупних програмних застосунків що надають можливість користування однією або двома необхідними за технічним завданням функціями, проте жодне з існуючих рішень не дає змоги одразу використовувати усі необхідні можливості.

## План робіт

01.02 – 01.03 Розроблення технічного завдання

02.03 – 10.03 Складання плану розробки

11.03 – 25.03 Початок розробки

26.03 Перший реліз (alpha-версія)

27.03 – 10.04 Подальша розробка

11.04 Другий реліз (beta-версія)

12.04 – 20.05 Розробка релізной версії програми

21.05 – 18.06 Тестування релізної версії

19.06 Презентація фінальної версії програмного продукту

# Вимоги до оточення

## Вимоги до апаратного забезпечення

Мінімальні вимоги до апаратного забезпечення:

* **Система** - Windows XP Sр3
* **Процесор** - тактова частотою не менше 300 MHz.
* **Оперативна пам’ять** - 128 Мб.
* **Жорсткий диск** - 100Мб

Рекомендовані вимоги до апаратного забезпечення:

* **Система** - Windows 7
* **Процесор** - 1 ГГц.
* **Оперативна пам’ять** - 1024Мб.
* **Жорсткий диск** - 100Мб
* **Графічний процесор** - 128 Мб пам'яті (адаптер з підтримкою DirectX 9), драйвери WDDM версії 1.0 і старше.

## Вимоги до програмного забезпечення

ОС Windows, починаючи з Windows 7 та вище.

## Вимоги до користувачів

Користувач повинен пройти базовий курс користування операційною системою Windows та розуміти використовуємо термінологію

# Архітектура системи

Система складається з шести головних частин:

1. System,
2. Memory,
3. Software,
4. Drives,
5. I/O Devices
6. Summary

Після проведення аналізу вимог було вирішено використовувати наступну архітектуру

## Варіанти користування

## 

Рис. 1 Діаграма варіантів використання

## Діаграма класів

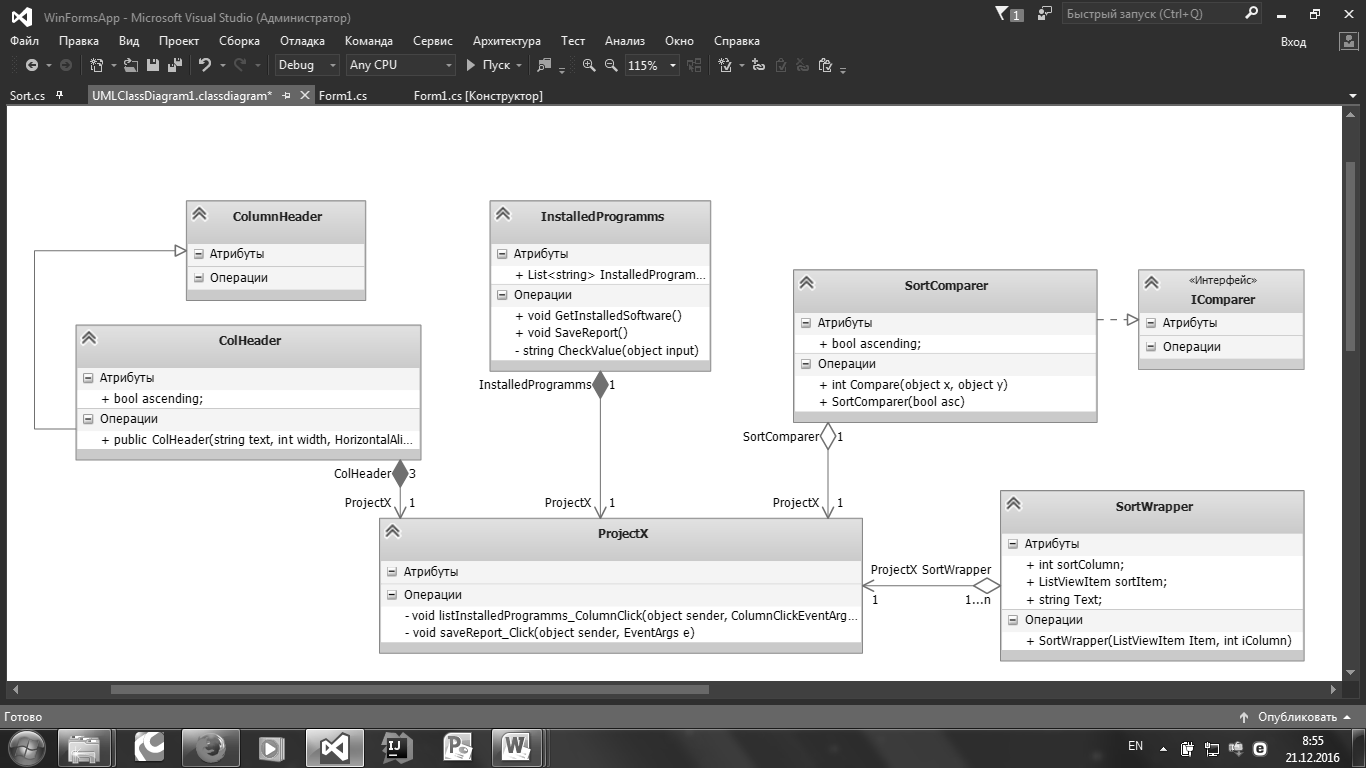


Рис.2 Діаграма класів

# Специфікація даних

## Опис формату та/або структури даних

## Опис сутностей

## Опис протоколу

# Функціональні вимоги

## Загальні вимоги

## Бібліотека підпрограм (класів)

# Вимоги до інтерфейсу

# Інші вимоги

## Вимоги до надійності

## Вимоги до продуктивності

# Проект програмної системи

## Засоби реалізації

## Модулі і алгоритми

## Структури даних

## Проект інтерфейсу

## Реалізація і тестування

# Висновки

# Список літератури

# Додатки