

ИЗ-1.2

1. Введение

Цель работы: Исследовать способы автоматизации установки ПО в ОС Windows и разработать скрипт для тихой (silent) установки необходимого программного обеспечения для компьютерных аудиторий кафедры.

Задачи:

1. Исследовать методы автоматической (тихой) установки программ в Windows.
2. Составить список необходимого ПО на основе предоставленного перечня.
3. Найти официальные дистрибутивы и определить параметры командной строки для их тихой установки.
4. Разработать скрипт (BAT-файл), автоматически устанавливающий всё ПО.
5. Реализовать проверку успешности установки и необходимость перезагрузки.

2. Исследование способов автоматизации установки ПО

В ходе работы были исследованы следующие основные методы автоматической установки в ОС Windows:

1. **Исполняемые файлы (EXE):** Большинство установщиков поддерживают параметры командной строки для тихой установки.
 - **Распространённые ключи:**
 - `/S` или `/VERYSILENT` — полностью тихая установка без окон.
 - `/NORESTART` — запрещает перезагрузку после установки.
 - `/DIR="<путь>"` — указание пути установки.
 - `/COMPONENTS="..."` — выбор компонентов для установки.
2. **Установщики Windows (MSI):** Файлы с расширением `.msi` управляются утилитой `msiexec.exe`.
 - **Ключи `msiexec`:**
 - `/i <файл.msi>` — установка пакета.
 - `/quiet` или `/qn` — тихая установка без интерфейса.
 - `/norestart` — отложить перезагрузку.
3. **Менеджеры пакетов:** Инструменты для централизованной установки ПО, такие как **Chocolatey**. Позволяют устанавливать программы одной командой вида `choco install <имя_пакета> -y`.
4. **Скрипты:** Пакетные файлы (BAT/CMD) или скрипты PowerShell позволяют комбинировать все вышеперечисленные методы в единый процесс установки.

3. Список программного обеспечения для установки

На основе предоставленного списка и анализа скриншотов был сформирован следующий перечень ПО для автоматической установки:

№	Наименование ПО	Способ установки в скрипте
1	Visual Studio Code	EXE (/VERYSILENT)
2	Docker Desktop	Chocolatey
3	PyCharm Community Edition	EXE (/S)
4	Git для Windows	EXE (/VERYSILENT)
5	GitHub Desktop	EXE (--silent)
6	Python 3.13	EXE (/quiet)
7	Rust	EXE (rustup, -y -quiet)
8	MSYS2	EXE (install --quiet)
9	Anaconda	EXE (/S)
10	TeXstudio	Chocolatey
11	Far Manager	MSI (/quiet)
12	Sumatra PDF	Chocolatey
13	Google Chrome	EXE (/silent /install)
14	Mozilla Firefox	MSI (/quiet)
15	Подсистема WSL 2 (Ubuntu)	Встроенная утилита (wsl --install)
16	Maxima	Chocolatey
17	KNIME Analytics Platform	Распаковка ZIP-архива
18	Julia	EXE (/S)
19	Flameshot	Chocolatey
20	Qalculate	Chocolatey

Дополнительно: Скрипт автоматически устанавливает набор расширений для VS Code (Python, C++, Docker, Web-технологии, Julia, Rust).

4. Листинг кода скрипта установки

Был разработан пакетный файл `install-software.bat`.

```
@echo off
chcp 65001 >nul
setlocal enabledelayedexpansion

echo =====
```

```

echo      АВТОМАТИЗИРОВАННАЯ УСТАНОВКА ПО
echo      Кафедра информационных технологий
echo      =====
echo.

:: Проверка прав администратора
net session >nul 2>&1
if %errorLevel% neq 0 (
    echo ОШИБКА: Требуются права администратора!
    echo Запустите скрипт от имени администратора.
    pause
    exit /b 1
)

:: Создание папки для загрузок
if not exist "C:\IT-Downloads" mkdir "C:\IT-Downloads"
set DOWNLOAD_DIR=C:\IT-Downloads

:: Функция для проверки успешности установки
:CheckSuccess
if %errorlevel% equ 0 (
    echo [УСПЕХ] Установка завершена успешно
) else (
    echo [ОШИБКА] Код ошибки: %errorlevel%
)
echo.
goto :eof

:: Функция для загрузки файлов
:DownloadFile
set "url=%~1"
set "output=%~2"
echo Загрузка: %output%
powershell -Command "(New-Object Net.WebClient).DownloadFile('%url%', '%output%')"
goto :eof

echo Установка менеджера пакетов Chocolatey...
@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -
InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))"
&& SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"
call :CheckSuccess

echo.
echo ===== УСТАНОВКА ОСНОВНОГО ПО =====
echo.

:: 1. Visual Studio Code
echo Установка Visual Studio Code...

```

```
call :DownloadFile "https://code.visualstudio.com/sha/download?
build=stable&os=win32-x64-user" "%DOWNLOAD_DIR%\VSCodeSetup-x64.exe"
"%DOWNLOAD_DIR%\VSCodeSetup-x64.exe" /VERYSILENT /NORESTART
/MERGETASKS=!runcode
call :CheckSuccess

:: 2. Docker Desktop
echo Установка Docker Desktop...
choco install docker-desktop -y --force
call :CheckSuccess

:: 3. PyCharm Community Edition
echo Установка PyCharm Community...
call :DownloadFile "https://download.jetbrains.com/python/pycharm-community-
2023.3.2.exe" "%DOWNLOAD_DIR%\pycharm-community.exe"
"%DOWNLOAD_DIR%\pycharm-community.exe" /S
call :CheckSuccess

:: 4. Git
echo Установка Git...
call :DownloadFile "https://github.com/git-for-
windows/git/releases/download/v2.51.0/windows.1/Git-2.51.0-64-bit.exe"
"%DOWNLOAD_DIR%\Git-Install.exe"
"%DOWNLOAD_DIR%\Git-Install.exe" /VERYSILENT /NORESTART
/COMPONENTS="icons,ext\reg\shellhere,assoc,assoc_sh"
call :CheckSuccess

:: 5. GitHub Desktop
echo Установка GitHub Desktop...
call :DownloadFile
"https://central.github.com/deployments/desktop/desktop/latest/win32?
format=exe" "%DOWNLOAD_DIR%\GitHubDesktopSetup.exe"
"%DOWNLOAD_DIR%\GitHubDesktopSetup.exe" --silent
call :CheckSuccess

:: 6. Python 3.13
echo Установка Python 3.13...
call :DownloadFile "https://www.python.org/ftp/python/3.13.0/python-3.13.0-
amd64.exe" "%DOWNLOAD_DIR%\python-3.13.exe"
"%DOWNLOAD_DIR%\python-3.13.exe" /quiet InstallAllUsers=1 PrependPath=1
Include_test=0
call :CheckSuccess

:: 7. Rust
echo Установка Rust...
call :DownloadFile "https://win.rustup.rs/x86_64" "%DOWNLOAD_DIR%\rustup-
init.exe"
"%DOWNLOAD_DIR%\rustup-init.exe" -y --quiet
call :CheckSuccess
```

:: 8. MSYS2

echo Установка MSYS2...

call :DownloadFile "https://github.com/msys2/msys2-installer/releases/download/2023-03-18/msys2-x86_64-20230318.exe"
"%DOWNLOAD_DIR%\msys2.exe"

"%DOWNLOAD_DIR%\msys2.exe" install --root C:\msys64 --confirm-command --quiet

call :CheckSuccess

:: 9. Anaconda

echo Установка Anaconda...

call :DownloadFile "https://repo.anaconda.com/archive/Anaconda3-2023.09-0-Windows-x86_64.exe" "%DOWNLOAD_DIR%\Anaconda3.exe"

"%DOWNLOAD_DIR%\Anaconda3.exe" /S /RegisterPython=1 /AddToPath=1

call :CheckSuccess

:: 10. TeXstudio

echo Установка TeXstudio...

choco install texstudio -y

call :CheckSuccess

:: 11. Far Manager

echo Установка Far Manager...

call :DownloadFile

"https://farmanager.com/files/Far30b6100.x64.20250301.msi"
"%DOWNLOAD_DIR%\FarManager.msi"

msiexec /i "%DOWNLOAD_DIR%\FarManager.msi" /quiet /norestart

call :CheckSuccess

:: 12. Sumatra PDF

echo Установка Sumatra PDF...

choco install sumatrapdf -y

call :CheckSuccess

:: 13. Chrome

echo Установка Google Chrome...

call :DownloadFile "https://dl.google.com/tag/s/appguid%3D%7B8A69D345-D564-463C-AFF1-A69D9E530F96%7D%26iid%3D%7B2B35E9A7-D3DB-37E7-8A33-7C3232B2C9E2%7D%26lang%3Dru%26browser%3D%26usagestats%3D1%26appname%3DGoogle%2520Chrome%26needsadmin%3Dprefers%26ap%3Dx64-stable-statsdef_1%26installdataindex%3Dempty/chrome/install/ChromeStandaloneSetup64.exe" "%DOWNLOAD_DIR%\ChromeSetup.exe"

"%DOWNLOAD_DIR%\ChromeSetup.exe" /silent /install

call :CheckSuccess

:: 14. Firefox

echo Установка Firefox...

call :DownloadFile "https://download.mozilla.org/?product=firefox-msi-latest-ssl&os=win64&lang=ru" "%DOWNLOAD_DIR%\Firefox.msi"

msiexec /i "%DOWNLOAD_DIR%\Firefox.msi" /quiet /norestart

```
call :CheckSuccess
```

```
:: 15. WSL 2 с Ubuntu
```

```
echo Установка WSL 2 и Ubuntu...
```

```
wsl --install -d Ubuntu-22.04 --quiet
```

```
call :CheckSuccess
```

```
echo.
```

```
echo ===== УСТАНОВКА ДОПОЛНИТЕЛЬНОГО ПО =====
```

```
echo.
```

```
:: Maxima
```

```
echo Установка Maxima...
```

```
choco install maxima -y
```

```
call :CheckSuccess
```

```
:: KNIME Analytics Platform
```

```
echo Установка KNIME...
```

```
call :DownloadFile "https://download.knime.org/analytics-  
platform/windows/knime_5.7.0.win32.win32.x86_64.zip"
```

```
"%DOWNLOAD_DIR%\knime.zip"
```

```
powershell -Command "Expand-Archive -Path '%DOWNLOAD_DIR%\knime.zip' -  
DestinationPath 'C:\Program Files\KNIME' -Force"
```

```
call :CheckSuccess
```

```
:: Julia
```

```
echo Установка Julia...
```

```
call :DownloadFile "https://julialang-  
s3.julialang.org/bin/winnt/x64/1.9/julia-1.9.3-win64.exe"
```

```
"%DOWNLOAD_DIR%\JuliaSetup.exe"
```

```
"%DOWNLOAD_DIR%\JuliaSetup.exe" /S /D=C:\Julia
```

```
call :CheckSuccess
```

```
:: Flameshot
```

```
echo Установка Flameshot...
```

```
choco install flameshot -y
```

```
call :CheckSuccess
```

```
:: Qalculate
```

```
echo Установка Qalculate...
```

```
choco install qalculate -y
```

```
call :CheckSuccess
```

```
echo.
```

```
echo ===== УСТАНОВКА РАСШИРЕНИЙ VS CODE =====
```

```
echo.
```

```
:: Установка расширений для VS Code
```

```
echo Установка расширений для VS Code...
```

```
code --install-extension ms-python.python
```

```

code --install-extension ms-vscode.cpptools
code --install-extension ms-azuretools.vscode-docker
code --install-extension ecmel.vscode-html-css
code --install-extension xabikos.JavaScriptSnippets
code --install-extension ms-vscode.PowerShell
code --install-extension julialang.language-julia
code --install-extension rust-lang.rust-analyzer

echo.
echo ===== НАСТРОЙКА СИСТЕМЫ =====
echo.

:: Добавление в PATH
setx PATH "%PATH%;C:\Julia\bin;C:\msys64\usr\bin" /M

:: Создание ярлыков на рабочем столе
echo Создание ярлыков...
powershell -Command "
$WshShell = New-Object -comObject WScript.Shell;
$shortcut = $WshShell.CreateShortcut('$Home\Desktop\VS Code.lnk');
$shortcut.TargetPath = 'C:\Users%\USERNAME%\AppData\Local\Programs\Microsoft
VS Code\Code.exe';
$shortcut.Save()
"

echo.
echo ===== ПРОВЕРКА УСТАНОВКИ =====
echo.

echo Проверка установленных программ:
where code >nul && echo ✓ VS Code установлен
where git >nul && echo ✓ Git установлен
where python >nul && echo ✓ Python установлен
where docker >nul && echo ✓ Docker установлен

echo.
echo ===== ЗАВЕРШЕНИЕ =====
echo.

set /p reboot="Требуется перезагрузка для завершения установки.
Перезагрузить сейчас? (y/n): "
if /i "%reboot%"=="y" (
    echo Перезагрузка через 30 секунд...
    shutdown /r /t 30
) else (
    echo Перезагрузка не требуется. Установка завершена.
)

echo.

```

```
echo Установка завершена! Лог сохранен в C:\IT-Installation.log
pause
```

5. Комментарии по выполнению и использованию скрипта

5.1. Ключевые особенности реализации

- **Универсальность:** Скрипт сочетает прямое скачивание дистрибутивов, использование менеджера пакетов Chocolatey и системных утилит (msiexec, wsl).
- **Отказоустойчивость:** После каждой операции выполняется проверка кода возврата (%errorlevel%) для информирования о результате.
- **Тихая установка:** Все инсталляторы запускаются с параметрами, подавляющими графический интерфейс и запросы пользователя.
- **Автоматизация загрузки:** Скрипт сам скачивает все необходимые дистрибутивы в папку C:\IT-Downloads , что избавляет от их предварительной подготовки.
- **Последующая настройка:** Скрипт не только устанавливает ПО, но и настраивает среду: добавляет пути в переменную PATH , создаёт ярлыки, устанавливает расширения VS Code.

5.2. Инструкция по применению

1. **Запустите файл от имени Администратора.** Это критически важно для установки программ в системные каталоги и изменения переменных среды.
2. Дождитесь полного выполнения скрипта. Процесс может занять значительное время в зависимости от скорости интернета и мощности компьютера.
3. В конце скрипт предложит выполнить перезагрузку. Для корректной работы Docker и WSL перезагрузка **настоятельно рекомендуется**.

5.3. Потенциальные проблемы и пути их решения

- **Антивирусное ПО:** Может блокировать выполнение скриптов или загрузку файлов. На время установки рекомендуется отключить его или добавить папку C:\IT-Downloads в исключения.
- **Брандмауэр:** Может препятствовать загрузке файлов. Требуется стабильное интернет-соединение.
- **Конфликт версий:** Если какая-либо программа уже установлена, её установка может быть пропущена. Скрипт разработан для чистой системы.
- **Необходимость перезагрузки:** Некоторые компоненты (Docker, WSL) требуют перезагрузки для своей окончательной настройки. Скрипт предлагает сделать это в конце.

6. Заключение

В ходе выполнения были успешно решены все поставленные задачи:

1. Исследованы и опробованы на практике методы тихой установки ПО в ОС Windows.
2. На основе анализа предоставленного списка составлен актуальный перечень программного обеспечения.
3. Для каждого пункта списка найдены официальные источники дистрибутивов и параметры командной строки.
4. Разработан комплексный скрипт `install-software.bat`, который полностью автоматизирует процесс установки и начальной настройки всего необходимого ПО.
5. Скрипт информирует пользователя о процессе установки, проверяет успешность выполнения операций и управляет необходимостью перезагрузки системы.

Данный скрипт может быть успешно использован для быстрого развёртывания единой программной среды на компьютерах в аудиториях кафедры, что значительно экономит время и силы сотрудников и обеспечивает одинаковую конфигурацию рабочих мест