

Programowanie systemowe - projekt

Aplikacja obsługująca tworzenie kopii zapasowych danych w środowisku systemu windows.

dr inż. Mariusz Wnęk

IPPP32:
Jakub Skowron
Jacek Barć
Radosław Loth

Sprawozdanie z projektu

1. Lista zastosowanych narzędzi programistycznych

Podczas realizacji projektu wykorzystano zintegrowane środowisko programistyczne Visual Studio 2019 wraz ze zintegrowanym systemem kontroli wersji git, edytorem i kompilatorem.

2. Tabela funkcjonalności

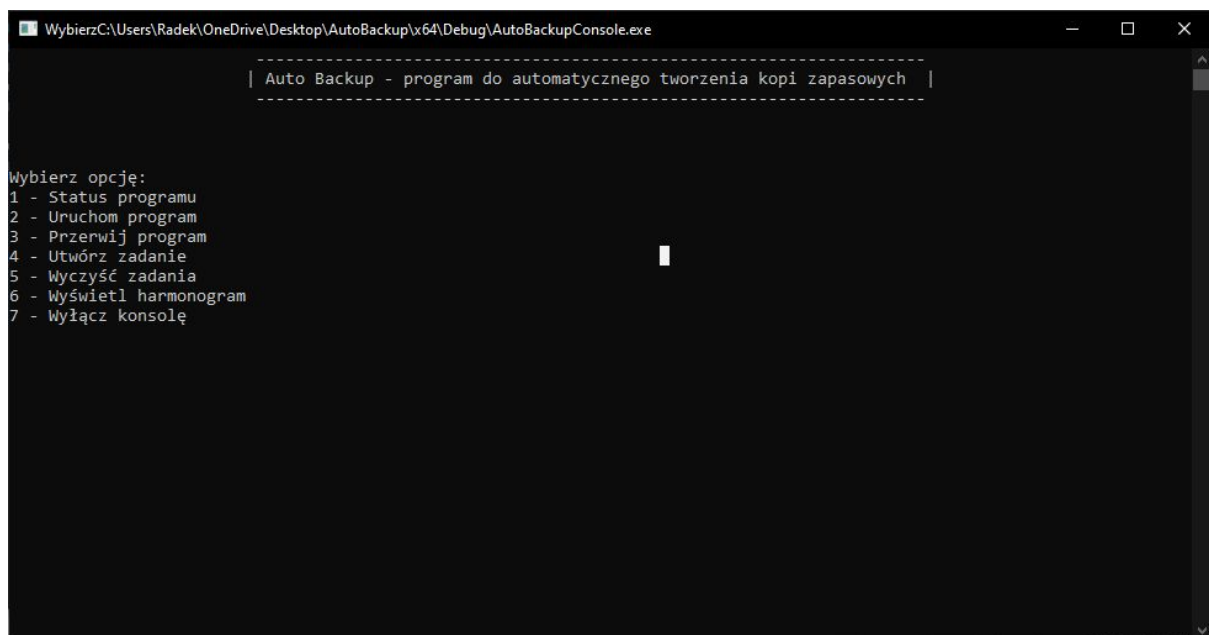
Funkcjonalność	Osoba odpowiedzialna	Status
Aplikacja konsolowa do zarządzania aplikacją wykonującą backupy	Jakub Skowron	ukończone
Zapis i odczyt z pliku harmonogramu oraz przekazanie struktury kopii zapasowej do wykonania	Jakub Skowron	ukończone
Przesunięcie kopii zapasowej o określony interwał	Jakub Skowron	ukończone
Wykonywanie kopii zapasowej w wielu wątkach	Radosław Loth	ukończone
Czyszczenie pliku harmonogramu	Radosław Loth	ukończone
Wykonywanie kopii zapasowej plików	Jacek Barć	ukończone
Wykonywanie kopii zapasowej folderów wraz z zawartością	Jacek Barć	ukończone
Wykonywanie kopii zapasowej z kompresją	Jacek Barć	nieukończone

3. Dokumentacja napisanego kodu źródłowego oraz opis funkcjonalności i przeznaczenia utworzonego oprogramowania

Opis przeznaczenia:

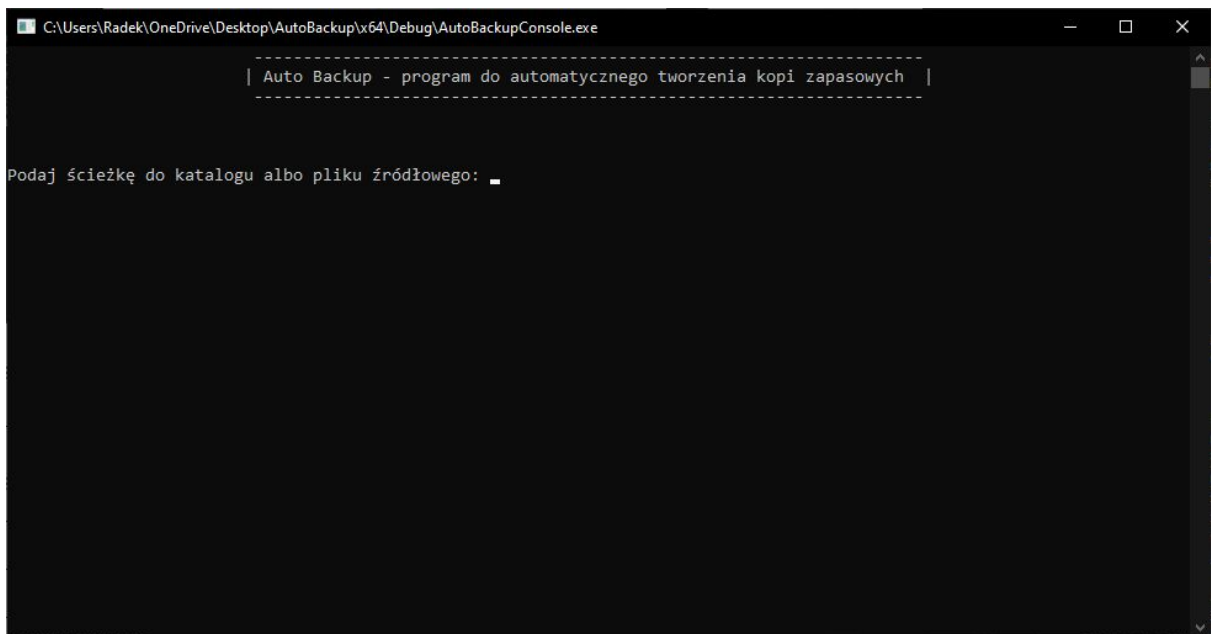
Auto Backup - program do automatycznego tworzenia kopii zapasowych.

Program umożliwia zaplanowanie oraz automatyczne wykonywanie kopii zapasowych.

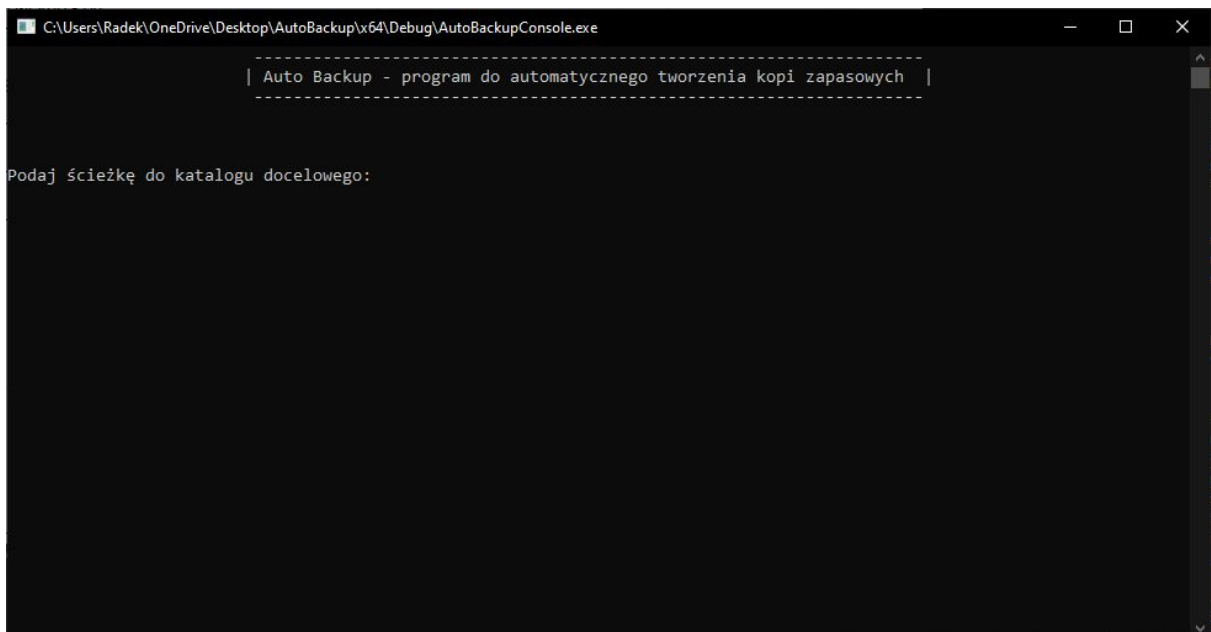


Opis funkcjonalności:

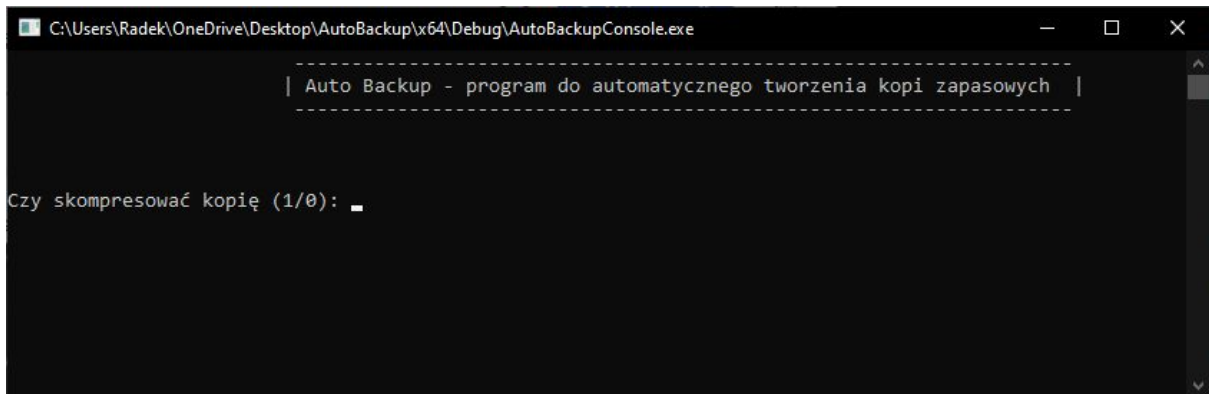
1. **Status programu** - pozwala na sprawdzenie czy proces odpowiadający za wywołanie wykonania kopii zapasowej jest uruchomiony w tle.
2. **Uruchom program** - pozwala na uruchomienie procesu odpowiadającego za wywołanie kopii zapasowej.
3. **Przerwij program** - pozwala na zatrzymanie procesu odpowiadającego za wywołanie kopii zapasowej.
4. **Utwórz zadanie** - pozwala na stworzenie nowego zaplanowanego działania.



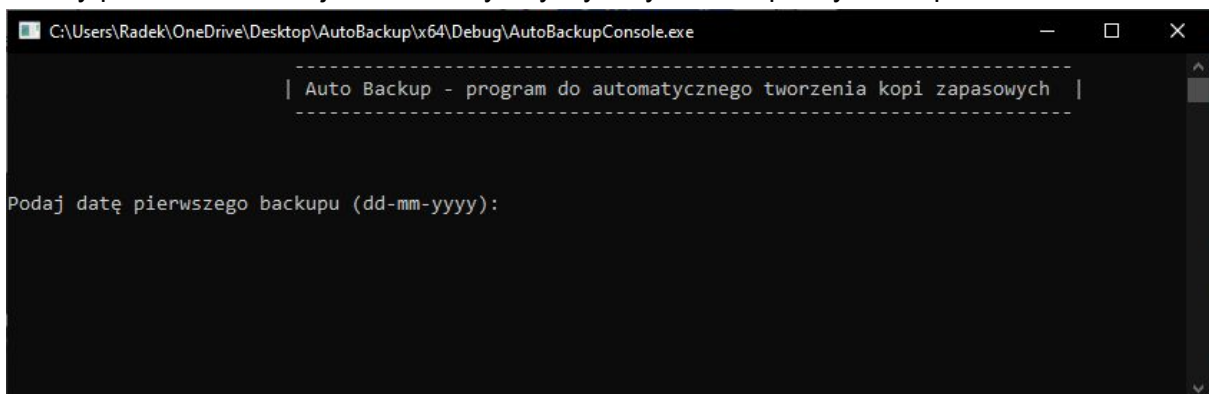
Należy podać ścieżkę do lokalizacji źródłowej, czyli tej której będzie wykonywana kopia zapasowa.



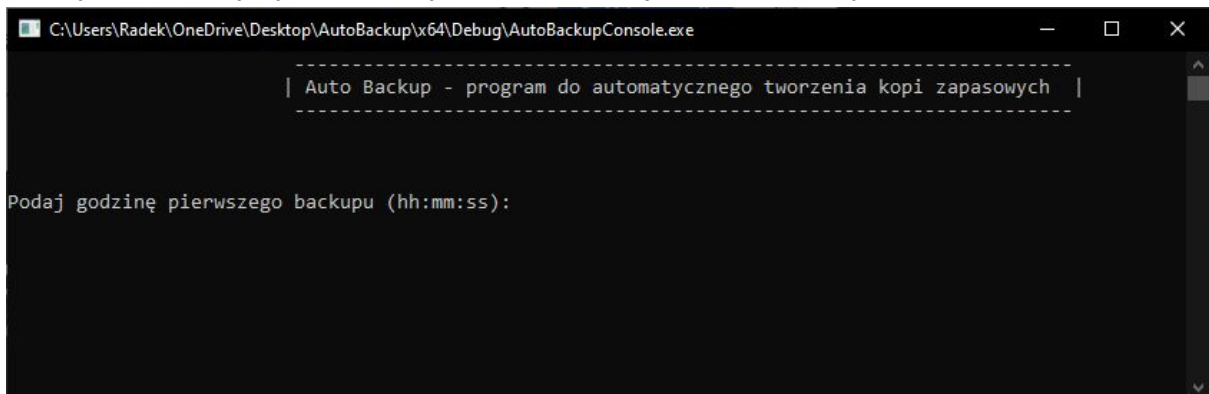
Należy podać ścieżkę lokalizacji docelowej, to znaczy miejsca przechowywania kopii zapasowej.



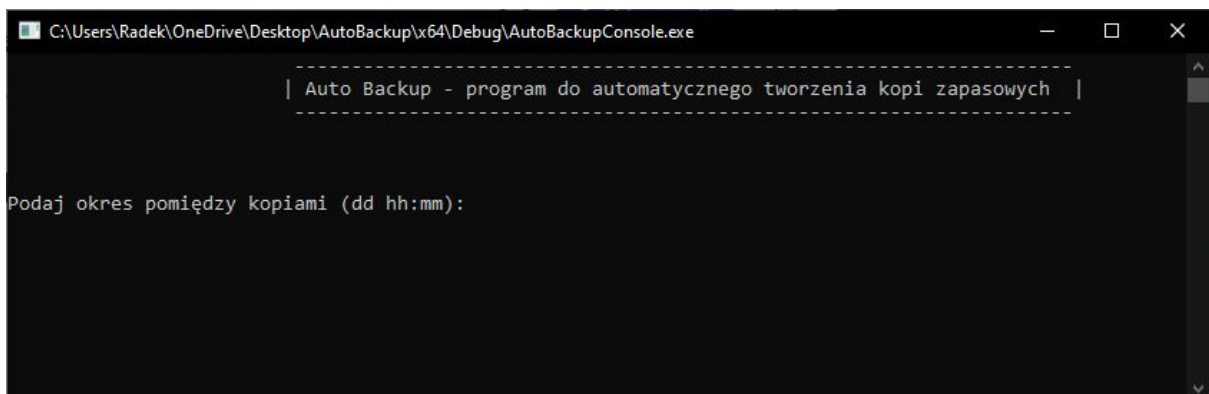
Należy podać wartość 1 jeżeli chcemy aby wykonywana kopia była kompresowana.



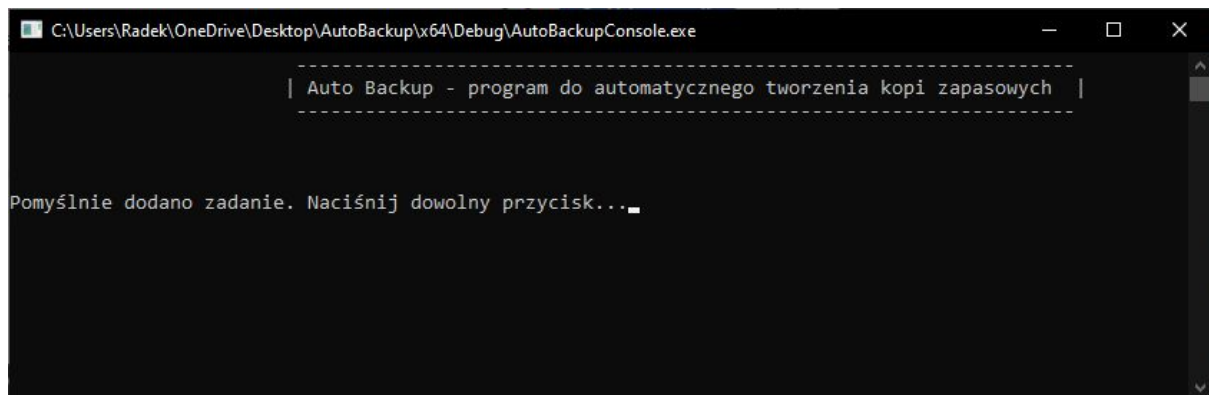
Należy podać datę wykonania się kopii zapasowej po raz pierwszy.



Należy podać godzinę wykonania się kopii zapasowej po raz pierwszy.



Należy podać ilość czasu która będzie dzieliła wykonanie się kolejnych kopii.



Zadanie zostało pomyślnie dodane do harmonogramu zadań.

5. **Wyczyść zadania** - pozwala na usunięcie zaplanowanych w harmonogramie wywołań wykonania kopii zapasowej.
6. **Wyświetl harmonogram** - Pozwala na wyświetlenie aktualnie obowiązującego harmonogramu wywołań wykonania kopii zapasowych.
7. **Zamknij konsolę** - Pozwala na zamknięcie okna konsoli do sterowania parametrami procesu roboczego.

Dokumentacja kodu programu:

1. AutoBackup:

- BackupProperties

- **struct BackupInterval**
Struktura przechowująca dane dotyczące interwału kopi zapasowej
- **struct BackupProperties**
Struktura przechowująca dane dotyczące kopi zapasowej
- **namespace backupprops**
Zwraca ciąg opisujący dane kopi zapasowej

- Logger

- **bool isBlockMode() { return bMode; }**
Zwraca tryb tworzenia logów
- **std::string getPath() { return fName; }**
Zwraca ścieżkę pliku logów
- **void error(std::string message);**
Dodaje błąd do logów
- **void info(std::string message);**
Dodaje informację do logów
- **void warning(std::string message);**
Dodaje ostrzeżenie do logów
- **void clear();**
Czyści logi
- **void waitForReady();**
Czeka na otwarcie pliku logów

- TaskExecutor

- **typedef struct threadData**
Struktura danych na potrzeby przekazania danych wątku do funkcji systemowej WINAPI CreateThread.
- **void execute(std::vector<BackupProperties> tasks);**
*Jako parametr przyjmuje kontener typu vector danych typu BackupProperties.
Funkcja powoduje uruchomienie tylu wątków w postaci klasy **MyThreadFunction** ile zostało przekazanych zadań do zrealizowania.*
- **static void DoBackup(std::string source, std::string destination, bool compress);**
Metoda wykonująca backup plików i folderów
- **DWORD WINAPI MyThreadFunction(LPVOID lpParam)**
*Funkcja wątku która wywołuje metode **DoBackup**. Pozwala to na realizowanie kilku zaplanowanych kopii w jednym czasie.*
- **void ErrorHandler(LPTSTR lpszFunction);**
Wyświetla przekazane komunikaty o błędach wykonania programu.

- TaskLoader

- **TaskLoader(std::string schedulerFileName, Logger* logger)**
: _scheduleFileName(schedulerFileName), _logger(logger)
Konstruktor klasy TaskLoader
- **std::vector<BackupProperties> loadTasks();**
Zwraca wektor zawierający informacje dotyczące kopi zapasowej

- TimeUtils

- **now()**
Zwraca ciąg z aktualną datą
- **addDaysToDate(std::string date, int days);**
Zwraca ciąg z datą pod dodaniu do niej zadanej ilości dni

2. AutoBackupConsole:

- AutoBackupProcess

- **bool start();**
Metoda uruchamiająca proces aplikacji kopi zapasowej w tle
- **bool stop();**
Metoda zatrzymująca proces aplikacji kopi zapasowej w tle
- **bool isRunning();**
Metoda sprawdzająca status procesu aplikacji

- BackupProperties

- **struct BackupInterval**
Struktura przechowująca dane dotyczące interwału kopi zapasowej
- **struct BackupProperties**
Struktura przechowująca dane dotyczące kopi zapasowej
- **namespace backupprops**
Zwraca ciąg opisujący dane kopi zapasowej

- Dialog

- **enum class Command**
Typu komend
- **Command Dialog::showMainDialog();**
Metoda zwracająca wybór użytkownika
- **void refreshConsole();**
Metoda odświeżająca konsolę
- **void printMainMenu();**
Metoda drukująca komendy do wyboru
- **void printTasks(std::vector<BackupProperties>& tasks);**
Metoda drukująca listę zadań do wykonania z harmonogramu
- **int readOption();**
- **void createNewBackupTask();**
Metoda dodająca nowe zadanie do harmonogramu
- **void saveTask(const BackupProperties& backup);**
Metoda zapisująca utworzone zadanie harmonogramu w pliku harmonogramu
- **void clearTasks();**
Usuwa zadania z harmonogramu zapisanego w pliku schedule.dat w katalogu uruchomieniowym programu
- **void showStatus();**
Metoda wyświetlająca status programu backupu działającego w tle
- **void showMessage(std::string text);**
Metoda wyświetlająca wiadomość na ekranie
- **void startAutoBackupProcess();**
Metoda uruchamiająca program kopi zapasowej w tle
- **void stopAutoBackupProcess();**
Metoda zatrzymująca działający program kopi zapasowej
- **void showSchedule();**
Metoda wyświetlająca harmonogram zadań
- **void pause();**
Metoda wstrzymująca konsolę, aż do reakcji użytkownika
- **std::vector<BackupProperties> loadTasksFromFile(std::string fileName);**
Metoda ładująca zadania kopii zapasowych z pliku
- **BackupProperties showNewTaskDialog();**
Metoda wyświetlająca dialog aby dodać zadanie kopii zapasowej
- **std::string showDateDialog();**
Metoda wyświetlająca dialog aby wprowadzić datę
- **std::string showTimeDialog();**
Metoda wyświetlająca dialog aby wprowadzić czas
- **BackupInterval showIntervalDialog();**
Metoda wyświetlająca dialog aby wprowadzić interwał
- **std::string showDirDialog(std::string text);**
Metoda wyświetlająca dialog, aby wprowadzić ścieżkę

- **bool showCompressDialog();**
Metoda wyświetlająca dialog pytający czy zastosować kompresję
 - **void showCancelDialog(std::string text);**
Metoda wyświetlająca dialog anulowania wprowadzania zadania
 - **bool scanAsBool();**
Metoda odczytująca standardowe wejście jako typ logiczny
- **Files**
 - **bool exists(std::string path)**
Metoda sprawdzająca czy plik o podanej ścieżce istnieje
- **NotImplementedException**
 - **NotImplementedException()**
Metoda zwracająca błąd o niezaimplementowanej funkcji
- **Validation**
 - **bool validateDate(string date)**
Metoda sprawdzająca poprawność daty (dd-mm-yyyy)
 - **bool validateTime(std::string time)**
Metoda sprawdzająca poprawność czasu (hh:mm:ss)
 - **bool validateInterval(std::string interval)**
Metoda sprawdzająca poprawność interwału (dd hh:mm)

4. Kod źródłowy wraz z numeracją linii oraz komentarze.

[Link do projektu na Githubie](#)

5. Skompilowany program, możliwy do uruchomienia na dowolnym komputerze.

[Release final · IT-Students-Web-Project/AutoBackup \(github.com\)](#)

Program AutoBackupConsole.exe należy uruchamiać z uprawnieniami administratora. Jest to konieczne do jego prawidłowego działania.