Opis projektu

Jakub Kuciński

7 lutego 2019

1 Krótki opis modułów

Program zbudowany jest z siedmiu modułów:
1. main
2. Interfejs
3. Baza_danych
4. Zestawienia_interfejs
5. Zestawienia_baza
6. Admin_interfejs
7. Admin_baza

Moduł main odpowiada za logowanie oraz wywoływanie funkcji odpowiadających wybranym przez użytkownika akcjom. W przypadku użytkownika standardowego wybranie opcji 0 powoduje zamknięcie aplikacji, wybranie opcji 1, 2 i 4 - wywołanie odpowiednich funkcji z modułu Interfejs, wybranie opcji 3 - wywołanie funkcji głównej modułu Zestawienia_interfejs. W przypadku administratora oraz dowolnej z dostępnych opcji, main wywołuje odpowiednie funkcje z modułu Admin_interfejs.

Moduł Interfejs odpowiada za wypisywanie komunikatów oraz wczytywanie danych od użytkownika w zakresie akcji 1, 2 i 4. W zależności od wczytanych od użytkownika infromacji wywołuje odpowiednie funkcje z modułu Baza_danych, które realizują wybrane przez użytkownika zdarzenia poprzez tworzenie odpowiednich zapytań do bazy danych oraz przetwarzanie otrzymanych wyników.

Moduł Zestawienia_interfejs odpowiada za wypisywanie komunikatów oraz wczytywanie danych od użytkownika w zakresie akcji 3, czyli generowania zestawień. Podobnie jak moduł Interfejs w zależności od wczytanych od użytkownika infromacji wywołuje odpowiednie funkcje z modułu Zestawienia_baza oraz pojedyńcze funkcje z modułu Baza_danych, które realizują wybrane przez użytkownika zdarzenia poprzez tworzenie odpowiednich zapytań do bazy danych oraz przetwarzanie otrzymanych wyników.

Moduł Admin_interfejs odpowiada za wypisywanie komunikatów oraz wczytywanie danych od użytkownika w zakresie wszystkich akcji dostępnych dla administratora. Podobnie jak moduły Interfejs i Zestawienia_interfejs w zależności od wczytanych od użytkownika infromacji wywołuje odpowiednie funkcje z modułu Admin_baza oraz pojedyńcze funkcje z modułów Baza_danych oraz Zestawienia_baza, które realizują wybrane przez użytkownika zdarzenia poprzez tworzenie odpowiednich zapytań do bazy danych oraz przetwarzanie otrzymanych wyników.

2 Krótki opis struktury i funkcji w poszczególnych modułach

2.1 main

Moduł main składa się głównie z wywołania funkcji wczytującej login i hasło użytkownika oraz w zależności od typu użytkownika wejścia do pętli przeznaczonej dla użytkownika standardowego lub admina. Pętle odpowiadają za wywoływanie funkcji wybranych przez użytkownika. Wyjście z pętli następuje, gdy użytkownik wybierze opcje wyjścia z programu.

2.2 Interfejs

Moduł Interfejs zawiera następujące funkcje:
1. void interfejs_logowanie(char login[52]);

2. int interfejs_funkcjonalnosci();
3. int interfejs_dodanie_nowego_rejestru(char * login);
4. int poprawna_data(char * data);
5. int interfejs_wyswietlanie_rejestrow(char * login);
6. int interfejs_zmiana_hasla(char * login);

1. interfejs_logowanie przyjmuje jako argument wskaźnik na tablicę o rozmiarze 52, do której ma zostać wczytany login użytkownika, funkcja pobiera od użytkownika nazwę login oraz hasło, sprawdza ich poprawność poprzez wywołanie odpowiednich funkcji z modułu Baza_danych czy dane są poprawne, funkcja kończy działanie, gdy zostaną podane poprawen dane.

2. interfejs_funkcjonalnosci odpowiada za wyświetlanie dostępnych funkcji dla użytkownika standardowego oraz wczytuje od użytkownika numer akcji, która ma zostać wywołana. Funkcja zwraca numer funkcji, która ma zostać wywołana.

3. interfejs_dodanie_nowego_rejestru przyjmuje jako argument login zalogowanego użytkownika oraz odpowiada za pobranie od użytkownika danych dotyczących nowego rejestru oraz wywołanie funkcji dodawania rejestru z modułu Baza_danych. Zwraca 0 gdy poprawnie zakończy swoje działanie.

4. int poprawna_data funkcja pomocnicza sprawdzająca poprawnośc podanej przez użytkownika daty podanej jako argument (typu *char) (czy spełniony jest warunek 2018.01.01<=data<=2019.12.31). Zwraca 0 gdy data jest niepoprawna. 1 wpp.

5. interfejs_wyswietlanie_rejestrow przyjmuje jako argument login zalogowanego użytkownika oraz odpowiada za wyświetlenie ostatnich 10 rejestrów użytkownika oraz ewentualne usunięcie wybranego (niekoniecznie spośród wypisanych) rejestru poprzez wywołanie odpowiednich funkcji z modułu Baza_danych. Zwraca 0 gdy poprawnie zakończy swoje działanie.

6. interfejs_zmiana_hasla przyjmuje jako argument login zalogowanego użytkownika oraz odpowiada za zmianę hasła użytkownika. Funkcja wczytuje od użytkownika nowe hasło oraz po uzyskaniu potwierdzenia dokonuje zmiany hasła poprzez wywołanie odpowiedniej funkcji z modułu Baza_danych. Zwraca 0 gdy poprawnie zakończy swoje działanie.

2.3 Baza_danych

Moduł Baza_danych zawiera następujące funkcje:
1. int Baza_logowanie(char * login, char * haslo);
2. int Baza_lista_przedmiotow();
3. int Baza_czy_id_przedmiotu(char * numer);
4. int Baza_lista_aktywnosci();
5. int Baza_czy_id_aktywnosci(char * numer);
6. int Baza_dodawanie_rejestru(char * login, char * id_przedmiotu, char * id_aktywnosci, char * czas_nauki, char * data);
7. int Baza_id_uzytkownika(char * login, char * id_uzytkownika);
8. int Baza_lista_rejestrow(char * login);
9. int Baza_ile_rejestrow(char * id_uzytkownika);
10. int Baza_czy_id_rejestru(char * numer, char * id_uzytkownika);
11. int Baza_usuwanie_rejestru(char * numer);
12. int Baza_zmiana_hasla(char * login, char * haslo);

1. Baza_logowanie sprawdza czy podane dane logowania (login i hasło) są w bazie danych. Zwraca 0 jeśli nie ma, 1 wpp.

2. Baza_lista_przedmiotow wyświetla listę przedmiotów w bazie danych. Zwraca 0 gdy poprawnie zakończy swoje działanie.

3. Baza_czy_id_przedmiotu sprawdza czy podane jako argument id ma odpowiadający przedmiot w bazie danych. Zwraca 0 jeśli nie ma, 1 wpp.

4. Baza_lista_aktywnosci wyświetla listę aktywności w bazie danych. Zwraca 0 gdy poprawnie zakończy swoje działanie.

5. Baza_czy_id_aktywnosci sprawdza czy podane jako argument id ma odpowiadającą aktywność w bazie danych. Zwraca 0 jeśli nie ma, 1 wpp.

6. Baza_dodawanie_rejestru dodaje rejestr o parametrach podanych w argumentach wywołania funkcji do bazy danych. Zwraca 0 gdy poprawnie zakończy swoje działanie.

7. Baza_id_uzytkownika wpisuje do podanego jako argument stringa id_uzytkownika odpowiadające podanemu jako argument loginowi - id użytkownika z bazy danych. Zwraca 0 jeśli nie ma odpiwadającego danemu loginowi id, 1 wpp.

8. Baza_lista_rejestrow wyświetla listę ostanich 10 rejestrów użytkownika o przekazanym w argumencie loginie. Zwraca 0 gdy poprawnie zakończy swoje działanie.

9. Baza_ile_rejestrow zwraca liczbę rejestów użytkownika o podanym w argumencie id użytkownika.

10. Baza_czy_id_rejestru sprawdza czy podane jako argument id ma odpowiadający rejestr w bazie danych o podanym w argumencie id użytkownika. Zwraca 0 jeśli nie ma, 1 wpp.

11. Baza_usuwanie_rejestru usuwa z bazy danych rejestr o podanym jako argument numerze. Zwraca 0 gdy poprawnie zakończy swoje działanie.

12. Baza_zmiana_hasla zmienia hasło podanego jako argument loginu na nowe podane jako argument hasło. Zwraca 0 gdy poprawnie zakończy swoje działanie.

2.4 Zestawienia_interfejs

Moduł Zestawienia_interfejs zawiera następujące funkcje:
1. int zestawienia_interfejs_sredni_czas(char * login);
2. int wczytaj_1_lub_2();
3. int wczytaj_przedmiot(char * przedmiot);
4. int wczytaj_aktywnosc(char * aktywnosc);
5. int zestawienia_interfejs_wyswietlenie_rejestrow(char * login);
6. int zestawienia_interfejs_wybor(char * login);
7. int zestawienia_interfejs_sredni_czas_przedmiotu(char * login);
8. int zestawienia_interfejs_sredni_czas_aktywnosci(char * login);
9. int wczytaj_0_1_2_3_4();

1. zestawienia_interfejs_sredni_odpowiada za wywołanie odpowiednich funkcji z modułu Zestawienia_interfejs. Wywoływane funkcje tworzą zestawienia z podziałem na poszczególne rodzaje przedmiotów i aktywności. W zależności od podanych przez użytkownika wartości wywoływane są funkcje tworzenia zestawień według rejestrów wszystkich użytkowników, zalogowanego użytkownika, wszystkich przedmiotów, wybranego przedmiotu, wszystkich aktywności, wybranej aktywności. Funkcja korzysta również z funkcji z modułu Baza_danych do wyświetlania list przedmiotów i aktywności. Jako argument przyjmuje login zalogowanego użytkownika. Zwraca 0 gdy poprawnie zakończy swoje działanie.

2. wczytaj_1_lub_2 wczytuje od użytkownika liczbę 1 lub 2 i zwraca wczytany wynik w postaci inta.

3. wczytaj_przedmiot wczytuje od użytkownika numer przedmiotu z bazy danych, zapisuje wynik do podanego jako argument char * przemiot. Zwraca 0 gdy poprawnie zakończy swoje działanie.

4. wczytaj_aktywnosc wczytuje od użytkownika numer aktywności z bazy danych, zapisuje wynik do podanego jako argument char * aktywnosc. Zwraca 0 gdy poprawnie zakończy swoje działanie.

5. zestawienia_interfejs_wyswietlenie_rejestrow odpowiada za wywołanie odpowiednich funkcji z modułu Zestawienia_baza. Wywoływane funkcje wypisują listę rejestrów z podziałem na poszczególne rodzaje przedmiotów i aktywności. W zależności od podanych przez użytkownika wartości wywoływane są funkcje wypisywa-

nia rejestrów wszystkich z wszystkich przedmiotów, wybranego przedmiotu, wszystkich aktywności, wybranej aktywności. Funkcja korzysta również z funkcji z modułu Baza_danych do wyświetlania list przedmiotów i aktywności. Zwraca 0 gdy poprawnie zakończy swoje działanie.

6. zestawienia_interfejs_wybor odpowiada za wypisanie dostępnych opcji tworzenia zestawień oraz wywołania odpowiadających im funkcji na podstawie danych wpisanych przez użytkownika. Jako argument przyjmuje login zalogowanego użytkownika. Zwraca 0 gdy poprawnie zakończy swoje działanie.

7. zestawienia_interfejs_sredni_czas_przedmiotu odpowiada za wywołanie odpowiednich funkcji z modułu Zestawienia_baza. Wywoływane funkcje tworzą zestawienia z podziałem na poszczególne rodzaje przedmiotów. W zależności od podanych przez użytkownika wartości wywoływane są funkcje tworzenia zestawień według rejestrów wszystkich użytkowników, zalogowanego użytkownika, wszystkich przedmiotów, wybranego przedmiotu. Funkcja korzysta również z funkcji z modułu Baza_danych do wyświetlania listy przedmiotów. Jako argument przyjmuje login zalogowanego użytkownika. Zwraca 0 gdy poprawnie zakończy swoje działanie.

8. zestawienia_interfejs_sredni_czas_aktywnosci odpowiada za wywołanie odpowiednich funkcji z modułu Zestawienia_baza. Wywoływane funkcje tworzą zestawienia z podziałem na poszczególne rodzaje aktywności. W zależności od podanych przez użytkownika wartości wywoływane są funkcje tworzenia zestawień według rejestrów wszystkich użytkowników, zalogowanego użytkownika, wszystkich aktywności, wybranej aktywności. Funkcja korzysta również z funkcji z modułu Baza_danych do wyświetlania list aktywności. Jako argument przyjmuje login zalogowanego użytkownika. Zwraca 0 gdy poprawnie zakończy swoje działanie.

9. wczytaj_0_1_2_3_4 wczytuje od użytkownika liczbę 1, 2, 3 lub 4 i zwraca wczytany wynik w postaci inta.

2.5 Zestawienia_baza

Moduł Zestawienia_baza zawiera następujące funkcje:
1. int wypisz(char * widok, char * str);
2. int wypisz_rejestry(char *str);
3. int Zestawienia_skrot_baza_111();
4. int Zestawienia_skrot_baza_112(char * aktywnosc);
5. int Zestawienia_skrot_baza_121(char * przedmiot);
6. int Zestawienia_skrot_baza_122(char * przedmiot, char * aktywnosc);
7. int Zestawienia_skrot_baza_211(char * id_uzytkownika);
8. int Zestawienia_skrot_baza_212(char * id_uzytkownika, char * aktywnosc);
9. int Zestawienia_skrot_baza_221(char * id_uzytkownika, char * przedmiot);
10. int Zestawienia_skrot_baza_222(char * id_uzytkownika, char * przedmiot, char * aktywnosc);
11. int Zestawienia_skrot_baza_rejestry_111(char * login);
12. int Zestawienia_skrot_baza_rejestry_112(char * login, char * aktywnosc);
13. int Zestawienia_skrot_baza_rejestry_121(char * login, char * przedmiot);
14. int Zestawienia_skrot_baza_rejestry_122(char * login, char * przedmiot, char * aktywnosc);
15. int Zestawienia_skrot_baza_przedmioty_11();
16. int Zestawienia_skrot_baza_przedmioty_12(char * przedmiot);
17. int Zestawienia_skrot_baza_przedmioty_21(char * id_uzytkownika);
18. int Zestawienia_skrot_baza_przedmioty_22(char * przedmiot, char * id_uzytkownika);
19. int Zestawienia_skrot_baza_aktywnosci_11();
20. int Zestawienia_skrot_baza_aktywnosci_12(char * aktywnosc);
21. int Zestawienia_skrot_baza_aktywnosci_21(char * id_uzytkownika);
22. int Zestawienia_skrot_baza_aktywnosci_22(char * aktywnosc, char * id_uzytkownika);

1. wypisz wypisuje zestwienie bazujące na widoku przekazanym przez parametr widok, który zawiera zapytanie w języku MySQL oraz na stringu str zawiarającego zapytanie w MySQL, poprzez komunikację z bazą danych. Zwraca 0 gdy poprawnie zakończy swoje działanie.

2. wypisz_rejestry wypisuje rejestry użytkownika według podanego w parametrze stringu str, który zawiera zapytanie w języku MySQL, poprzez komunikację z bazą danych. Zwraca 0 gdy poprawnie zakończy swoje działanie.

3-10. Zestawienia_skrot_baza_abc Funkcja tworzy zapytanie w MySQL o zestawienie średniego czasu nauki z podziałem na rodzaje aktywności i przedmiotów oraz w zależnośći od parametrów z rejestrów wszystkich użytkowników lub zalogowanego użytkownika. Gdy a=1 funkcja tworzy zapytanie w MySQL o zestawienie z rejestrów wszystkich użytkowników, gdy a=2 z rejestrów zalogowanego użytkownika (argument id_uzytkownika), b=1 ze wszystkich przedmiotów, b=2 z wybranego przemiotu (argument przedmiot), c=1 ze wszystkich aktywności, c=2 z wybranej aktywności (argument aktywnosc). Funkcja wywołuje funkcję wypisz z tego samego modułu. Zwraca 0 gdy poprawnie zakończy swoje działanie.

11-14. Zestawienia_skrot_baza_rejestry_1bc Funkcja tworzy zapytanie w MySQL o zestawienie rejestrów z podziałem na rodzaje aktywności i przedmiotów. Gdy b=1 funkcja tworzy zapytanie w MySQL o zestawienie ze wszystkich przedmiotów, b=2 z wybranego przemiotu (argument przedmiot), c=1 ze wszystkich aktywności, c=2 z wybranej aktywności (argument aktywnosc). Funkcja wywołuje funkcję wypisz_rejestry z tego samego modułu. Zwraca 0 gdy poprawnie zakończy swoje działanie.

15-18. Zestawienia_skrot_baza_przedmioty_ab Funkcja tworzy zapytanie w MySQL o zestawienie średniego czasu nauki z podziałem na rodzaje przedmiotów oraz w zależnośći od parametrów z rejestrów wszystkich użytkowników lub zalogowanego użytkownika. Gdy a=1 funkcja tworzy zapytanie w MySQL o zestawienie z rejestrów wszystkich użytkowników, gdy a=2 z rejestrów zalogowanego użytkownika (argument id_uzytkownika), b=1 ze wszystkich przedmiotów, b=2 z wybranego przemiotu (argument przedmiot). Funkcja wywołuje funkcję wypisz z tego samego modułu. Zwraca 0 gdy poprawnie zakończy swoje działanie.

19-22. Zestawienia_skrot_baza_aktywnosci_ac Funkcja tworzy zapytanie w MySQL o zestawienie średniego czasu nauki z podziałem na rodzaje aktywności oraz w zależnośći od parametrów z rejestrów wszystkich użytkowników lub zalogowanego użytkownika. Gdy a=1 funkcja tworzy zapytanie w MySQL o zestawienie z rejestrów wszystkich użytkowników, gdy a=2 z rejestrów zalogowanego użytkownika (argument id_uzytkownika), c=1 ze wszystkich aktywności, c=2 z wybranej aktywności (argument aktywnosc). Funkcja wywołuje funkcję wypisz z tego samego modułu. Zwraca 0 gdy poprawnie zakończy swoje działanie.

2.6 Admin_interfejs

Moduł Admin_interfejs zawiera następujące funkcje:
1. int Admin_interfejs_funkcjonalnosci();
2. int Admin_interfejs_dodawanie_uzytkownika();
3. int Admin_interfejs_dodawanie_przedmiotu();
4. int Admin_interfejs_dodawanie_aktywnosci();
5. int Admin_interfejs_zmiana_nazwy_uzytkownika();
6. int Admin_interfejs_zmiana_nazwy_przedmiotu();
7. int Admin_interfejs_zmiana_nazwy_aktywnosci();
8. int Admin_interfejs_usuwanie_uzytkownika();
9. int Admin_interfejs_usuwanie_przedmiotu();
10. int Admin_interfejs_usuwanie_aktywnosci();
11. int czy_login_poprawny(char * login);

1. Admin_interfejs_funkcjonalnosci odpowiada za wyświetlanie dostępnych funkcji dla administratora oraz wczytuje od użytkownika numer akcji, która ma zostać wywołana. Funkcja zwraca numer funkcji, która ma zostać wywołana.

2. Admin_interfejs_dodawanie_uzytkownika wyświetla listę zajętych loginów poprzez wywołanie funkcji z modułu Admin_baza, a następnie wczytuje login i hasło nowego użytkownika i wywołuje funkcję dodawania nowego użytkownika z modułu Admin_baza. Zwraca 0 gdy poprawnie zakończy swoje działanie.

3. Admin_interfejs_dodawanie_przedmiotu wyświetla listę istniejących przemiotów poprzez wywołanie funkcji z modułu Baza_danych, a następnie wczytuje nowy przedmiot i wywołuje funkcję dodawania nowego przedmiotu z modułu Admin_baza. Zwraca 0 gdy poprawnie zakończy swoje działanie.

4. Admin_interfejs_dodawanie_aktywnosci wyświetla listę istniejących aktywności poprzez wywołanie funkcji z modułu Baza_danych, a następnie wczytuje nową aktywność i wywołuje funkcję dodawania nowej aktywności z modułu Admin_baza. Zwraca 0 gdy poprawnie zakończy swoje działanie.

5. Admin_interfejs_zmiana_nazwy_uzytkownika wypisuje listę loginów i ich id przez wywołanie funkcji z modułu Admin_baza oraz pobiera od użytkownika numer użytkownika, którego login ma zostać zmieniony, a następnie jego nowy login. Potem wywołuje funkcję zmieniającą login użytkownika z modułu Admin_baza. Zwraca 0 gdy poprawnie zakończy swoje działanie.

6. Admin_interfejs_zmiana_nazwy_przedmiotu wypisuje listę przedmiotów przez wywołanie funkcji z modułu Baza_danych oraz pobiera od użytkownika numer przedmiotu, którego nazwa ma zostać zmieniona. Potem wywołuje funkcję zmieniającą nazwę przedmiotu z modułu Admin_baza. Zwraca 0 gdy poprawnie zakończy swoje działanie.

7. Admin_interfejs_zmiana_nazwy_aktywnosci wypisuje listę aktywności przez wywołanie funkcji z modułu Baza_danych oraz pobiera od użytkownika numer aktywności, której nazwa ma zostać zmieniona. Potem wywołuje funkcję zmieniającą nazwę aktywności z modułu Admin_baza. Zwraca 0 gdy poprawnie zakończy swoje działanie.

8. Admin_interfejs_usuwanie_uzytkownika wypisuje listę loginów i ich id przez wywołanie funkcji z modułu Admin_baza oraz pobiera od użytkownika numer użytkownika, który ma zostać usunięty z bazy danych. Potem wywołuje funkcję usuwającą użytkownika z modułu Admin_baza. Zwraca 0 gdy poprawnie zakończy swoje działanie.

9. Admin_interfejs_usuwanie_przedmiotu wypisuje listę przedmiotów przez wywołanie funkcji z modułu Baza_danych oraz pobiera od użytkownika numer przedmiotu, który ma zostać usunięty. Potem wywołuje funkcję usuwającą przedmiot z modułu Admin_baza. Zwraca 0 gdy poprawnie zakończy swoje działanie.

10. Admin_interfejs_usuwanie_aktywnosci wypisuje listę aktywności przez wywołanie funkcji z modułu Baza_danych oraz pobiera od użytkownika numer aktywności, która ma zostać usunięta. Potem wywołuje funkcję usuwającą aktywność z modułu Admin_baza. Zwraca 0 gdy poprawnie zakończy swoje działanie.

11. czy_login_poprawny funkcja pomocnicza wczytująca login (używana też do wczytywania hasła) do char * login podanego jako argument. Zwraca 0 gdy poprawnie zakończy swoje działanie.

2.7 Admin_baza

Moduł Admin_baza zawiera następujące funkcje:
1. int Admin_baza_czy_zajety_login(char * login);
2. int Admin_baza_czy_przedmiot_istnieje(char * przedmiot);
3. int Admin_baza_czy_aktywnosc_istnieje(char * aktywnosc);
4. int Admin_baza_czy_id_uzytkownika_istnieje(char * id_uzytkownika);
5. int Admin_baza_lista_loginow();
6. int Admin_baza_dodawanie_nowego_uzytkownika(char * login, char * haslo);
7. int Admin_baza_dodawanie_nowego_przedmiotu(char * przedmiot);
8. int Admin_baza_dodawanie_nowej_aktywnosci(char * aktywnosc);
9. int Admin_baza_zmiana_nazwy_uzytkownika(char * id_uzytkownika, char * login);
10. int Admin_baza_zmiana_nazwy_przedmiotu(char * id_przedmiotu, char * nazwa_przedmiotu);
11. int Admin_baza_zmiana_nazwy_aktywnosci(char * id_aktywnosci, char * nazwa_aktywnosci);
12. int Admin_baza_usuwanie_uzytkownika(char * id_uzytkownika);
13. int Admin_baza_usuwanie_przedmiotu(char * id_przedmiotu);
14. int Admin_baza_usuwanie_aktywnosci(char * id_aktywnosci);

1. Admin_baza_czy_zajety_login sprawdza czy podany jako argument login znajduje się w bazie danych. Zwraca 0 jeśli nie, 1 wpp.

2. Admin_baza_czy_przedmiot_istnieje sprawdza czy podany jako argument przedmiot (nazwa) znajduje się w bazie danych. Zwraca 0 jeśli nie, 1 wpp.

3. Admin_baza_czy_aktywnosc_istnieje sprawdza czy podana jako argument aktywność (nazwa) znajduje się w bazie danych. Zwraca 0 jeśli nie, 1 wpp.

4. Admin_baza_czy_id_uzytkownika_istnieje sprawdza czy podane jako argument id ma odpowiadający login w bazie danych. Zwraca 0 jeśli nie ma, 1 wpp.

5. Admin_baza_lista_loginow wyświetla listę loginów w bazie danych. Zwraca 0 gdy poprawnie zakończy swoje działanie.

6. Admin_baza_dodawanie_nowego_uzytkownika funkcja dodaje do bazy danych nowego użytkownika o określonych w parametrach funkcji loginie i haśle. Zwraca 0 gdy poprawnie zakończy swoje działanie.

7. Admin_baza_dodawanie_nowego_przedmiotu funkcja dodaje do bazy danych nowy przedmiot o nazwie określonej w parametrze funkcji. Zwraca 0 gdy poprawnie zakończy swoje działanie.

8. Admin_baza_dodawanie_nowej_aktywnosci funkcja dodaje do bazy danych nową aktywność o nazwie określonej w parametrze funkcji. Zwraca 0 gdy poprawnie zakończy swoje działanie.

9. Admin_baza_zmiana_nazwy_uzytkownika funkcja zmienia login użytkownika o podanym w parametrze id_uzytkownika na nazwę podaną w parametrze (char * login). Zwraca 0 gdy poprawnie zakończy swoje działanie.

10. Admin_baza_zmiana_nazwy_przedmiotu funkcja zmienia nazwę przedmiotu o podanym w parametrze id_przedmiotu na nazwę podaną w parametrze (char * nazwa_przedmiotu). Zwraca 0 gdy poprawnie zakończy swoje działanie.

11. Admin_baza_zmiana_nazwy_aktywnosci funkcja zmienia nazwę aktywności o podanym w parametrze id_aktywnosci na nazwę podaną w parametrze (char * nazwa_aktywnosci). Zwraca 0 gdy poprawnie zakończy swoje działanie.

12. Admin_baza_usuwanie_uzytkownika funkcja usuwa użytkownika o podanym w parametrze id_uzytkownika z bazy danych wraz ze wszystkimi powiązanymi z tym użytkownikiem rejestrami. Zwraca 0 gdy poprawnie zakończy swoje działanie.

13. Admin_baza_usuwanie_przedmiotu funkcja usuwa przedmiot o podanym w parametrze id_przedmiotu z bazy danych wraz ze wszystkimi powiązanymi z tym przedmiotem rejestrami. Zwraca 0 gdy poprawnie zakończy swoje działanie.

14. Admin_baza_usuwanie_aktywnosci funkcja usuwa aktywność o podanym w parametrze id_aktywnosci z bazy danych wraz ze wszystkimi powiązanymi z tą aktywnością rejestrami. Zwraca 0 gdy poprawnie zakończy swoje działanie.