

Programowanie w języku JAVA

Zajęcia laboratoryjne 2

1. Zadania do wykonania

Zadanie 1. Stwórz nowy projekt w IntelliJ Idea. (File -> New -> Project). Z listy po lewej stronie wybierz pierwszą opcję - Java. Przy tworzeniu projektu nie zaznaczaj opcji wykorzystania gotowych szablonów.

Zadanie 2. W folderze src dodaj nową paczkę. (Kliknij prawym przyciskiem myszy na folderze, a następnie: new -> package). Nazwij ją **com.swingapp**.

Zadanie 3. W tej paczce dodaj nowe okno aplikacji. (Kliknij prawym przyciskiem myszy na paczce, a następnie new -> GUI Form) nazwij je **AddEmployee**.

Zadanie 4. W klasie AddEmployee dodaj publiczną statyczną metodę main. Poprawne wykonanie wszystkich kroków powinno umożliwić uruchomienie aplikacji. W konsoli wyjściowej po uruchomieniu powinien zostać zwrócony kod 0.

Zadanie 5. W metodzie main dodaj kod, który utworzy nowe okno i ustawi jego widoczność.

- Stwórz nowy obiekt **JFrame**. W konstruktorze jako parametr wpisz tytuł okna.
- Wykorzystaj metodę `setContentPane` żeby powiązać puste okno z zawartością. (podaj jako parametr nową instancję panelu: **new AddEmployee.panel1**)
- Ustaw zamknięcie aplikacji jako akcję, która ma się wykonać po zamknięciu okna. (Wykorzystaj metodę `setDefaultCloseOperation`)
- Wywołaj metodę, która przygotowuje okno przed wyświetleniem: **pack**.
- Ustaw widoczność okna na wartość logiczną true.

Zadanie 6. Dodaj nowy przycisk do okna. (Otwórz plik Employee.form. Z panelu po prawej stronie wybierz **JButton** i przeciągnij go na obszar okna).

Zadanie 7. Aby powiązać przycisk z akcją która ma nastąpić po jego naciśnięciu dodaj **ActionListener** do przycisku. (Kliknij w niego prawym przyciskiem myszy, a następnie wybierz Create Listener -> Action listener). Spowoduje to dodanie metody w klasie AddEmployee.

Zadanie 8. W oknie dodaj nowe pole tekstowe. (Przeciągnij na obszar okna `JTextField`)



Zadanie 9. W metodzie odpowiedzialnej za reakcję na naciśnięcie przycisku dopisz kod, który zawartość pola tekstowego (metoda `getText` pobiera zawartość pola) zapisze do pliku `First.txt`. Jeżeli pole testowe jest puste aplikacja powinna stworzyć nowe okienko informujące o błędzie. (Wykorzystaj metodę: `JOptionPane.showMessageDialog(null, "Tekst wiadomości");`)

Zadanie 10. Dodaj do projektu nową paczkę. nazwij ją **com.classes**.

Zadanie 11. Dodaj do paczki nowy typ wyliczeniowy - enum o nazwie `ContractType`. (new Java Class, a następnie wybierz z listy Enum). Dodaj w nim 2 wartości: **B2B**, **EmployementContract**.

Zadanie 12. Do paczki dodaj nową klasę **Employee**. Dopisz w niej pola: `name`, `surname`, `email`, `salary`, `retired`, `contractType` (`String`, `String`, `String`, `int`, `Boolean`, `ContractType`). Dodaj w klasie odpowiednie gettery i settery.

Zadanie 13. Zmodyfikuj okno tak aby posiadało:

- 4 `TextField`
- 1 `Button`
- 2 `RadioButton`
- 1 `CheckBoxField`

Zadanie 14. Dla obu `JRadiobuttonów` dodaj `ActionListenery`. Napisz kod, który po zaznaczeniu pierwszego `JRadioButton`a odznaczy drugi oraz odwrotnie.

Zadanie 15. Zmodyfikuj `ActionListener` dla przycisku tak aby tworzył nową instancję klasy `Employee` z danych wprowadzonych przez użytkownika. (Pola tekstowe do pól `name`, `surname`, `email` i `salary` - metoda `Integer..ValueOf`. `JCheckBox` do `retired`. `JRadioButton`ony do wyboru rodzaju zatrudnienia)

Zadanie 16*. Dodaj kod, który przed utworzeniem nowego pracownika sprawdzi poprawność wypełnionych pól w aplikacji.

Zadanie 17*. Automatycznie generowane nazwy przez środowisko niedostatecznie dobrze opisują przeznaczenie kontrolki. Pozmieniaj nazwy tak aby informowały o tym z jakim polem klasy są skojarzone (np `textField1` -> `textFieldEmployeeName`).

Zadanie 18*. Pola tekstowe aplikacji nie zostały w żaden sposób opisane. Nad każdym z nich dodaj `JLabel` z tekstem objaśniającym pole.