

Nazwa
kwalifikacji:

Projektowanie, programowanie i testowanie aplikacji

Oznaczenie
kwalifikacji:

INF.04

Numer zadania:

02

Kod arkusza:

INF.04-02-26.01-SG

Wersja arkusza:

SG

| Lp. | Elementy podlegające ocenie/kryteria oceny |
|------------|---|
| R.1 | Rezultat 1: Implementacja, kompilacja, uruchomienie programu |
| | <i>Uwaga: kryteria należy odnieść do aplikacji konsolowej, jeżeli ta nie istnieje, zastosować do aplikacji mobilnej. Kryteria dotyczą wyłącznie samodzielnie napisanego kodu. Wystarczy, że sprawdzaną cechę zastosowano dla większości(90%) przypadków w kodzie</i> |
| R.1.1 | Kod źródłowy zapisany w sposób czytelny: instrukcje w osobnych liniach, stosowane spacje pomiędzy operatorami, konsekwentnie stosowana wybrana konwencja dla nawiasów klamrowych |
| R.1.2 | Kod zapisany z wcięciami dla zagnieżdżeń bloków |
| R.1.3 | Polskie lub angielskie, znaczące nazewnictwo metod |
| R.1.4 | Polskie lub angielskie, znaczące nazewnictwo pól. Wyjątkami od reguły są zmienne bufor, tmp, iteratory pętli itp. Kryterium nie jest spełnione, gdy nazwy zmiennych nic nie znaczą, np. x, fun, foo, tab, tablica |
| R.1.5 | Typy zmiennych pasują do problemu np. typ tekstowy dla treści pytania, odpowiedzi i nazwy obrazu, char dla udzielonej odpowiedzi, bool dla określenia czy odpowiedź jest prawdziwa (w przypadku języka Python typ wynika z przypisanych danych) |
| R.1.6 | Podjęta próba uruchomienia kodu, co jest udokumentowane zrzutem ekranowym przedstawiającym uruchomiony program lub jego kompilację |
| R.1.7 | Program nawiązuje zrozumiałą komunikację z użytkownikiem. Wystarczy, że wyświetla dane (quiz) lub informację o poprawności udzielonej odpowiedzi w dowolnej aplikacji. Jeżeli kod nie uruchamia się z powodu błędów kompilacji - sprawdzić w kodzie aplikacji |
| R.2 | Rezultat 2: Aplikacja konsolowa |
| | <i>Uwaga: kryteria należy sprawdzić w kodzie programu, sprawdzane elementy muszą być zapisane zgodnie ze składnią. Poprawność nadania nazwy klasie jest sprawdzana tylko w R.2.1</i> |
| R.2.1 | Program zawiera klasę Pytanie oraz klasę potomną PytanieZamkniete |
| R.2.2 | Klasa Pytanie jest abstrakcyjna (zawiera metodę abstrakcyjną lub słowo abstract, lub konstruktor niepubliczny, dziedziczy po klasie abc.ABC) |
| R.2.3 | Klasa Pytanie zawiera pola protected: dwa napisowe oraz jedno logiczne (w Python podkreślnik przed nazwą) |
| R.2.4 | Klasa PytanieZamkniete zawiera pola private: trzy napisowe oraz jedno typu znakowego (w Python dwa podkreślniki przed nazwą) |
| R.2.5 | Klasa Pytanie zawiera konstruktor dwuargumentowy, argumenty są przypisane odpowiednio do pól napisowych oraz pole logiczne jest ustawione na false |
| R.2.6 | Klasa PytanieZamkniete zawiera konstruktor sześćargumentowy, który wywołuje konstruktor rodzica (np. super.__init__, :base(...), super(...)) oraz przypisuje treści odpowiedzi i znak odpowiedzi prawidłowej (A, B lub C) do odpowiednich pól klasy |
| R.2.7 | W klasie Pytanie zdefiniowana metoda abstrakcyjna typu logicznego (słowo kluczowe abstract lub Python: @abstractmethod i słowo pass). Metoda nie ma implementacji oraz w klasie PytanieZamkniete istnieje metoda nadpisująca metodę abstrakcyjną z klasy Pytanie (w C# i Java obowiązkowo słowo override) |

| | |
|------------|--|
| R.2.8 | W klasie <i>PytanieZamkniete</i> jest zaimplementowana metoda, która sprawdza czy odpowiedź z argumentu zgadza się z poprawną, jeśli tak, przypisuje zmiennej logicznej z klasy <i>Pytanie</i> wartość true; zwracana jest wartość pola logicznego |
| R.3 | Rezultat 3: Aplikacja mobilna |
| | <i>Uwaga: należy uwzględnić różnice pomiędzy emulacjami - takie cechy jak marginesy, wielkości bloków itp. nie należy brać pod uwagę. Kryteria 3.1 ÷ 3.6, 3.10 sprawdzić w kodzie źródłowym, sprawdzane elementy muszą być zapisane zgodnie ze składnią. Gdy aplikacja nie uruchamia się, a zdający zapisał zrzuty ekranu z uruchomienia aplikacji należy sprawdzić powód braku kompilacji. Jeśli występują błędy w plikach źródłowych zdającego kryteria 3.7 ÷ 3.9 nie są spełnione. Jeżeli błędy występują w innych plikach lub bibliotekach sprawdzić w kodzie oraz na zrzucie ekranu. Dla Android Studio dopuszcza się także rozwiązanie w języku Kotlin</i> |
| R.3.1 | Zastosowany język znaczników XML/XAML lub inny do opisu interfejsu użytkownika oraz zdefiniowana przynajmniej jedna kontrolka |
| R.3.2 | Zdefiniowane kontrolki: obraz z pliku <i>zad1.jpg</i> , pole tekstowe z treścią pytania: „Które to schronisko?” trzy pola radio z odpowiedziami, oraz przycisk DALEJ |
| R.3.3 | Kolor tła okna lub rozkładu #2E7CB8, rozmiar czcionki z treścią pytania jest większy niż pozostałe treści, w jednym momencie można wybrać tylko jedno pole radio |
| R.3.4 | Umieszczone treść pytań i odpowiedzi w tablicy, liście lub innej kolekcji. Aplikacja zawsze działa na wszystkich elementach kolekcji np. za pomocą pętli |
| R.3.5 | Program obsługuje zdarzenie kliknięcia przycisku w sposób właściwy dla danego środowiska programistycznego. Wystarczy, że kontrolka przycisku jest odpowiednio powiązana ze zdarzeniem |
| R.3.6 | Program odwołuje się do przynajmniej jednej kontrolki w sposób zgodny z danym środowiskiem programistycznym |
| R.3.7 | Aplikacja kompiluje się i uruchamia w emulatorze, co jest udokumentowane zrzutem ekranu jej układ jest zgodny z obrazem 1 lub obrazem 2 w arkuszu egzaminacyjnym (<i>w uruchomionej aplikacji lub na zrzucie i obowiązkowo w kodzie</i>) |
| R.3.8 | Po wybraniu przycisku jeżeli istnieje następne pytanie w quizie, jest ono wyświetlane oraz wszystkie przyciski radio nie są zaznaczone (<i>w uruchomionej aplikacji lub na zrzucie i obowiązkowo w kodzie</i>) |
| R.3.9 | Po wybraniu przycisku jeżeli jest to ostatnie pytanie, jest wyświetlane pierwsze pytanie (<i>w uruchomionej aplikacji lub na zrzucie i obowiązkowo w kodzie</i>) |
| R.3.10 | Jeżeli wybrana odpowiedź jest prawidłowa, inkrementowana jest liczba punktów |
| R.4 | Rezultat 4: Testy aplikacji i dokumentacja |
| | <i>Uwaga: Zrzuty ekranu z kryterium 4.3 muszą zawierać cały obszar ekranu z widocznym paskiem zadań. Dokumentacja z kryterium 4.3 musi być zgodna ze stanem faktycznym z kodu źródłowego, nawet jeżeli w kodzie są błędy logiczne. Dokumentacja z kryterium 4.6 zapisana jest w pliku egzamin</i> |
| R.4.1 | W aplikacji konsolowej jest wykonany test inicjacji obiektu klasy <i>Pytanie</i> - test nie powiódł się (<i>kod nie skompilował się</i>), co jest udokumentowane na zrzucie ekranu |
| R.4.2 | W aplikacji konsolowej jest wykonany test obiektu klasy <i>PytanieZamkniete</i> : obiekt inicjowano danymi wczytanymi z klawiatury, udzielono odpowiedzi, program wypisał czy odpowiedź jest prawidłowa (<i>metoda sprawdzająca poprawność</i>) co jest udokumentowane na zrzucie ekranu |
| R.4.3 | Zapisany przynajmniej jeden zrzut ekranu z uruchomienia lub kompilacji aplikacji mobilnej, na zrzucie widoczne jest środowisko, w którym powstała aplikacja |
| R.4.4 | Skonfigurowane repozytorium wersji Git (email <i>egzamin@poczta.pl</i> , nazwa użytkownika - numer zdającego) co jest udokumentowane zrzutem ekranu |
| R.4.5 | Wykonane co najmniej dwa zapisy wersji do repozytorium wersji Git co jest udokumentowane zrzutem ekranu |
| R.4.6 | Dokumentacja zawiera: nazwę systemu operacyjnego, nazwy środowisk programistycznych, emulatora, nazwy języków programowania |
| R.4.7 | Przygotowana jest dokumentacja w postaci płyty: oba projekty są spakowane oraz zmodyfikowane pliki są na zewnątrz archiwów |