

Programowanie obiektowe - zadania

1. Napisz klasę Book, która będzie posiadała numer ISBN, tytuł oraz autora. Następnie stwórz parę książek, dodaj je do tablicy, a następnie stwórz drugą tablicę, która będzie zawierała elementy z tej pierwszej, ale tylko takie, których tytuł zaczyna się na 'H'. Nie pozwól, żeby nowa tablica zawierała null. Zobaczysz to jak ją wydrukujesz.
2. Napisz interface Function, w którym zdefiniujesz metodę `calculate()`, która przyjmie jako parametr `int` i zwróci `int`. Następnie napisz klasę HalfCalculator, która będzie implementowała ten interface, a metoda `calculate()` będzie dzieliła podany argument na 2. Napisz też klasę QuarterCalculator, która będzie implementowała ten interface, a metoda `calculate()` będzie dzieliła podany argument na 4. Wywołaj tę metodę, stosując polimorfizm, to znaczy, stwórz 2 obiekty, których referencje będą zadeklarowane jako Function. Następnie wydrukuj na ekranie wynik dla kilku podanych przykładów. Zrób ponownie to samo, tylko tym razem, zastąp interface Function klasą o nazwie FunctionClass.

Później stwórz tablicę na 10 elementów, gdzie tablica będzie typu Function i dodaj do niej obiekty na przemian HalfCalculator i QuarterCalculator. Przechodząc przez wszystkie elementy tablicy, wywołaj metodę `calculate()` na wyniku poprzednim i dodawaj do siebie kolejne otrzymane wyniki zaczynając od jakiejś dużej liczby, np 12642364. Wynik wydrukuj na ekranie.

3. Napisz klasę Square reprezentującą kwadrat, w której zawrzesz 2 konstruktory, jeden domyślny, gdzie bok kwadratu będzie wynosił 5. Drugi, w którym użytkownik może określić tę wartość. Do tego dodaj getter oraz setter. Pamiętaj o enkapsulacji. Dodaj metodę liczącą obwód i metodę liczącą pole kwadratu. Nadpisz metodę `equals()`. Nadpisz metodę `toString()`. Użyj następnie tej klasy, aby stworzyć kilka kwadratów o różnych bokach i umieścić je w tablicy. dodaj parę zduplikowanych kwadratów. Następnie wykorzystując algorytm do usuwania duplikatów, który został napisany w zadaniach z tablicami, usuń zduplikowane kwadraty z tablicy. (Algorytm zakładał sortowanie elementów, załóż zatem, że porównanie kwadratów, który jest większy, a który mniejszy nastąpi po jego polu powierzchni).
4. Napisz program, który odwzoruje relacje jakie mogą występować na uczelni między studentem, nauczycielem i egzaminem z pytaniami. Najpierw egzamin jest tworzony przez nauczyciela, który tworzy np. 5 pytań, o jakiejś treści i 3 odpowiedziach, z czego tylko jedna jest poprawna. Każde pytanie ma zapisaną poprawną odpowiedź i to ustala nauczyciel. Tylko nauczyciel może tę wartość odczytać (enkapsulacja). Następnie nauczyciel rozdaje taki egzamin swoim trzem studentom i każdy z nich odpowiada na pytania (podczas tworzenia studenta zdefiniuj, na które pytanie odpowie poprawnie lub poszukaj w internecie jak wylosować wartość z podanego przedziału).

Po udzieleniu odpowiedzi, studenci przekazują pytania do nauczyciela, który egzaminy sprawdza i ocenia razem z określeniem kto jaką dostał ocenę. Zapewnij, aby student nie mógł odwołać się do informacji, których nie powinien zobaczyć (tzn. nie mógł dostać się do wartości, która odpowiedź w danym pytaniu jest poprawna).

Napisz to tak, aby klasa studenta oraz klasa nauczyciela mogły współdzielić te same cechy (pokazywałem mechanizmy jak to osiągnąć). Zdefiniuj również interfejsy Examined oraz Examiner, w których będzie określone, co muszą być w stanie zrobić egzaminator (np. stworzyć egzamin, albo dokonać jego oceny) oraz osoba egzaminowana (np. napisać egzamin).

Do każdego kroku dodaj drukowanie na ekranie, które pokaże jakie kroki są wykonywane w danym momencie, przykładowo tak jak poniżej:

```
Creating teacher: Nauczyciel Nauczycielski
Creating student: Rafał Rafalski
Creating student: Roman Romański
Creating student: Michał Michalski
Teacher created 3 exams
Student: Rafał Rafalski answering question: 1st Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 3rd answer, index:[2]
Student: Rafał Rafalski answering question: 2nd Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 1st answer, index:[0]
Student: Rafał Rafalski answering question: 3rd Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 3rd answer, index:[2]
Student: Rafał Rafalski answering question: 4th Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 2nd answer, index:[1]
Student: Rafał Rafalski answering question: 5th Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 2nd answer, index:[1]
Student: Roman Romański answering question: 1st Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 2nd answer, index:[1]
Student: Roman Romański answering question: 2nd Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 1st answer, index:[0]
Student: Roman Romański answering question: 3rd Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 2nd answer, index:[1]
Student: Roman Romański answering question: 4th Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 3rd answer, index:[2]
Student: Roman Romański answering question: 5th Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 1st answer, index:[0]
Student: Michał Michalski answering question: 1st Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 2nd answer, index:[1]
Student: Michał Michalski answering question: 2nd Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 3rd answer, index:[2]
Student: Michał Michalski answering question: 3rd Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 1st answer, index:[0]
Student: Michał Michalski answering question: 4th Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 2nd answer, index:[1]
Student: Michał Michalski answering question: 5th Question content!, possible answers: [1st answer, 2nd answer, 3rd answer], answered: 3rd answer, index:[2]
Teacher evaluating Student: Rafał Rafalski exam
Question: 1, content: 1st Question content!, goodAnswerIndex: 1, givenAnswerIndex: 2
Question: 2, content: 2nd Question content!, goodAnswerIndex: 2, givenAnswerIndex: 0
Question: 3, content: 3rd Question content!, goodAnswerIndex: 0, givenAnswerIndex: 2
Question: 4, content: 4th Question content!, goodAnswerIndex: 1, givenAnswerIndex: 1
Question: 5, content: 5th Question content!, goodAnswerIndex: 0, givenAnswerIndex: 1
Student: Rafał Rafalski points scored: 1
Teacher evaluating Student: Roman Romański exam
Question: 1, content: 1st Question content!, goodAnswerIndex: 1, givenAnswerIndex: 1
Question: 2, content: 2nd Question content!, goodAnswerIndex: 2, givenAnswerIndex: 0
Question: 3, content: 3rd Question content!, goodAnswerIndex: 0, givenAnswerIndex: 1
Question: 4, content: 4th Question content!, goodAnswerIndex: 1, givenAnswerIndex: 2
Question: 5, content: 5th Question content!, goodAnswerIndex: 0, givenAnswerIndex: 0
```

```
Student: Roman Romański points scored: 2
Teacher evaluating Student: Michał Michalski exam
Question: 1, content: 1st Question content!, goodAnswerIndex: 1, givenAnswerIndex: 1
Question: 2, content: 2nd Question content!, goodAnswerIndex: 2, givenAnswerIndex: 2
Question: 3, content: 3rd Question content!, goodAnswerIndex: 0, givenAnswerIndex: 0
Question: 4, content: 4th Question content!, goodAnswerIndex: 1, givenAnswerIndex: 1
Question: 5, content: 5th Question content!, goodAnswerIndex: 0, givenAnswerIndex: 2
Student: Michał Michalski points scored: 4
```