

Programowanie Java
Anna Gogolińska
Zestaw 3

Zadania 1-3 dotyczą jednego projektu.

Zad 1. Wykorzystać klasę *Figura* z poprzedniego zestawu zadań.

Stworzyć klasy potomne *Kolo* (zawierającą dodatkowo pole *promien*) i *Prostokat* (zawierające dodatkowo pola *bokA* i *bokB*), metody obliczające pole i obwód danej figury, oraz konstruktory ustawiające wartości wszystkich dodatkowych pól i inicjalizujące obliczenia pola i obwodu. Zmodyfikuj również metodę *toString()*.

Stworzyć klasę potomną klasy *Prostokat* o nazwie *Kwadrat* (wraz z odpowiednim konstruktorem).

Zad 2. Dodaj do klas *Figura*, *Prostokat*, *Kwadrat*, *Kolo* metodę *powieksz(int a)*, która powiększa wszystkie wymiary *a*-krotnie i ponownie oblicza ich pola.

W *main()* stwórz 5-elementową tablicę obiektów zdefiniowaną jako *Figura[]* i dodaj do niej kilka zainicjalizowanych obiektów typów potomnych.

Wykonaj na obiektach z tej tablicy iteracyjnie metodę *powieksz(-)* oraz wypisz informacje o tych elementach.

Metoda zwiększająca wymiary musi zostać dodana do klasy *Kolo* i *Prostokat*. *Kwadrat* będzie dziedziczył ją z klasy *Prostokat*. W klasie *Figura* metoda może pozostać pusta.

Zad 3. Stwórz klasę *Stos*, która będzie dziedziczyć z klasy *ArrayList* i będzie reprezentować stos dla obiektów typu *Object*.

Klasa ma zawierać następujące metody (do ich implementacji użyć metod *ArrayList*):

- *wstaw(-)*, która wstawia element do struktury,
- *pobierz()*, która pobiera element i zwraca jego wartość jako wynik działania metody,
- *czyPusty()*, który sprawdza czy istnieje przynajmniej jeden przechowywany element

W *main()* poza tablicą stworzyć również *Stos* zawierający obiekty klas potomnych dla *Figura*. Zdjąć wszystkie obiekty z stosu i dla każdego z nich wywołać *toString()*.

Zadania samodzielne

Zad 4.

Zdefiniować klasę *Okret* zawierającą długości boków, ilość pasażerów, wyporność, unikalny numer okrętu, trzy wersje konstruktora przeciążonego, metodę obliczającą pole powierzchni (należy założyć, że jest w kształcie prostokąta). Następnie zdefiniować klasy:

- *Niszczyciel* dziedziczącą z klasy *Okret* i uzupełnioną o informację o sile rażenia i ilość dział,
- *Lotniskowiec* dziedziczącą z klasy *Niszczyciel* i uzupełnioną o dane dotyczące ilości samolotów, które może pomieścić.

Do obu klas (*Niszczyciel* i *Lotniskowiec*) dodać metodę *toString()* wypisującą odpowiednie informacje o statkach.

Na koniec, zdefiniować klasę *Flota* zawierającą:

- Listę (*ArrayList*) z elementami typu *Okret*, która będzie wypełniana dynamicznie poprzez utworzoną do tego celu metodę *dodajDoFloty(Okret)*
- Metodę *silaFloty()* zliczającą siłę rażenia, ilość pasażerów i dział w całej flocie
- W *main()* stworzyć obiekt typu *Flota* i wypełnić losowymi danymi. Wypisać informacje o statkach (użyć *toString()* i *silaFloty()*).

Zad 5.

Utworzyć klasę ONP obliczającą wartość wyrażenia podanego w odwrotnej notacji polskiej. W tym celu należy wykorzystać stos utworzony w zadaniu 3. W metodzie *main(-)* należy odczytać wyrażenie ze standardowego wejścia oraz stworzyć obiekt klasy ONP z wykorzystaniem jedyne go konstruktora przyjmującego jako parametr obiekt String zawierający wyrażenie. Ponadto klasa ONP powinna posiadać metodę *wyznaczWartosc()*, która zwróci wartość wyrażenia. Należy założyć, że w wyrażeniu będą tylko liczby całkowite, jeden operator jednoargumentowy - zmieniający znak liczby oraz dwa operatory dwuargumentowe + i * reprezentujące dodawanie i mnożenie.

Przykład

Wejście:

2 3 + 5 *

Wyjście:

30