

Stwórz aplikację do obsługi pralki. Spełniającą, poniższe kryteria:

1. Klasa pralka przechowuje numer i nazwę aktualnego programu prania. Program jest wybierany z listy 10 predefiniowanych programów.
2. Pralka implementuje metody `setProgram(int)` i `getProgram()` oraz metody `nextProgram()`, `previousProgram()`. Metoda `setProgram` ustawia program, `getProgram` zwraca go. `nextProgram` zmienia na kolejny (tak jak okrągłe pokrętko w pralce) i `previousProgram` poprzedni. Gdy użytkownik wybierze przekroczy zakres programów, przeskakuje na koniec/początek w zależności od wybranej opcji.
3. Pralka posiada zmienną `temp` która przechowuje wartość temperatury jako liczbę zmiennoprzecinkową ze skokiem 0,5 stopnia. Temperatura jest z zakresu 0 – 90 stopni. Istnieje możliwość przestawienia pralki na stopnie Farenheit – wówczas zakres się zmienia na 32 – 194, a skok do pełnych 1. Przełączenie można wykonać w dowolnej chwili, co za tym idzie obecnie ustawiona temperatura powinna być konwertowana i zaokrąglana (jeżeli to konieczne) do drugiego systemu jednostek.
4. Pralka implementuje metody `setTemp` `getTemp`. Metody działają na temperaturze zgodnie z systemem jednostek, który obecnie jest ustawiony.
5. Dopisz metody `tempUp()` i `tempDown()` (skok 0 0,5 stopnia Celsjusza, 1 stopień Farenhaita). Zakresy są nie przekraczalne. Przy próbie zmiany temperatury poza zakres powinien się pojawić błąd o przekroczonym zakresie, a wartość powinna się nie zmienić.
6. Zmiana temperatury powoduje wyświetlenie się komunikatu o temperaturze np. „current temp 75 °C”. Podpowiedź tabela ASCII ☺
7. Pamiętaj o odpowiedniej obsłudze błędów i o zabezpieczeniu wszystkich zakresów.
8. Pralka przechowuje wartość 0 – 1000 predkości wirowania. Skok obrotów o 100. Zakres zachowuje się podobnie jak w programach. Każdy program ma ustaloną domyślną wartość zalecanego wirowania, która jest ustawiana po wybraniu programu. Metody `setV` `getV` i `upV` i `downV`. Program „Delikatne”, nie pozwoli zwiększyć wirowania powyżej 500 obrotów.
9. Pralka posiada funkcję `showStatus` która wypisuje komunikat złożony z wszystkich wartości, które przechowuje pralka. Np.: numer programu 4, temperatura 60 °C
10. Każda z pralek ma możliwość przesunięcia startu programu, a każdy program posiada przybliżoną datę trwania.
11. Stwórz klasy pralek Beko, Wirpool, Amica. Pralki te zachowują się tak samo jak pralka z małymi zmianami
12. Pralka beko ma skok temperatury w nie o 0,5 stopnia tylko o 1 °C i o 2 ° F.
13. Pralka wirpool ma 15 programów prania.
14. Pralka Amica i Beko, mają wbudowaną wagę. Pozwala ona na sprawdzenie, czy maksymalna ładowność pralki nie została przekroczona (max ładunek, jest określony dla konkretnego modelu pralki).
15. Stwórz metodę, która pozwoli na oszacowanie realnego czasu trwania cyklu. Szacowany czas jest obliczany z prostego wzoru: Domyślny czas * (aktualna waga/ maksymalna ładowność). Wartość ma być zwracana w raporcie z pkt 9
16. Każda z pralek posiada metodę, która zwraca dane historyczne z 30 ostatnich cykli. Informacje jakie są zawarte w historii to: program, temperatura, prędkość wirowania. Beko i Amica mają dodatkowo wagę prania z danego cyklu i procent załadunku.
17. W metodzie `main` stwórz metodę do optymalizacji prania. Metoda ma prześledzić całą historię prania i po załadunku i programie określić czy któreś pranie można było połączyć. Jeżeli każdy z cykli ma ponad 90% załadunku, wyświetl informacje o braku optymalizacji. W przypadku tych samych programów z mniejszą ładownością wyświetl informacje:
„Zaoszczędziłbyś piorąc rzadziej programem „Nawa programu” w temperaturze „Najniższa temperatura”.

Na koniec punktów 10 i 15 w metodzie `main` stwórz obiekty wszystkich pralek, ustaw i program, temperaturę, podnieś i obniż temperaturę, ustaw prędkość obrotów i dla wszystkich wyświetl status.