

Zadanie E – Owocowy ogród

Punktów do uzyskania: **10**

Zadanie polega na implementacji klas obsługujących owocowy ogród.

- Ogólny opis klas
 - Ogród
 - Jest odwzorowany w klasie o nazwie GARDEN_CLASS.
 - Może obejmować dowolną ilość drzewek.
 - Drzewko
 - Jest odwzorowane w klasie o nazwie WOOD_CLASS.
 - Posiada unikalny nieujemny numer całkowity.
 - Ma określoną nieujemną wysokość.
 - Może obejmować dowolną nieujemną ilość gałęzi.
 - Gałąź
 - Jest odwzorowana w klasie o nazwie BRANCH_CLASS.
 - Ma określoną długość.
 - Może obejmować dowolną ilość owoców.
 - Owo:
 - Jest odwzorowany w klasie o nazwie FRUIT_CLASS.
 - Ma określoną wagę.
 - Wartości użyte dotąd oraz wymienione poniżej zawsze mieszczą się w zakresie typu **unsigned int**:
 - Ilość drzewek.
 - Ilość wszystkich gałęzi.
 - Ilość wszystkich owoców.
 - Suma wag wszystkich owoców.
 - Numer drzewka
 - Wysokość drzewka.
 - Długość gałęzi.
 - Waga owocu.
- Wymagane metody klasy GARDEN_CLASS.
 - **unsigned int** getWoodsTotal (**void**)
Zwraca ilość drzewek w ogrodzie.
 - **unsigned int** getBranchesTotal (**void**)
Zwraca ilość gałęzi w ogrodzie.
 - **unsigned int** getFruitsTotal (**void**)
Zwraca ilość owoców ogrodu.
 - **unsigned int** getWeightsTotal (**void**)
Zwraca wagę owoców w ogrodzie.
 - **void** plantWood (**void**)
Dodaje nowe drzewko o najniższym możliwym identyfikatorze, zerowej wysokości i zerowej ilości gałęzi.
 - **void** extractWood (**unsigned int**)
Usuwa drzewko o numerze danym w argumencie o ile istnieje.
 - **void** growthGarden (**void**)
Dojrzewa każdego drzewko według metody growthWood dla klasy WOOD_CLASS.
 - **void** fadeGarden (**void**)
Wiednie każde drzewko według metody fadeWood dla klasy WOOD_CLASS.
 - **void** harvestGarden (**unsigned int**)
Zrywa wszystkie owoce o wadze nie mniejszej od danej argumentem zgodnie z metodą harvestWood dla klasy WOOD_CLASS.
 - **void** cloneWood (**unsigned int**)
 - Dodaje nowe drzewko o najniższym możliwym numerze będące klonem drzewka o numerze danym argumentem - o ile istnieje.

- Uwaga: w implementacji możliwe użycie wyłącznie konstruktorów kopiujących.
- WOOD_CLASS* getWoodPointer (**unsigned int**)
Zwraca wskaźnik do drzewka o numerze danym w argumencie lub wartość NULL, gdy wymaganego drzewka brak.
- Wymagane metody klasy WOOD_CLASS.
 - **unsigned int** getBranchesTotal (**void**)
Zwraca ilość gałęzi drzewka.
 - **unsigned int** getFruitsTotal (**void**)
Zwraca ilość owoców drzewka.
 - **unsigned int** getWeightsTotal (**void**)
Zwraca sumę wag wszystkich owoców drzewka.
 - **unsigned int** getNumber (**void**)
Zwraca numer drzewka.
 - **unsigned int** getHeight (**void**)
Zwraca wysokość drzewka.
 - **void** growthWood (**void**)
 - Zwiększa wysokość drzewka o wartość 1.
 - Każdą dotychczasową gałąź drzewka dojrzewa według metody growthBranch dla klasy BRANCH_CLASS.
 - Osiągnięcie wysokości drzewka równej każdej dodatniej wielokrotności liczby 3 powoduje powstanie nowej gałęzi o zerowej długości i zerowej ilości owoców.
 - **Uwaga: Na określonej wysokości może rosnąć jedna gałąź.**
 - **void** fadeWood (**void**)
 - Zmniejsza wysokość drzewka o wartość 1.
 - Każda gałąź więdnie według metody fadeBranch dla klasy BRANCH_CLASS.
 - Każda gałąź powyżej zmniejszonej wysokości zostaje usunięta.
 - Drzewko o zerowej wysokości nie zostaje usunięte i może dojrzewać.
 - **void** harvestWood (**unsigned int**)
Wszystkie owoce wszystkich gałęzi o wadze nie mniejszej od danej argumentem zostają zerwane zgodnie z metodą harvestBranch dla klasy BRANCH_CLASS.
 - **void** cutWood (**unsigned int**)
 - Przycina drzewko do wysokości danej argumentem.
 - Wszystkie gałęzie powyżej zmniejszonej wysokości zostają usunięte.
 - **void** cloneBranch (**BRANCH_CLASS***)
 - Pierwsza gałąź (licząc od najniższej wysokości drzewka) o zerowej długości zostaje zastąpiona klonem gałęzi wskazanej argumentem.
 - Uwaga: w implementacji możliwe użycie wyłącznie konstruktorów kopiujących.
 - GARDEN_CLASS* getGardenPointer (**void**)
Zwraca wskaźnik do ogrodu, w którym rośnie dane drzewko, lub wartość NULL gdy drzewko istnieje bez ogrodu.
 - BRANCH_CLASS* getBranchPointer (**unsigned int**)
Zwraca wskaźnik do gałęzi wyrastającej na wysokości danej argumentem lub wartość NULL, gdy brak gałęzi na zadanej wysokości.
- Wymagane metody klasy BRANCH_CLASS.
 - **unsigned int** getFruitsTotal (**void**)
Zwraca ilość owoców na gałęzi.
 - **unsigned int** getWeightsTotal (**void**)
Zwraca sumaryczną wagę owoców na gałęzi.

- **unsigned int** getHeight (**void**)
Zwraca wysokość na jakiej rośnie gałąź.
- **unsigned int** getLength (**void**)
Zwraca długość gałęzi.
- **void** growthBranch (**void**)
 - Zwiększa długość gałęzi o wartość 1.
 - Każda dotychczasowy owoc zwiększa wagę o 1.
 - Osiągnięcie długości równej każdej dodatniej wielokrotności liczby 2 powoduje powstanie nowego owocu o wadze wynoszącej 0.
 - **Uwaga: Na określonej długości może rosnąć jeden owoc.**
- **void** fadeBranch (**void**)
 - Zmniejsza długość gałęzi o wartość 1.
 - Zmniejsza wagę każdego owocu o wartość 1.
 - Usuwa każdy owoc powyżej zmniejszonej długości.
 - Gałąź o zerowej długości nie zostaje usunięta i może dojrzewać.
- **void** harvestBranch (**unsigned int**)
Zrywa wszystkie owoce o wadze nie mniejszej od danej argumentem zgodnie z metodą pluckFruit dla klasy FRUIT_CLASS.
- **void** cutBranch (**unsigned int**)
 - Przycina gałąź do długości danej argumentem.
 - Usuwa wszystkie owoce na długości powyżej zmniejszonej.
- FRUIT_CLASS* getFruitPointer (**unsigned int**)
Zwraca wskaźnik do owocu wyrastającego w odległości danej argumentem lub wartość NULL, gdy owocu w zadanej odległości brak.
- WOOD_CLASS* getWoodPointer (**void**)
Zwraca wskaźnik do drzewa obejmującego daną gałąź.
- Wymagane metody klasy FRUIT_CLASS.
 - **unsigned int** getLength (**void**)
Zwraca długość na jakiej rośnie owoc.
 - **unsigned int** getWeight (**void**)
Zwraca wagę owocu.
 - **void** growthFruit (**void**)
Zwiększa wagę owocu o wartość 1.
 - **void** fadeFruit (**void**)
Zmniejsza wagę owocu o wartość 1.
 - **void** pluckFruit (**void**)
Zeruje wagę owocu nie usuwając go.
 - BRANCH_CLASS* getBranchPointer (**void**)
Zwraca wskaźnik do gałęzi obejmującej dany owoc.
- Dodatkowe uwarunkowania:
 - Wszystkie pola muszą mieć hermetyzację **private**.
 - Wszystkie metody muszą mieć hermetyzację **public**.
 - Zabronionymi są:
 - Zmienne globalne.
 - Nawiasy kwadratowe.
 - Słowa **friend** i **static**.
 - Podprogramy niestanowiące metod, czyli ogólnie podprogramy zewnętrzne.
 - Własne zewnętrzne klas inne niż cztery wymagane.
 - Kontenery.
 - Identyfikatory zaczynające się znakiem podkreślenia.
 - Brak wygórowanych wymagań czasowych poza zakazem stosowania algorytmów o złożoności liniowej w sytuacji, gdy istnieje rozwiązanie o złożoności stałej, czy zakazem stosowania algorytmów o złożoności kwadratowej, gdy istnieje rozwiązanie o złożoności liniowej.