

Napišite program koji predstavlja sadnju vrta. U vrtu se može saditi cvijeće i stabla. Sam vrt se sastoji od nekoliko parcela. Pojedina parcela može biti namijenjena ili sadnji cvijeća ili sadnji stabala. Vrt održava vrtljar koji mora svaku sadnicu koju dobije pokušati posaditi u vrt na odgovarajuću parcelu. Nakon što je biljka posađena treba ju zaliti da bi narasla.

Treba definirati:

paket **plants**

- **Plantable** – Sučelje, predstavlja biljke koje se mogu posaditi u vrt.
 - `void addWater();` Metoda kojom se zalijeva biljka.
- **Tree, Flower** – Apstraktne klase koje implementiraju **Plantable**, predstavljaju tip biljke. Biljke tipa **Flower** je potrebno jednom zaliti da bi narasle, biljke tipa **Tree** je potrebno zaliti dva puta.
- Nekoliko proizvoljnih konkretnih klasa koje predstavljaju razno cvijeće i stabla. Potrebno je imati barem dvije biljke svake vrste.
 - Svaka konkretna klasa treba implementirati metodu `toString()` tako da ispisuje naziv biljke malim slovima ako biljka nije narasla, velikim slovima ako je.

paket **garden**

- **Plot** – Sučelje koje predstavlja parcele u vrtu.
 - **CAPACITY** - Konstanta koja kaže koliko biljaka parcela može primiti (sve parcele u vrtu su iste veličine i stablo zauzima jednako mjesta kao i cvijet).
 - `boolean addPlant(int i, Plantable p) throws InvalidPlantType, SpaceOccupiedException;` – Metoda kojom se na zadano mjesto na parceli sadi zadana biljka. Iznimku `InvalidPlantType` je potrebno baciti ako tip biljke ne odgovara tipu parcele, `SpaceOccupiedException` ako na zadanom mjestu već postoji neka druga biljka.
 - `Plantable getPlant(int i);` – Vraca biljku posađenu na zadanom mjestu u parceli, `null` ako je mjesto prazno.
 - `int countPlants();` - Vraća broj biljaka posađenih na ovoj parceli.
- **Flowerbed, Orchard** - Konkretne klase koje predstavljaju tip parcele, implementiraju sučelje **Plot**. U parcelu tipa **Flowerbed** se smiju saditi samo biljke tipa **Flower**, u **Orchard** samo **Tree**.
 - Sadrže array svog dozvoljenog tip biljke veličine **CAPACITY**.
 - Implementiraju `toString` tako da ispišu sve biljke koje sadrže (konkretno biljke ispisati pozivom njihove `toString` metode). Ako na nekom mjestu nema biljke, to mjesto treba biti označeno kao prazno u ispisu.

- **Garden** – Konkretna klasa koja predstavlja cijeli vrt. Sadrži array parcela.
 - Implementira `toString` metodu tako da ispiše sve parcele koje vrt sadrži pozivom `parceline toString` metode. Molim, formatirajte ispis cvijeća, parcela i vrta tako da bude pregledan.

paket **gardener**

- **Gardener** - Konkretna klasa koja predstavlja vrtlara. Sadrži metode koje predstavljaju vrtlareve radnje u vrtu. Od klasa iz paketa `plants` i `garden`, ova klasa smije koristiti samo `Plantable`, `Plot` i `Garden` – treba raditi sa sučeljima i apstraktnim klasama, a ne konkretnim biljkama i tipovima parcela. Ako se naknadno dodaju novi tipovi biljaka ili parcela, postojeća implementacija vrtlara bi trebala i dalje ispravno raditi.
 - `plant (Garden g, Plantable p) throws NoRoomInGardenException;` Metoda treba pokušati posaditi biljku `p` na prvo slobodno mjesto u vrtu `g`. (vrtlar treba pretražiti redom parcele i pokušati naći slobodno mjesto u nekoj koja odgovara tipu biljke koja se sadi). Iznimka `NoRoomInGardenException` se baca ukoliko u vrtu nema slobodnog mjesta za biljku.
 - `waterPlants (Garden g) ;` Jednom zalijeva sve biljke u vrtu.
 - `admirePlants (Garden g) ;` U standardni izlaz ispisuje sadržaj vrta (poziv `toString` na zadanom vrtu).

Osim toga, program treba sadržavati klasu za pokretanje koja sadrži `main` metodu i u kojoj se:

- Instancira vrt s nekoliko parcela i vrtlarom tako da u vrtu bude barem jedna parcela svake vrste (`Flowerbed` i `Orchard`).
 - Kreiranje vrta se može podržati prema podacima iz datoteke (tako da npr. datoteka sa sadržajem FFOF opisuje vrt s 3 `Flowerbed`-a i 1 `Orchard`-om). Za čitanje datoteke možete koristiti.

```
...
Path path = Paths.get("C:\\hello.txt");
List<String> fileLines = Files.readAllLines(path)
...
```

- Nije nužno da se vrt učitava iz datoteke, kreiranje možete i hardkodirati u `main` metodi prije početka zadavanja naredbi vrtlaru ili možete čak dodati uređenje parcela kao novu funkcionalnost vrtlara.
- Omogućuje davanje naredbi vrtlaru koristeći:

```
...
String command = null;

while ((command = in.readLine()) != null && !command.equals("Q")) {
```

```

//kod koji za naredbu koju korisnik unese poziva odgovarajuću
//metodu na vrtlaru
}
...

```

- Potrebno je podržati naredbe:
 - 'W' – zalijevanje biljaka waterPlants
 - 'L' – gledanje vrta, poziva admirePlants i ispisuje sadržaj vrta u standardni izlaz (System.out.println).
 - 'PX' – sadi biljku 'X', instancira se biljka oznake 'X' i poziva plant.
 Ukoliko u vrtu nema mjesta za biljku, program treba javiti odgovarajuću potruku i nastaviti izvršavanje. Popis biljki koje se mogu saditi (potrebno je podržati barem dva cvijeta i dva stabla):
 - Cvijeće (Flower) :
 - Amarant - Amaranthus - "A"
 - Hortenzija - Hydrangea "H"
 - Neven - Calendula "C"
 - Narcis - Daffodil "D"
 - Frezija - Freesia "F"
 - Tulipan - Tulip "T"
 - Ruža - Rose "R"
 - Jasmin - Jasmine "J"
 - Ljubičica - Violet "V"
 - Iris - Iris "I"
 - Stabla (Tree) :
 - Magnolija - Magnolia "M"
 - Hrast - Oak "O"
 - Bukva - Beech "B"
 - Bor - Pine "P"
 - Smreka - Spruce "S"
 - Negundovac - Acer Negundo "N"
 - Brijest - Elm "E"
 - Kesten - Chestnut "K"
 - Lipa - Basswood - "B"
 - Glog - Hawthorne "H"
 - 'Q' – Završetak programa.
 - Nakon zadavanja naredbe, vrtlar treba odraditi zadanu naredbu i javiti što je odradio. Ako se unese neka naredba koja nije podržana, program treba javiti odgovarajuću poruku i nastaviti izvođenje (čeka sljedeći zadatak).

Primjer kako bi moglo izgledati pokretanje programa za vrt s jednim Flowerbed-om i jednim Orchard-om. Počinje se s praznim vrtom od dvioje parcele. U vrt se posade jedno stablo i jedan cvijet te se zaliju. Nakon zalijevanja, cvijet je narastao, a stablo nije (jer su mu potrebna dva zalijevanja).

The gardener has entered the garden:

Garden:

0. Flowerbed:

0. *empty* | 1. *empty* | 2. *empty* | 3. *empty* | 4. *empty* |

1. Orchard:

0. *empty* | 1. *empty* | 2. *empty* | 3. *empty* | 4. *empty* |

Enter command: P(lant)[Plant type] - e.g. PO to (P)lant an (O)ak | W(ater plants) | L(ook at garden) | Q(uit)

PB

Gardener: I've planted the beech.

Enter command: P(lant)[Plant type] - e.g. PO to (P)lant an (O)ak | W(ater plants) | L(ook at garden) | Q(uit)

PR

Gardener: I've planted the rose.

Enter command: P(lant)[Plant type] - e.g. PO to (P)lant an (O)ak | W(ater plants) | L(ook at garden) | Q(uit)

W

Gardener: Drink my children, drink and GROW! HAHAAHAHA!!

Enter command: P(lant)[Plant type] - e.g. PO to (P)lant an (O)ak | W(ater plants) | L(ook at garden) | Q(uit)

L

The gardener admires the garden

Garden:

0. Flowerbed:

0. ROSE | 1. *empty* | 2. *empty* | 3. *empty* | 4. *empty* |

1. Orchard:

0. beech | 1. *empty* | 2. *empty* | 3. *empty* | 4. *empty* |

Enter command: P(lant)[Plant type] - e.g. PO to plant an Oak | W(ater plants) | L(ook at garden) | Q(uit)

Q

The gardener leaves the garden.

Nazivi klasa, varijabli, metoda itd. trebaju biti napisani na engleskom jeziku i pridržavati se Java Naming Convention

(<https://www.oracle.com/technetwork/java/codeconventions-135099.html>).

Sve public metode trebaju imati javadoc, poželjno je da ga imaju i ostale metode.

Poruke koje se program ispisi mogu biti na hrvatskom ili engleskom jeziku (ali sve na istom). Format je proizvoljan, no mora prikazati sve tražene informacije.

Za implemetaciju će vam možda biti potreban i koncept 'cast'-anja koji mislim da nismo prošli na predavanju. Kratki opis i primjer možete naći ovdje:

<https://docs.oracle.com/javase/tutorial/java/landl/subclasses.html>.