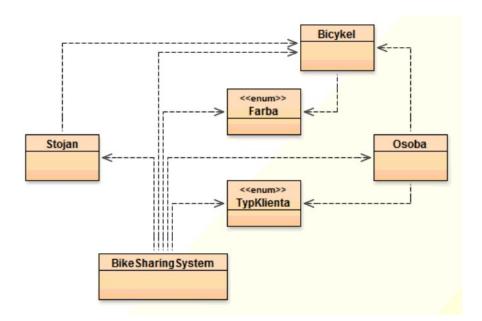
ZADANIE

Podľa priloženého UML diagramu a popisu metód naprogramujte v prostredí BlueJ jednoduchý informačný systém pre službu bike-sharing!



UML diagram

	osik: boolean)		vVstojane: int)
Bioykel	+ new(paFarba: Farba, paCisloBioykla: int, paMaNakupnyKosik: boolean) + getFarba(): Farba + getCisloBioykla(): int + setInfoMaNakupnyKosik(): boolean + getInfoMaNakupnyKosik(): boolean + toString(): String	BikeSharingSystem	+ new/paPocetStojanov: int, paVelkostStojana: int, paPocetBicyklovVstojane: int) + urciPoplatok(paTypKilenta: TypKilenta): double + urciPoplatok(paTypKilenta: TypKilenta): double + uypisZoznamRegistrovanychUserov(): void + uypisZoznamBicyklovVstojanoch(): void + uypisZoznamBicyklovVstojanoch(): void + uypisZoznamPozicanychBicyklovSsobami(): void + uypisZoznamPozicanychBicyklovSsobami(): void + urstBicykel(paOsoba: Osoba, paIndexStojanu: int): void + pozicaiBicykel(paOsoba: Osoba, paIndexStojanu: int): void
	o: TypKlienta)		+ + + + + + + +
Osoba	+ new(paMeno: String, paPriezvisko: String, paTvp: TvpKlienta) + getMenoPriezvisko(): String + getTypKlienta(): TypKlienta + getKredit(): double + zvysKredit(paSuma: double): void + znizKredit(paSuma: double): void + getMojBioykel(): Bioykel + setMojBioykel(): Bioykel + toString(): String	Stojan	+ new(paKapacita: int) + getKapacita(): int + getAktualnyPooetBioyklov(): int + existujeVolneMiesto(): boolean + uvolniBioykel(): Bioykel + pridajBioykel(paBioykel: Bioykel): boolean + getZoznamBioyklov(): ArrayList <bioykel></bioykel>
ta ta			nt): Farba
«enum» TypKlienta	+ Standard + Student + Senior + VIP	«enum»	+ Biela + Modra + Cervena + Zelena + getFarba(index: int): Farba

Atribúty si navrhnite podľa svojho uváženia!

Stručný popis jednotlivých metód

(pokiaľ nie je uvedené inak, každá metóda je za 1 bod, atribúty sú spoločne za 1 bod)

(1b) Enumerácia TypKlienta

- Enumerácia, ktorá obsahuje nasledovné hodnoty: Standard, Student, Senior, VIP

(2b) Enumerácia Farba

- Enumerácia, ktorá obsahuje nasledovné hodnoty: Biela, Modra, Cervena, Zelena
- Okrem toho obsahuje metódu Farba **getFarba**(int index), ktorá vráti farbu na základe indexu (poradia)

(10b) Trieda Osoba

- Objekty tejto triedy budú reprezentovať konkrétnych klientov (cyklistov)
- Deklarujte potrebné atribúty (1b) a metódy:
 - o Osoba(String paMeno, String paPriezvisko, TypKlienta paTyp)
 - Parametrický konštruktor triedy
 - o void **zvysKredit**(double paSuma)
 - Metóda na zvýšenie kreditu (ošetriť nesprávne hodnoty)
 - o void **znizKredit**(double paSuma)
 - Metóda na zníženie kreditu po zaplatení poplatku za požičanie bicykla (ošetriť nesprávne hodnoty)
 - o SET metódy:
 - void setMojBicykel(Bicykel paBicykel)
 - o GET metódy:
 - String getMenoPriezvisko()
 - TypKlienta getTypKlienta()
 - double getKredit()
 - Bicykel getMojBicykel()
 - Metóda String toString()

(7b) Trieda Bicvkel

Objekty tejto triedy budú reprezentovať bicykle, ktoré sa požičiavajú

- Deklarujte potrebné atribúty (1b) a metódy:
 - o **Bicykel**(Farba paFarba, int paCisloBicykla, boolean paMaNakupnyKosik)
 - Parametrický konštruktor triedy
 - SET metódy:
 - void setInfoMaNakupnyKosik(boolean paNovaHodnota)
 - GET metódy:
 - Farba getFarba()
 - int getCisloBicykla()
 - boolean getInfoMaNakupnyKosik()
 - Metóda String toString()

(12b) Trieda Stojan

- Objekty tejto triedy budú reprezentovať konkrétne stojany na bicykle
- Deklarujte potrebné atribúty (1b) a metódy:
 - o Stojan(int paKapacita)
 - Parametrický konštruktor triedy
 - GET metódy:
 - int getKapacita()
 - (2b) int getAktualnyPocetBicyklov()
 - boolean existujeVolneMiesto()
 - o (2b) Bicykel **uvolniBicykel**()
 - Metóda vyberie zo stojanu bicykel a vráti referenciu naň.
 - o (2b) boolean **pridajBicykel**(Bicykel paBicykel)
 - Metóda nájde voľné miesto v stojane a vloží tam bicykel. Následne vráti true. Ak sa voľné miesto nenájde, metóda vráti false.
 - o (2b) ArrayList<Bicykel> getZoznamBicyklov()
 - Metóda vytvorí zoznam bicyklov umiestnených v danom stojane.

(18b) Trieda BikeSharingSystem

- Objekt tejto triedy bude reprezentovať konkrétny systém
- Deklarujte potrebné atribúty (1b) a metódy:
 - o (3b) **BikeSharingSystem**(int paPocetStojanov, int paVelkostStojana, int paPocetBicyklovVstojane)
 - Parametrický konštruktor triedy
 - o private double **urciPoplatok**(TypK lienta paTypK lienta)
 - Metóda vráti výšku poplatku za každé požičanie bicykla pre konkrétny typ klienta (napríklad: štandardná cena je 50 centov, seniori majú 20 centov, študenti 30 centov, VIP klienti zadarmo)
 - o void vypisZoznamRegistrovanychUserov()
 - o void pridajUsera(Osoba paOsoba)
 - o (2b) void vypisZoznamBicyklovVstojanoch()
 - Metóda prejde všetky stojany a pre každý vypíše zoznam bicyklov, ktoré sú v ňom.
 - void vypis Priemerny Zisk Zjazdy()
 - o (2b) void vvpisZoznamPozicanychBicyklovSosobami()
 - Metóda prejde zoznam registrovaných používateľov a ak má používsteľ požičaný bicykel, vypíšu sa informácie o klientovi a tiež o bicykli.
 - o (3b) void vratBicykel(Osoba paOsoba, int paIndexStojanu)
 - Metóda skontroluje, či má osoba skutočne požičaný bicykel. Pokúsi sa ho vrátiť do stojanu na voľné miesto, pričom, ak by sa to nepodarilo, bicykel zostane u klienta.
 - o (3b) void **pozicajBicykel**(Osoba paOsoba, int paIndexStojanu)
 - Metóda skontroluje, či si klient môže požičať bicykel. Nesmie mať požičané dva a viac. Ak sa v požadovanom stojane nájde voľný bicykel (nemusí sa), skontroluje sa, či má klient dostatok kreditu, aby si bicykel požičal. Ak áno, je mu pridelený bicykel (odstráni sa zo stojanu) a ak nie, používateľ bicykel nedostane.