**Objektiniai uždaviniai**

1. Sukurti klasę **Kibiras1**. Sukurti protected savybę **akmenuKiekis**. Parašyti šiai savybei metodus **prideti1Akmeni()** **pridetiDaugAkmenu($kiekis)** ir metodą **kiekPririnktaAkmenu()**. Sukurti kibiro objektą ir pademonstruoti akmenų rinkimą į kibirą ir rezultatų išvedimą.
2. Sukurti klasę **Piniginė**. Sukurti dvi privačias savybes **popieriniaiPinigai** ir **metaliniaiPinigai**. Parašyti metodą **ideti($kiekis)**, kuris prideda pinigus į piniginę. Jeigu kiekis nedidesnis už 2, tai prideda prie **metaliniaiPinigai**, jeigu kitaip- prie **popieriniaiPinigai**. Parašykite metodą **skaiciuoti()**, kuris suskaičiuotų ir atspausdintų **popieriniaiPinigai** ir **metaliniaiPinigai** sumą. Sukurti klasės objektą ir pademonstruoti veikimą. Nesvarbu kokios popierinės kupiūros ir metalinės monetos egzistuoja realiame pasaulyje.
3. *(STATIC)* Sukurkite klasę kaip pirmame uždavinyje ir pavadinkite **Kibiras2**. Patobulinkite pridėdami statinę privačią savybę **akmenuKiekisVisuoseKibiruose**. Ši savybė turi rodyti kiek akmenų surinkta visuose **Kibiras2** objektuose. Sukurkite geterį objekte, ir statinį geterį klasėje, kuris išvestų **akmenuKiekisVisuoseKibiruose** reikšmę. Sukurkite tris kibirus ir pademonstruokite veikimą.
4. *(EXTENDS)* Sukurkite klasę kaip pirmame uždavinyje ir pavadinkite **Kibiras3**. Sukurkite dar vieną klasę **KibirasNePo1**, kuri extendina klasę **Kibiras3**. **KibirasNePo1** turi naudoti visus tėvinius metodus, bet metodas **Prideti1Akmeni()** turi pridėti ne vieną o atsitiktinį akmenų kiekį nuo 2 iki 5. Sukurkite **KibirasNePo1** objektą ir pademonstruokite veikimą.
5. Sukurti klasę **Stikline**. Sukurti privačią savybę **turis** ir **kiekis**. Parašyti metodą **ipilti($kiekis)**, kuris keistų savybę **kiekis**. Jeigu stiklinės tūris yra mažesnis nei pilamas kiekis- kiekis netelpa ir būna lygus tūriui. Parašyti metodą **ispilti()**, kuris grąžiną kiekį. Pilant išpilamas visas kiekis, tas kas netelpa, nuteka per stalo viršų. Sukurti tris stiklinės objektus su tūriais: 200, 150, 100. Didžiausią pripilti pilną ir tada ją ispilti į mažesnę stiklinę, o mažesnę į dar mažesnę.
6. Sukurti klasę **Grybas**. Sukurti klasę **Krepsys**. **Krepsys** turi konstantą **dydis** lygią 500. **Grybas** turi tris privačias savybes: **valgomas, sukirmijes, svoris**. Kuriant **Grybo** objektą jo savybės turi būti atsitiktinai priskiriamos taip: **valgomas**- true arba false, **sukirmijes**- true arba false ir **svoris**- nuo 5 iki 45. Eiti grybauti, t.y. Kurti naujus **Grybas** objektus, jeigu nesukirmijęs ir valgomas dėti į **Krepsi** objektą, kol bus pririnktas pilnas krepšys nesukirmijusių ir valgomų grybų (gali būti biški daugiau nei **dydis**).
7. Patobulinti 2 uždavinio piniginę taip, kad būtų galima skaičiuoti kiek piniginėje yra monetų ir kiek banknotų. Parašyti metodą **monetos()**, kuris skaičiuotų kiek yra piniginėje monetų ir metoda **banknotai()** - popierinių pinigų skaičiavimui.
8. *(STATIC)* Sukurkite klasę **MarsoPalydovas**. Kontroliuokite objekto kūrimą iš klasės naudodami statinį metodą. Padarykite taip, kad iš viso būtų galima sukurti tik du objektus iš šitos klasės. Pirmam sukuriamam objektui įrašykite privačią savybę **title** lygią stringui “Deimas”, o antram tokią pat savybę tik lygią stringui “Fobas”. Bandant sukurti trečią objektą, turėtų būti grąžinamas vienas iš anksčiau sukurtų objektų parinktas atsitiktine tvarka.
9. *(STATIC)* Sukurti klasę **Tenisininkas**. Klasė **Tenisininkas** turi privačią savybę **vardas**, privačią savybę **kamuoliukas** (true jei turi ir false jei ne) privačią static savybę **zaidejas1**, privačią static savybę **zaidejas2** (žaidėjų objektams saugoti) Klasė turi tokius metodus:

A. Public **arTuriKamuoliuka()**;

B. Public **perduotiKamuoliuka()** Perduoda kamuoliuką kitam **Tenisininkas** objektui;

C. Public static **zaidimoPradzia()** Perduoda kamuoliuką atsitiktiniam žaidėjo objektui;

Sukurti du **Tenisininkas** objektus. Kamuoliuko neturi nei vienas. Iškviesti statinį metodą **zaidimoPradzia()** ir kažkuriam žaidėjui priskirti kamuoliuką. Žaidėjo objekto metodu **perduotiKamuoliuka()** perduoti kamuoliuką kitam žaidėjui. Žaidėjas, kuris neturi kamuoliuko, perduoti kitam negali.