**Ülesanne TUTBank – kokku maksimaalselt 10 punkti (iga osaülesanne – max 2 p).**

TTÜ on asutanud oma panga ja programmi ülesandeks on pangakaartide käitlemise süsteemi modelleerimine. Lisaks tavalisele pangakaardi funktsionaalsusele töötab pangakaart ka sooduskaardina makse tegemisel TTÜ linnaku territooriumil olevates müügikohtades.

1. Iga pangakaardiga peab olema seotud viiekohaline kasutajatunnus (juhuarv vahemikus 10000 kuni 99999). **Garanteerida, et väljastatud kaartide kasutajatunnused oleksid unikaalsed.**
2. Igal pangakaardil peab olema võimalus järgmiste operatsioonide tegemiseks:
* Algjäägi seadmine
* Vaba jäägi päring
* Makse teostamine
* Soodustuse protsendi arvutamine maksel TTÜ müügikohas
* Pangakaardiga seotud info esitus tekstilisel kujul

**Kui operatsiooni teostamine pole võimalik, peab vastav meetod tagastama spetsiaalse väärtuse, mille kontroll ja veateate väljastamine peab toimuma testprogrammis main()** .

1. Pangakaarte on kolme tüüpi: tudengikaart, õppejõukaart ja muu personali kaart. Põhiliseks erinevuseks nende vahel on reegel soodustuse protsendi arvutamiseks:
2. Tudengil
* Kui kaubagrupp on 3, 7, 8 või 11, siis on soodustus 10%
* Kui kaubagrupp on 4 või 15, siis on soodustus 7%
1. Õppejõul
	* Kui kaubagrupp on 2, 7 või 9, siis on soodustus 11%
	* Kui kaubagrupp on 1 või 12, siis on soodustus 5%
2. Muu personal
* Kui kaubagrupp on 6 või 13, siis on soodustus 8%

Lisage baasklassi meetod ostumakse tegemiseks (argumentideks kaubagrupp ja kauba tavahind). Testklassis demonstreerige kõigi meetodite toimimist erinevat tüüpi pangakaartide korral.

1. Looge klass, milles on meetod uue pangakaardi tekitamiseks (argumentideks pangakaardi tüüp ja algjääk). Klass peab järgima mustrit [*Factory Method Pattern*](http://www.javatpoint.com/factory-method-design-pattern)(vt. ka harjutust HotDrinks). Pangakaardi tüüpi peaks kirjeldama spetsiaalne *enum* andmetüüp. **Testklassis moodustage erinevat tüüpi pangakaartidest üks kollektsioon ja demonstreerige polümorfismi toimimist kõigi kollektsiooni kuuluvate kaartide jaoks.**
2. Tudengikaart on tehtud eriti tugevast materjalist ja sellega saab avada korgitud (õlle-)pudeleid. Looge vastav meetod, mille argumendiks on joogi nimetus (**NB! Loodav meetod peab eksisteerima ainult tudengikaartide klassis!**). Testprogrammis demonstreerige selle meetodi toimimist kõigi kollektsioonis olevate tudengikaartide korral. NB! See on ka ainuke koht programmis, kus **objekti tüübi kindlaks tegemiseks** tohib (ja tulebki!) kasutada *if*-lauset ja allateisenduse (*downcast*) operatsiooni dynamic\_cast!

LISAÜLESANNE (max 2 punkti). Iga loodava klassi lähtetekst paigutada eraldi **moodulisse** (**.h** ja **.cpp** faili).

**TÄHTAEG: 8. Detsember**

**Üldised nõuded kodutöödele**.

* Programm peab töötama!
* Programm teeb täpselt seda ja ainult seda, mis on nõutud ülesandes.
* Programmi ülesehitus peab baseeruma OOP 3 vaalal: [kapseldamine (ka realisatsiooni varjamine!)](http://en.wikipedia.org/wiki/Encapsulation_%28object-oriented_programming%29), tuletamine, polümorfism.
* Programmi lähtetekst **peab** olema „[trepitud](http://foorum.hinnavaatlus.ee/viewtopic.php?t=294507)“ («[лесенкой](http://www.amse.ru/courses/algorithms6/03/Silina/03.php)») – treppimise *stiili* võite valida ise. Treppimata lähtekoodiga programm ei kuulu hindamisele! Ka klasside, nende väljade ja meetodite **nimede** juures tuleks järgida mingit ühte [stiili](https://en.wikibooks.org/wiki/C%2B%2B_Programming/Weblinks#C.2B.2B_Coding_Conventions).
* Kogu programm (eriti aga kõigi klasside kõik väljad ja meetodid) peab olema varustatud asjakohaste kommentaaridega.
* [Copy/Paste stiilis programmeerimine](https://en.wikipedia.org/wiki/Copy_and_paste_programming) on karistatav trahvipunktidega! Alternatiiviks on korduvalt kasutatava koodi paigutamine alamprogrammi või meetodisse.
* Kui töö kaitsmisel tudeng ei oska seletada oma programmi ülesehitust ja selle toimimise loogikat, siis ka funktsionaalselt õige lahenduse korral saab tudeng hindeks mitte rohkem kui 1/3 ülesande maksimumpunktidest.