View->Web Layout – For the testCases to appear better

PROBLEMS

# 1.

Издавачката куќа FINKI-Education издава онлајн и печатени книги. За секоја книга се чуваат податоци за ISBN бројот (низа од најмногу 20 знаци), насловот (низа од најмногу 50 знаци), авторот (низа од најмногу 30 знаци) и основната цена изразена во $ (реален број). Класата за опишување на книгите е апстрактна **(5 поени)**.

За секоја онлајн книга дополнително се чуваат податоци за url од каде може да се симне (динамички резервирана низа од знаци) и големината изразена во MB (цел број). За секоја печатена книга дополнително се чуваат податоци за масата изразена во килограми (реален број) и дали ја има на залиха (логичка променлива). **(5 поени)**

За секој објект од двете изведени класи треба да бидат на располагање следниве методи:

* Метод bookPrice, за пресметување на продажната цена на книгата на следниот начин: **(10 поени)**
  + За онлајн книга - цената се зголемува за 20% од основната цена ако книгата е поголема од 20MB
  + За печатена книга - цената се зголемува за 15% од основната цена ако масата на книгата е поголема од 0.7kg
* Преоптоварен оператор > за споредба на две книги од каков било вид според нивната цена. **(5 поени)**
* Преоптоварен оператор << за печатење на податоците за книгите во формат: **(5 поени)**

Да се имплементира функција mostExpensiveBook со потпис:

void mostExpensiveBook (Book\*\* books, int n)

во која се печати вкупниот број на онлајн, односно, печатени книги во проследената низа посебно. **(5 поени)** Потоа се наоѓа и печати најскапата книга. **(5 поени)**

Да се обезбедат сите потребни функции за правилно функционирање на програмата. **(5 поени)**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 4  3  1  0-312-31677-1  The Moscow Vector  Robert Ludlum  7  www.finki-education/olinebooks/book1.pdf  1  2  007-6092006565  Thinking in C++  Bruce Eckel  52  1.2  1  1  978-0672326974  C++ Primer Plus  Stephen Prata  20  www.finki-education/olinebooks/book2.pdf  30 | ====== Testing method mostExpensiveBook() ======  FINKI-Education  Total number of online books: 2  Total number of print books: 1  The most expensive book is:  007-6092006565: Thinking in C++, Bruce Eckel 59.8 | ====== Testing method mostExpensiveBook() ======  FINKI-Education  Total number of online books: 2  Total number of print books: 1  The most expensive book is:  007-6092006565: Thinking in C++, Bruce Eckel 59.8 |  |
|  | 2  978-0672326974  C++ Primer Plus  Stephen Prata  20  www.finki-education/olinebooks/book2.pdf  30  000-0672326974  111-0672326974 | ====== Testing OnlineBook CONSTRUCTORS ======  CONSTRUCTOR  978-0672326974: C++ Primer Plus, Stephen Prata 24  COPY CONSTRUCTOR  978-0672326974: C++ Primer Plus, Stephen Prata 24  000-0672326974: C++ Primer Plus, Stephen Prata 24  OPERATOR =  000-0672326974: C++ Primer Plus, Stephen Prata 24  111-0672326974: C++ Primer Plus, Stephen Prata 24 | ====== Testing OnlineBook CONSTRUCTORS ======  CONSTRUCTOR  978-0672326974: C++ Primer Plus, Stephen Prata 24  COPY CONSTRUCTOR  978-0672326974: C++ Primer Plus, Stephen Prata 24  000-0672326974: C++ Primer Plus, Stephen Prata 24  OPERATOR =  000-0672326974: C++ Primer Plus, Stephen Prata 24  111-0672326974: C++ Primer Plus, Stephen Prata 24 |  |
|  | 3  2  007-6092006565  Thinking in C++  Bruce Eckel  52  1.2  1  978-1118823774  C++ for Dummies  Stephen R. Davis  21  2.2  10 | ====== Testing PrintBook class ======  CONSTRUCTOR  OPERATOR <<  007-6092006565: Thinking in C++, Bruce Eckel 59.8  CONSTRUCTOR  OPERATOR <<  978-1118823774: C++ for Dummies, Stephen R. Davis 24.15  OPERATOR >  Rezultat od sporedbata e:  007-6092006565: Thinking in C++, Bruce Eckel 59.8 | ====== Testing PrintBook class ======  CONSTRUCTOR  OPERATOR <<  007-6092006565: Thinking in C++, Bruce Eckel 59.8  CONSTRUCTOR  OPERATOR <<  978-1118823774: C++ for Dummies, Stephen R. Davis 24.15  OPERATOR >  Rezultat od sporedbata e:  007-6092006565: Thinking in C++, Bruce Eckel 59.8 |  |
|  | 1  2  0-312-31677-1  The Moscow Vector  Robert Ludlum  7  www.finki-education/olinebooks/book1.pdf  1  978-0672326974  C++ Primer Plus  Stephen Prata  20  www.finki-education/olinebooks/book2.pdf  30 | ====== Testing OnlineBook class ======  CONSTRUCTOR  OPERATOR <<  0-312-31677-1: The Moscow Vector, Robert Ludlum 7  CONSTRUCTOR  OPERATOR <<  978-0672326974: C++ Primer Plus, Stephen Prata 24  OPERATOR >  Rezultat od sporedbata e:  978-0672326974: C++ Primer Plus, Stephen Prata 24 | ====== Testing OnlineBook class ======  CONSTRUCTOR  OPERATOR <<  0-312-31677-1: The Moscow Vector, Robert Ludlum 7  CONSTRUCTOR  OPERATOR <<  978-0672326974: C++ Primer Plus, Stephen Prata 24  OPERATOR >  Rezultat od sporedbata e:  978-0672326974: C++ Primer Plus, Stephen Prata 24 |  |

Passed

# 2.

Да се дефинира класа Vozac во која се чуваат информации за:

* име (низа од максимум 100 знаци)
* возраст (цел број)
* број на трки (цел број)
* дали е ветеран (да/не).

Да се преоптоварат:

* операторот << за печатење кој ќе ги печати името, возраст, бројот на трки и дали е ветеран (се печати VETERAN)
* операторот == за споредување на два возачи според заработувачката по трка.

Од оваа класа да се изведат две нови класи Avtomobilist и Motociklist.

За автомобилистот дополнително се чува:

* цена на автомобилот (децимален број).

За мотоциклистот дополнително се чува:

* моќност на мотор (цел број).

Заработувачката по трка на возачите се пресметува како:

* за автомобилостот заработувачката по трка е $\frac{CENA\_{AVTOMOBIL}}{5}$
* за мотоциклистот заработувачката по трка е $MOKJNOST\_NA\_MOTOR \* 20$

Секој автомобилист плаќа данок на заработка. Да се напишат соодветни методи за пресметување данок:

* за автомобилисти: ако бројот на трки е поголем од 10 стапката на данок е 15% од заработката, инаку 10% од заработката.
* за мотоциклисти: ако е ветеран стапката на данок е 25% од заработката, инаку 20% од заработката.

Да се напше надворешна функција soIstaZarabotuvachka која како аргументи прима низа од покажувачи од класата Vocac, нивниот број, како и покажувач кон ојбект од возач и враќа како резултат бројот на возачи кои имаат иста заработувачка по трка со проследениот возач.

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 5  1  Hamilton 30 95 0 55000  Vetel 26 88 1 800  Barrichello 38 198 0 810  Rossi 32 130 1 800  Lorenzo 24 45 0 900  VozacX 38 198 1 800 | === DANOK ===  Hamilton  30  95  1650  Vetel  26  88  VETERAN  4000  Barrichello  38  198  3240  Rossi  32  130  VETERAN  4000  Lorenzo  24  45  3600  === VOZAC X ===  VozacX  38  198  VETERAN  === SO ISTA ZARABOTUVACKA KAKO VOZAC X ===  2 | === DANOK ===  Hamilton  30  95  1650  Vetel  26  88  VETERAN  4000  Barrichello  38  198  3240  Rossi  32  130  VETERAN  4000  Lorenzo  24  45  3600  === VOZAC X ===  VozacX  38  198  VETERAN  === SO ISTA ZARABOTUVACKA KAKO VOZAC X ===  2 |  |
|  | 5  4  Hamilton 30 95 0 55000  Vetel 26 88 1 8000  Barrichello 38 198 0 8100  Rossi 32 130 1 8000  Lorenzo 24 45 0 9000  VozacX 38 198 1 800 | === DANOK ===  Hamilton  30  95  1650  Vetel  26  88  VETERAN  240  Barrichello  38  198  243  Rossi  32  130  VETERAN  240  Lorenzo  24  45  36000  === VOZAC X ===  VozacX  38  198  VETERAN  === SO ISTA ZARABOTUVACKA KAKO VOZAC X ===  0 | === DANOK ===  Hamilton  30  95  1650  Vetel  26  88  VETERAN  240  Barrichello  38  198  243  Rossi  32  130  VETERAN  240  Lorenzo  24  45  36000  === VOZAC X ===  VozacX  38  198  VETERAN  === SO ISTA ZARABOTUVACKA KAKO VOZAC X ===  0 |  |
|  | 5  3  Hamilton 30 95 0 55000  Vetel 26 88 0 14000  Barrichello 38 198 1 30000  Rossi 32 130 1 850  Lorenzo 24 45 0 900  VozacX 38 198 1 850 | === DANOK ===  Hamilton  30  95  1650  Vetel  26  88  420  Barrichello  38  198  VETERAN  900  Rossi  32  130  VETERAN  4250  Lorenzo  24  45  3600  === VOZAC X ===  VozacX  38  198  VETERAN  === SO ISTA ZARABOTUVACKA KAKO VOZAC X ===  1 | === DANOK ===  Hamilton  30  95  1650  Vetel  26  88  420  Barrichello  38  198  VETERAN  900  Rossi  32  130  VETERAN  4250  Lorenzo  24  45  3600  === VOZAC X ===  VozacX  38  198  VETERAN  === SO ISTA ZARABOTUVACKA KAKO VOZAC X ===  1 |  |

Passed

# 3.

Дадена е класа Kurs во која се чуваат информации за име на курс (низа од знаци) и број на кредити (цел број).

Дадена е класа Student што содржи инфомрации за: индекс на студентот (цел број), низа од оценки на студентот (динамички алоцирана низа на оценките кои претставуваат броеви од 5 до 10) и број на оценки.

Дадена е класа Predavach што содржи инфомрации за: име на предавачот (динамички алоцирана низа од знаци), листа од курсеви кои ги предава предавачот (низа од објекти од класата Kurs) и број на курсеви (цел број).

Да се креира класа Demonstrator, со која се претставуваат студентите држат лабораториските вежби на некои курсеви. Објектите од оваа класа треба да содржат инфомрации за: индекс на студентот, оценки на студентот, број на оценки, име на демонстраторот, листа од курсеви, број на курсеви чии лабораторисски вежби ги држи студентот и број на часови во неделата кога студентот држи лабораториски вежби (цел број). **(5 поени)**

За секој студент да се овозможат следните функции:

* getBodovi() - која враќа цел број кој го претставува број на бодови за даден студент. Студентите кои не се демонстратори имаат бодови кои го претставуваат процентот на преодни оценки на студентот. (На пример студент со оценки: 5 6 7 ќе има 66 бодови (цел дел од 66.666...) затоа што во 66% од оценките има оценка поголема од 5 ). Кај секој демонстратор на овие бодовите од оценките се додаваат бодовите од лабораториските вежби: (20\*C)/N, каде N e бројот на курсеви кои ги држи, C бројот на часови во неделата кога студентот држи лабораториски вежби. Во случај кога некој демонстратор не држи ниту еден курс се фрла исклучокот **NoCourseException**. Справување со исклучокот треба да реализира онаму каде што е потребно и притоа да се испечати соодветна порака за грешка "Demonstratorot so indeks XXXX ne drzi laboratoriski vezbi", каде XXXX е индексот на демонстраторот. **(15 поени)**
* pecati()- во која се печати само индексот на студентот ако студентот не е демонстратор, а во случај кога студентот е демонстратор во продолжение се печатат информации за курсевите чии лабораториски вежби ги држи демонстраторот. **(10 поени)**

Форматот за печатење е:

Indeks: ime (Kurs1 Krediti1 ECTS, Kurs2 Krediti2 ECTS,...)

Да се имплементираат следните глобални функции:

* Student& vratiNajdobroRangiran(Student \*\* studenti, int n ) што враќа референца кон студентот кој има најмногу бодови од листата на дадените n студенти (studenti). Да забележиме дека оние демонстратори кои не држат лабораториски вежби ќе земеме дека имаат 0 бодови. Да забележиме и дека во примерите секогаш има точно еден студент кој има најголем број на бодови. **(15 поени)**
* void pecatiDemonstratoriKurs (char\* kurs, Student\*\* studenti, int n) - која од дадена листа на студенти, ќе ги испечати само оние кои држат лабораториски вежби на курсот kurs. **(10 поени)**

Комплетна функционалност на програмата. **(5 поени)**

Да забележиме дека веќе постоечките класи Kurs, Student и Predavach може да се дополнуваат и менуваат. Погледнете ги дадените класи. Во нив покрај конструкторите дадени се и други функциите кои можат да се користат.

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 2  11133 3 10 5 9 | -----TEST pecati-----  11133 | -----TEST pecati-----  11133 |  |
|  | 4  11020 5 5 6 7 5 8  A.Angelovski 2 OOP 6 Angliski 2  10 | -----TEST getVkupnaOcenka-----  Broj na bodovi: 160 | -----TEST getVkupnaOcenka-----  Broj na bodovi: 160 |  |
|  | 7  4  1 11111 3 10 5 8  2 11444 4 5 6 6 6 P.Petkovski 0 0  1 22222 3 10 9 8  2 33333 5 5 6 5 5 5 A.Angelovski 5 OOP 6 Angliski 2 Kalkulus 6 DMatematika 6 Infromatika 3 10 | -----TEST vratiNajdobroRangiran-----  Demonstratorot so indeks 11444 ne drzi laboratoriski vezbi  Maksimalniot broj na bodovi e:100  Najdobro rangiran:22222 | -----TEST vratiNajdobroRangiran-----  Demonstratorot so indeks 11444 ne drzi laboratoriski vezbi  Maksimalniot broj na bodovi e:100  Najdobro rangiran:22222 |  |
|  | 3  11133 3 10 5 9 | -----TEST getVkupnaOcenka-----  Broj na bodovi: 66 | -----TEST getVkupnaOcenka-----  Broj na bodovi: 66 |  |
|  | 6  11020 5 10 6 7 10 10  A.Angelovski 4 OOP 6 Angliski 2 Kalkulus 6 Dmatematika 6  12 | -----TEST Student i Demonstrator-----  11020: A.Angelovski (OOP 6ECTS, Angliski 2ECTS, Kalkulus 6ECTS, Dmatematika 6ECTS)  Broj na bodovi: 160 | -----TEST Student i Demonstrator-----  11020: A.Angelovski (OOP 6ECTS, Angliski 2ECTS, Kalkulus 6ECTS, Dmatematika 6ECTS)  Broj na bodovi: 160 |  |
|  | 8  5  2 55555 5 5 6 9 7 10 B.Boskovska 3 OOP 6 Angliski 2 DMatematika 6 10  2 44444 3 5 7 9 V.Velkovski 3 Informatika 3 OOP 6 Angliski 2 12  1 11111 3 10 5 8  1 22222 3 10 9 8  2 33333 5 5 6 10 10 10 A.Angelovski 4 Angliski 2 Kalkulus 6 DMatematika 6 Infromatika 3 10  OOP | -----TEST pecatiDemonstratoriKurs-----  Demonstratori na OOP se:  55555: B.Boskovska (OOP 6ECTS, Angliski 2ECTS, DMatematika 6ECTS)  44444: V.Velkovski (Informatika 3ECTS, OOP 6ECTS, Angliski 2ECTS) | -----TEST pecatiDemonstratoriKurs-----  Demonstratori na OOP se:  55555: B.Boskovska (OOP 6ECTS, Angliski 2ECTS, DMatematika 6ECTS)  44444: V.Velkovski (Informatika 3ECTS, OOP 6ECTS, Angliski 2ECTS) |  |
|  | 4  11020 5 5 6 5 5 5  A.Angelovski 5 OOP 6 Angliski 2 Kalkulus 6 DMatematika 6 Infromatika 3  10 | -----TEST getVkupnaOcenka-----  Broj na bodovi: 60 | -----TEST getVkupnaOcenka-----  Broj na bodovi: 60 |  |
|  | 5  11020 5 10 6 7 5 8  A.Angelovski 5 OOP 6 Angliski 2 Kalkulus 6 DMatematika 6 Infromatika 3  10 | -----TEST pecati -----  11020: A.Angelovski (OOP 6ECTS, Angliski 2ECTS, Kalkulus 6ECTS, DMatematika 6ECTS, Infromatika 3ECTS) | -----TEST pecati -----  11020: A.Angelovski (OOP 6ECTS, Angliski 2ECTS, Kalkulus 6ECTS, DMatematika 6ECTS, Infromatika 3ECTS) |  |
|  | 1  11020 5 5 6 7 5 8  A.Angelovski 2 OOP 6 Angliski 2  10 | -----TEST Demonstrator-----  Objekt od klasata Demonstrator e kreiran | -----TEST Demonstrator-----  Objekt od klasata Demonstrator e kreiran |  |
|  | 7  3  1 11111 3 10 5 8  1 22222 3 10 9 8  2 33333 5 5 6 10 10 10  A.Angelovski 5 OOP 6 Angliski 2 Kalkulus 6 DMatematika 6 Infromatika 3  10 | -----TEST vratiNajdobroRangiran-----  Maksimalniot broj na bodovi e:120  Najdobro rangiran:33333: A.Angelovski (OOP 6ECTS, Angliski 2ECTS, Kalkulus 6ECTS, DMatematika 6ECTS, Infromatika 3ECTS) | -----TEST vratiNajdobroRangiran-----  Maksimalniot broj na bodovi e:120  Najdobro rangiran:33333: A.Angelovski (OOP 6ECTS, Angliski 2ECTS, Kalkulus 6ECTS, DMatematika 6ECTS, Infromatika 3ECTS) |  |

Passed

# 4.

Да се креира апстрактна класа Pizza за опишување пици. **(5 поени)** За секоја пица се чуваат следните информации:

* име (низа од максимум 20 знаци)
* состојки (низа од максимум 100 знаци)
* основна цена (реален број)

Од оваа класа да се изведат класите FlatPizza и FoldedPizza за опишување на рамни и преклопени пици соодветно **(5 поени)**.

За секоја рамна пица дополнително се чува големина (enum - една од три можности: мала, голема и фамилијарна). За секоја преклопена пица дополнително се чува информација дали тестото е од бело брашно или не (boolean).

За секоја пица треба да се обезбеди метод за пресметување на нејзината продажна цена:

* цената на рамната пица се пресметува така што основната цена се зголемува за 10%, 20% и 30% за мала, голема и фамилијарна пица соодветно.
* цената на преклопената пица се пресметува така што основната цена се зголемува за 10% ако е тестото е од бело брашно, а во спротивно за 30%. **(10 поени)**

Да се преоптоварат следните оператори:

* оператор << - за печатење сите податоци за пиците во следниов формат:
* За рамна пица:[име]: [состојки], [големина] - [продажната цена на пицата].
* За преклопена пица: [име]: [состојки], [wf - ако е со бело брашно / nwf - ако не е со бело брашно] - [продажната цена на пицата]

**(5 поени)**

* оператор < - за споредување на пиците од каков било вид според нивната продажна цена. **(5 поени)**

Да се дефинира глобална функција еxpensivePizza што на влез прима низа од покажувачи кон објекти од класата Pizza и нивниот број, а како резултат ги печати информациите за пицата со највисока продажна цена. Ако има повеќе пици со иста највисока цена, се печати првата. **(10 поени)**

Да се дефинираат потребните конструктори и методи во класите за правилно функционирање на програмата. **(5 поени)**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 5  Margarita  Tomato sauce, cheese, mozzarella, basil, oregano  100  2  Margarita  Tomato sauce, cheese, mozzarella, basil, oregano  100  0  Capricciosa calzone  Tomato sauce, cheese, ham, fresh mushrooms  140  Capricciosa calzone  Tomato sauce, cheese, ham, fresh mushrooms  140 | Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, family - 130  Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, small - 110  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, wf - 154  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, nwf - 182  Lower price:  110  130  110  154 | Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, family - 130  Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, small - 110  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, wf - 154  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, nwf - 182  Lower price:  110  130  110  154 |  |
|  | 6  6  1  Peperoni  Tomato sauce, cheese, kulen sausage, oregano  250  0  1  Capricciosa  tomato sauce, cheese, ham, fresh mushrooms, oregano  250  0  1  Prosciutto  tomato sauce, cheese, prosciutto, oregano  310  0  2  Capricciosa calzone  Tomato sauce, cheese, ham, fresh mushrooms  140  1  Veggie  tomato sauce, cheese, tomatoes, peppers, onion, olives, fresh mushrooms, oregano  270  0  1  Caprese  tomato sauce, cheese, mozzarella, tomatoes, pesto, garlic, oregano  310  0 | Peperoni: Tomato sauce, cheese, kulen sausage, oregano, small - 275  Capricciosa: tomato sauce, cheese, ham, fresh mushrooms, oregano, small - 275  Prosciutto: tomato sauce, cheese, prosciutto, oregano, small - 341  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, nwf - 182  Veggie: tomato sauce, cheese, tomatoes, peppers, onion, olives, fresh mushrooms, oregano, small - 297  Caprese: tomato sauce, cheese, mozzarella, tomatoes, pesto, garlic, oregano, small - 341  The most expensive pizza:  Prosciutto: tomato sauce, cheese, prosciutto, oregano, small - 341 | Peperoni: Tomato sauce, cheese, kulen sausage, oregano, small - 275  Capricciosa: tomato sauce, cheese, ham, fresh mushrooms, oregano, small - 275  Prosciutto: tomato sauce, cheese, prosciutto, oregano, small - 341  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, nwf - 182  Veggie: tomato sauce, cheese, tomatoes, peppers, onion, olives, fresh mushrooms, oregano, small - 297  Caprese: tomato sauce, cheese, mozzarella, tomatoes, pesto, garlic, oregano, small - 341  The most expensive pizza:  Prosciutto: tomato sauce, cheese, prosciutto, oregano, small - 341 |  |
|  | 3  Capricciosa calzone  Tomato sauce, cheese, ham, fresh mushrooms  140 | Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, wf - 154 | Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, wf - 154 |  |
|  | 6  4  1  Margarita  Tomato sauce, cheese, mozzarella, basil, oregano  100  2  1  Margarita  Tomato sauce, cheese, mozzarella, basil, oregano  100  0  2  Capricciosa calzone  Tomato sauce, cheese, ham, fresh mushrooms  140  2  Capricciosa calzone  Tomato sauce, cheese, ham, fresh mushrooms  140 | Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, family - 130  Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, small - 110  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, wf - 154  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, nwf - 182  The most expensive pizza:  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, nwf - 182 | Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, family - 130  Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, small - 110  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, wf - 154  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, nwf - 182  The most expensive pizza:  Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, nwf - 182 |  |
|  | 2  Margarita  Tomato sauce, cheese, mozzarella, basil, oregano  100  2 | Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, family - 130 | Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, family - 130 |  |
|  | 1  Margarita  Tomato sauce, cheese, mozzarella, basil, oregano  100 | Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, small - 110 | Margarita: Tomato sauce, cheese, mozzarella, basil, oregano, small - 110 |  |
|  | 4  Capricciosa calzone  Tomato sauce, cheese, ham, fresh mushrooms  140 | Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, nwf - 182 | Capricciosa calzone: Tomato sauce, cheese, ham, fresh mushrooms, nwf - 182 |  |

Passed

# 5.

Да се креира класа Customer за опишување на купувачи на една книжара. За секој купувач се чуваат податоци за:

* името (низа од 50 знаци),
* електронска адреса (низа од 50 знаци),
* вид на купувач (стандардни, лојални или вип),
* основен попуст (цел број изразен во проценти),
* дополнителен попуст (цел број изразен во проценти) и
* број на купени производи. **(5 поени)**

Сите лојални купувачи со покажување на клуб картичка имаат право на основниот попуст при купување од книжарата. Основниот попуст е **ист** за сите лојални купувачи и изнесува 10%. Оваа вредност **може да се смени** со одлука на раководството на книжарата. Дополнителниот попуст е **фиксен** и може да го користат само вип купувачите и изнесува 20%. Стандардните купувачи немаат право на попуст **(5 поени)**.

За оваа класа да се имплементира оператор << за печатење на купувач во формат:

[ime\_na\_kupuvac]

[email\_na\_kupuvac]

[broj\_na\_proizvodi]

[vid\_na\_kupuvac] [popust\_sto\_moze\_da\_go\_koristi]

каде попустот е 0% ако купувачот е стандарден, вредноста на основниот попуст, ако е лојален или збир од основниот и дополнителниот попуст, ако купувачот е вип. **(5 поени)**

Да се креира класа за онлајн книжара FINKI-bookstore во која се чува низа од регистрирани купувачи (динамички алоцирана низа) и број на купувачи. **(5 поени)** За класата да се обезбедат:

* operator+= **(10 поени)** за додавање купувач во листата од купувачи, но само ако веќе не е дел од неа (ако во листата нема купувач со иста електронска адреса). Ако во листата постои корисник со иста електронска адреса, треба да се генерира исклучок UserExistsException. Потребно е да се обезбеди справување со исклучокот во функцијата main на означеното место. Во ваква ситуација се печати порака "The user already exists in the list" **(5 поени)**.
* Функција update со која сите стандардни купувачи со купени повеќе од 5 производи стануваат лојални, а сите лојални купувачи со над 10 купени производи, стануваат вип **(5 поени)**.
* Оператор << за печатење на информациите за сите регистрирани купувачи **(5 поени)**.

Да се обезбедат сите дополнителни методи потребни за правилно функционирање на програмата. **(10 поени)**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 6  5  Albert Einstein  einstein@finki.mk  1  2  Isaac Newton  newton@finki.mk  2  10  Galileo Galilei  galilei@finki.mk  1  12  Stephen Hawking  hawking@finki.mk  0  17  Nikola Tesla  tesla@finki.mk  0  7 | ===== Test Case - update method ======  Update:  Albert Einstein  einstein@finki.mk  2  loyal 10  Isaac Newton  newton@finki.mk  10  vip 30  Galileo Galilei  galilei@finki.mk  12  vip 30  Stephen Hawking  hawking@finki.mk  17  loyal 10  Nikola Tesla  tesla@finki.mk  7  loyal 10 | ===== Test Case - update method ======  Update:  Albert Einstein  einstein@finki.mk  2  loyal 10  Isaac Newton  newton@finki.mk  10  vip 30  Galileo Galilei  galilei@finki.mk  12  vip 30  Stephen Hawking  hawking@finki.mk  17  loyal 10  Nikola Tesla  tesla@finki.mk  7  loyal 10 |  |
|  | 3  3  Albert Einstein  einstein@finki.mk  1  2  Isaac Newton  newton@finki.mk  2  10  Galileo Galilei  galilei@finki.mk  0  10 | ===== Test Case - FINKI-bookstore ======  Albert Einstein  einstein@finki.mk  2  loyal 10  Isaac Newton  newton@finki.mk  10  vip 30  Galileo Galilei  galilei@finki.mk  10  standard 0 | ===== Test Case - FINKI-bookstore ======  Albert Einstein  einstein@finki.mk  2  loyal 10  Isaac Newton  newton@finki.mk  10  vip 30  Galileo Galilei  galilei@finki.mk  10  standard 0 |  |
|  | 4  3  Albert Einstein  einstein@finki.mk  1  2  Isaac Newton  newton@finki.mk  2  10  Galileo Galilei  galilei@finki.mk  0  10  Stephen Hawking  hawking@finki.mk  2  17 | ===== Test Case - operator+= ======  OPERATOR +=  Albert Einstein  einstein@finki.mk  2  loyal 10  Isaac Newton  newton@finki.mk  10  vip 30  Galileo Galilei  galilei@finki.mk  10  standard 0  Stephen Hawking  hawking@finki.mk  17  vip 30 | ===== Test Case - operator+= ======  OPERATOR +=  Albert Einstein  einstein@finki.mk  2  loyal 10  Isaac Newton  newton@finki.mk  10  vip 30  Galileo Galilei  galilei@finki.mk  10  standard 0  Stephen Hawking  hawking@finki.mk  17  vip 30 |  |
|  | 2  Albert Einstein  einstein@finki.mk  1  2 | ===== Test Case - Static Members ======  ===== CONSTRUCTOR ======  Albert Einstein  einstein@finki.mk  2  loyal 10  Albert Einstein  einstein@finki.mk  2  loyal 5 | ===== Test Case - Static Members ======  ===== CONSTRUCTOR ======  Albert Einstein  einstein@finki.mk  2  loyal 10  Albert Einstein  einstein@finki.mk  2  loyal 5 |  |
|  | 1  Albert Einstein  einstein@finki.mk  1  2 | ===== Test Case - Customer Class ======  ===== CONSTRUCTOR ======  Albert Einstein  einstein@finki.mk  2  loyal 10 | ===== Test Case - Customer Class ======  ===== CONSTRUCTOR ======  Albert Einstein  einstein@finki.mk  2  loyal 10 |  |
|  | 5  4  Albert Einstein  einstein@finki.mk  1  2  Isaac Newton  newton@finki.mk  2  10  Galileo Galilei  galilei@finki.mk  0  10  Stephen Hawking  hawking@finki.mk  2  17  Al Ein  einstein@finki.mk  0  3 | ===== Test Case - operator+= (exception) ======  OPERATOR +=  The user already exists in the list!  Albert Einstein  einstein@finki.mk  2  loyal 10  Isaac Newton  newton@finki.mk  10  vip 30  Galileo Galilei  galilei@finki.mk  10  standard 0  Stephen Hawking  hawking@finki.mk  17  vip 30 | ===== Test Case - operator+= (exception) ======  OPERATOR +=  The user already exists in the list!  Albert Einstein  einstein@finki.mk  2  loyal 10  Isaac Newton  newton@finki.mk  10  vip 30  Galileo Galilei  galilei@finki.mk  10  standard 0  Stephen Hawking  hawking@finki.mk  17  vip 30 |  |

Passed

# 6.

Потребно е да се имплементира класа за компјутерска игра (Game), што содржи информации за:

* име на играта (низа од макс. 100 знаци)
* цена на играта (децимален број)
* дали е играта купена на распродажба (bool променлива).

Од класата Game да се изведе класа SubscriptionGame, што дополнително ќе чува:

* месечен надоместок за играње (децимален број).
* датум кога играта е купена (месец и година како позитивни цели броеви)

За класите Game и SubscriptionGame да се преоптоварат операторите за печатење (<<) и читање (>>). Да се дефинира и операторот == кој ќе споредува игри според нивното име.

Да се дефинира класа за корисник (User) во која се чуваат:

* кориснично име на корисникот (низа од макс. 100 знаци)
* колекција од игри кои се купени од корисникот (динамички алоцирана низа).

Да се преоптовари операторот += кој ќе овозможи додавање на нова игра во колекцијата на игри. Притоа ако корисникот ја има веќе купено играта, потребно е да се креира исклучок од типот ExistingGame. Класата за имплементација на исклучоци потребно е има соодветен конструктор и метода message за печатење на порака на екран.

Да се креира и метода total\_spent() во класата User која ќе пресметува колку пари корисникот потрошил за својата колекција од игри. Доколку играта е купена на распродажба, цената на играта е 30% од стандарната цена. Доколку играта е од типот SubscriptionGame, потребно е да се вкалкулира и сумата потрошена за месечниот надоместок (број\_на\_изминати\_месеци x цена\_на\_месечен\_надоместок) без да се земе во предвид моменталниот месец (мај 2018).

Да се преоптовари и оператоторот за печатење на корисникот, која печати информации во сл. формат (види тест примери 7 до 12):

User: username

- Game: PUBG, regular price: $70, bought on sale

- Game: Half Life 2, regular price: $70 - Game: Warcraft 4, regular price: $40, monthly fee: $10, purchased: 1-2017

Листа на дел од методите со нивни прототипови кои се користат во main:

ЕxistingGame::message()

Game::operator==(Game&)

User::operator+=(Game&)

User::get\_game(int)

User::total\_spent(int)

User::get\_name()

User::get\_games\_number()

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 4  World of Warcraft  40  0  10  1  2017 | Testing operator>> for SubscriptionGame  Game: World of Warcraft, regular price: $40, monthly fee: $10, purchased: 1-2017 | Testing operator>> for SubscriptionGame  Game: World of Warcraft, regular price: $40, monthly fee: $10, purchased: 1-2017 |  |
|  | 1  Sims  50  0 | Testing class Game and operator<< for Game  Game: Sims, regular price: $50 | Testing class Game and operator<< for Game  Game: Sims, regular price: $50 |  |
|  | 1  Half Life  70  1 | Testing class Game and operator<< for Game  Game: Half Life, regular price: $70, bought on sale | Testing class Game and operator<< for Game  Game: Half Life, regular price: $70, bought on sale |  |
|  | 3  Half Life 2  70  1 | Testing operator>> for Game  Game: Half Life 2, regular price: $70, bought on sale | Testing operator>> for Game  Game: Half Life 2, regular price: $70, bought on sale |  |
|  | 2  PUBG  100  1  5  5  2018 | Testing class SubscriptionGame and operator<< for SubscritionGame  Game: PUBG, regular price: $100, bought on sale, monthly fee: $5, purchased: 5-2018 | Testing class SubscriptionGame and operator<< for SubscritionGame  Game: PUBG, regular price: $100, bought on sale, monthly fee: $5, purchased: 5-2018 |  |
|  | 7  BugsBunny  2  1  Half Life 2  70  0  2  Warcraft 4  40  0  10  1  2017 | Testing total\_spent method() for User  User: BugsBunny  - Game: Half Life 2, regular price: $70  - Game: Warcraft 4, regular price: $40, monthly fee: $10, purchased: 1-2017  Total money spent: $270 | Testing total\_spent method() for User  User: BugsBunny  - Game: Half Life 2, regular price: $70  - Game: Warcraft 4, regular price: $40, monthly fee: $10, purchased: 1-2017  Total money spent: $270 |  |
|  | 5  BugsBunny  1  1  Half Life 2  70  0 | Testing class User and operator+= for User  User: BugsBunny  - Game: Half Life 2, regular price: $70 | Testing class User and operator+= for User  User: BugsBunny  - Game: Half Life 2, regular price: $70 |  |
|  | 6  BugsBunny  3  1  Half Life 2  70  0  2  Warcraft 4  40  0  10  1  2017  1  Half Life 2  70  1 | Testing exception ExistingGame for User  The game is already in the collection  User: BugsBunny  - Game: Half Life 2, regular price: $70  - Game: Warcraft 4, regular price: $40, monthly fee: $10, purchased: 1-2017 | Testing exception ExistingGame for User  The game is already in the collection  User: BugsBunny  - Game: Half Life 2, regular price: $70  - Game: Warcraft 4, regular price: $40, monthly fee: $10, purchased: 1-2017 |  |
|  | 5  BugsBunny  2  1  Half Life 2  70  0  2  World od Warcraft  40  0  10  1  2017 | Testing class User and operator+= for User  User: BugsBunny  - Game: Half Life 2, regular price: $70  - Game: World od Warcraft, regular price: $40, monthly fee: $10, purchased: 1-2017 | Testing class User and operator+= for User  User: BugsBunny  - Game: Half Life 2, regular price: $70  - Game: World od Warcraft, regular price: $40, monthly fee: $10, purchased: 1-2017 |  |
|  | 7  DaffyDuck  1  2  Warcraft 4  40  0  10  4  2018 | Testing total\_spent method() for User  User: DaffyDuck  - Game: Warcraft 4, regular price: $40, monthly fee: $10, purchased: 4-2018  Total money spent: $50 | Testing total\_spent method() for User  User: DaffyDuck  - Game: Warcraft 4, regular price: $40, monthly fee: $10, purchased: 4-2018  Total money spent: $50 |  |
|  | 6  BugsBunny  2  1  Half Life 2  70  1  2  Warcraft 4  40  0  10  1  2017 | Testing exception ExistingGame for User  User: BugsBunny  - Game: Half Life 2, regular price: $70, bought on sale  - Game: Warcraft 4, regular price: $40, monthly fee: $10, purchased: 1-2017 | Testing exception ExistingGame for User  User: BugsBunny  - Game: Half Life 2, regular price: $70, bought on sale  - Game: Warcraft 4, regular price: $40, monthly fee: $10, purchased: 1-2017 |  |
|  | 2  World of Warcraft  40  0  10  1  2018 | Testing class SubscriptionGame and operator<< for SubscritionGame  Game: World of Warcraft, regular price: $40, monthly fee: $10, purchased: 1-2018 | Testing class SubscriptionGame and operator<< for SubscritionGame  Game: World of Warcraft, regular price: $40, monthly fee: $10, purchased: 1-2018 |  |

Passed

# 7.

Дадена е дел од дефиницијата на класата Karticka. За секоја картичка се чуваат информации за **трансакциска сметка** (низа од 15 знаци) , **пин код** (цел број) и дали има **повеќе пин кодови** (булова променлива).

За секој картичка може да се пресмета тежината за пробивање на картичката. Тежината на пробивање е бројот на цифрите на пин кодот.

Специјалните картички покрај еден задолжителен имаат уште неколку пин кодови . За овие корисници да се моделира соодветна класа SpecijalnaKaricka во која покрај **дополнителните пин кодови** (динамичко алоцирана низа од цели броеви) се чува и **бројот на пин кодовите** (цел број). Со дополнителните пин кодови се менува и тежината на пробивање на картичката и таа се зголемува за бројот на дополнителни пин кодови.

Максималниот број на дополнителни пин кодови кој може да има било која картичка е P. P има фиксна вредност 4 за сите картички и оваа вредност не може да се промени.

Во класите Karticka и SpecijalnaKarticka треба да се имплементираат потребните конструктори, функции за поставување и преземање и деструктор. **(5 поени)**

За сите објекти од класите треба да се обезбедат следните функции:

* **преоптоварен оператор <<** во кој се печатат информации за картичката во формат: smetka: tezina **(5 поени)**
* tezinaProbivanje() – ја пресметува тежината за пробивање на картичката **(5 поени)**

Во класата SpecijalnaKarticka дефинирај ја функцијата:

* **преоптоварен оператор +=** за додавање на нов пин код **(5 поени)**

Ако се направи обид да се внесат повеќе од дозволениот број на пин кодови во безбедносниот код да се фрли исклучок (објект од класата OutOfBoundException). Фатете го исклучокот во главната функција каде што е потребно. Откако ќе го фатите отпечатете соодветна порака за грешка (*Brojot na pin kodovi ne moze da go nadmine dozvolenoto*) и справете се со исклучокот така да додадениот пин код не се зема предвид **(10 поени)**

Дадена е дел од дефиницијата на класата Banka во која се чуваат информаци за името на банката (низа од 30 знаци) и за картичките издадени од банката (низа од 20 покажувачи кон класата Karticka) како и бројот на такви картички.

Велиме дека картичката издадена од банката може да се пробие ако тежината за пробивање е најмногуLIMIT. Вредноста LIMIT е членка на класата Banka, има почетна вредност 7 и истата може да се промени со функцијата setLimit(). За сите банки оваа вредност е иста. **(5 поени)**

Во класата Banka имплементирај ги функциите:

* функција pecatiKarticki() во која се печатат сите картички кои можат да се пробијат, секој во посебен ред. Претходно во првиот ред се печати: *Vo bankata XXXXX moze da se probijat kartickite:* **(5 поени)**.
* функција dodadiDopolnitelenPin(char \* smetka, int novPin) во која на картичката со дадена трансакциона сметка се додава нов дополнителен пин код. Ако тоа не е можно функцијата не прави ништо **(15 поени)**.

Комплетна функционалност **(5 поени)**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 2  ab1232432 345 0  bh4555432 876 1  4  ab1232432 2  ab1112432 100  ab1232432 56  ab1211111 88 | Vo bankata Komercijalna moze da se probijat kartickite:  ab1232432: 3  bh4555432: 3 | Vo bankata Komercijalna moze da se probijat kartickite:  ab1232432: 3  bh4555432: 3 |  |
|  | 2  ab1232432 345 1  bh4555432 876 1  4  ab1232432 2  ab1112432 100  ab1232432 56  ab1211111 88 | Vo bankata Komercijalna moze da se probijat kartickite:  ab1232432: 5  bh4555432: 3 | Vo bankata Komercijalna moze da se probijat kartickite:  ab1232432: 5  bh4555432: 3 |  |
|  | 2  ab1232432 345 0  bh4555432 876576 0  0 | Vo bankata Komercijalna moze da se probijat kartickite:  ab1232432: 3 | Vo bankata Komercijalna moze da se probijat kartickite:  ab1232432: 3 |  |
|  | 2  ab1232432 345 0  bh4555432 8766 0  0 | Vo bankata Komercijalna moze da se probijat kartickite:  ab1232432: 3  bh4555432: 4 | Vo bankata Komercijalna moze da se probijat kartickite:  ab1232432: 3  bh4555432: 4 |  |
|  | 2  ab1232432 345 1  bh4555432 8766 1  2  ab1232432 2  ab1232432 34 | Vo bankata Komercijalna moze da se probijat kartickite:  ab1232432: 5  bh4555432: 4 | Vo bankata Komercijalna moze da se probijat kartickite:  ab1232432: 5  bh4555432: 4 |  |
|  | 2  ab1232432 345 1  bh4555432 8766 1  4  ab1232432 2  ab1232432 34  ab1232432 56  ab1232432 67 | Vo bankata Komercijalna moze da se probijat kartickite:  bh4555432: 4 | Vo bankata Komercijalna moze da se probijat kartickite:  bh4555432: 4 |  |

Passed

# 8.

Да се креира класа за опишување на концерти. За еден концерт се чуваат информации за (5 поени):

● назив (низа од најмногу 20 знаци)

● локација (низа од најмногу 20 знаци)

● сезонски попуст (реален број)

● цена на билет во денари (децимален број)

Сите податоци во класата треба да се приватни. Сезонскиот попуст е ист за сите концерти и може да се менува од страна на менаџерите во зависност од сезоната. За таа цел да се обезбеди функција за менување на сезонскиот попуст. Сезонскиот попуст е изразен во проценти и почетната вредност е 20 проценти. (5 поени)

Основната цена на еден билет за концерт се пресметува со методот:

● cena() – која ја враќа цената со сезонскиот попуст (5 поени)

За потребите на еден летен фестивал, треба да се обезбедат посебен вид на забавни концерти, електронски концерти. За секој eлектронски концерт дополнително се чуваат инфромации за името на DJ-от(динамички алоцирана низа од знаци), времетраење на настапот во часови (реален број) и логичка променлива за дали се работи за дневна или ноќна забава (дневна-true/ноќна-false). (5 поени)

За секој електронски концерт треба да биде на располагање метод за пресметување на цената за билет (5 поени):

● cena() - основната цена се зголемува во зависност од времетраењето на електронскиот концерт. Ако времетраењето на концертот е подолго од 5 часа, основната цена се зголемува за 150 денари. Ако времетраењето е подолго од 7 часа, цената се зголемува за 360 денари. Доколку се работи за дневна забава, цената се намалува за 50, а доколку се работи за ноќна забава цената се зголемува за 100 денари.

Да се имплементираат следните функции:

· void najskapKoncert(Koncert \*\* koncerti, int n) – во која ќе се испечати називот и цената на најскапиот концерт во низата. Дополнително, да се испечати и колку од концертите се електронски, а колку не во формат:

[Електронски концерти: X од вкупно Y.] (10 поени)

· bool prebarajKoncert(Koncert \*\* koncerti, int n, char \* naziv, bool elektronski) – во која ќе се прабарува концерт кој имаат ист назив како променливата naziv. Доколку променливата elektronski е true, треба да се пребарува само низ електронските концерти, во спротивно се пребаруваат сите концерти. Доколку е пронајден концертот, се печати неговиот назив и цена и функцијата враќа true. Доколку не е пронајден концертот, функцијата враќа false. Можно е да има повеќе концерти со ист назив. (10 поени)

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 6 | Najskap koncert: SeaDance 7230  Elektronski koncerti: 3 od vkupno 5 | Najskap koncert: SeaDance 7230  Elektronski koncerti: 3 od vkupno 5 |  |
|  | 3  TomorrowLand  Belgium  8432.2  Axwell  5.6  0 | Kreiran e elektronski koncert so naziv TomorrowLand i sezonskiPopust 0.2 | Kreiran e elektronski koncert so naziv TomorrowLand i sezonskiPopust 0.2 |  |
|  | 4  TomorrowLand  Belgium  8432.2  Axwell  5.6  0 | Cenata na elektronskiot koncert so naziv TomorrowLand e: 6995.76 | Cenata na elektronskiot koncert so naziv TomorrowLand e: 6995.76 |  |
|  | 2  MkcKoncert  Skopje  333.3 | Osnovna cena na koncertot so naziv MkcKoncert e: 266.64 | Osnovna cena na koncertot so naziv MkcKoncert e: 266.64 |  |
|  | 7  1 | Ne e pronajden  Area 280  Pronajden | Ne e pronajden  Area 280  Pronajden |  |
|  | 8 | Najskap koncert: TomorrowLand 1260  Elektronski koncerti: 2 od vkupno 4 | Najskap koncert: TomorrowLand 1260  Elektronski koncerti: 2 od vkupno 4 |  |
|  | 1  Axwell  Skopje  450 | Kreiran e koncert so naziv: Axwell | Kreiran e koncert so naziv: Axwell |  |

Passed

# 9.

Да се имплементира класа Trud во која се чуваат информации за: **(5 поени)**

* вид на труд (еден знак и тоа C за конференциски труд, J за труд во списание)
* година на издавање (цел број).

Да се имплементира класа Student во која се чува: **(5 поени)**

* името на студентот (низа од најмногу 30 карактери)
* индекс (цел број)
* година на упис (цел број)
* листа на оцени од положени предмети (низа од цели броеви)
* број на положени предмети (цел број)

За оваа класа да се имплементираат следните методи:

* функција rang() што пресметува просек од положените испити на студентот **(5 поени)**
* оператор << за печатење на студентот во формат: **(5 поени)**

Индекс Име Година на упис ранг

Да се имплементира класа PhDStudent во која покрај основните информации за студентот дополнително се чува: **(5 поени)**

* листа од објавени трудови (динамички резервирана низа од објекти од класата Trud)
* бројот на трудови (цел број)

Во оваа класа да се препокрие соодветно функцијата rang() така што на просекот од положените испити ќе се додаде и збирот од поените од објавените трудови на PhD студентот. Во зависност од видот на трудот, секој универзитет има посебен начин на бодување на трудовите. Начинот на бодување е ист за сите PhD студенти. Иницијално да се смета дека конференциски труд се бодува со 1 поен, а труд во списание со 3 поени. Универзитетот има можност да ги менува вредностите на бодовите. **(5 поени + 5 поени)**

За оваа класа да се обезбеди:

* оператор += за додавање нов објект од класата Trud во листата **(5 поени)**. Ако се направи обид да се внесе труд што е издаден порано од годината на упис на студентот да се фрли исклучок (објект од класата Exception). Справувањето со исклучокот треба да се реализира во главната функција main каде што е потребно, но и во конструктор ако е потребно. Ако бил генериран исклучок треба да се отпечати соодветна порака за грешка **"Ne moze da se vnese dadeniot trud"**, а новиот труд нема да се внесе во листата на трудови од студентот. **(10 поени)**

**Напомена:** Сите променливи на класите се чуваат како приватни.

Да се обезбедат сите потребни функции за правилно функционирање на програмата. **(5 поени)**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 2  3  1  PhDStudent\_1  111222  2012  5  10 10 10 9 10  1  j  2015  1  PhDStudent\_2  123456  2003  5  10 9 10 8 10  3  c  2003  j  2003  j  2005  0  Student  12234  2010  3  10 10 10  123456  C  2004 | ===== Testiranje na operatorot += ======  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 17.4  12234 Student 2010 10 | ===== Testiranje na operatorot += ======  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 17.4  12234 Student 2010 10 |  |
|  | 4  PhDStudent  123456  2003  5  10 9 10 8 10  3  c  2003  j  2001  J  2005 | ===== Testiranje na isklucoci ======  Ne moze da se vnese dadeniot trud  123456 PhDStudent 2003 13.4 | ===== Testiranje na isklucoci ======  Ne moze da se vnese dadeniot trud  123456 PhDStudent 2003 13.4 |  |
|  | 6  3  1  PhDStudent\_1  111222  2012  5  10 10 10 9 10  1  j  2015  1  PhDStudent\_2  123456  2003  5  10 9 10 8 10  3  c  2003  j  2003  j  2005  0  Student  12234  2010  3  10 10 10  2  5 | ===== Testiranje na static clenovi ======  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10  Lista na site studenti:  111222 PhDStudent\_1 2012 14.8  123456 PhDStudent\_2 2003 21.4  12234 Student 2010 10 | ===== Testiranje na static clenovi ======  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10  Lista na site studenti:  111222 PhDStudent\_1 2012 14.8  123456 PhDStudent\_2 2003 21.4  12234 Student 2010 10 |  |
|  | 1  Student  12234  2010  3  10 10 10  PhDStudent  123456  2003  5  10 9 10 8 10  3  c  2003  j  2008  j  2005 | ===== Testiranje na klasite ======  12234 Student 2010 10  123456 PhDStudent 2003 16.4 | ===== Testiranje na klasite ======  12234 Student 2010 10  123456 PhDStudent 2003 16.4 |  |
|  | 2  3  1  PhDStudent\_1  111222  2012  5  10 10 10 9 10  1  j  2015  1  PhDStudent\_2  123456  2003  5  10 9 10 8 10  3  c  2003  j  2003  j  2005  0  Student  12234  2010  3  10 10 10  12234  C  2004 | ===== Testiranje na operatorot += ======  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10  Ne postoi PhD student so indeks 12234  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10 | ===== Testiranje na operatorot += ======  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10  Ne postoi PhD student so indeks 12234  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10 |  |
|  | 5  3  1  PhDStudent\_1  111222  2012  5  10 10 10 9 10  1  j  2015  1  PhDStudent\_2  123456  2003  5  10 9 10 8 10  3  c  2003  j  2003  j  2005  0  Student  12234  2010  3  10 10 10  111222  C  2010 | ===== Testiranje na isklucoci ======  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10  Ne moze da se vnese dadeniot trud  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10 | ===== Testiranje na isklucoci ======  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10  Ne moze da se vnese dadeniot trud  Lista na site studenti:  111222 PhDStudent\_1 2012 12.8  123456 PhDStudent\_2 2003 16.4  12234 Student 2010 10 |  |

Passed

# 10.

За потребите на електронскиот огласник ФИНКИ-Огласи треба да се развие класа Oglas со информации за наслов (текстуална низа од максимум 50 знаци), категорија (текстуална низа од максимум 30 знаци), опис (текстуална низа од максимум 100 знаци) и цена изразена во евра (реален број). **(5 поени)**

За оваа класа да се обезбедат:

* Оператор > за споредба на двa огласи според цената **(5 поени)**
* Оператор << за печатење на податоците за огласот во форма: **(5 поени)**

[наслов]

[опис]

[цена] evra

Да се развие класа Oglasnik во која се чуваат податоци за име на огласникот (текстуална низа од максимум 20 знаци), низа од огласи (динамички резервирана низа од објекти од класата Oglas) и број на огласи во низата (цел број) **(5 поени)**

За оваа класа да се обезбедат:

* Оператор += за додавање нов оглас во низата од огласи. **(5 поени)** Ако цената на огласот што се внесува е негативна, треба да се генерира исклучок NegativnaVrednost (објект од класата NegativnaVrednost што посебно треба да се дефинира). Во ваков случај се печати порака **“Oglasot ima nevalidna vrednost za cenata i nema da bide evidentiran!“** и не се додава во низата. **(5 поени)**
* Оператор << за печатење на огласите во огласникот. **(5 поени)** Печатењето треба да биде во следниот формат:

[Име на огласникот]

[наслов1] [опис1] [цена1]

[наслов2] [опис2] [цена2]

...

* Функција void oglasOdKategorija(const char \*k) со која се печатат сите огласи од категоријата k што е проследена како влезен аргумент на методот. **(5 поени)**
* Функција void najniskaCena() со која се печати огласот што има најниска цена. Ако има повеќе огласи со иста најниска цена, да се испечатат податоците за првиот од нив. **(5 поени)**

Сите променливи во класите се **приватни**.

Да се обезбедат сите потребни методи за правилно функционирање на програмата. **(5 поени)**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 2  BMW 520 FULL OPREMA  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  13000  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 | -----Test Oglas & operator > -----  Prviot oglas e poskap. | -----Test Oglas & operator > -----  Prviot oglas e poskap. |  |
|  | 6  FINKI-Oglasnik  7  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  12000  BMW 520 FULL OPREMA  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  13000  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Avtomobili  Godina: 2013 Kilometraza: 150 000 - 159 999  12900  NOV STAN OD 102M2 NA VODNO  Stanovi  Broj na sobi: 4  99000  NOV DVOSOBEN STAN 35m2 VLAE  Stanovi  Broj na sobi: 2  36000  Luksuzen stan vo Aerodrom  Stanovi  Broj na sobi: 5, Povrsina 90m2  89000  Luksuzen stan vo Ohrid  Stanovi  Broj na sobi: 4, Povrsina 70m2  50000 | -----Test najniskaCena -----  Oglas so najniska cena:  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra | -----Test najniskaCena -----  Oglas so najniska cena:  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra |  |
|  | 2  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  12000  BMW 520 FULL OPREMA  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  13000 | -----Test Oglas & operator > -----  Prviot oglas ne e poskap. | -----Test Oglas & operator > -----  Prviot oglas ne e poskap. |  |
|  | 1  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 | -----Test Oglas & operator <<-----  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra | -----Test Oglas & operator <<-----  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra |  |
|  | 4  FINKI-Oglasnik  7  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  12000  BMW 520 FULL OPREMA  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  13000  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Avtomobili  Godina: 2013 Kilometraza: 150 000 - 159 999  12900  NOV STAN OD 102M2 NA VODNO  Stanovi  Broj na sobi: 4  99000  NOV DVOSOBEN STAN 35m2 VLAE  Stanovi  Broj na sobi: 2  36000  Luksuzen stan vo Aerodrom  Stanovi  Broj na sobi: 5, Povrsina 90m2  89000  Luksuzen stan vo Ohrid  Stanovi  Broj na sobi: 4, Povrsina 70m2  50000  Stanovi | -----Test oglasOdKategorija -----  Oglasi od kategorijata: Stanovi  NOV STAN OD 102M2 NA VODNO  Broj na sobi: 4  99000 evra  NOV DVOSOBEN STAN 35m2 VLAE  Broj na sobi: 2  36000 evra  Luksuzen stan vo Aerodrom  Broj na sobi: 5, Povrsina 90m2  89000 evra  Luksuzen stan vo Ohrid  Broj na sobi: 4, Povrsina 70m2  50000 evra | -----Test oglasOdKategorija -----  Oglasi od kategorijata: Stanovi  NOV STAN OD 102M2 NA VODNO  Broj na sobi: 4  99000 evra  NOV DVOSOBEN STAN 35m2 VLAE  Broj na sobi: 2  36000 evra  Luksuzen stan vo Aerodrom  Broj na sobi: 5, Povrsina 90m2  89000 evra  Luksuzen stan vo Ohrid  Broj na sobi: 4, Povrsina 70m2  50000 evra |  |
|  | 7  FINKI-Oglasnik  9  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  12000  BMW 520 FULL OPREMA  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  12000  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Avtomobili  Godina: 2013 Kilometraza: 150 000 - 159 999  12900  NOV STAN OD 110M2 NA VODNO  Stanovi  Broj na sobi: 4  99000  NOV DVOSOBEN STAN 33m2 VLAE  Stanovi  Broj na sobi: 2  -36000  Luksuzen stan vo Aerodrom  Stanovi  Broj na sobi: 5, Povrsina 90m2  89000  Luksuzen stan vo Ohrid  Stanovi  Broj na sobi: 4, Povrsina 74m2  50000  KUKA OD 250M2 VO KISELA VODA  Kuki  Prekrasen pogled, Broj na sobi > 10, Povrsina 400m2  156000  Moderno namestena vila itno  Kuki  Broj na sobi 6, Povrsina 130m2  80000  Stanovi | -----Test All -----  Oglasot ima nevalidna vrednost za cenata i nema da bide evidentiran!  FINKI-Oglasnik  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra  BMW 520 FULL OPREMA  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Godina: 2013 Kilometraza: 150 000 - 159 999  12900 evra  NOV STAN OD 110M2 NA VODNO  Broj na sobi: 4  99000 evra  Luksuzen stan vo Aerodrom  Broj na sobi: 5, Povrsina 90m2  89000 evra  Luksuzen stan vo Ohrid  Broj na sobi: 4, Povrsina 74m2  50000 evra  KUKA OD 250M2 VO KISELA VODA  Prekrasen pogled, Broj na sobi > 10, Povrsina 400m2  156000 evra  Moderno namestena vila itno  Broj na sobi 6, Povrsina 130m2  80000 evra  Oglasi od kategorijata: Stanovi  NOV STAN OD 110M2 NA VODNO  Broj na sobi: 4  99000 evra  Luksuzen stan vo Aerodrom  Broj na sobi: 5, Povrsina 90m2  89000 evra  Luksuzen stan vo Ohrid  Broj na sobi: 4, Povrsina 74m2  50000 evra  Oglas so najniska cena:  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra | -----Test All -----  Oglasot ima nevalidna vrednost za cenata i nema da bide evidentiran!  FINKI-Oglasnik  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra  BMW 520 FULL OPREMA  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Godina: 2013 Kilometraza: 150 000 - 159 999  12900 evra  NOV STAN OD 110M2 NA VODNO  Broj na sobi: 4  99000 evra  Luksuzen stan vo Aerodrom  Broj na sobi: 5, Povrsina 90m2  89000 evra  Luksuzen stan vo Ohrid  Broj na sobi: 4, Povrsina 74m2  50000 evra  KUKA OD 250M2 VO KISELA VODA  Prekrasen pogled, Broj na sobi > 10, Povrsina 400m2  156000 evra  Moderno namestena vila itno  Broj na sobi 6, Povrsina 130m2  80000 evra  Oglasi od kategorijata: Stanovi  NOV STAN OD 110M2 NA VODNO  Broj na sobi: 4  99000 evra  Luksuzen stan vo Aerodrom  Broj na sobi: 5, Povrsina 90m2  89000 evra  Luksuzen stan vo Ohrid  Broj na sobi: 4, Povrsina 74m2  50000 evra  Oglas so najniska cena:  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra |  |
|  | 5  FINKI-Oglasnik  7  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  12000  BMW 520 FULL OPREMA  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  -13000  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Avtomobili  Godina: 2013 Kilometraza: 150 000 - 159 999  12900  NOV STAN OD 102M2 NA VODNO  Stanovi  Broj na sobi: 4  99000  NOV DVOSOBEN STAN 35m2 VLAE  Stanovi  Broj na sobi: 2  36000  Luksuzen stan vo Aerodrom  Stanovi  Broj na sobi: 5, Povrsina 90m2  -89000  Luksuzen stan vo Ohrid  Stanovi  Broj na sobi: 4, Povrsina 70m2  50000 | -----Test Exception -----  Oglasot ima nevalidna vrednost za cenata i nema da bide evidentiran!  Oglasot ima nevalidna vrednost za cenata i nema da bide evidentiran!  FINKI-Oglasnik  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Godina: 2013 Kilometraza: 150 000 - 159 999  12900 evra  NOV STAN OD 102M2 NA VODNO  Broj na sobi: 4  99000 evra  NOV DVOSOBEN STAN 35m2 VLAE  Broj na sobi: 2  36000 evra  Luksuzen stan vo Ohrid  Broj na sobi: 4, Povrsina 70m2  50000 evra | -----Test Exception -----  Oglasot ima nevalidna vrednost za cenata i nema da bide evidentiran!  Oglasot ima nevalidna vrednost za cenata i nema da bide evidentiran!  FINKI-Oglasnik  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Godina: 2013 Kilometraza: 150 000 - 159 999  12900 evra  NOV STAN OD 102M2 NA VODNO  Broj na sobi: 4  99000 evra  NOV DVOSOBEN STAN 35m2 VLAE  Broj na sobi: 2  36000 evra  Luksuzen stan vo Ohrid  Broj na sobi: 4, Povrsina 70m2  50000 evra |  |
|  | 3  FINKI-Oglasnik  3  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  12000  BMW 520 FULL OPREMA  Avtomobili  Godina: 2011 Kilometraza: 140 000 - 149 999  13000  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Avtomobili  Godina: 2013 Kilometraza: 150 000 - 159 999  12900 | -----Test Oglasnik, operator +=, operator << -----  FINKI-Oglasnik  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra  BMW 520 FULL OPREMA  Godina: 2011 Kilometraza: 140 000 - 149 999  13000 evra  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Godina: 2013 Kilometraza: 150 000 - 159 999  12900 evra | -----Test Oglasnik, operator +=, operator << -----  FINKI-Oglasnik  VW PASSAT 2.0 TDI 140 KS BLUEMOTION  Godina: 2011 Kilometraza: 140 000 - 149 999  12000 evra  BMW 520 FULL OPREMA  Godina: 2011 Kilometraza: 140 000 - 149 999  13000 evra  AUDI A3 2.0 TDI LIMOUSINE PRO LINE  Godina: 2013 Kilometraza: 150 000 - 159 999  12900 evra |  |

Passed

# 11.

Потребно е да се моделира продажба на карти за прикажување на претстави во некој културен центар. За таа цел, да се направи класа DELO. За класата се чуваат следните атрибути:

* Име на делото (низа од 50 знаци)
* Година кога е напишано (цел број)
* Земја на потекло (низа од 50 знаци)

За класата да се направат потребниот конструктор, get и set методи. За класата DELO да се преоптовари операторот == кој ќе споредува дали две дела се исти (две дела се исти ако имаат исто име) **(5 поени)**.

Да се дефинира класа PRЕTSTAVA. За класата се чуваат информации за:

* Делото кое се прикажува (објект од класата DELO)
* Број на продадени карти (цел број)
* Дата на прикажување (низа од 15 знаци)

За класата да се дефинира потребниот конструктор, set и get методите Дополнително, за класата PRETSTAVA да се напише метода cena која треба да ја врати цената на една карта. **(5 поени)** Цената зависи од староста на делото и од земјата на потекло и се пресметува по следната формула: **Цена = N + М**, каде М = 50ден. ако делото е напишано во 20 или 21 век, М = 75ден. ако делото е напишано во 19 век и М = 100ден. ако делото е напишано пред 19 век. N = 100ден ако делото е од Италија, N = 150ден ако делото е од Русија. Делата од останатите земји имаат N = 80ден.

Претставите можат да бидат опера и балет. За балетите се чува дополнителен атрибут за цената на балетот која се додава на оригиналната цена на картата и таа вредност е иста за сите балетски претстави. На почеток изнесува 150 денари, со можност да се променува ако така одлучи менаџментот на културниот центар. **(10 поени)**

Да се дефинира надворешна функција prihod која како аргумент прима низа од покажувачи кон претстави кои се прикажале во еден месец и големина на низата, а како резултат треба да го пресмета и врати вкупниот приход на културниот центар. **(10 поени)**

Да се дефинира надворешна функција brojPretstaviNaDelo која како аргумент прима низа од покажувачи кон претстави кои се прикажале во еден месец, големина на низата и едно дело. Оваа функција пресметува на колку од дадените претстави е прикажано даденото дело. **(5 поени)**

*(Напомена: Во низата од претстави која се проследува како аргумент во функциите prihod и brojPretstaviNaDelo може да имаме повеќе претстави за едно исто дело)*

Целосна функционалност на задачата **(5 поени)**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 4  4  0 Anastasia 1967 Rusija 250 10.05.2016  1 Rigoletto 1851 Italija 120 11.05.2016  0 Anastasia 1967 Rusija 100 12.05.2016  1 Beggar 1728 Anglija 60 13.05.2016 | ======TEST CASE 4=======  154300 | ======TEST CASE 4=======  154300 |  |
|  | 2  0 Anastasia 1967 Rusija 250 10.05.2016  1 Rigoletto 1851 Italija 120 11.05.2016 | ======TEST CASE 2=======  350  175 | ======TEST CASE 2=======  350  175 |  |
|  | 6  5  0 Anastasia 1967 Rusija 250 10.05.2016  1 Rigoletto 1851 Italija 120 11.05.2016  0 Anastasia 1967 Rusija 100 12.05.2016  1 Beggar 1728 Anglija 60 13.05.2016  1 Rigoletto 1851 Italija 150 16.05.2016  Rigoletto 1851 Italija | ======TEST CASE 6=======  2 | ======TEST CASE 6=======  2 |  |
|  | 3  Anastasia 1967 Rusija  Anastasia 1967 Rusija  Rigoletto 1851 Italija | ======TEST CASE 3=======  Isti se  Ne se isti | ======TEST CASE 3=======  Isti se  Ne se isti |  |
|  | 1  0 Anastasia 1967 Rusija 250 10.05.2016  1 Rigoletto 1851 Italija 120 11.05.2016 | ======TEST CASE 1=======  Anastasia  Rigoletto | ======TEST CASE 1=======  Anastasia  Rigoletto |  |
|  | 5  200  4  0 Anastasia 1967 Rusija 250 10.05.2016  1 Rigoletto 1851 Italija 120 11.05.2016  0 Anastasia 1967 Rusija 100 12.05.2016  1 Beggar 1728 Anglija 60 13.05.2016 | ======TEST CASE 5=======  171800 | ======TEST CASE 5=======  171800 |  |

Passed

# 12.

Да се креира класа за опишување на еден сервис за мобилни уреди. За секој мобилен телефон се чуваат податоци за модел (низа од 100 знаци), тип на уред (смартфон или компјутер), потребни часови за основна проверка (децимален број), година на производство (цел број). **(5 поени)**

Сите уреди имаат исто времетраење за основна проверка и истата изнесува 1 час. Оваа вредност може да се смени со одлука на сервисот. Дополнително, уредите кои се понови од 2015 година имаат потреба од дополнителени 1.5 часа за проверка. При проверка, лаптопите имаат потреба од дополнителни 2 часа за софтверска дијагностика **(5 поени)**.

За оваа класа да се имплементира оператор << за печатење на уред во формат:

[model*na*uredotl]  
[tip*na*ured] [casovi*za*proverka]

Каде часовите за проверка се оние од основната проверка+ дополнителни 2 часа доколку уредот е понов од 2015 + 2 часа доколу уредот е лаптоп **(5 поени)**

Да се креира класа MobileServis во која се чува адреса (низа од 100 знаци), низа од уреди (динамички алоцирана низа) и број на уреди.**(5 поени)**

За класата да се обезбедат:

operatoror+= за додавање на нов уред во сервисот.(5 поени) Не смее да се дозволи годината на производство на уредот да биде поголема од 2019 или помала од 2000. Ако се направи обид за додавање на уред со невалидна година, треба да се генерира исклучок InvalidProductionDate. **(10 поени)**

Функција pecatiUredi со која се печатат сите уреди со времето потребно за нивната проверка во сервисот **(5 поени)**.

Да се обезбедат сите дополнителни методи потребни за правилно функционирање на програмата. **(5 поени)**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 3  TelService  6  iPhone  0  2020  Samsung  0  1999  Huawei  0  1990  Toshiba  1  2016  OnePlus  0  2009  Fujitsu  1  2010 | ===== Testiranje na isklucoci ======  Невалидна година на производство  Невалидна година на производство  Невалидна година на производство  Ime: TelService  Toshiba  Laptop 5  OnePlus  Mobilen 1  Fujitsu  Laptop 3 | ===== Testiranje na isklucoci ======  Невалидна година на производство  Невалидна година на производство  Невалидна година на производство  Ime: TelService  Toshiba  Laptop 5  OnePlus  Mobilen 1  Fujitsu  Laptop 3 |  |
|  | 2  iStyle  10  iPhone  0  2019  Samsung  0  2015  Huawei  0  2013  Toshiba  1  2016  OnePlus  0  2009  Fujitsu  1  2010  Dell  1  2019  Sony  0  2007  Hp  1  2019  Alcatel  0  2011 | ===== Testiranje na operatorot += ======  Ime: iStyle  iPhone  Mobilen 3  Samsung  Mobilen 1  Huawei  Mobilen 1  Toshiba  Laptop 5  OnePlus  Mobilen 1  Fujitsu  Laptop 3  Dell  Laptop 5  Sony  Mobilen 1  Hp  Laptop 5  Alcatel  Mobilen 1 | ===== Testiranje na operatorot += ======  Ime: iStyle  iPhone  Mobilen 3  Samsung  Mobilen 1  Huawei  Mobilen 1  Toshiba  Laptop 5  OnePlus  Mobilen 1  Fujitsu  Laptop 3  Dell  Laptop 5  Sony  Mobilen 1  Hp  Laptop 5  Alcatel  Mobilen 1 |  |
|  | 6  MobiClick  14  iPhone  0  2019  Samsung  0  2015  Huawei  0  2009  Toshiba  1  2016  OnePlus  0  2009  Fujitsu  1  2010  Dell  1  2019  Sony  0  2007  Hp  1  2019  Alcatel  0  2011  Simens  0  1990  Macintosh  1  1900  Lenovo  1  2016  Lg  0  2014 | ===== Testiranje na kompletna funkcionalnost ======  Невалидна година на производство  Невалидна година на производство  Ime: MobiClick  iPhone  Mobilen 5  Samsung  Mobilen 3  Huawei  Mobilen 3  Toshiba  Laptop 7  OnePlus  Mobilen 3  Fujitsu  Laptop 5  Dell  Laptop 7  Sony  Mobilen 3  Hp  Laptop 7  Alcatel  Mobilen 3  Lenovo  Laptop 7  Lg  Mobilen 3 | ===== Testiranje na kompletna funkcionalnost ======  Невалидна година на производство  Невалидна година на производство  Ime: MobiClick  iPhone  Mobilen 5  Samsung  Mobilen 3  Huawei  Mobilen 3  Toshiba  Laptop 7  OnePlus  Mobilen 3  Fujitsu  Laptop 5  Dell  Laptop 7  Sony  Mobilen 3  Hp  Laptop 7  Alcatel  Mobilen 3  Lenovo  Laptop 7  Lg  Mobilen 3 |  |
|  | 5  MobileService  10  iPhone  0  2019  Samsung  0  2015  Huawei  0  2009  Toshiba  1  2016  OnePlus  0  2009  Fujitsu  1  2010  Dell  1  2019  Sony  0  2007  Hp  1  2019  Alcatel  0  2011 | ===== Testiranje na static clenovi ======  Ime: MobileService  iPhone  Mobilen 3  Samsung  Mobilen 1  Huawei  Mobilen 1  Toshiba  Laptop 5  OnePlus  Mobilen 1  Fujitsu  Laptop 3  Dell  Laptop 5  Sony  Mobilen 1  Hp  Laptop 5  Alcatel  Mobilen 1  ===== Promena na static clenovi ======  Ime: MobileService  iPhone  Mobilen 4  Samsung  Mobilen 2  Huawei  Mobilen 2  Toshiba  Laptop 6  OnePlus  Mobilen 2  Fujitsu  Laptop 4  Dell  Laptop 6  Sony  Mobilen 2  Hp  Laptop 6  Alcatel  Mobilen 2 | ===== Testiranje na static clenovi ======  Ime: MobileService  iPhone  Mobilen 3  Samsung  Mobilen 1  Huawei  Mobilen 1  Toshiba  Laptop 5  OnePlus  Mobilen 1  Fujitsu  Laptop 3  Dell  Laptop 5  Sony  Mobilen 1  Hp  Laptop 5  Alcatel  Mobilen 1  ===== Promena na static clenovi ======  Ime: MobileService  iPhone  Mobilen 4  Samsung  Mobilen 2  Huawei  Mobilen 2  Toshiba  Laptop 6  OnePlus  Mobilen 2  Fujitsu  Laptop 4  Dell  Laptop 6  Sony  Mobilen 2  Hp  Laptop 6  Alcatel  Mobilen 2 |  |
|  | 1  Nokia  0  2018  MobiClick | ===== Testiranje na klasite ======  Nokia  Mobilen 3 | ===== Testiranje na klasite ======  Nokia  Mobilen 3 |  |
|  | 4  MobileStar  10  iPhone  0  2019  Samsung  0  2011  Huawei  0  2016  Toshiba  1  2016  OnePlus  0  2025  Fujitsu  1  2010  Dell  1  2019  Sony  0  2015  Hp  1  2019  Alcatel  0  2011 | ===== Testiranje na konstruktori ======  Невалидна година на производство  Ime: MobileStar  iPhone  Mobilen 3  Samsung  Mobilen 1  Huawei  Mobilen 3  Toshiba  Laptop 5  Fujitsu  Laptop 3  Dell  Laptop 5  Sony  Mobilen 1  Hp  Laptop 5  Alcatel  Mobilen 1 | ===== Testiranje na konstruktori ======  Невалидна година на производство  Ime: MobileStar  iPhone  Mobilen 3  Samsung  Mobilen 1  Huawei  Mobilen 3  Toshiba  Laptop 5  Fujitsu  Laptop 3  Dell  Laptop 5  Sony  Mobilen 1  Hp  Laptop 5  Alcatel  Mobilen 1 |  |

Passed

# 13.

Да се креира класа која претставува слика (Image) во рамките на еден компјутер. За секоја датотека се чуваат **(5 поени)**:

* име на сликата (дин. алоцирана низа од знаци). Предефинирано поставено на untitled.
* име на корисникот кој е сопственик на датотеката (низа од макс. 50 знаци). Предефинирано поставено на unknown.
* димензии на сликата (2 цели броеви кои претставуваат ширина и висина во пиксели). Предефинирано поставени на 800.

Од класата Imageда се изведе класата TransparentImage, која претставува слика која содржи поддржува транспарентност. За транспарентна слика дополнително се чува :

* дали поддржува нивоа на транспарентност (bool променлива, предефинирано не поддржува транспарентност).

За класите Image и TransparentImage да биде достапна метода (fileSize)за пресметка на големината на сликата (во бајти). Големината се пресметува на сл. начин **(5 поени)**:

* за објектите од класата Image, големината на датотеката е еднаква на:висината \* ширината \* 3.
* за TransparentImage, доколку поддржува нивоа на транспарентност, големината на датотеката е еднаква на ширина \* висина \* 4. Доколку не поддржува транспарентност, големината на сликата се пресметува како ширина \* висина + бр.\_на\_пиксели\_присутни\_во\_сликата / 8.

За класите Image и TransparentImage да се преоптоварат следниве оператори **(5 поени)**:

* оператор << кој ги печати податоците од датотеката во сл. формат:

ime\_fajl avtor golemina\_na\_fajlot\_vo\_bajti

* оператор > кој ги споредува сликите според нивната големина.

Да се дефинира класа Folder, што репрезентира директориум во кој може да се поставуваат слики. За секој фолдер се чува **(5 поени)**:

* име на фолдерот (низа од макс 255 знаци)
* име на корисникот кој е сопственик на фолдерот (низа од макс. 50 знаци). Предефинирано поставено на unknown.
* низа од покажувачи кон Image објекти (макс. 100 во рамките на еден фолдер).

Да се имплементираат следниве методи **(5 поени)**:

* метода за пресметка на големината на фолдерот (folderSize). Големината на фолдерот е сума од големините од сите слики во рамките на фолдерот.
* метода за пронаоѓање на наголемата слика во рамките на фолдерот (getMaxFile). Методата враќа покажувач кон најголемата слика во фолдерот (сметано во бајти).

За класата Folder да се имплементира и оператор += кој додава објекти од типот Image/TransparentImage во рамките на фолдерот **(5 поени)**.

Да се креираат следниве функционалности за класите **(5 поени)**:

* operator << - со кој се печатат податоците за фолдерот во формат *(5 поени)*:

ime\_folder sopstvenik --

* 1. ime*slika avtor golemina*na*fajlot*vo\_bajti
  2. ime*slika avtor golemina*na*fajlot*vo*bajti -- golemina*na*folder*vo\_bajti
* operator [] - кој враќа покажувач кон соодветната слика во фолдерот. Доколку не постои слика на тој индекс, треба да се врати NULL **(5 поени)**.

Да се креира и глобална функција max\_folder\_size која го прима низа од објекти од типот Folder и го враќа фолдерот кој има зафаќа најмногу простор (во бајти). (**5 поени**)

Да се овозможи правилно функционирање на класите (потребни set или get методи/оператори/конструктори/деструктори) според изворниот код кој е веќе зададен. Сите податочни членови на класите се protected. (**5 поени**)

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 5  folder  user  1  2  image1  u1  100 100  1  1  2  image1  u1  100 100  0  0 | image1 u1 40000 | image1 u1 40000 |  |
|  | 7  3  dd  uu  1  1  fileName1  user1  50 50  1  2  textfile  pero  100 100  1  1  2  textfile  kiro  100 100  0  0  directory2  ana  1  1  fajl  ana  10 10  0  oop  alek  0 | dd uu  --  fileName1 user1 7500  textfile pero 40000  textfile kiro 11250  --  Folder size: 58750 | dd uu  --  fileName1 user1 7500  textfile pero 40000  textfile kiro 11250  --  Folder size: 58750 |  |
|  | 4  folder  user  1  2  image1  u1  100 100  1  1  2  image1  u1  100 100  0  0 | folder user  --  image1 u1 40000  image1 u1 11250  --  Folder size: 51250 | folder user  --  image1 u1 40000  image1 u1 11250  --  Folder size: 51250 |  |
|  | 1  filename  username  10  10 | untitled unknown 1920000  filename unknown 1920000  filename username 1920000  filename username 300 | untitled unknown 1920000  filename unknown 1920000  filename username 1920000  filename username 300 |  |
|  | 3  folder  user | folder user  --  --  Folder size: 0 | folder user  --  --  Folder size: 0 |  |
|  | 4  folder  user  1  1  image1  u1  1000 1000  1  1  image1  u1  100 100  0 | folder user  --  image1 u1 3000000  image1 u1 30000  --  Folder size: 3030000 | folder user  --  image1 u1 3000000  image1 u1 30000  --  Folder size: 3030000 |  |
|  | 4  folder  user  1  1  image1  u1  100 100  1  2  image1  u1  10 10  0  0 | folder user  --  image1 u1 30000  image1 u1 112  --  Folder size: 30112 | folder user  --  image1 u1 30000  image1 u1 112  --  Folder size: 30112 |  |
|  | 6  dirName  user  1  1  image1  user1  100 100  1  2  trasnparentImage  pero  100 100  1  1  2  trasparentImage2  kiro  500 500  0  0  1 | trasnparentImage pero 40000 | trasnparentImage pero 40000 |  |
|  | 2  transpImage  user  10  10  0 | untitled unknown 2560000  transpImage user 112 | untitled unknown 2560000  transpImage user 112 |  |
|  | 6  dirName  user  1  1  image1  user1  100 100  1  2  trasnparentImage  pero  100 100  1  1  2  trasparentImage2  kiro  500 500  0  0  2 | trasparentImage2 kiro 281250 | trasparentImage2 kiro 281250 |  |
|  | 5  folder  user  1  2  image1  u1  100 100  1  1  2  image1  u1  10 10  0  0 | image1 u1 40000 | image1 u1 40000 |  |

Passed

# 14.

Во рамките на еден телекомуникациски оператор, СМС пораките се делат на регуларни и специјални. За секојa СМС порака се знае:

* основна цена за една порака до 160 знаци (реален број)
* претплатнички број на кој е испратена (низа од знаци со должина 12)

Класата за опишување на СМС пораки е **апстрактна (5 поени)**.

За секоја Регуларна СМСдополнително се чуваат податоци за текстот на пораката и тоа дали е користена роаминг услуга (bool променлива). За секоја Специјална СМС дополнително се чуваат податоци за тоа дали е наменета за хуманитарни цели (bool променлива). **(5 поени)**

За секој објект од двете изведени класи треба да бидат на располагање следниве методи:

Метод SMS\_cena(), за пресметување на цената на испратена СМС порака на следниот начин: **(10 поени)**

* За регуларна СМС - цената се зголемува за 300% од основната цена ако е испратена од странство, а 18% од дома и цената се формира врз база на тоа во колку СМС пораки ќе го собере текстот што сакаме да го испратиме. Една регуларна порака може да собере најмногу 160 знаци. Притоа, доколку се надмине 160-от знак, се започнува нова порака и цената се пресметува како за две или повеќе пораки (пр. за 162 знаци, цената на СМС е иста како и за 320 знаци).
* За специјална СМС - цената се зголемува за 150% од основната цена ако пораката HE е наменета за хуманитарни цели. Доколку е наменета за тоа, данокот од 18% не се пресметува на цената.
* Пресметувањето 18% данок на цената на сите СМС пораки е фиксен и не се менува, додека пак процентот од 300% за регуларни и 150% за специјални СМС е променлив и во зависност од операторот може да се менува. Да се обезбеди механизам за можност за нивно менување. **(5 поени)**

Преоптоварен оператор << за печатење на податоците за СМС пораките во формат: **(5 поени)**

Тел.број: цена

Да се имплементира функција vkupno\_SMS со потпис:

void vkupno\_SMS(SMS\*\* poraka, int n)

во која се печати вкупниот број на регуларни СМС пораки и нивната вкупна цена, како и бројот на специјални СМС пораки и нивната вкупна цена во проследената низа посебно. **(15 поени)**

Да се обезбедат сите потребни функции за правилно функционирање на програмата **(5 поени)**.

**For example:**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 4  0038971123456  5  Test poraka za testiranje na promena na procentot?  1  0038971123456  5  Test poraka za testiranje na promena na procentot?  1  350 | ====== Testing RegularSMS class with a changed percentage======  Tel: 0038971123456 - cena: 20den.  Tel: 0038971123456 - cena: 22.5den. | ====== Testing RegularSMS class with a changed percentage======  Tel: 0038971123456 - cena: 20den.  Tel: 0038971123456 - cena: 22.5den. |  |
|  | 3  3  2  0500123456  45  0  2  0038971140300  200  1  2  0038975140300  100  1 | ====== Testing method vkupno\_SMS() ======  Vkupno ima 0 regularni SMS poraki i nivnata cena e: 0  Vkupno ima 3 specijalni SMS poraki i nivnata cena e: 412.5 | ====== Testing method vkupno\_SMS() ======  Vkupno ima 0 regularni SMS poraki i nivnata cena e: 0  Vkupno ima 3 specijalni SMS poraki i nivnata cena e: 412.5 |  |
|  | 2  2  0038971111222  15.8  0  0038975140300  100  1 | ====== Testing SpecialSMS class ======  CONSTRUCTOR  OPERATOR <<  Tel: 0038971111222 - cena: 39.5den.  CONSTRUCTOR  OPERATOR <<  Tel: 0038975140300 - cena: 100den. | ====== Testing SpecialSMS class ======  CONSTRUCTOR  OPERATOR <<  Tel: 0038971111222 - cena: 39.5den.  CONSTRUCTOR  OPERATOR <<  Tel: 0038975140300 - cena: 100den. |  |
|  | 3  5  2  0500123456  45  0  1  0038975123456  5  Test poraka!  1  1  0038971123456  5  Test poraka za testiranje na dolzinata na porakata. Ovaa poraka treba da bide pogolema od 160 karakteri. Kolkava e cenata na ovaa poraka iako e ispratena od domasen operator!? Test poraka za testiranje na dolzinata na porakata. Ovaa poraka treba da bide pogolema od 160 karakteri. Kolkava e cenata na ovaa poraka iako e ispratena od domasen operator!?  0  2  0038971111222  45  0  2  0038975140300  100  1 | ====== Testing method vkupno\_SMS() ======  Vkupno ima 2 regularni SMS poraki i nivnata cena e: 37.7  Vkupno ima 3 specijalni SMS poraki i nivnata cena e: 325 | ====== Testing method vkupno\_SMS() ======  Vkupno ima 2 regularni SMS poraki i nivnata cena e: 37.7  Vkupno ima 3 specijalni SMS poraki i nivnata cena e: 325 |  |
|  | 1  2  0038975123456  5  Test poraka!  1  0038971123456  5  Test poraka za testiranje na dolzinata na porakata. Ovaa poraka treba da bide pogolema od 160 karakteri. Kolkava e cenata na ovaa poraka iako e ispratena od domasen operator!?  0 | ====== Testing RegularSMS class ======  CONSTRUCTOR  OPERATOR <<  Tel: 0038975123456 - cena: 20den.  CONSTRUCTOR  OPERATOR <<  Tel: 0038971123456 - cena: 11.8den. | ====== Testing RegularSMS class ======  CONSTRUCTOR  OPERATOR <<  Tel: 0038975123456 - cena: 20den.  CONSTRUCTOR  OPERATOR <<  Tel: 0038971123456 - cena: 11.8den. |  |
|  | 5  0038971111222  20  0  0038971111222  20  0  180 | ====== Testing SpecialSMS class with a changed percentage======  Tel: 0038971111222 - cena: 50den.  Tel: 0038971111222 - cena: 56den. | ====== Testing SpecialSMS class with a changed percentage======  Tel: 0038971111222 - cena: 50den.  Tel: 0038971111222 - cena: 56den. |  |
|  | 1  2  0038975123456  5  Test poraka!  0  0038971123456  5.5  Testirame isprakjanje na poraka.  0 | ====== Testing RegularSMS class ======  CONSTRUCTOR  OPERATOR <<  Tel: 0038975123456 - cena: 5.9den.  CONSTRUCTOR  OPERATOR <<  Tel: 0038971123456 - cena: 6.49den. | ====== Testing RegularSMS class ======  CONSTRUCTOR  OPERATOR <<  Tel: 0038975123456 - cena: 5.9den.  CONSTRUCTOR  OPERATOR <<  Tel: 0038971123456 - cena: 6.49den. |  |

Passed

# 15.

Дадена е дел од дефиниција на класата StudentKurs. За секој студент кој слуша некој курс се чуваат информации за неговото **име** (низа од знаци), **оценка од писмен дел** (цел број) и **дали студентот сака да биде испрашуван и усно** (булова променлива).

Оцената за писмениот дел е вредност од 1 до МАX. MAX има почетна вредност 10 која е иста е за сите и може да се промени со соодветна функција setMAX(). **(5 поени)**

Кај оние студенти кои сакаат да бидат испрашувани усно и добиваат и описна оценка. Имплементирај соодветна класа StudentKursUsno во која се чуваат дополнителни информации за описна оценка на усното испрашување (динамичко алоцирана низа од знаци). Описни оценки може да бидат: **odlicen**, **dobro**, **losho**.... Ако описната оценка е odlicen тогаш оцената ќе биде најмногу за оценки 2 повисока од оценката од писмениот дел, ако е dobro оценката ќе биде најмногу за 1 повисока, а ако е losho за 1 пониска. Ако студентот добие некоја друга описна оценка, неговата оценка ќе остане иста со оценката од писмениот дел.

Во класите StudentKurs и StudentKursUsno треба да се имплементираат потребните конструктори, функции за поставување и преземање и деструктор. **(5 поени)**

За сите објекти од класите треба да се обезбедат следните функции:

* **преоптоварен оператор <<** во кој се печатат информации за студентот на курсот во формат: Ime --- ocenka **(5 поени)**
* оcenka() – ја враќа оценката на студентот на курсот **(5 поени)**

Во класата StudentKursUsno дефинирај ја функцијата:

* преоптоварен оператор += за поставување на описна оценка на студентот**(5 поени)**

Ако се направи обид да се внесе описна оценка во која покрај букви има и други знаци треба да се фрли исклучок (објект од класата BadInputException). Фатете го исклучокот во главната функција каде што е потребно. Откако ќе го фатите отпечатете соодветна порака за грешка (Greshna opisna ocenka) и справете се со исклучокот така што тие знаци ќе отстранат од стрингот со описната оценка (ако описната оценка била dо1ba0r ќе се промени во dоbar). **(10 поени)**

Дадена е дел од дефиницијата на класата KursFakultet во која се чуваат информаци за името на курсот (низа од 30 знаци) и за студентите запишани на курсот (низа од 20 покажувачи кон класата StudentKurs) како и бројот на студенти запишани на курсот.

Еден студент кој запишал некој курс ќе го положи курсот ако има барем MINOCENKA. Вредноста MINOCENKA е членка на класата StudentKurs и има фиксна вредност 6 која не може да се промени. **(5 поени)**

Во класата KursFakultet имплементирај ги функциите:

* функција pecatiStudenti() во која се печатат сите студенти кои го положиле курсот, секој во посебен ред. Претходно во првиот ред се печати: Kursot XXXXX go polozile: **(5 поени)**.
* функција postaviOpisnaOcenka(char \* ime, char\* opisnaOcenka) во која на студентот на курсот со даденото име му се поствува описна оценка. Ако тоа не е можно функцијата не прави ништо **(15 поени)**.

Комплетна функционалност **(5 поени**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 2  Tanja 4 0  Vasko 8 1  2  Vasko do7br4o  Ana losho | Greshna opisna ocenka  Kursot OOP go polozile:  Vasko --- 9 | Greshna opisna ocenka  Kursot OOP go polozile:  Vasko --- 9 |  |
|  | 2  Tanja 4 0  Vasko 8 1  1  Vasko do7br4o | Greshna opisna ocenka  Kursot OOP go polozile:  Vasko --- 9 | Greshna opisna ocenka  Kursot OOP go polozile:  Vasko --- 9 |  |
|  | 2  Tanja 4 0  Vasko 9 1  1  Vasko dobro | Kursot OOP go polozile:  Vasko --- 9 | Kursot OOP go polozile:  Vasko --- 9 |  |
|  | 2  Tanja 4 1  Vasko 9 0  1  Tanja odlicen | Kursot OOP go polozile:  Tanja --- 6  Vasko --- 9 | Kursot OOP go polozile:  Tanja --- 6  Vasko --- 9 |  |
|  | 2  Tanja 4 0  Vasko 9 0  0 | Kursot OOP go polozile:  Vasko --- 9 | Kursot OOP go polozile:  Vasko --- 9 |  |
|  | 2  Tanja 4 0  Vasko 9 1  1  Vasko losho | Kursot OOP go polozile:  Vasko --- 8 | Kursot OOP go polozile:  Vasko --- 8 |  |

Passed

# 16.

Туристичка агенција нуди различни понуди за транспорт на патници. За секоја понуда се чуваат податоци за дестинацијата (низа од знаци), основна цена (цел број) и податок за растојанието до дестинацијата изразено во km (цел број).

Од класата Transport да се изведат класите AvtomobilTransport и KombeTransport за моделирање на понудите за транспорт кои нудат транспорт со автомобил или комбе, соодветно. За секоја понуда за транспорт со автомобил се чува податок за тоа дали во понудата има платен шофер (булова променлива), а за транспорт со комбе бројот на луѓе кои може да се превезат во комбето (цел број). **(5 поени)**

За секој објект од двете изведени класи треба да бидат на располагање следниве методи:

* конструктор со аргументи што одговараат на податочните членови и set и get методи за податочните членови **(5 поени)**
* метод cenaTransport, за пресметување на цената на понудата на следниот начин:
  + За транспорт со автомобил - основната цена се зголемува за 20% ако има платен шофер
  + За транспорт со комбе - основната цена ќе се намали зависно од бројот на патници во комбето. За секој патник во комбето цената се намалува за 200 **(10 поени)**
* преоптоварен оператор < за споредба на две понуди за транспорт на патник според растојанието до дестинацијата. **(5 поени)**

Да се имплементира функција pecatiPoloshiPonudi што на влез прима низа од понуди за транспорт на патник, бројот на елементите од низата и една понуда T. Функцијата ја печати дестинацијата, растојанието до дестинацијата и цената за понудите од низата кои се поскапи од понудата T сортирани во растечки редослед по растојанието до дестинацијата (погледни го повикот на функцијата) **(10 поени)**

Комплетна функционалност на кодот **(5 поени)**

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 3  1 Belgrad 3200 900 1  1 Bitola 2500 400 0  2 Sofija 5000 1040 2 | Bitola 400 2500  Belgrad 900 3840  Sofija 1040 4600 | Bitola 400 2500  Belgrad 900 3840  Sofija 1040 4600 |  |
|  | 2  1 Belgrad 3200 900 1  1 Bitola 2500 400 0 | Bitola 400 2500  Belgrad 900 3840 | Bitola 400 2500  Belgrad 900 3840 |  |
|  | 2  1 Belgrad 3200 900 1  1 Bitola 1500 400 0 | Belgrad 900 3840 | Belgrad 900 3840 |  |
|  | 4  2 Krushevo 1200 400 4  1 Belgrad 3200 900 1  1 Bitola 2500 400 0  2 Sofija 5000 1040 2 | Bitola 400 2500  Belgrad 900 3840  Sofija 1040 4600 | Bitola 400 2500  Belgrad 900 3840  Sofija 1040 4600 |  |

Passed

# 17.

Да се имплементира апстрактна класа FudblaskaEkipa во која се чува: **(5 поени)**

* име на тренерот на екипата (максимум 100 знаци)
* бројот на постигнати голови на последните 10 натпревари, последниот натпревар е на позиција 9, предпоследниот на позиција 8, итн (поле од 10 цели броеви)

Од класата FudblaskaEkipa да се изведат класите Klub и Reprezentacija. За секој клуб дополнително се чува податок за името и бројот на титули што ги има освоено, а за репрезентацијата се чуваат податоци за името на државата и вкупен број на меѓународни настапи. За овие класи да се имплементираат следните методи:

* соодветен конструктор **(5 поени)**
* оператор << за печатење на стандарден излез во формат [IME\_NA\_KLUB/DRZHAVA]\n[TRENER]\n[USPEH]\n **(5 поени)**
* преоптоварен оператор+= за додавање на голови од последниот натпревар (внимавајте секогаш се чуваат головите само од последните 10 натпревари) **(10 поени)**
* метод uspeh, за пресметување на успехот на тимот на следниот начин:
  + За Klub се пресметува како збир од збирот на головите од последните 10 натпревари помножен со 3 и бројот на титули помножен со 1000 (на пр. голови = {2, 0, 1, 3, 2, 0, 1, 4, 2, 3} и број на титули = 3, достигнување = 18 \* 3 + 3 \* 1000 = 3054)
  + За Reprezentacija како збир од збирот на головите од последните 10 натпревари помножен со 3 и бројот на меѓународни настапи помножен со 50 (на пр. голови = {2, 0, 1, 3, 2, 0, 1, 4, 2, 3} и број на меѓународни настапи=150, достигнување = 18 \* 3 + 150 \* 50 = 7554) **(10 поени)**
* преоптоварен оператор > за споредба на две фудбласки екипи од каков било вид (клубови или репрезентации) според успехот **(5 поени)**

Да се имплементира функција najdobarTrener што на влез прима низа од покажувачи кон FudblaskaEkipa и големина на низата и го печати тимот со најголем успех **(10 поени)**.

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 5  0  Zinedin Zidane  2 2 2 3 4 1 5 3 0 3  Real Madrid CF  2  0  Luis Enrique  3 4 5 1 3 0 6 5 4 3  FC Barcelona  6  1  Visente Del Boske  1 1 4 0 2 2 3 1 3 5  Spain  2  1  Antonio Conte  0 0 1 4 2 3 4 1 2 2  Italy  3  0  Claudio Ranieri  3 2 2 3 1 2 2 0 3 4  Leicester City FC  1  3  2  3  2  1 | ===== SITE EKIPI =====  Real Madrid CF  Zinedin Zidane  2075  FC Barcelona  Luis Enrique  6102  Spain  Visente Del Boske  166  Italy  Antonio Conte  207  Leicester City FC  Claudio Ranieri  1066  ===== DODADI GOLOVI =====  dodavam golovi: 3  dodavam golovi: 2  dodavam golovi: 3  dodavam golovi: 2  dodavam golovi: 1  ===== SITE EKIPI =====  Real Madrid CF  Zinedin Zidane  2078  FC Barcelona  Luis Enrique  6099  Spain  Visente Del Boske  172  Italy  Antonio Conte  213  Leicester City FC  Claudio Ranieri  1060  ===== NAJDOBAR TRENER =====  FC Barcelona  Luis Enrique  6099 | ===== SITE EKIPI =====  Real Madrid CF  Zinedin Zidane  2075  FC Barcelona  Luis Enrique  6102  Spain  Visente Del Boske  166  Italy  Antonio Conte  207  Leicester City FC  Claudio Ranieri  1066  ===== DODADI GOLOVI =====  dodavam golovi: 3  dodavam golovi: 2  dodavam golovi: 3  dodavam golovi: 2  dodavam golovi: 1  ===== SITE EKIPI =====  Real Madrid CF  Zinedin Zidane  2078  FC Barcelona  Luis Enrique  6099  Spain  Visente Del Boske  172  Italy  Antonio Conte  213  Leicester City FC  Claudio Ranieri  1060  ===== NAJDOBAR TRENER =====  FC Barcelona  Luis Enrique  6099 |  |
|  | 2  0  Luis Enrique  2 0 1 3 2 0 1 4 2 3  FC Barcelona  3  1  Joachim Low  2 0 1 3 2 0 1 4 2 3  Germany  150  5  2  79 91 80 100 88  France  3  115  97 | ===== SITE EKIPI =====  FC Barcelona  Luis Enrique  3054  Germany  Joachim Low  7554  ===== DODADI GOLOVI =====  dodavam golovi: 5  dodavam golovi: 2  ===== SITE EKIPI =====  FC Barcelona  Luis Enrique  3063  Germany  Joachim Low  7554  ===== NAJDOBAR TRENER =====  Germany  Joachim Low  7554 | ===== SITE EKIPI =====  FC Barcelona  Luis Enrique  3054  Germany  Joachim Low  7554  ===== DODADI GOLOVI =====  dodavam golovi: 5  dodavam golovi: 2  ===== SITE EKIPI =====  FC Barcelona  Luis Enrique  3063  Germany  Joachim Low  7554  ===== NAJDOBAR TRENER =====  Germany  Joachim Low  7554 |  |

Passed