Испит по Напреден развој на софтвер
(01.09.2012)
термин 1, група 1

<u>Задача 1 (50 поени) (Да се имплементира во C++)</u>

Целта на оваа задача е да се направи програма во која за даден **телефонски број** и за дадени направени **телефонски разговори** од него во даден месец ќе се креира **телефонска сметка** и истата ќе се прикаже на екран во следниот формат:

Telefonski broj: _XXX - XXXX_ Ime : _____
Broj na napraveni razgovori: _____ Vkupno vreme: ____ minuti _____ sekundi
Vkupno za naplata _____

Потребно е да се имплементираат следните класи за кои се чуваат дадените информации:

TelefonskiBroj:	**Razgovor:**	**TelefonskaSmetka:**
• Име на сопственик (динамичка алоцирана низа од знаци) • Префикс од најмногу 3 цифри (цел број) • Телефонски број (позитивен цел број)	• Телефонскиот број кој се повикува за разговор (објект од класата TelefonskiBroj) • Време на разговор изразено во секунди (цел број)	• Сопственик на телефонската сметка (објект од класата TelefonskiBroj) • Низа од разговори (низа од најмногу 20 објекти од класата Razgovor) • Број на направени разговори (цел број) • Претплата (цел број кој е константно 500 денари)

Во наведените три класи да се преоптоварат сору конструктор, подразбирлив конструктор, конструктор со аргументи и да се напише соодветен деструктор. По потреба може да се додадат set() и get() методи, како и други помошни функции.

Да се преоптовари операторот << за печатење во класите TelefonskiBroj и TelefonskaSmetka така што излезот за бројот да е во формат XXX – XXXX Ime: ___, а за сметката да соодветствува на она што е погоре прикажано.
Да се преоптовари операторот != во класата TelefonskiBroj. Два телефонски број ќе се смета дека се исти ако тие имаат ист префикс и ист телефонски број.

Покрај овие да се имплементираат и следните функции:
getMinuti() во Razgovor – Го враќа бројот на минути во еден разговор
presmetajVkupnaCena() во TelefonskaSmetka– Ја враќа цената на сметката зависно од низата направени разговори. На таа цена се додава претплатата. Притоа, тарифата по минута кон сите броеви е 7 денари.
getBrojNaRazgovoriKon(TelefonskaSmetka ts) во TelefonskiBroj – го враќа бројот на разговори направени од дадениот телефонски број со телефонскиот број од сметката која се предава како аргумент на функцијата.

Во функцијата main() :
- Да се креира објект naIvana за сметката на Ивана. Ивана има телефонски број 072 2343. Во конструкторот да се додадат 3 разговори кои ги направила Ивана со податоци по ваш избор.
- Да се отпечати сметката на Ивана со операторот <<.

<u>Задача 2 (50 поени) (Да се имплементира во Java)</u>

Една реченица е составена од низа зборови. Дел од зборовите може да бидат глаголи, именки или придавки. Согласно концептите од објектно ориентирано програмирање да се имплементираат следните класи заедно со сите get() и set() методи и конструктори со аргументи (податоците не се **public**) :

Zbor:	**Imenka:**	**Glagol:**	**Rechenica:**
• Текстуален запис (стринг) • Реден број во реченица (цел број)	• Текстуален запис (стринг) • Реден број во реченица (цел број) • Род (знак – m, z, s) • Дали е еднина (вистинита вредност)	• Текстуален запис (стринг) • Реден број во реченица (цел број) • Време (-1:минато 0: сегашност 1:иднина)	• Зборови во реченица (поле од зборови) • n (број на елементи)

За сите зборови се чува нивниот текстуален запис и нивниот реден број во дадена реченица. Именките се зборови кои може да имаат еднина и множина. Сите класи на зборови кои може да имаат и множина го имплементираат интерфејсот BrojliviZborovi кој ја има функцијата vratiDaliEdnina(). Таа враќа дали зборот е во еднина

Во сите класи да се имплементираат методите toString()и clone() . Во класата Rechenica да се овозможат следните функции:

- pecatiMinatoVreme() – се печатат сите глаголи во минато време во реченицата.
- odrediNajdolgZborSoBukva(char a) – го враќа најдолгиот збор кој ја содржи буквата зададена како аргумент.
- vratiPodredena() – се креира нова трансформирана реченица која се враќа со функцијата. (со примена на методата clone()). Во новата реченица зборовите се подредени по редниот број и тоа во растечки редослед. На прво место е првиот збор.

Ако бројот за времето на глаголите е невалиден да се фрли општ исклучок – објект од класата Exception. По потреба може да се додаваат функции.

Моделирањето на класите треба да овозможи дадената main() функција да се извршува без грешки.

Zbor[] zborovi= new Zbor[5]; zborovi[0]=new Imenka(“fakultet”,3,’m’,true); zborovi[1]=new Glagol(“sedeshe”,6,-1); zborovi[2]=new Glagol(“uci”,2,-2); zborovi[3]=new Zbor(“na”,1); zborovi[4]=new Glagol(“uceshe”,4,-1); Rechenica r=new Rechenica (zborovi); r. pecatiMinatoVreme (); System.out.println("Najdolgiot zbor so bukvata a e : “ + r. odrediNajdolgZborSoBukva ()); System.out.println(r. vratiPodredena ()); System.out.println(r);	sedeshe uceshe Najkratkiot zbor so bukvata a e : fakultet na uci faklutet uceshe sedeshe fakultet sedeshe uci na uceshe

Испит по Напреден развој на софтвер
(01.09.2012)
термин 1, група 2

<u>Задача 1 (50 поени) (Да се имплементира во C++)</u>
Целта на оваа задача е да се направи програма во која за даден **пејџер број** и за дадени пратени **пораки** од него во даден месец ќе се креира **пејџер сметка** и истата ќе се прикаже на екран во следниот формат:

Broj: _XXX.XXX.XXX_ Ime : _____
Broj na prateni poraki: _____ Vkupno bukvi: ____ samoglaski _____drugi bukvi
Vkupno za naplata _____

Потребно е да се имплементираат следните класи за кои се чуваат дадените информации :

PejgherBroj:	**Poraka:**	**PejgherSmetka:**
• Име на сопственик (поле од најмногу 15 знаци) • Префикс од најмногу 3 цифри (цел број) • Број во средината со 3 цифри (цел број) • Суфикс од најмногу 3 цифри (цел број)	• Број на пејџер кон кој се повикува праќа пораката (објект од класата PejgherBroj) • Текст на порака (поле од најмногу 50 знаци) • Број на букви во пораката (цел број)	• Сопственик на пејџер сметка (објект од класата PejgherBroj) • Низа од пораки (динамички алоцирана низа од објекти од класата Poraka) • Број на направени разговори (цел број) • Претплата (цел број кој е ист за сите)

Во наведените три класи да се преоптоварат сору конструктор, подразбирлив конструктор, конструктор со аргументи и да се напише соодветен деструктор. По потреба може да се додадат set() и get() методи, како и други помошни методи.

Да се преоптовари операторот << за печатење во класите PejgherBroj и PejgherSmetka така што излезот за бројот да е во формат XXX.XXX.XXX Ime:___, а за сметката да соодветствува на она што е погоре прикажано.
Да се преоптовари операторот == во класата PejgherBroj. Два пејџер броеви ќе се смета дека се исти ако тие имаат ист префикс и ист суфикс.

Покрај овие да се имплементираат и следните методи:
- getSamoglaski() во Poraka – го враќа бројот на самогласки во пораката.
- presmetajVkupnaCena() во PejgherSmetka– ја враќа цената на сметката зависно од низата пратени пораки. На таа цена се додава претплатата. За една порака се наплаќа 10 денари.
- daliImaPorakaKonBroj(PejgherSmetka ts) во PejgherBroj –враќа дали е пратена порака од дадениот пејџер до пејџер бројот од сметката која се предава како аргумент (префиксот и суфиксот да им е ист).

Во функцијата main() :
- Да се креира објект naAce за сметката на Аце, кој месечно плаќа 400 денари претплата. Аце има пејџер број 072.234.333. Во конструкторот да се додадат 3 пораки кои ги пратил Аце со податоци по ваш избор.
- Да се отпечати сметката на Аце со операторот <<.

<u>Задача 2 (50 поени) (Да се имплементира во Java)</u>

Една реченица е составена од низа зборови. Дел од зборовите може да бидат глаголи, лични заменки или придавки. Согласно концептите од објектно ориентирано програмирање да се имплементираат следните класи заедно со сите get() и set() методи и конструктори со аргументи (податоците не се **public**) :

Zbor:	**Glagol:**	**Pridavka:**	**Rechenica:**
• Текстуален запис (стринг) • Реден број во реченица (цел број)	• Текстуален запис (стринг) • Реден број во реченица (цел број) • Лице (цел број – 1, 2, 3)	• Текстуален запис (стринг) • Реден број во реченица (цел број) • Дали е еднина (вистинита вредност) • Род (знак – m, z, s)	• Зборови во реченица (поле од зборови) • n (број на елементи)

За сите зборови се чува нивниот текстуален запис и нивниот реден број во дадена реченица. Сите класи на зборови кои може да имаат и лице го имплементираат интерфејсот LiceviZborovi кој ја има функцијата vratiLice(). Таа враќа во кое лице е зборот

Во сите класи да се имплементираат методите toString() и clone(). Во класата Rechenica да се овозможат следните функции:

- pecatiMashkiRodVreme() – се печатат сите придавки во машки род во реченицата. .
- odrediNajkratokZborSoBukva(char a) – го враќа најкраткиот збор кој ја содржи буквата зададена како аргумент. .
- vratiPodredena() – се креира нова трансформирана реченица која се враќа со функцијата. (со примена на методата clone()). Во новата реченица зборовите се поредени по редниот број и тоа по опаѓачки редослед. На прво место е последниот збор.

Ако знакот за родот на придавките е невалиден да се фрли општ исклучок – објект од класата Exception. По потреба може да се додаваат функции. .

Моделирањето на класите треба да овозможи дадената main() функција да се извршува без грешки.

Zbor[] zborovi= new Zbor[5]; zborovi[0]=new Pridavka(“ubav”,3,true,’m’); zborovi[1]=new Glagol(“sedeshe”,6,3); zborovi[2]=new Glagol(“ucam”,2,1); zborovi[3]=new Zbor(“na”,1); zborovi[4]=new Glavol(“ucish”,4,2); Rechenica r=new Rechenica (zborovi); r. pecatiMinatoVreme (); System.out.println("Najkratok zbor so bukvata a e : “ + r. odrediNajkratokZborSoBukva (‘a’)); System.out.println(r.vratiPodredena ()); System.out.println(r);	Ubav Najkratkiot zbor so bukvata a e : na sedeshe ucish ubav ucam na ubav sedeshe ucam na ucish

Испит по Напреден развој на софтвер
(01.09.2012)
термин 2, група 1

<u>Задача 1 (50 поени) (Да се имплементира во C++)</u>

Потребно е да се имплементираат следните класи за кои се чуваат дадените информации :

Avtor:	**Kniga:**	**Polica:**
• Име на автор (динамичка алоцирана низа од знаци) • Број на напишани книги (цел број)	• Автор на книгата (објект од класата Avtor) • Број на страници (цел број) • Маса (децимален број)	• Низа од книги (динамички алоцирана низа од покажувачи кон објекти од класата Kniga) • Број на книги (цел број) • Максимална носивост на полицата (децимален број кој е константно 23,4кг)

Во наведените три класи да се преоптоварат copy конструктор, подразбирлив конструктор, конструктор со аргументи и да се напише соодветен деструктор. По потреба може да се додадат set() и get() методи, како и други помошни методи.

Да се преоптовари операторот >> за вчитување во класите Avtor и Kniga така што за внесување на една книга да се добива следниот изглед: .
Vnesete podatoci za knigata:
Vnesi avtor: _____
Vnesi broj napishani knigi: ___
Vnesi broj na stranici:_____
Vnesi masa: _____

Да се преоптовари операторот += за додавање на книга на полицата. .

Покрај овие да се имплементираат и следните методи:

- dodadiNaBrojNapishani(int br) во Avtor – на бројот на напишани книги се додава вредноста на аргументот, но само ако аргументот има позитивна вредност.
- daliMozeDaSeDodadeKniga(Kniga k) во Polica – враќа дали може да се додаде книгата зададена како аргумент на полицата. Таа може да се додаде ако со таа книга вкупната маса нема да ја надмине носивоста на полицата.
- getNajdolgaKniga() во Polica – ја враќа последната книга која има најмногу страници. .

Во функцијата main() :

- Да се креира објект policaMala за една полица. Да се креираат три објекти од класата Kniga и да се иницијализираат со операторот >>.
- Да се додадат на полицата првите 2 креирани книги со операторот +=. .
- На авторот на првата книга да се додадат уште 3 напишани книги (dodadiNaBrojNapishani).
- Да се провери дали може третата книга да се додаде на полицата (daliMozeDaSeDodadeKniga).

<u>Задача 2 (50 поени) (Да се имплементира во Java)</u>

Една комбинација на лото е составена од низа броеви. Броевите се или цели или децимални. Согласно концептите од објектно ориентирано програмирање да се имплементираат следните класи заедно со сите get() и set() методи и конструктори со аргументи (податоците не се **public**) .:

Broj:	**DecimalenBroj:**	**CelBroj:**	**LotoKombinacija:**
• Боја (стринг) • Реден број во комбинацијата (цел број)	• Боја (стринг) • Реден број во комбинацијата (цел број) • Вредност (децимален број)	• Боја (стринг) • Реден број во комбинацијата (цел број) • Вредност (цел број)	• Броеви во комбинација (поле од броеви) • n (број на елементи)

За сите броеви се чува нивната боја и нивниот реден број во дадена комбинација. Целите броеви можат да бидат парни или непарни. Сите класи кои можат да бидат парни или непарни го имплементираат интерфејсот vrednostiSoParnost кој ја има функцијата vratiDaliParen(). Таа враќа дали бројот е парен. .

Во сите класи да се имплементираат методите toString() и clone() . Во класата LotoKombinacija да се овозможат следните функции:

- pecatiPozitivniDecimalniVreme() – се печатат сите позитивни децимални броеви. .
- odrediSuma(int a,String s) –ја враќа сумата на броевите од комбинацијата кои имаат иста боја како аргументот и имаат реден број помал од аргументот. .
- vratiTransformirana() – се креира нова трансформирана комбинација која се враќа со функцијата. (со примена на методата clone()). Во новата комбинација редниот број на најголемиот децимален број треба да се замени со редниот број на најмалиот цел број.

Ако редниот број е негативен да се фрли општ исклучок – објект од класата Exception. По потреба може да се додаваат функции. .

Моделирањето на класите треба да овозможи дадената main() функција да се извршува без грешки.

Broj[] broevi= new Broj[5]; broevi [0]=new DecimalenBroj(“sina”,-1,34.2); broevi [1]=new DecimalenBroj(“crvena”,2,-12.3); broevi [2]=new CelBroj(“crvena”,3,-2); broevi [3]=new CelBroj(“crvena”,4,92); broevi [4]=new CelBroj(“sina”,5,11); LotoKombinacija l=new LotoKombinacija (broevi); l. pecatiPozitivniDecimalniVreme (); System.out.println("Sumata za crvena boja i broj 5 e : “ + r. odrediSuma (“crvena”,5)); System.out.println(r. vratiTransformirana ()); System.out.println(r);	34.2 Sumata za crvena boja i broj 5 e : 77.7 **3**:34.2 2: -12.3 **1**: -2 4: 92 5: 11 1:34.2 2: -12.3 3: -2 4: 92 5: 11

Испит по Напреден развој на софтвер
(01.09.2012)
термин 2, група 2

<u>Задача 1 (50 поени) (Да се имплементира во C++)</u>
Потребно е да се имплементираат следните класи за кои се чуваат дадените информации .:

Kompozitor:	**Pesna:**	**CD:**
• Име на композитор (низа од најмногу 30 знаци) • Број на напишани песни (цел број)	• Име на композитор (објект од класата Kompozitor) • Број на секунди (цел број) • Мемориска зафатеност во KB (децимален број)	• Низа од песни (динамички алоцирана низа од покажувачи кон објекти од класата Pesna) • Број на песни (цел број) • Максимален мемориски капацитет (децимален број кој е ист за сите)

Во наведените три класи да се преоптоварат сору конструктор, подразбирлив конструктор, конструктор со аргументи и да се напише соодветен деструктор. По потреба може да се додадат set() и get() методи, како и други помошни методи ..

Да се преоптовари операторот >> за вчитување во класите Kompozitor и Pesna така што за внесување на една книга да се добива следниот изглед: .
Vnesete podatoci za pesnata:
Vnesi kompozitor: _____
Vnesi broj napishani pesni: ___
Vnesi broj na sekundi:_____
Vnesi memoriska zafatenost: _____

Да се преоптовари операторот += за додавање на песна на полицата. .

Покрај овие да се имплементираат и следните методи:
- odzemiNaBrojNapishani(int br) во Kompozitor – на бројот на напишани песни се одзема вредноста на аргументот. Ако се добие негативна вредност на бројот на песни, тој се поставува на 0.
- daliMozeDaSeDodadePesna(Pesna k) во CD – враќа дали може да се додаде песната зададена како аргумент на CD-то. Таа може да се додаде ако со таа песна вкупната мемориска зафатеност нема да го надмине максималниот мемориски капацитет.
- getNajkratkaPesna() во CD – ја враќа последната песна која има трае најмалку секунди.

Во функцијата main() :
- Да се креира објект cdGolemo за едно CD од марка со капацитет 4323KB. Да се креираат три објекти од класата Pesna и да се иницијализираат со операторот >>.
- Да се додадат во CD-то првите 2 креирани песни со операторот +=. .
- На композиторот на првата песна да се одземат уште 2 напишани песни (odzemiNaBrojNapishani).
- Да се провери дали може третата песна да се додаде на CD-то (daliMozeDaSeDodadePesna).

<u>Задача 2 (50 поени) (Да се имплементира во Java)</u>

Една комбинација на лото е составена од низа броеви. Броевите се или цели или децимални. Согласно концептите од објектно ориентирано програмирање да се имплементираат следните класи заедно со сите get() и set() методи и конструктори со аргументи (податоците не се **public**) .:

Broj:	**DecimalenBroj:**	**CelBroj:**	**LotoKombinacija:**
• Боја (стринг) • Реден број во комбинацијата (цел број)	• Боја (стринг) • Реден број во комбинацијата (цел број) • Вредност (децимален број)	• Боја (стринг) • Реден број во комбинацијата (цел број) • Вредност (цел број)	• Броеви во комбинација (поле од броеви) • n (број на елементи)

За сите броеви се чува нивната боја и нивниот реден број во дадена комбинација. Целите и децималните броеви можат да бидат позитивни или негативни. Тие класи го имплементираат интерфејсот vrednostiSoZnak кој ја има функцијата vratiApsolutnaVrednost(). Таа ја враќа апсолутната вредност на бројот. .

Во сите класи да се имплементираат методите toString() и clone(). Во класата LotoKombinacija да се овозможат следните функции:

- pecatiPozitivniCeli() – се печатат сите позитивни цели броеви. .
- odrediSuma(int a,String s) –ја враќа сумата по апсолутна вредност на броевите од комбинацијата кои имаат иста боја како аргументот и имаат реден број поголем од аргументот. .
- vratiTransformirana() – се креира нова трансформирана комбинација која се враќа со функцијата. (со примена на методата clone()). Во новата комбинација редниот број на најмалиот децимален број треба да се замени со редниот број на најголемиот цел број. .

Ако редниот број е негативен да се фрли општ исклучок – објект од класата Exception. По потреба може да се додаваат функции. .

Моделирањето на класите треба да овозможи дадената main() функција да се извршува без грешки.

Broj[] broevi= new Broj[5]; broevi [0]=new DecimalenBroj(“sina”,-1,34.2); broevi [1]=new DecimalenBroj(“crvena”,2,-12.3); broevi [2]=new CelBroj(“crvena”,3,-2); broevi [3]=new CelBroj(“crvena”,4,92); broevi [4]=new CelBroj(“sina”,5,11); LotoKombinacija l=new LotoKombinacija (broevi); l. pecatiPozitivniDecimalniVreme (); System.out.println("Sumata za crvena boja i broj 5 e : “ + r. odrediSuma (“crvena”,2)); System.out.println(r. vratiTransformirana ()); System.out.println(r);	92 11 Sumata za crvena boja i broj 2 e : 94 1:34.2 **4**: -12.3 1: -2 **2**: 92 5: 11 1:34.2 2: -12.3 3: -2 4: 92 5: 11