

Contents

Requirements:.....	1
Task:	2
Example usage:	4
Reccomendations and tips:.....	5

Requirements:

- Do the exam yourself. If you cant do everything then simplify it and do only the things you can but do it yourself. Minimum requirements for passing:
 - At least half of the methods from Basic Mobile are working properly.
 - There is one derived class which follows class inheritance rules.
 - There are a few tests.
 - Few objects are created in main method.
- Plan enough time for doing the exam and avoid [student syndrome](#), do not start 1-2 days before your defending time, its not enough. If you get stuck with something then its better to let the brain 'rest' and continue the next day.
- Follow clean code principles. Make the code also look good (remove unwanted empty lines, unused variables, align braces etc).
- Write unit tests for methods where you think it is neccessary. Minimum number of tests: 12. Reccomendation: test the methods that calculate smth or find a value etc.
- Avoid code duplication. Use small(er) helper methods.
- Add an interface. Think which methods should be public and which not.
- Method should not be longer than 20-30 lines. There can be exceptions but with a good reason (long case-swith sentences for example).
- In main method create all 3 phones and test different (calling, sending sms etc) methods and usage scenarios!
- Create a separate file for each class.
- Think where an exception could happen and add (at least) one try-catch block in your code.
- Think of the situation when data is not valid; what to check and when and what warnings to display.
- Project type: console application.

Task:

You are hired by a company to create software that helps to keep track of phones call register (*kõneregister*) and print info about cost for a period. Period ends when data is reset (data is reset with reset method).

What is common for all phones:

- Calls can be made, sms'es can be sent and calls can be answered only from valid phone numbers. Think of the conditions (what make a phone number to be valid or not) yourself. If number is not valid then display a message.
- Calling emergency number 112 is always free. When calling to that number then also display a notification: „You are calling to an emergency number.“
- All phones have a unique serial number with format:
company code, 2 last numbers of the year of the creation and up to 4 digits which show what was the number of the phone created.
Example: First JazzPhone created this year will have a code: JZ180001, second one JZ180002, third one JZ180003, hundreth JZ180100 etc. Save used codes to a file! When creating a new code continue the numers from the file.
Generation of the serial number should done with creation of the phone and this value cannot be changed later.

Basic Mobile phone

Basic phone has methods:

- 1) Method for adding favourite phonenumber. Different phones can save a different amount of favourite numbers. If max number is reached then display a notification „Max amount of favourite numbers reached, cannot add more“. All favourite numbers have to be unique; if user tries to add the same number twice then display a warning. Favourite numbers are reset (cleared) with reset method.
Example: AddFavNumber(12345) -> „12345 was added to favourite numbers“.
AddFavNumber(1234) -> „12345 is already in favourite numbers, cant add it twice“.
- 2) Method for calling to other phones. This method takes 2 parameters: number to call to and call duration in seconds. Method prints call info.
Example: Call(1234, 40) - > „Outgoing call to 1234 with duration 40“
 - Add a second call method which takes index from favourites list and duration as parameters. If there is no such item present then display a warning.
Example: Call(0, 40) - > „Outgoing call to 12345 with duration 40“
Example: Call(5, 40) - > „There is no contact with index 5 in favourites.“
- 3) Method for answering an incoming call. This method takes 2 parameters: number that the call is coming from and call duration in seconds. Method prints call's info.
Example: Answer(1234, 40) - > „Incoming call from number 1234 with duration 40“
- 4) Method for sending an sms. This method takes 2 parameters: number to send the sms to and the text of the sms. Method prints: „Sending an sms to number: ...“ + 15 first characters of the

text. Standard length for one sms is 160 characters. Keep this in mind when calculating price. If text's length is 168 characters then this counts as 2 sms-es.

Example: SendSms(1234, „Kuule osta palun piima ka“) -> „Sending sms „Kuule osta palu..“ to number 1234.

- Add a second call method which takes index from favourites list and text of the sms as parameters.
- 5) Method for displaying a call register. This method prints:
- a. Total number of incoming calls and their duration and cost (if there is any).
 - b. Total number of outgoing calls and their duration cost.
 - c. Total number of sms-es sent and their cost.
 - d. Total cost for a period of time. (Period ends with reset method).

Duration has to be in minutes and cost in euros.

Add some formatting (line breaks etc) so that it would be easy to view/read the data.

- 6) Method for finding and printing the most frequently contacted number (contacting: answering a call, sending a sms or calling).
- 7) Method for printing phone info (name, serial number and all favourite numbers).
- 8) Method for resetting data: all info about incoming and outgoing calls and sent sms-es is deleted. This means that one period ends and next one starts.

Calling abroad and country codes.

Some phones allow calling abroad and answering calls from other countries. Answering a call from another country costs money (answering calls within Estonia is free)!

If calling abroad is possible, then methods 1-3 also print the country's name. Country codes are stored in a file „countryCodes.txt“ (you can find this file from ained.ttu.ee). Based on the country code from that file find country's name. If an incoming call is coming from abroad then display a warning.

Example:

Answer(„+35812345“, 60) -> „Warning! Call from: Finland. Incoming call from number +35812345 with duration 60 seconds“

Call(„+35812345“, 60“) -> „Destination: Finland. Outgoing call to +35812345 with duration 60“.

When call register is displayed (4th method from basic phone) then foreign calls should be displayed as a separate category.

3 Different phones:

Different phones have different prices and options. JazzPhone is basic phone. OrangeCalls and PhoneBlazt also allow foreign calls.

With orange phone if you send less than 5 smses in a period then price for all sms-es is 0. If you send more than 5 then price for all smses is 0.6 cents (If you send 8 sms-es then total price for sms is $8 \cdot 0.6 = 4.8$ euros).

	JazzPhone	OrgangeCalls	PhoneBlazt
Minute price for domestic call	0.5 cents	<100 minutes: 0.9 cents >= 100 minutes: 0.3 cents	0.3 cents
Price for sms	< 8 sms: 50 cents >=9 sms: 40 cents	< 5 sms: free >=6 sms: 0.6 cents	< 10 sms: 0.4 cents >=11 sms: 0.2 cents
Minute price for foreign call minute (both incoming and outgoing)	Foreign calls not supported	0.9	1.1
Company code	JP	OC	PB
Max amount of fav numbers	3	5	unlimited

Example usage:

Here is an example for calling each basic phone method (1-8).

JazzPhone jazzPhone = new **JazzPhone**(); creates phone and checks serials from file. If there are no previous phones created then creates serial JZ180001 and adds it to serials file.

1. jazzPhone.AddFavouriteNr("+37256241779"); -> „Added „+37256241779 to favourites“ adds this number to favourites
2. jazzPhone.Call("+3725612345", 60);-> „Outgoing call to +3725612345 with duration 60 seconds“
jazzPhone.Call("tere", 60);-> „Cannot call, invalid number!“
jazzPhone.Call(0, 60); -> „Outgoing call to +37256241779 with duration 60 seconds“
3. jazzPhone.Answer("+3725612345", 90); -> „Incoming call from number +3725612345 with duration 90 seconds“
4. jazzPhone.SendSms("+3725612345", "Jou mis teed");-> „Sending sms „Jou mis teed“ to number +3725612345“
jazzPhone.SendSms(0, "See on nüüd nii pikk tekst, et maksma peab kahe sõnumi hinna. Aga natuke peab tekst veel pikem olema, et saaks ikka 160 tähemärki täis. Aga kohe on 165 tähte koos."); -> „Sending sms „See on nüüd nii“ to number +37256241779“
5. jazzPhone.DisplayData();->
„Number of incoming calls is 1 with duration 1.5 minutes. “
„Number of outgoing calls is 2 with duration 2 minutes and cost 0.6 euros. “
„Number of smses sent is is 2 and cost is 1 euro. “
„Total cost is: 1.6 euros“
6. jazzPhone.FindMostFrequentContact();-> „Most frequently contacted number is +3725612345: 3 times“.
7. jazzPhone.PrintPhoneInfo();-> „Model is Jazz Phone, serial is JZ180001. Favourite numbers are: +37256241779“
8. jazzPhone.ResetData(); -> „Data is reset!“

```
jazzPhone.DisplayData();->
„Number of incoming calls is 0 with duration 0 minutes. “
„Number of outgoing calls is 0 with duration 0 minutes and cost 0 euros.“
„Number of smses sent is is 0 and cost is 0 euro.“
„Total cost is: 0 euros“
```

Reccomendations and tips:

- Reading data from file is memory consuming operation and should be done as few times as possible. Reading in the country code values should be done only once!
- Think of the sctructure of the country codes file. Try to find some rules that every line in the file follows and read in the values according to that. Think what is best data type for storing the values. (Every country code has one corresponding country name).
- Finding the number that is contacted the most frequently: one option is to use a dictionary where: *phone number* is the key and *number of times contacted* is the value. If phone number is already present in dictionary then increase the times it has been contacted. If not then add it to dictionary with value 1 (because it is the first time its contacted).
- You can choose the data types to use yourself. Choosing correct data types makes everything much easier. Think what you need to do with a variable and then think of a suiting data type.
- Method texts and method names do not need to look exactly the same as in the examples here; they should be similar but you can use different text and format.
- Use loops where possible (instead of copy-pasting the same row x times).
- Private/protected methods cannot be tested with nUnit. Internal methods can.