

CHALMERS

EXAMINATION / TENTAMEN

Urkod	Course name/kursnamn		
3	<i>Object-oriented programming</i>		
	Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
T043-0041-JR0	23-10-23	12	5

o mobile or other similar electronic equipment available during the examination.
e har mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under

CHALMERS	Anonymous code Anonym kod DITO43-0041-JRO	Points for question (to be filled in by teacher) Poäng på uppgiften (ifylltes av lärare)	Consecutive page no. Löpande sid nr Question no. Uppgift nr
			1
			1
1.	$x = 17 \quad y = 8 \quad z = 10$	/	(+13)
2.	$x = 6 \quad y = 1 \quad z = 0$	/	
3.	$x = 6 \quad y = 3 \quad z = 4$	/	
4.	$x = 6 \quad y = 8 \quad z = 10$	/	
5.	$p_1 = 3 \quad p_2 = 13$	X	
6.	$x = 16 \quad y = 1 \quad z = 9$	/	
7.	$x = 16 \quad y = 8 \quad z = 10$	/	
8.	$x = 16 \quad y = 1 \quad z = 9$	$x = 16 \quad y = 8 \quad z = 10$	/
9.	$x = 16 \quad y = 1 \quad z = 9$	/	
10.	$x = 16 \quad y = 1 \quad z = 9$	/	
11.	$p_1 = 17 \quad p_2 = 21$	X	
12.	$x = 1 \quad y = 1 \quad z = 9$	/	
13.	$x = 1 \quad y = 8 \quad z = 1$	/	
14.	$x = 1 \quad y = 8 \quad z = 10$	/	
15.	$p_1 = 7 \quad p_2 = 13$	/	
16.	(empty line after the last println)		

CHALMERS	Anonymous code Anonym kod	Points for question (to be filled in by teacher) Poäng på uppgiften (ifylls av lärare)	Consecutive page no. Löpande sid nr Question no. Uppgift nr
			2 1

NOT FOR GRADING! Scratch.

LINES.

MEMORY

$$\boxed{1} \quad y =$$

$$z =$$

$$\boxed{2} \quad y =$$

$$z =$$

$$\boxed{3} \quad y =$$

$$z =$$

$$\boxed{P} \quad x =$$

$$39 \ 35 \ 37 \ 40 \ 42 \ 43 \ 45 \ 152 \ 66 \ 81$$

$$1 \ 8 \ 0 \ 3 \ 4 \ 1 \ 8 \ 9 \ 10 \ 8$$

$$P_2 \ 8 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1$$

$$P_1 \ 9 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1$$

$$P_1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1$$

$$P_1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1$$

$$P_1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1$$

$$P_1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1$$

REFERENCES

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$\boxed{42} P_1; P_3$

$\boxed{43} (P_2; B)$

$\frac{m \times C_{nT}(x, y)}{x \times \%2 == 0}$

$$1. Q_3 + Q_2 - Q_4 - Q_8 \quad x = 17 \quad y = 8 \quad z = 10 \quad (l_{38}; P_2)$$

$$2. x = 6 \quad y = 1 \quad z = 0 \quad (l_{38}; P_2)$$

$$3. x = 6 \quad y = 3 \quad z = 9 \quad (l_{38}; P_2)$$

$$4. x = 6 \quad y = 8 \quad z = 10 \quad (l_{38}; P_2)$$

$$5. P_1 = 3 \quad P_2 = 13 \quad (l_{38}; P_2)$$

$$6. x = 16 \quad y = 1 \quad z = 9 \quad (l_{38}; P_2)$$

$$7. x = 16 \quad y = 8 \quad z = 10 \quad (l_{38}; P_2)$$

$$8. x = 16 \quad y = 8 \quad z = 10 \quad (l_{38}; P_2)$$

$$9. x = 16 \quad y = 1 \quad z = 9 \quad (l_{38}; P_2)$$

$$10. x = 16 \quad y = 1 \quad z = 9 \quad (l_{38}; P_2)$$

$$11. P_1 = 17 \quad P_2 = 27 \quad (l_{38}; P_2)$$

$$12. x = 1 \quad y = 1 \quad z = 9 \quad (l_{38}; P_2)$$

$$13. x = 1 \quad y = 8 \quad z = 1 \quad (l_{38}; P_2)$$

$$14. x = 1 \quad y = 8 \quad z = 10 \quad (l_{38}; P_2)$$

$$15. P_1 = 7 \quad P_2 = 14 \quad (l_{38}; P_2)$$

CHALMERS	Anonymous code Anonym kod DITO43-0041-JRO	Points for question (to be filled in by teacher) Poäng på uppgiften (fylls av lärare)	Consecutive page no. Löpande sid nr 3
			Question no. Uppgift nr 2

2.1.

+6

```

public boolean equals(Object obj){
    if(obj==null) {
        return false;
    } else if(obj==this) {
        return true;
    } else if(obj instanceof Player) {
        Player player = (Player) obj;
        if(player.username.equals(this.username))
        {
            return true; } } else {
        return false;
    }
}

```

+1
Could have been simplified

2.2.

The class is not well encapsulated. Even though all the attributes are set to be private, we have getters for all of them and setters for all non-final. Also, setters does not verify values of the parameters and just assign them to attributes. This leads to the same issue, if the (context page)

CHALMERS	Anonymous code Anonym kod DITO43-0041-JRC	Points for question (to be filled in by teacher) Poäng på uppgiften (fylltes av lärare)	Consecutive page no. Löpande sid nr Question no. Uppgift nr
			4 2

2.2. (cont)

+8p

attributes were all public: the values inside them can be changed by anyone in the program, as well as the values assigned can be invalid (such as blank for name or negatives for wins or losses). Also, the should be no setter for wins and losses, as we have strict rules where they change automatically and only by 1.

2.3.

→ Great answer!

Suggestion 1) Improve constructor.

* Right now, the constructor in the class is very vulnerable to bugs. Victories and losses are assigned there, even though they should start at 0; role is not being checked if it is one of the 3 existing roles, and despite the .is Empty() check, the string values can still be blank (just a couple of spaces, which will not be detected by .is Empty()). To improve fix this problem, I propose these:

- write a custom function to check if the String value is empty or blank or null.
- stop taking victories and losses (cont next page)

CHALMERS	Anonymous code Anonym kod DIT043-0041-JRC	Points for question (to be filled in by teacher) Poäng på uppgiften (ifylls av lärare)	Consecutive page no. Löpande sid nr Question no. Uppgift nr
----------	---	---	--

5

2

as parameters, and just assign both of them to 0 in the constructor.

- change role attribute to an enum, and check in the constructor (and everywhere else, where the role is assigned as a String) if this string value is a value of enum, and then assign an appropriate value from enum.

Suggestion 2) toString() method.

Right now, Player class uses standard default Object toString method, the return value of which is not following specification. To fix it, we need to override it with a new toString method, which returns specified String output (`V: <victories>, L: <losses>, Role: <role>`).

Q.4

2.4 on the next page

+15p

CHALMERS	Anonymous code Anonym kod DITO43-0041-JRO	Points for question (to be filled in by teacher) Poäng på uppgiften (ifylls av lärare)	Consecutive page no. Löpande sid nr Question no. Uppgift nr
			6 2

2.4.

```

public class RankedPlayer extends Player {
    private double rate; +1
    private String rank; -3 not needed throws Exception
    public Rank(String username,
                String name, String role) {
        super(username, name, role);
        this.rank = "Noob rank";
        this.rate = 0.0; +1
    }
    public String getRank() {
        return this.rank;
    }
    public double getRate() {
        return this.rate;
    }
    public void updRate() { +3
        How? If those are in the superclass?
        // call this method, whenever which number
        // of victories or losses are changed
        // in the methods of superclass
        int totalGames = this.getVictories()
                      + this.getLosses();
        int wins = this.getVictories();
        rate = (double) wins / totalGames;
    }
}

```

(cont next page)

CHALMERS	Anonymous code Anonym kod DITO43-0041-JRO	Points for question (to be filled in by teacher) Poäng på uppgiften (fylls av lärare)	Consecutive page no. Löpande sid nr Question no. Uppgift nr
			7 8

2.4. (cont)

```

// truncation
rate = rate * 100;
int intRate = (int) rate;
rate = (double) intRate / 100.0;
if (rate >= 0.95) {
    rank = "Diamond rank";
} else if (rate >= 0.7) {
    rank = "Gold rank";
} else if (rate >= 0.5) {
    rank = "Silver rank";
} else if (rate >= 0.2) {
    rank = "Bronze rank";
} else {
    rank = "Noob rank";
}

```

→ Code was unnecessary complicated.

→ Some good parts though.

{

{

```

public void resetRate() {
    // call this method, whenever number of
    // wins or losses is not incremented,
    // But set to a new value or reset to 0.
    int totalGames = this.getVictories()
        + this.getLosses();
    if (totalGames == 0) {
        rank = "Noob rank";
        rate = 0.0;
    } else

```

(cont next page)

updRate()

{

{

public String toString(){

return super.toString() +

"Rank: " + rank;

different.

{

{

CHALMERS	Anonymous code Anonym kod	Points for question (to be filled in by teacher) Poäng på uppgiften (fylls av lärare)	Consecutive page no. Löpande sid nr Question no. Uppgift nr
	DITO43-0041-JRO		9 3

3.1.

The LeaderBoard class is well encapsulated. All its attributes are private, it does not implement any setters for any of the attributes of itself or related classes, otherwise implementing special methods, that correlates to specification (such as addPlayer, addWin, addLoss). Even though it does not allow to create a different type of player (depending on the implementation of add Player either Ranked Player or just Player), it is not a problem of encapsulation. But the specification of leaderboard.

3.4.

To allow LeaderBoard objects to be stored in files, the class itself, Player class, and RankedPlayer class (if it can be add to the LeaderBoard) must all implement a special interface Serializable, that shows, that the object of these classes can be stored in files.

CHALMERS	Anonymous code Anonym kod DITO43-0041-JRO	Points for question (to be filled in by teacher) Poäng på uppgiften (fylltes av lärare)	Consecutive page no. Löpande sid nr Question no. Uppgift nr 3
----------	---	--	---

3.2

```

public Map<String, int> getTableOfRoles() {
    Map<String, int> roles = new Map<String,
        int>();
    for( Player player : leaderboards) {
        if(
            String role = player.getRole();
            if(!roles.contains(role)) {
                roles.add(role, 1);
            } else {
                roles.set(role) = roles.get(role)+1;
                roles.set(role, roles.get(role)+1);
            }
        }
        return roles;
    }
}

```

for this and the next question I assume that my suggestion in task 1 for turning roles to type of an Enum was rejected.

CHALMERS	Anonymous code Anonym kod	Points for question (to be filled in by teacher) Poäng på uppgiften (fylls av lärare)	Consecutive page no. Löpande sid nr
	DIT043-C041-JRO		11 3

3-3.

(x7)

Leaderboard	
- LeaderBoard : ArrayList<Player>	
+ Leaderboards()	+ 1
+ add Player(String username, String name, String role): void	(throws Exception)
+ remove Player(String username): boolean	
+ add Win(String username): void	
+ add Loss(String username): void	
+ getTableOfRoles(): Map<String, int>	
+ getWins(String username): int	
+ getLosses(String username): int	

Player	
- username: String (final)	
- name: String	
- victories: int	
- losses: int	+ 1
- role: String	
+ Player(String username, String name, String Role) (throws Exception)	
+ getUsername(): String	
+ getName(): String	
+ getVictories: int	
+ getLosses: int	
+ getRole: String	
+ setName: void	
+ setVictories: void	
+ setLosses: void	
+ getRole: void	

Ranked Player	
- rate: double	
- rank: String	
+ RankedPlayer(String username, String name, String Role)	(throws Exception)
+ updRate(): void	
+ resetRate(): void	
+ toString: String	

Leaderboard and Player classes form a composition, with

Leaderboard being a

collection of + 2 components, and Players' component. It allows us to access and regulate the access of other parts of program to this component.

+ equals(Object obj): Boolean
+ toString(): String

Inheriting RankedPlayer

from Player allows us to reuse similar methods in both classes, with adding new methods to subclass and overriding some. Also, it potentially allows us to use polymorphism in Leaderboard class. + 5

CHALMERS	Anonymous code	Points for question (to be filled in by teacher)	Consecutive page no. Löpande sid nr
	Anonym kod DIT043-0041-JRO	Poäng på uppgiften (ifylls av lärare)	Question no. Uppgift nr 4

```

public static String compress(String origStr) {
    String newStr = "";
    char curChar = origStr.charAt(0);
    int cnt = 1;
    for(int i=1; i<origStr.length(); i++){
        if(origStr.charAt(i) == curChar){
            cnt++;
        } else{
            newStr = newStr + curChar + cnt;
            curChar = origStr.charAt(i);
            cnt = 1;
        }
    }
    newStr = newStr + curChar + cnt;
    if(newStr.length() < origStr.length()){
        return newStr;
    } else{
        return origStr;
    }
}

```