

# תוכנה 1 – אביב תשע"ד

## תרגיל מספר 9

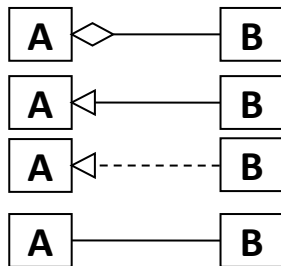
### הורשה

#### הנחיות כלליות:

קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.

- הגשת התרגיל תעשה במערכת ה-moodle בלבד (<http://moodle.tau.ac.il/>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש aviv יקרא הקובץ aviv\_hw9.zip). קובץ ה-zip יכיל:
  - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
  - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש, כולל תיקיות החבילה.
  - ג. קובץ PDF בשם answers.pdf המכיל את התשובות לשאלות.

**הערה כללית:** בתרגיל זה אתם מתבקשים, בין היתר, לשרטט דיאגרמות של מחלקות. השתמשו בסימונים הבאים בלבד:



- **aggregation** (יחס של הכלה (למשל, ל-A יש שדה מטיפוס B))
- **ירשה** (B מחלקה הירשת את A):
- **מימוש** (B מחלקה המממשת מנשק הירש את המנשק A):
- **association** (קשר כללי שאינו נופל בקטגוריות הקודמות. למשל, A משתמש במשתנה מטיפוס B באחת המתודות).

**יש לציין:** בתוך כל מלבן <<interface>>, <<abstract>> או <<class>>, ואת שם המנשק או המחלקה.

**אין צורך לציין:** מספרים ושמות שדות על יחסי אגרגציה ואסוציאציה; שמות מתודות ושדות בתוך מלבני המחלקות; יחסים "עקיפים" בין מחלקות (כלומר, אם C יורש מ-B שירש מ-A, אין צורך לציין קשר בין C ל-A אלא אם יש ביניהם קשר ישיר בנוסף, למשל של הכלה)

**יצירת הדיאגרמות:** ניתן לעשות זאת דרך Word, PowerPoint, תוכנת הציור המועדפת עליכם או לסרוק שרטוט (בכתב ברור!).



# Starfleet Command

בתרגיל זה נבנה מערכת תוכנה לניהול צי חלליות עתידיני.

בתחילה נבנה מחלקות שייצגו את אנשי הצוות ואת סוגי החלליות השונים תוך שימוש במנשקים, מחלקות אבסטרקטיות והורשה. לאחר מכן, ניצור אובייקטים של מחלקות אלו ולבסוף נדפיס מספר דוחות המציגים חיתוכי מידע שונים על צי החלליות שלנו כגון עלות אחזקה כוללת, כוח-אש כולל של חלליות הצי ועוד.

## Starfleet Personnel

צי החלל של פדרציית הכוכבים המאוחדת כולל 2 סוגי אנשי צוות (Crew-Members):

1. Crewman – איש צוות רגיל.
2. Officer – איש צוות שהינו קצין (בעל דרגת קצונה).

הנה תיאור השירותים בהם תתמוך כל אחת מהמחלקות המייצגות את סוגי אנשי הצוות הנ"ל:

### Crewman

שם השירות	טיפוס החזרה	הסבר
<code>getName()</code>	String	שם איש הצוות (מחרוזת המציינת שם ייחודי לכל איש צוות).
<code>getAge()</code>	int	גילו של איש הצוות (בשנות כדור-הארץ, למשל 28).
<code>getYearsInService()</code>	int	מספר שנות השירות של איש הצוות (בשנות כדור-הארץ, למשל 10).

### Officer

איש צוות שהינו קצין יכלול את כל התכונות של איש צוות רגיל, ובנוסף תהיה לו גם דרגת קצונה:

שם השירות	טיפוס החזרה	הסבר
<code>getRank()</code>	OfficeRank	הדרגה של הקצין (מיוצג ע"י Enum בשם OfficerRank).

שימו לב:

- בהמשך נגדיר מנשק בשם `CrewMember` אשר ייצג איש צוות מסוג כלשהו.
- הטיפוס `OfficerRank` הוא Enum המגדיר קבועים המציינים את דרגות הקצונה. טיפוס זה נתון לכם.
- ההתייחסות בלשון זכר לאנשי הצוות היא מטעמי קיצור בלבד.

## Starfleet Ships

צי החלל של פדרציית הכוכבים המאוחדת כולל 4 סוגי חלליות:

1. Exploration Ship – חללית מחקר המשוטטת בגלקסיה וחוקרת כוכבים ותופעות טבע.
2. Fighter – חללית קרב (Battleship) קטנה ומהירה.
3. Bomber – חללית קרב (Battleship) גדולה בעלת עוצמת אש אדירה.
4. Transport Ship – חללית תובלה המאפשרת שינוע נוסעים ומטען בין בסיסי חלל.

להלן פירוט המאפיינים של סוגי החלליות השונות:

## Spaceship

נתחיל בתיאור השירותים המשותפים לכל סוגי החלליות. עבור כל חללית (להלן Spaceship) נגדיר את השירותים הבאים:

שם השירות	טיפוס החזרה	הסבר
<code>getName()</code>	String	שם החללית (מחרוזת המציינת שם ייחודי לכל חללית).
<code>getCommissionYear()</code>	int	שנת ייצור (בשנות כדור הארץ, למשל 2241).
<code>getMaximalSpeed()</code>	float	מהירות מקסימלית (שבר בין 0 ל-10).
<code>getFirePower()</code>	int	סכום כוח-אש של כל כלי הנשק המותקנים בחללית (מספרים שלמים, ביחידות של כוח-אש). לכל חללית יש כוח-אש בסיסי מובנה של 10 יחידות כוח-אש. בחלליות קרב מתווסף כוח-אש נוסף מכלי הנשק המותקנים על החללית.
<code>getCrewMembers()</code>	Set<CrewMember>	חברי הצוות המאיישים את החללית (CrewMember הינו מנשק המייצג איש-צוות מסוג כלשהו).
<code>getAnnualMaintenanceCost()</code>	int	עלות אחזקה שנתית כוללת (מספרים שלמים) ביחידות של דולר-פדרציה. נתון זה יחושב באופן שונה לכל סוג חללית על פי המפורט בהמשך.

## ExplorationShip

עבור חללית מחקר נגדיר את כל השירותים של חללית המובאים לעיל, בתוספת ההגדרות הבאות:

שם השירות	טיפוס החזרה	הסבר
<code>getNumberOfResearchLabs()</code>	int	מספר מעבדות המחקר המותקנות על חללית המחקר. עלות האחזקה של כל מעבדה היא <b>2000</b> דולר לשנה.
<code>getAnnualMaintenanceCost()</code>	int	עלות האחזקה השנתית כוללת של ספינת מחקר מורכבת מסכום הרכיבים הבאים: <ul style="list-style-type: none"> <li>עלות אחזקה שנתית בסיסית לספינה מחקר (5000 דולר)</li> <li>עלות אחזקה שנתית של המעבדות (מספר המעבדות * 2000 דולר).</li> </ul>

## Fighter

חללית קרב מהירה. נגדיר עבורה את כל השירותים של חללית בנוסף להגדרות הבאות:

שם השירות	טיפוס החזרה	הסבר
<code>getWeaponArray()</code>	<code>List&lt;Weapon&gt;</code>	רשימת כלי הנשק המותקנים על חללית הקרב.  עבור כל נשק ( <code>Weapon</code> ) נשמור את הנתונים הבאים: <ul style="list-style-type: none"> <li>שם כלי הנשק.</li> <li>כוח-אש (ביחידות כוח-אש).</li> <li>עלות אחזקה שנתית (בדולרים).</li> </ul>
<code>getFirePower()</code>	<code>int</code>	כוח האש המצטבר של חללית קרב הינו סכום כוח-האש של כל הנשקים המותקנים, בנוסף לכוח האש המובנה של כל חללית.
<code>getAnnualMaintenanceCost()</code>	<code>int</code>	עלות האחזקה השנתית של חללית קרב מסוג <code>Fighter</code> מורכבת מסכום הרכיבים הבאים: <ul style="list-style-type: none"> <li>עלות אחזקה שנתית בסיסית לספינת קרב <code>Fighter</code> (2000 דולר).</li> <li>עלות אחזקה השנתית של כלי הנשק (סכום עלות האחזקה של כל כלי הנשק המותקנים על חללית הקרב).</li> <li>עלות אחזקת מנועי החללית כתלות במהירות החללית המקסימלית (<math>1000 * MaximalSpeed</math>, מעוגל לשלמים).</li> </ul>

## Bomber

חללית קרב כבדה בעל יכולת הפצה מרשימה. נגדיר עבורה את כל השירותים של חללית בנוסף להגדרות הבאות:

שם השירות	טיפוס החזרה	הסבר
<code>getWeaponArray()</code>	<code>List&lt;Weapon&gt;</code>	רשימת כלי הנשק המותקנים על חללית הקרב.  עבור כל נשק נשמור את הנתונים הבאים: <ul style="list-style-type: none"> <li>שם כלי הנשק.</li> <li>כוח-אש (ביחידות כוח-אש).</li> <li>עלות תחזוקה שנתית (בדולרים).</li> </ul>
<code>getFirePower()</code>	<code>int</code>	כוח האש המצטבר של חללית קרב הינו סכום כוח-האש של כל הנשקים המותקנים, בנוסף לכוח האש המובנה של כל חללית.
<code>getNumberOfTechnicians()</code>	<code>int</code>	מספר הטכנאים המוצבים על החללית (מספר שלם בטווח 0-5).
<code>getAnnualMaintenanceCost()</code>	<code>int</code>	עלות האחזקה השנתית של חללית קרב מסוג <code>Bomber</code> מורכבת מסכום הרכיבים הבאים: <ul style="list-style-type: none"> <li>עלות אחזקה שנתית בסיסית לספינת קרב מסוג <code>Bomber</code> (6000 דולר).</li> </ul>

- עלות האחזקה השנתית של כלי הנשק (עלות האחזקה של כל כלי הנשק המותקנים על חללית הקרב). שימו לב – כל טכנאי המוצב על החללית מוזיל את עלויות האחזקה השנתיות על כלי הנשק ב-10%. כלומר, עלות תחזוקת כלי הנשק מופחתת בשיעור של 0-50% כתלות במספר הטכנאים. יש לעגל את המחיר לשלמים אחרי חישוב ההוזלה ביחס לסכום עלויות כלי הנשק.

## Transport Ship

עבור חללית תובלה נגדיר את כל השירותים של חללית המובאים לעיל, בתוספת ההגדרות הבאות:

שם השירות	טיפוס החזרה	הסבר
<code>getCargoCapacity()</code>	int	יכולת נשיאת מטען, ביחידות של מגה-טון (מספר שלם)
<code>getPassengerCapacity()</code>	int	יכולת נשיאת נוסעים, ביחידות של מספר נוסעים (מספר שלם)
<code>getAnnualMaintenanceCost()</code>	int	עלות האחזקה השנתית הכוללת של ספינת תובלה מורכבת מסכום הרכיבים הבאים: <ul style="list-style-type: none"> <li>• עלות אחזקה שנתית בסיסית לספינת תובלה (4000 דולר).</li> <li>• עלות של 3 דולר לכל מגה-טון של יכולת נשיאת מטען (כלומר <math>CargoCapacity * 3</math> דולר).</li> <li>• עלות של 2 דולר פר יכולת נשיאת נוסע (כלומר <math>PassengerCapacity * 2</math> דולר).</li> </ul>

## מה עליכם לעשות ?

### 1. הגדרת ממשקים (Interfaces) (5%)

#### הגדרת הממשק `CrewMember`

- הגדירו ממשק בשם `CrewMember` אשר ייצג איש צוות בצי החלל. על הממשק לכלול את המתודות `getName()`, `getAge()`, `getYearsInService()`.
- על כל מחלקה המייצגת איש צוות לממש ממשק זה.

#### הגדרת הממשק `Spaceship`

- הגדירו ממשק בשם `Spaceship` אשר ייצג חללית בצי החלל. על הממשק לכלול את המתודות המאפיינות חללית כלשהי (היעזרו בטבלה המתארת `Spaceship` כללי לעיל).
- על כל מחלקה המייצגת חללית לממש ממשק זה.

- ✓ הגדרת ממשק מאפשרת לנו לעבוד בצורה אחידה עם מחלקות שונות המממשות אותו. למשל - נוכל ליצור אוסף פולימורפי המכיל אובייקטים של חלליות מסוגים שונים ולגשת אליהם בצורה אחידה דרך המתודות המוגדרות בממשק `Spaceship`.
- ✓ יש לממש את הממשקים ואת כל שאר המחלקות בתרגיל זה תחת החבילה `starfleet`.

**2. הגדרת עץ ההורשה (5%)**

- נתחו את הדמיון בין המחלקות השונות שהוגדרו לעיל עבור אנשי צוות ועבור חלליות, ובנו עצי הורשה מתאימים אשר יכללו מנשקים, מחלקות אבסטרקטיות, מחלקות קונקרטיות ומחלקות עזר אם קיימות.
- יחסי ההורשה בין המחלקות אמורים למנוע שכפול קוד בין מחלקות.
- שרטטו את היחסים בין המחלקות השונות על פי המוגדר בראש התרגיל והגישו את דיאגרמת המחלקות בקובץ התשובות.
- וודאו שהשרטוט שלכם מכיל את כל המחלקות הקונקרטיות המפורטות בתחילת הסעיף הבא.

**3. מימוש המחלקות (40%)**

בהתבסס על עצי ההורשה אותם הגדרתם ולפי פירוט המתודות שהובא בטבלאות לעיל, ממשו את המחלקות הבאות בתוך החבילה *starfleet*:

- 1) Crewman
- 2) Officer
- 3) ExplorationShip
- 4) Fighter
- 5) Bomber
- 6) TransportShip
- 7) Weapon

- במידה ובחרתם להגדיר מחלקות אבסטרקטיות, ממשו גם אותן.
- בנאים - לכל מחלקה ניצור בנאי המקבל את כל הפרמטרים הנדרשים לאתחול שדות המחלקה. ממשו את הבנאים על פי החתימות הבאות (ניתן להניח שהקלט לבנאים תקין):

```
public Crewman(String name, int age, int yearsInService)
```

```
public Officer(String name, int age, int yearsInService, OfficerRank rank)
```

```
public ExplorationShip(String name, int commissionYear, float maximalSpeed,
Set<CrewMember> crewMembers, int numberOfResearchLabs)
```

```
public Fighter(String name, int commissionYear, float maximalSpeed,
Set<CrewMember> crewMembers, List<Weapon> weaponArray)
```

```
public Bomber(String name, int commissionYear, float maximalSpeed,
Set<CrewMember> crewMembers, List<Weapon> weaponArray, int numberOfTechnicians)
```

```
public TransportShip(String name, int commissionYear, float maximalSpeed,
Set<CrewMember> crewMembers, int cargoCapacity, int passengerCapacity)
```

```
public Weapon(String name, int firePower, int annualMaintenanceCost)
```

- מתודת toString() – בכל אחת מהמחלקות עליכם לממש מתודת toString() המחזירה מחרוזת המתארת את נתוני המחלקה.
  - המחוזות תתחיל בשם המחלקה, ואח"כ מוסטים לימין ע"י טאב בודד יופיעו נתוני המחלקה לפי הסדר והפורמט המודגמים בהמשך (סדר הופעת השדות יהיה: שדות המחלקה המשותפים לכל סוגי החלליות, אח"כ תופיע עלות האחזקה השנתית, ולאחר מכן השדות הספציפיים לאותה המחלקה).
  - מתודת ה- toString() עשויה לקרוא למתודה באותו שם במחלקת האם.
  - הקפידו שהמחרוזות שנוצרות יהיו זהות לאלו המוצגות בסוף התרגיל או בקובץ הפלט הנלווה.

הנה דוגמא למחרוזת המיוצרת ע"י מתודת ה- toString של מחלקת TransportShip:

```
TransportShip
  Name=USS Peres
  CommissionYear=2396
  MaximalSpeed=5.1
  FirePower=10
  CrewMembers=23
  AnnualMaintenanceCost=20000
  CargoCapacity=2000
  PassengerCapacity=5000
```

#### • דריסת המתודות equals() ו- hashCode()

- **public boolean** equals(Object obj)
- **public int** hashCode()

על מנת שנוכל לאחסן אובייקטים של מחלקות שיצרנו במבני נתונים המבוססים על HashTable יש לדרוש את המתודות equals() (הבודקת זהות מול אובייקט אחר) ואת המתודה hashCode() (המחזירה ערך גיבוב). וודאו שכל מחלקה המייצגת איש צוות או חללית מכילה דריסה של 2 מתודות אלו (אך הימנעו משכפול קוד מיותר תוך שימוש בהורשה).

✓ שימו לב ששדה השם מהווה ערך מזהה ייחודי עבור אנשי צוות ועבור חלליות.

✓ היעזרו באקליפס ליצירה אוטומטית של מתודות אלו ( **Source>Generate hashCode() and equals()...** ), אך וודאו שאתם מבינים את הקוד שנוצר.

#### • תמיכה במיון של אובייקטים מסוג איש צוות או חללית

ייתכן ותרצו שהמחלקות שיצרתם יממשו את הממשק Comparable כדי שניתן יהיה להשתמש בהן עם מתודות או מבני נתונים הדורשים הגדרת יחס סדר (כגון Collection.sort או כמפתחות של TreeMap). לחילופין, תוכלו בהמשך להגדיר מחלקת עזר חיצונית המממשת את הממשק Comparator ולספק אותה כמגדירת יחס סדר למתודה או לבנאי של מבנה הנתונים הרלבנטי.

#### הערות כלליות לסעיף זה:

- אתם רשאים להוסיף שדות, מתודות ומחלקות עזר נוספות בכל אחת מהמחלקות שלכם כל זמן שאתם לא פוגעים בחתימות ובממשק המוגדרים לעיל.
- שימו לב לנראות השדות בכל אחד משלבי היררכיית הירושה. לא ניתן לגשת לשדות המוגדרים כפרטיים במחלקת האם.
- הקפידו להשתמש בקבועים כשאלו נדרשים.

**4. StarfleetManager (50%)**

מחלקה זו (עבורה נתון לכם השלד) תכיל מספר מתודות סטטיות המקבלות אוסף חלליות ומחזירות חיתוכים שונים על פי הפירוט הבא:

```
1. public static List<String>
   getSpaceshipDescriptionsByCommissionYear(List<Spaceship> fleet)
```

(5%) המתודה תחזיר רשימה של מחרוזות המתארות את חלליות הצי, כאשר החלליות ממוינות לפי שנת הייצור שלהן בסדר עולה. כל איבר ברשימה המוחזרת יהיה מחרוזת שהינה תוצר של מתודת ה- toString() של אובייקט החללית המתאים.

✓ לצורך מיון החלליות על פי שנת הייצור עליכם להגדיר מחלקה שתממש את המנשק Comparator ולספק אותה כפרמטר למתודה Collections.sort.

```
2. public static Map<String, Integer> getInstanceNumberPerClass
   (List<Spaceship> fleet)
```

(5%) המתודה תחזיר מפה המכילה עבור כל שם מחלקה של חללית את מספר האובייקטים שנוצרו מהמחלקה (רק אם נוצרו, אין צורך לכלול מחלקות שלא נוצרו מהן אובייקטים).

✓ ניתן להשתמש במתודה getClass() על כל אובייקט כדי לדעת מאיזו מחלקה הוא (מקבלים חזרה אובייקט מסוג Class ואז ניתן לקבל את שם המחלקה באמצעות המתודה (getSimpleName()). גם האופרטור instanceof עשוי להיות שימושי במימוש מתודה זו.

✓ שימו לב שניתן לממש מתודה זו בשתי דרכים: האחת כוללת מעבר על רשימת החלליות הניתנת כפרמטר למתודה, והשניה כוללת החזקת מונים סטטיים באחת או יותר מהמחלקות הרלבנטיות וקידומם בבנאי(ם) בעת יצירת אובייקט חדש. וודאו שאתם מבינים את שתי הדרכים ובחרו בזו שנראית לכם יותר אלגנטית לצורך המימוש שלכם.

```
3. public static int getTotalMaintenanceCost (List<Spaceship> fleet)
```

(5%) המתודה תחזיר את סך כל עלויות האחזקה של כל חלליות הצי ע"י סכימת עלויות האחזקה של כל חללית בצי.

```
4. public static int getTotalFleetFirePower (List<Spaceship> fleet)
```

(5%) המתודה תחזיר את סך כל כוח-האש של חלליות הצי ע"י סכימת כוח-האש של כל חללית בצי.

```
5. public static float getFleetAverageOfMaximalSpeed (List<Spaceship>
   fleet)
```

(5%) המתודה תחזיר את ממוצע המהירות (המקסימלית) של חלליות הצי.

```
6. public static Set<String> getFleetWeaponNames (List<Spaceship>
   fleet)
```

(5%) המתודה תחזיר אוסף מסוג קבוצה המכיל מחרוזות המייצגות את שמות כלי הנשק השונים (ללא חזרות) המותקנים על חלליות הצי.



```
7. public static List<String> getShipsNamesSortedByMaintenanceCost
(List<Spaceship> fleet)
```

(5%) המתודה תחזיר את רשימת שמות החלליות כשהן ממוינות בסדר יורד על פי עלות האחזקה השנתית שלהן (שם החללית בעלת העלות הגבוהה ביותר תופיע ראשונה).

```
8. public static int getTotalNumberOfFleetCrewMembers (List<Spaceship>
fleet)
```

(5%) המתודה תחזיר את מספר אנשי הצוות הכולל בצי (סכום אנשי הצוות המוצבים בכל חללית)

```
9. public static float getAverageAgeOfFleetOfficers (List<Spaceship>
fleet)
```

(5%) המתודה תחזיר את הגיל הממוצע של קציני הצי (ניתן להניח שעל כל חללית קיים קצין אחד לפחות).

```
10. public static Map<Officer, Spaceship>
getHighestRankingOfficerPerShip (List<Spaceship> fleet)
```

(5%) המתודה תמצא את הקצין בעל הדרגה הבכירה ביותר המוצב על כל חללית בצי, ותחזיר מפה הממפה לכל קצין כזה את החללית בה הוא מוצב.

שימו לב שהטיפוס OfficerRank שמסופק לכם, הוא Enum אשר מונה את דרגות הקצונה על פי סדר הבכירות שלהן. Enum בג'אוה מממש את הממשק Comparable ועל כן ניתן למיין לפיו.

## StarfleetManagerTester .5

המחלקה StarfleetManagerTester מייצרת צי של חלליות על צוותיהן ומשתמשת בכל המחלקות והמתודות שכתבתם כדי להדפיס דוח מסכם למסך. לאחר שסיימתם את מימוש כל המחלקות, הריצו את המחלקה StarfleetManagerTester שמסופקת לכם בשלמותה ובדקו שהפלט המודפס על ידכם זהה לפלט המובא בהמשך (בכל מקום שבו מופיעה הזחה, ניתן להניח שמדובר בטאב בודד).

- ✓ המחלקה StarfleetManagerTester מייצרת צי חלליות וצוותים בצורה אוטומטית (אך לא רנדומאלית, על מנת שתוכלו לקבל פלט זהה לשלנו בכל הרצה).
- ✓ שמות אנשי הצוות והחלליות אמורים להיות ייחודיים ועל כן יצרנו שמות מהצורה "James #121".

בהצלחה !

\*\*\* STARFLEET COMMAND OFFICIAL REPORT \*\*\*

Fleet Ships by Commission Year:

TransportShip

Name=USS Peres  
 CommissionYear=2396  
 MaximalSpeed=5.1  
 FirePower=10  
 CrewMembers=23  
 AnnualMaintenanceCost=20000  
 CargoCapacity=2000  
 PassengerCapacity=5000

Bomber

Name=USS Defiant  
 CommissionYear=2416  
 MaximalSpeed=9.9  
 FirePower=120  
 CrewMembers=236  
 AnnualMaintenanceCost=6150  
 WeaponArray=[Weapon [name=Laser Canons, firePower=10, annualMaintenanceCost=100], Weapon [name=Photon Torpedoes,  
 firePower=100, annualMaintenanceCost=200]]  
 NumberOfTechnicians=5

Fighter

Name=USS Yamaguchi  
 CommissionYear=2423  
 MaximalSpeed=6.0  
 FirePower=290  
 CrewMembers=139  
 AnnualMaintenanceCost=8580  
 WeaponArray=[Weapon [name=Laser Canons, firePower=10, annualMaintenanceCost=100], Weapon [name=Quantum  
 Torpedoes, firePower=120, annualMaintenanceCost=200], Weapon [name=TAU Phasers, firePower=150,  
 annualMaintenanceCost=280]]

ExplorationShip

Name=USS Andromeda  
 CommissionYear=2451  
 MaximalSpeed=7.2  
 FirePower=10  
 CrewMembers=78  
 AnnualMaintenanceCost=27000  
 NumberOfResearchLabs=11

ExplorationShip

Name=USS Enterprise  
 CommissionYear=2532  
 MaximalSpeed=4.5  
 FirePower=10  
 CrewMembers=46  
 AnnualMaintenanceCost=27000  
 NumberOfResearchLabs=11

TransportShip

Name=USS Lantree  
 CommissionYear=24571  
 MaximalSpeed=5.1  
 FirePower=10  
 CrewMembers=9  
 AnnualMaintenanceCost=33000  
 Cargo Capacity=3000  
 Passenger Capacity=10000

Ship Counts by Type:

1 Bomber  
 2 ExplorationShip  
 1 Fighter  
 2 TransportShip

Weapon Types:

Photon Torpedoes  
 TAU Phasers  
 Quantum Torpedoes  
 Laser Canons

Ship names sorted by annual maintenance cost:

USS Lantree  
 USS Andromeda  
 USS Enterprise  
 USS Peres  
 USS Yamaguchi  
 USS Defiant

Highest ranking officer per ship:

Captain Picard #133 USS Peres  
 Captain FitzRoy #151 USS Lantree  
 Captain Sparrow #178 USS Yamaguchi  
 Captain Kathryn #93 USS Enterprise  
 Captain Beverly #319 USS Defiant  
 Captain Riker #14 USS Andromeda

Fleet Totals:

Total Fire Power: 450  
 Average Fleet Speed: 6.30  
 Total fleet crew members: 531  
 Average age of fleet officers: 42.17  
 Total Annual Maintenance Cost: 121730