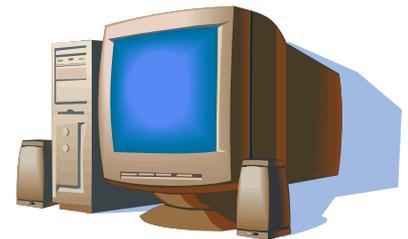


情報①

1学期 第8⇒9回

Project.8「作品制作プロジェクト③」



今日の授業のながれ

- **Project.8「作品制作プロジェクト③」**
 - ◆ 作品説明書について
 - ◆ 救済措置について
- 配布資料
 - Project.8.zip

クリアファイルを返却します。
名前を呼ばれたら取りに来てください。



出欠について

- 授業ではクリアファイルと併せ、[WebClass]でも「出席」を取ります
 - 授業開始[10分後]まで⇒「出席」扱い
 - 以降[授業時間内]⇒「遅刻」扱い
 - 以降[授業終了後]⇒「欠席」扱い

- WebClass上部「出席」⇒「2025/× ×/× × 出席確認」⇒「開始」⇒「出席します！」

教材 マイレポート 成績▼ **出席** その他▼ コース▼

学生モード 解除

出席

教材名	状態	回数制限	パスワード
» 2022/01/25 出席確認	出席	1回	-

合計 1回
必要出席数 1回

出席:1
遅刻:0
欠席:0

課題の取り組み方

作品づくりの心得

- 今日の作業目標を決める
 - 例:シューティングゲームの場合
 - 自機をジョイスティックで動かせるようにする
 - 余裕があれば, 弾を発射できるようにしておく
- 作業を進める
 - 問題点があれば, 私や友達に相談してみる
 - こまめに保存をする!



今日の目標

- **ゲームの中核部分のプログラミングを続ける**
 - 最低限のゲームとして成立させる機能が中核部分
- 例:シューティングゲーム
 - **中核部分**
 - 自機の操作、弾丸の発射
 - 敵機の移動と弾丸とのあたり判定
 - **オプション**
 - 自機と敵とのあたり判定(微妙なラインだが、なくてもゲームとして成り立つ可能性は高い)
 - アイテムによるパワーアップ
 - ゲームオーバーやスコア
 - タイム制限

今後の予定



作品制作の流れ

(第7回)

(第8回)

(第9回)

(第10回)

Project
5

シート返却

作品
制作

①

作品
制作

②

作品
制作

③

作品
制作

④

企画シート
提出

作品提出

前半

後半

今ココ！

【前回までの復習】

作品制作プロジェクト について

一学期の最終作品

- 選択「情報①」の授業では、1学期の成績は平常点（授業内課題や提出物など）及び、最終提出作品で決まります。

最終作品の課題

Eclipse(Java)で作成したミニゲーム
(イメージとしてはWebサイトのミニゲーム)

例：ドラえもんシューティング



プログラム(もの)を作るときの手順

企画シート

どんなプログラムを作るか分析する (企画・仕様)



概要設計

どうやってそのプログラムを作るか考える (設計)



プログラムを作る (実装)



作ったプログラムを検証する (テスト)

データの受け渡し

● 作品の提出方法

- ① 作品データをzipファイルでエクスポート
- ② 授業用サイト(Classroom)にアップロード

● 自宅での作業方法

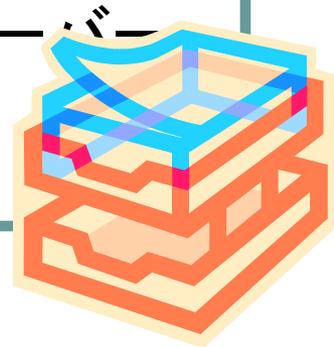
- ③ 授業用サイト(Classroom)からダウンロード
- ④ 作品データzipファイルを解凍してインポート
- ⑤ 授業用サイト(Classroom)に再アップロード

作品説明書について

説明書について

- 説明書は『ワード』で作成
- 簡単な概要(タイトル名,ゲームの目的)
- **遊び方**(初見の人が見て分かる操作方法)
- 例)
 - ① Javaを実行するとスタート
 - ② 左右キー(←、→)でドラえもんを左右に動かす
 - ③ Enterボタンを押して敵を倒すと得点ゲット
 - ④ タイムオーバーもしくは星に当たるとゲームオーバー
 - ⑤ 右上の×を押して終了

などなど



説明書について

- 説明書は『ワード』で作成

- 簡単な概要(何をしたか(目的))

- **遊バ**

最終作品と同時に

提出してもらいます

(授業用サイトに見本有！)

- 例)

- ① Java

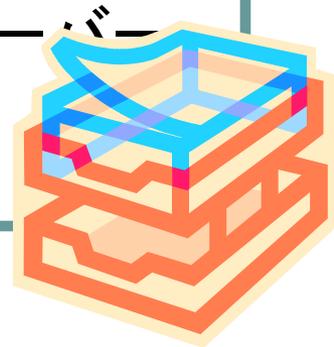
- ② 左右キー (と動かす

- ③ Enterボタンと押しを倒し点ゲッ

- ④ タイムオーバーもは星に当たるとゲームオーバー

- ⑤ 右上の×を押して終了

などなど



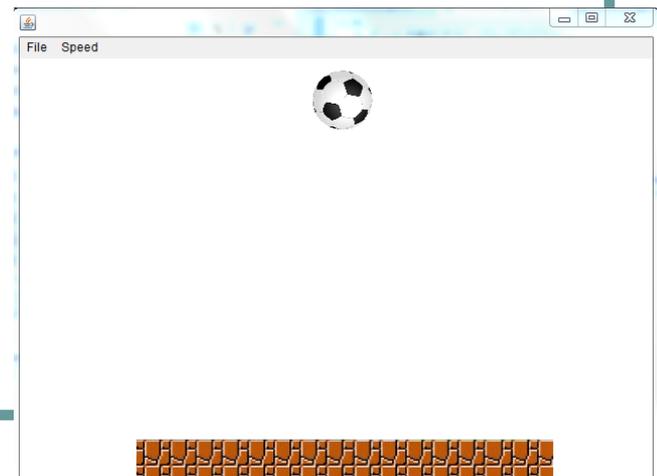
Project.6

「ゲームサンプル①」

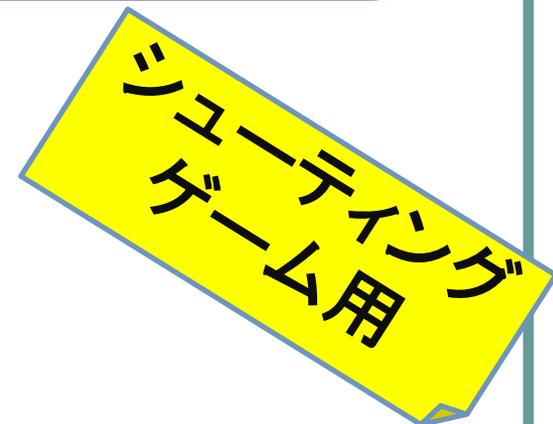
①ボールを動かすサンプル

- 初期化
 - タートルを消す
 - 画面の大きさと位置を調整
 - 初期スピードの調整(縦方向は自由落下)
 - ボールを真ん中, 上の方に配置
 - ブロックを真ん中, 下の方の配置
- アニメーションループ
 - 待つ
 - 処理
 - ボールを動かす
 - ブロックにあたらったら反射させる
 - キーボードでボールを左右に動かす
 - 再描画

アクション
ゲーム用



②シューティングゲームのサンプル



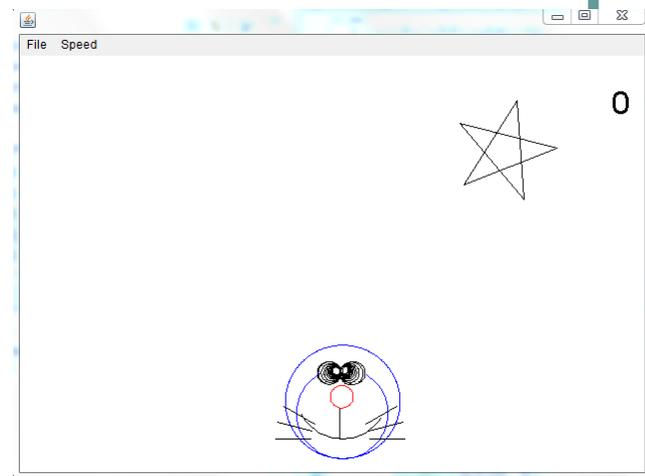
● 初期化

- タートルを消す
- 画面の調整
- キャラクターの生成
- 得点板の生成
- 弾を適切な大きさに調整
- キャラクターの位置を調整
- 得点板の位置を調整
- 得点を初期化

● アニメーションループ

- 待つ
- 処理
 - 敵を動かす(流れ星と一緒に)
 - もし、敵が右端だったら左端にワープ
 - ドラえもんを左右に動かす
 - 上が押されたら、弾を出す(弾をドラえもんと同じ位置にする)
 - 弾を上を動かす
 - もし、弾が敵に当たっていたら得点を増やす
 - 得点板の数字を更新

● 再描画



Project.7

「ゲームサンプル②」

Project.7のサンプルについて

● サンプル

- ① **ClickSample.java**
 - あるオブジェクトがクリックされたか判定するサンプル
- ② **JumpSample.java**
 - ジャンプの動作を実現するためのサンプル
- ③ **MotionSample.java**
 - あるオブジェクトを一定期間表示させるサンプル
 - ※ **Whileループの中でNewしていないことがポイント**
- ④ **MoveEnemySample.java**
 - (インベーダーゲームのような)敵の動きのサンプル
- ⑤ **SlowTimerSample.java**
 - ゆっくり動くタイマーのサンプル

● ツール

- **KeyCodeChecker.java**
 - キーコードを調べるツール

前は5つのサンプルと
1つのツールが入ってます！

作品制作プロジェクト

③

ゲームサンプル について

- 授業で説明しきれなかったけれど、ゲーム作成に関わる内容を解説します。
- ゲームサンプル用ファイル
 - Project6
 - Project7(今回)
 - Project8(今回)

今学期のサンプル
はこれでおしまい

サンプルとできること

- オブジェクトを隠す・出現させる
 - 弾にあたったら、敵を消す など
- オブジェクトの見た目を別のオブジェクトにする
 - パワーアップすると、プレイヤーの見た目が変わる
- 文字を表示させる
 - タイムやスコア、説明などを表示させる
- キー入力を調べる
 - キャラの移動 など
- マウス入力を調べる
 - クリックによる操作 など
- 2つのオブジェクトの衝突判定
 - 敵キャラとプレイヤーのあたり判定 などなど

ゲームサンプル配布

授業用サイトから
ダウンロード可能

Project.8

「ゲームサンプル③」

Project.8のサンプルについて

- サンプル

- GameOverSample.java

ゲームオーバー画面のサンプル(条件の指定)

- SoundSample.java【おまけ】

音楽や効果音をつけるためのサンプル

今学期のサンプル(Project)
の配布はこれでおしまい！

①GameOverSample

①ゲームオーバーサンプル

- ゲームオーバー画面を用意して隠す

```
// ゲームオーバー画面の調整
gameOverScreen.warp(250, 220); // 中央に移動する
gameOverScreen.hide(); // 開始時は隠す
```

- ゲームオーバーの条件を調べて処理を行う

```
// 残りタイムがなくなったらゲームオーバー
if (time < 0) {
    gameOverScreen.show(); // ゲームオーバー画面を表示
    update(); // アニメーションループから脱出するので、画面更新をする
    break; // アニメーションループから脱出する
}
```



新しい命令について

新しい命令

- **break;**
 - アニメーションループから脱出する

直前でupdate();する

```
// 残り時間がなくなったらゲームオーバー
if (time < 0) {
    gameOverScreen.show(); // ゲームオーバー画面を表示
    update(); // アニメーションループから脱出するので、画面更新をする
    break; // アニメーションループから脱出する
}
```

【おまけ】

②SoundSample

②音楽の再生サンプル

● はじめに

- 再生できるファイルの種類は「**MP3形式、WAVE形式、MIDI形式**」
- 画像ファイルと同じ位置(フォルダ)に音楽ファイルを置く
- 素材はなるべくフリーの物を使用

【おすすめ素材サイト】 <http://www.01earth.net/>

- 著作物を使う場合は参照・引用をプログラム内コメントに記述
- プログラムの行頭に必ず次の1行を加える。

```
import blib.sound.BSound;
```

②音楽の再生サンプル

● BGMについて

- BGMなど再生時間が長い音を鳴らす場合は、BSound オブジェクトを作成
- 作成したBSoundオブジェクトに対して後述の play() や stop() などの命令を行う

例)

```
BSound bgm = new BSound("bgm.mp3");  
bgm.play();
```

音声に関連した新しい命令

- **play();**
 - 音を1度だけ再生します
- **loop();**
 - 音をループ再生します
 - 音の再生が終わると、**stop()**が呼ばれるまでもう一度はじめから再生し続けます
- **stop();**
 - 音を停止します
- **isPlaying();**
 - 音が再生中かどうか調べます
 - if文の条件式の中に書く

音声に関連した新しい命令

- **getVolume();**
 - 音量を取得。
 - 取得する音量は0(消音)から100(最大音量)
- **setVolume(??);**
 - 音量を??に設定。
 - 設定できる音量は0(消音)から100(最大音量)
- **getDefaultVolume();**
 - 初期設定の音量を取得。
 - 取得する音量は0(消音)から100(最大音量)

②音楽の再生サンプル

● 効果音について

- 効果音など再生時間が短い音を鳴らす場合は次ページの簡易的な命令を使うとよい
- その際、音量を指定して再生することもできる
- ただし、簡易的な命令で音を再生した場合、音を途中で停止したり、音量を変更したりすることができない

音声に関連した新しい命令

- **BSound.play(??);**
 - 指定した音ファイルを再生
- **BSound.play(??, ??);**
 - 1番目の引数で指定した音ファイルを、2番目の引数で指定した音量(0-100)で再生

②音楽の再生サンプル

● その他の諸注意について

- 効果音等の再生時に反応速度が重要な場合、あらかじめ音を読み込んで早くできる
- ただし、サイズが大きなファイル(再生時間が長い)を読み込む場合、それだけメモリが必要

● **BSound.load(??);**

- 指定した音ファイルを読み込みます。

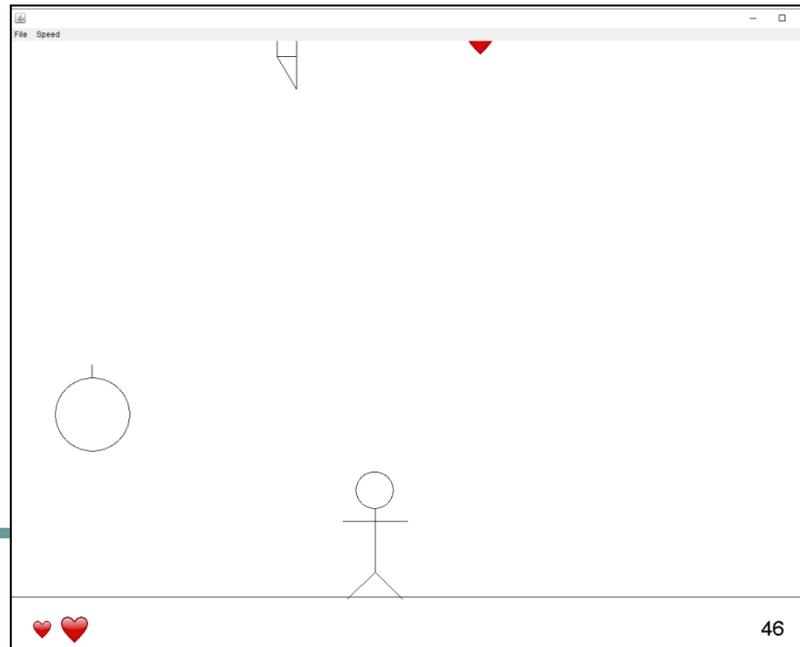


【おまけ】
音楽関連

音楽関連

● 選択「情報」OB:上野 友裕 先輩

- 2017年3月『学習院高等科』卒業
- 2021年3月『東京音楽大学作曲科』卒業
- ゲーム作品に自作BGMを付けたりしてました



音楽関連

- 現在、**フリーランスの作編曲家**として活動
 - 奥華子さんのアルバム曲の編曲や某有名作品のオーケストラアレンジなど
- 音楽教室など幅広く活動してます、何か音楽関係でありましたらお気軽にご相談を♪
 - ピアノレッスン
 - 作曲基礎レッスン
 - 出張DTMLレッスン

など

作編曲家

上野 友裕



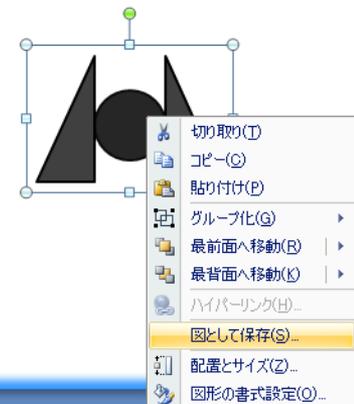
Tel: 090-1700-8864
LINE ID: instermentsplayer
E-mail: tomohiroueno6451@gmail.com
<https://twitter.com/ueno6451>

画像の作成について

画像作成の方法

- 周囲を透明にした画像を作成したい場合は Power Pointを使うと便利
- 絵を書いたら、右クリックして「グループ化」し、右クリックして「図として保存」

その他の透過画像の作り方は
授業用サイトを参照



タートルの命令まとめ

タートル(オブジェクト)への命令

● 基本命令

- `rt(〇〇)` タートル(オブジェクト)を〇〇度右に回らせる
- `lt(〇〇)` タートル(オブジェクト)を〇〇度左に回らせる
- `fd(〇〇)` タートル(オブジェクト)を〇〇歩前に進ませる
- `bk(〇〇)` タートル(オブジェクト)を〇〇歩後に進ませる

- `up()` ペンを上げる(軌跡を書かない)
- `down()` ペンを下げる(軌跡を書く)

- `color(java.awt.Color.色の指定)` ペンの色を変える

【参考】 black(黒) blue(青) green(緑) red(赤)
white(白) yellow(黄色) cyan(水色) gray(灰色)
pink(ピンク) orange(オレンジ) magenta(赤紫)
darkGray(濃い灰色) lightGray(明るい灰色)



その他の基本命令

● その他の基本命令

- `input()`
 - ユーザからの入力を受け付ける
- `print(“出力する文字列”や変数)`
 - 文字列等をコンソールに出力する
- `random(数)`
 - 指定した数の乱数を発生(「0～指定した数-1」まで)
- `System.exit(0)`
 - プログラムの最後を書いてウィンドウ画面を閉じる
- `show()` タートル(オブジェクト)を出現させる
- `hide()` タートル(オブジェクト)を隠す



オブジェクトの生成に関する命令

● オブジェクトの生成に関する命令

- Turtle [名前] = new Turtle()
 - タートル(オブジェクト)を生成し、名付ける
- [図形のファイル名(クラス)] [名前] = new [図形のファイル名]()
 - 図形オブジェクトを生成し、名付ける
- ImageTurtle [名前] = new ImageTurtle("[画像のファイル名]")
 - 指定の画像を初期値とする画像オブジェクトを生成し、名付ける
- TextTurtle [名前] = new ImageTurtle("[文字列]")
 - 指定の文字列を初期値とする文字オブジェクトを生成し、名付ける



アニメーションのための命令

● アニメーションのための命令

- **sleep([秒の指定]);**
指定された秒数だけ止まる

- **update();**
画面を書き換える

(※ 画面を書き換えないと、アニメーションが動かない！)

- **break;**
アニメーションループから脱出する

```
// 残りタイムがなくなったらゲームオーバー  
if (time < 0) {  
    gameOverScreen.show(); // ゲームオーバー画面を表示  
    update(); // アニメーションループから脱出するので、画面更新をする  
    break; // アニメーションループから脱出する  
}
```

直前でupdate();する



座標に関する命令

- 座標に関する命令

- **warp([x座標, y座標])**

指定された(x, y)座標にオブジェクトをワープ(瞬間移動)

【参考】ウィンドウ画面の表示位置を変更する
window.warp(数値, 数値);

- **getX()** (オブジェクトの)現在のx座標を取得
- **getY()** (オブジェクトの)現在のy座標を取得



新しい命令まとめ (ゲーム制作編)

出現/隠すに関する命令

- 出現/隠すに関する命令
 - **show()** オブジェクトを出現させる
 - **hide()** オブジェクトを隠す
 - **isShow()**
 - 現在、隠されているか、現れているかを調べる
 - **show(??)**
 - ??にtrueと書けば、出現させ、??にfalseと書けば、隠す
 - **show(!isShow())**
 - 隠れていれば、出現させ、出現していれば、隠す



大きさに関する命令

● 大きさに関する命令

● `.size([横の長さ], [縦の長さ])`

- そのオブジェクトを横の長さ、縦の長さを、指定された大きさにします。
- 命令に数字2つ必要なので、カンマで区切るのを忘れずに。

【参考】ウィンドウ画面のサイズを調整する
`window.size(数値, 数値);`

- `.getWidth()` 現在の横幅を取得します。
- `.getHeight()` 現在の縦幅を取得します。



大きさに関する命令

● 大きさ変更に関する命令

- **.scale([倍率])** 指定した倍率の大きさに拡大・縮小
- **.large([長さ])** 指定した長さだけ縦横を幅を大きくします
- **.small([長さ])** 指定した長さだけ縦横の幅を小さくします
- **.wide([長さ])** 指定した長さだけ横幅を大きくします
- **.narrow([長さ])** 指定した長さだけ横幅を小さくします
- **.tall([長さ])** 指定した長さだけ縦幅を大きくします
- **.little([長さ])** 指定した長さだけ縦幅を小さくします



その他の命令

- 見た目に関する命令
 - **looks(??)**
 - オブジェクトを??に入れられたオブジェクトとおなじ見た目に変身させます。
- テキストの中身を変える命令
 - **text("文字列"など)**
 - テキストオブジェクトの文字を(文字列など)に変更
 - (??)には、『"文字列"(や変数など)』を入れることが可能



キー・マウス入力に関する命令

- 押されているキーコード(キーボードの番号)を調べる
 - **key()**
 - 何も押されていない時は, キーコードが-1
 - 「左:37上:38右:39下:40」
- マウス位置を調べる命令
 - **mouseX()**
 - マウスのx座標を取得します。
 - **mouseY()**
 - マウスのy座標を取得します。



マウス入力を調べる命令

- マウス入力に関する命令
 - **mouseDown()**
 - いまマウスが押されているかどうかを調べます
 - **leftMouseDown()**
 - いまマウスの左ボタンが押されているか調べる
 - **rightMouseDown()**
 - いまマウスの右ボタンが押されているか調べる
 - **doubleClick():**
 - マウスがダブルクリックされたかどうかをる
 - 必ず, if文などの条件式の中に書く



当たり判定に関する命令

- オブジェクトの接触に関する命令

- **intersects(??)**

- オブジェクトが??(オブジェクト)と当たっているかどうか調べる
- 必ず, if文などの条件式の中に書く

- **contains(??, ??)**

- オブジェクトが座標(??, ??)に含まれているかどうか調べる
- 必ず, if文などの条件式の中に書く



本日の演習

演習の順番

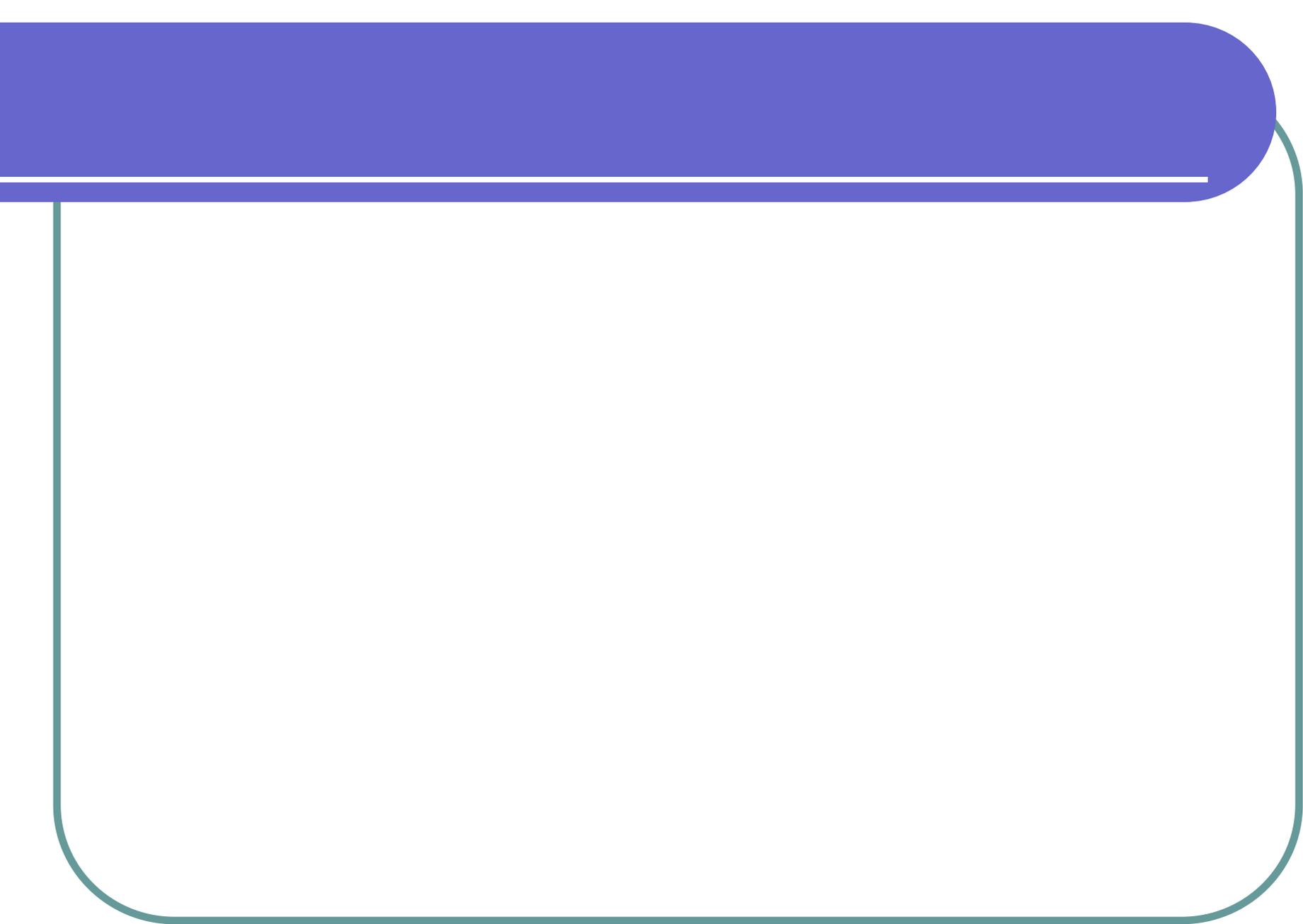
配布された**サンプル**の意味を理解する

- ① 返却された**企画シート等**を確認する
- ② 作品制作をする(返却物を見ながら)



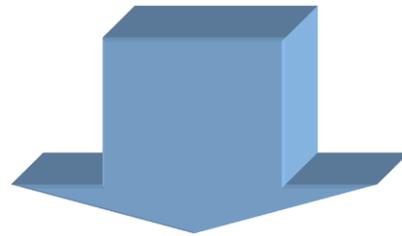
企画シート 返却

救済措置について



オンライン提出

- **7月9日(水)23:59まで**作品データの更新を受け付けます！
- 提出方法は授業内提出のときと同様です



オンライン提出する場合でも、
授業の最後に**必ず提出**してもらいます！

- ただし **7月8日(火)**は私用のため対応不可

- 質問等あれば、前日(7/7)までに解決！！

- そこで、、、

期末試験終了後

「7/7(月) 12:00～16:00」PC教室開放

(重要) 学期末の提出物について

- 提出物
 - 1. 作品のデータファイル
 - 作品ファイル (zip形式で圧縮)
- 最終締め切り
 - 7月9日 (水) 23:59 予定
 - 締め切り厳守 (※採点・成績処理の関係上)

**【参考】作業環境整備
～自宅での作業方法～**

自宅での作業方法

- この授業ではGoogleClassRoomからオンライン提出なので、自宅での作業環境が整っていると、授業外作業や自宅から提出できるためおすすめ！
- 逆に難しい人は必ず授業内で完成させる！
- 「USBメモリなど(やクラウドサービス)」を用いて、Eclipseフォルダ丸ごと持ち運ぶのも便利♪
(ノートパソコン持ち込みも可)

データの受け渡し

● 作品の提出方法

- ① 作品データをzipファイルでエクスポート
- ② 授業用サイト(Classroom)にアップロード

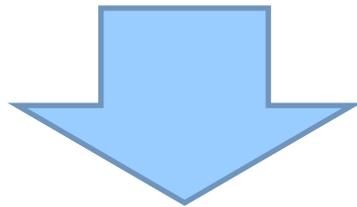
● 自宅での作業方法

- ③ 授業用サイト(Classroom)からダウンロード
- ④ 作品データzipファイルを解凍してインポート
- ⑤ 授業用サイト(Classroom)に再アップロード

提出の練習を
してみよう！

バックアップについて

授業用サイトにサインインしてください



授業用サイトから
バックアップ提出



(1) 作品提出

データの受け渡し

● 作品の提出方法

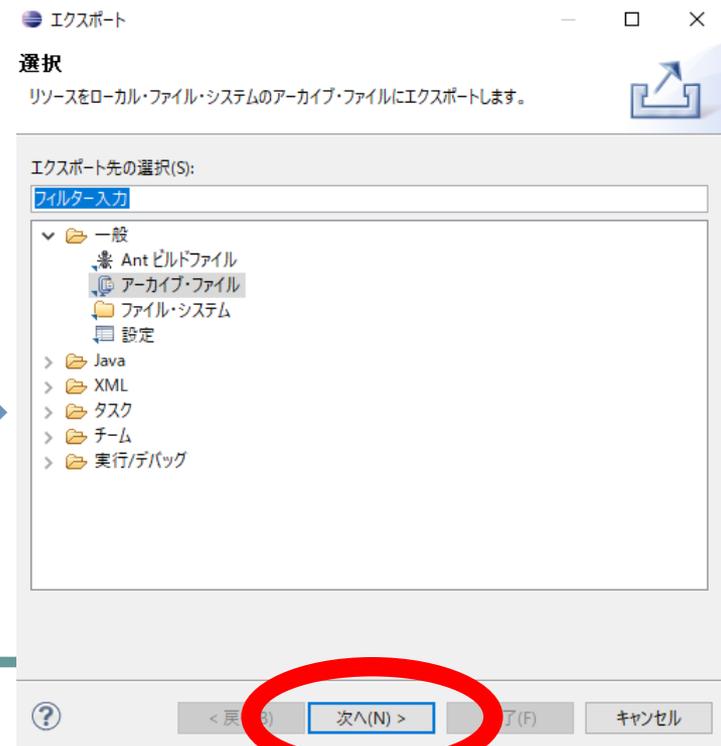
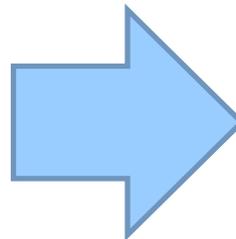
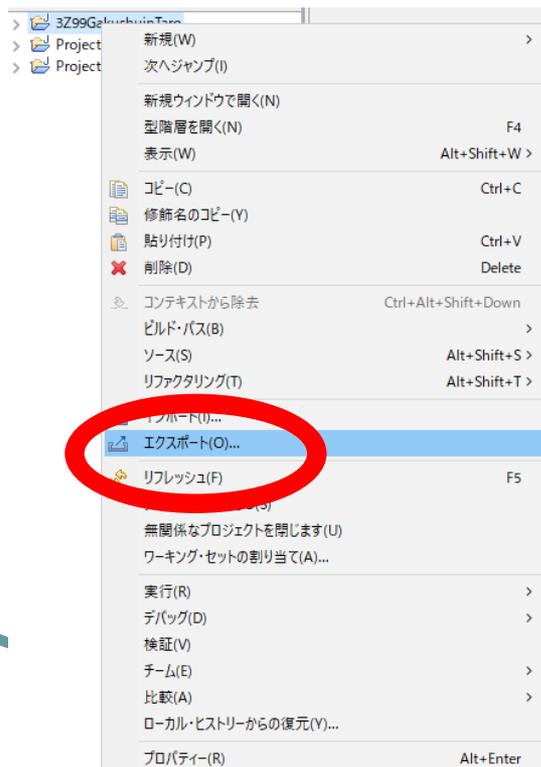
- ① 作品データをzipファイルでエクスポート
- ② 授業用サイト(Classroom)にアップロード

● 自宅での作業方法

- ③ 授業用サイト(Classroom)からダウンロード
- ④ 作品データzipファイルを解凍してインポート
- ⑤ 授業用サイト(Classroom)に再アップロード

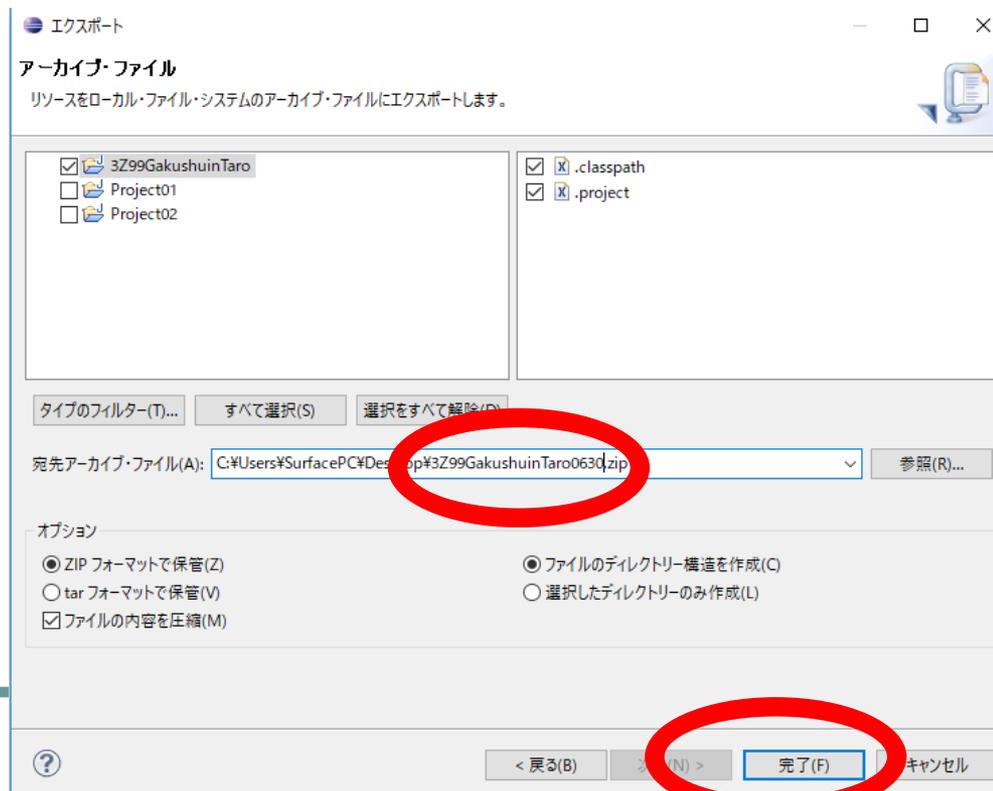
① 作品データのエクスポート

- 「作品プロジェクト(2Z99名前)」を右クリック→【リフレッシュ】
- 「作品プロジェクト」を右クリック→「エクスポート」→「一般」から「アーカイブファイル」を選択



① 作品データのエクスポート

- 「宛て先アーカイブファイル」の右【参照】→保存場所(デスクトップ)を選択し、ファイル名を「**組番号名前日付**(例:2Z99〇〇0630.zip)」→【完了】を押す(※zipファイルが作成される)



② 作品データのアップロード

- 授業用サイトの「作品の提出」及び「作品説明書の提出」クリック

<https://www-cc.gakushuin.ac.jp/~e110238/info1/index.html>

✓ 作品提出先

1.  「[作品 \(zipファイル\)](#)」及び「[作品説明書 \(Wordファイル\)](#)」の提出

- Google Classroomに移動
- Googleにログイン(Gmailアカウント)
- 2025年度 選択「情報1」1学期末課題の仮提出



学習院太郎 さんが新しい課題を投稿しました: 2020年度 選択「情報2」1学期末課題
6月13日 (最終編集: 18:15)



利用方法

●【Google Classroom】

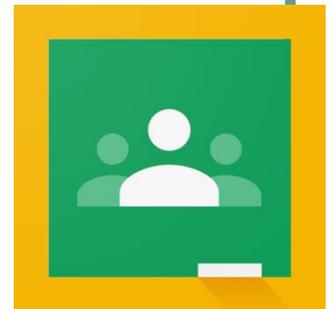
- Googleが公開しているWebサイト
- 遠隔授業での課題提出や「コメントの送受信」ができる
- Googleアカウント(Gmail)が必要
 - 持っていない人はこの際に作成！

【アクセス方法】

- Webブラウザで以下のサイトにアクセス

<https://classroom.google.com/>

または「Google Classroom」で検索



利用方法

Classroom で指導と学習を管理する

Classroom は、生徒と教師による課題の管理、コラボレーションの促進、コミュニケーションの改善に役立ちます。

①

Classroom にアクセス

Classroom へのログインで問題が発生した場合は、[こちらのヘルプ記事](#)をご確認ください。

②

Google

ログイン

ドキュメントに移動する

メールアドレスまたは電話番号

メールアドレスを忘れた場合

ご自分のパソコンでない場合は、In Private ウィンドウ
を使用してログインしてください。詳細

アカウントを作成

次へ

Gmailアカウント

- ① Classroomにアクセス○をクリック
- ② Googleにログイン
(Gmailアカウント)
- ③ Classroomが起動



② 作品データのアップロード

- 「あなたの課題」の「追加または作成」⇒ファイル
- 【zipファイル】と【説明書のWordファイル】を選択してアップロード

The screenshot illustrates the process of uploading files to a Google Drive task. On the left, the 'あなたの課題' (Your task) menu is open, showing the '追加または作成' (Add or create) button. The 'ファイル' (Files) option is highlighted with a red circle. A blue arrow points to the file selection interface, which shows two files selected: '3Z99GakushuinTaro0630.zip' and '3Z99作品説明書.docx'. On the right, the 'アップロード' (Upload) button is circled in red, indicating the next step in the process.

②作品データのアップロード

あなたの課題 割り当て済み

3Z99GakushuinTa...
圧縮アーカイブ ×

3Z99作品説明書.d...
Word ×

+ 追加または作成

提出

限定公開のコメント

限定公開コメントを追加 ▶

- 画面右側の「あなたの課題」の「追加または作成」⇒提出をクリック
- 『限定公開のコメント』に組番号名前、作業状況を追加例)
2Z99学習院、未完成なので、後日提出します!

次回予告

次回予告

- 作品制作プロジェクト④

クリアファイルを
提出して帰る事！

以上

NEXT
TIME

おつかれさまでした。
それでは、また次回！！

